University of Amsterdam

Master's Thesis

Improving Initialization in MO-RV-GOMEA Optimization for Cervical Cancer Brachytherapy Treatment Planning

Author:
Wiebe Timmers

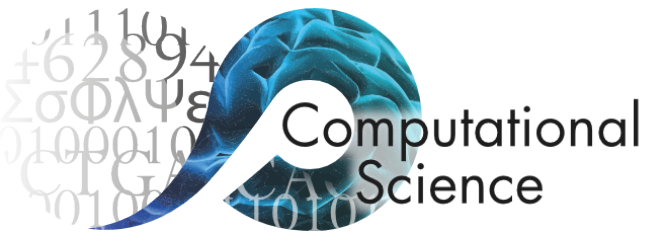
Examiner:
Mark Hoogendoorn
Supervisors:
Renzo Scholman
Evangelos Kostoulas
Peter Bosman
Tanja Alderliesten
Assessor:
Tanja Alderliesten

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Computational Science

in the

Computational Science Lab
Informatics Institute

May 2024



Declaration of Authorship

I, Wiebe Timmers, declare that this thesis, entitled 'Improving Initialization in MO-RV-GOMEA Optimization for Cervical Cancer Brachytherapy Treatment Planning' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at the University of Amsterdam.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date: 8 May 2024

"What I cannot create, I do not understand."

Richard P. Feynman

UNIVERSITY OF AMSTERDAM
Faculty of Science

Abstract

Informatics Institute

Master of Science in Computational Science

Improving Initialization in MO-RV-GOMEA Optimization for Cervical Cancer Brachytherapy Treatment Planning

by Wiebe Timmers

In High-Dose-Rate BrachyTherapy (HDR-BT), a form of internal radiotherapy, the time to construct a treatment plan is desired to be as short as possible. To reduce the treatment planning time, a shift from manual to automatic treatment planning has been made over the past decades. Specifically, automatic treatment planning for HDR-BT is aimed to output treatment plans that satisfy a clinical protocol and are produced within a limited time frame. An optimization algorithm that is particularly well-suited for performing such a task is the Multi-Objective Real-Valued Gene pool Optimal Mixing Evolutionary Algorithm (MO-RV-GOMEA). Recent applications of MO-RV-GOMEA to prostate HDR-BT treatment planning have been shown to produce high-quality treatment plans efficiently. However, the application to cervical HDR-BT treatment planning has exemplified the need to further decrease the run-time required by MO-RV-GOMEA to produce a clinically viable treatment plan solution (evaluated by the so-called golden corner). We propose an ensembled Feedforward Neural Network model with Exponential search and Interpolation output correction (eFNN+ExpInt) to generate a set of treatment plan solutions that can be used to initialize MO-RV-GOMEA. The eFNN+ExpInt model's main purpose is to resemble a proof-of-concept initialization that can reduce the required MO-RV-GOMEA optimization time toward clinically viable treatment plans. Our results show that the required MO-RV-GOMEA optimization time to reach a clinically viable plan is significantly reduced for the majority of tested patients by the use of eFNN+ExpInt initialization. Specifically, the required average runtime (over 25 patient fractions) to reach the golden corner was 10.662 seconds for eFNN initialization, 9.297 seconds for eFNN+ExpInt initialization and 11.156 seconds for Heuristic+ExpInt initialization as opposed to 21.597 seconds for the current initialization benchmark. A warm start initialization such as eFNN+ExpInt can enable MO-RV-GOMEA to converge faster to a clinically viable solution and can open opportunities for more sophisticated and time-costly patient treatment planning solutions.

Acknowledgements

I would like to thank both my daily supervisors (Renzo and Evangelos) for their unconditional support and mentorship over the past 9 months. I want to especially thank Renzo for his guidance on MO-RV-GOMEA and its application to brachytherapy, and for making this work technically possible by means of the BRIGHT software. I want to especially thank Evangelos for his guidance on the machine learning part of the project, which was specifically important for the eFNN optimization and the early phases of the project. Both Renzo and Evangelos contributed significantly to the ideas presented in this thesis. Additionally, I would like to thank anyone in the EI group who contributed to any useful discussion. Lastly, I would like to thank Peter and Tanja (coordinating/-monthly supervisors) for their guidance and for giving me the opportunity to do research in the evolutionary intelligence domain.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

UvA	**U**niversiteit **v**an **A**msterdam
HDR-BT	**H**igh-**D**ose **R**ate **B**rachy**T**herapy
MRI	**M**agnetic **R**esonance **I**maging
CT	**C**omputated **T**omography
CTV	**C**linical **T**arget **V**olume
OAR	**O**rgan **A**t **R**isk
GTV-RES	**G**ross **T**umor **V**olume **RES**idue
HR-CTV	**H**igh-**R**isk **C**linical **T**arget **V**olume
IR-CTV	**I**ntermediate-**R**isk **C**linical **T**arget **V**olume
DVI	**D**ose **V**olume **I**ndices
BRIGHT	**BR**achytherapy via artificially **I**ntelligent **G**OMEA-**H**euristic based **T**reatment planning
MO-RV-GOMEA	**M**ulti-**O**bjective **R**eal-**V**alued **G**ene-pool **O**ptimal **M**ixing **E**volutionary **A**lgorithm
GPU	**G**raphic **P**rocessing **U**nit
ML	**M**achine **L**earning
EA	**E**volutionary **A**lgorithm
LPM	**L**inear **P**enalty Model
LP	**L**inear **P**rogramming
QPM	**Q**uadratic **P**enalty **M**odel
DTMR	**D**well **T**ime **M**odulation **R**estriction
DVM	**D**ose **V**olume **M**odel
MTD	**M**ean-**T**ail-**D**ose
DVH	**D**ose **V**olume **H**istogram
CNN	**C**onvolutional **N**eural **N**etwork

GAN	**G**enerative **A**dversarial **N**etwork
FiLM	**F**eature-wise **L**inear **M**odulation
TP-GAN	**T**reatment **P**lanning with **G**enerative **A**dversarial **N**etworks
ITPN	**I**ntelligent **T**reatment **P**lanner **N**etwork
LT	**L**inkage **T**ree
FOS	**F**amily **O**f **S**ubsets
GOM	**G**ene pool **O**ptimal **M**ixing
DC	**D**ose **C**alculation
LUMC	**L**eiden **U**niversity **M**edical **C**enter
LCI	**L**east **C**overage **I**ndex
LSI	**L**east **S**paring **I**ndex
LAI	**L**east **A**dded **I**ndex
GC	**G**olden **C**orner
PRNG	**P**seudo-**R**andom **N**umber **G**enerators
CNG	**C**haotic **N**umber **G**enerators
CIGAR	**C**ase-**I**njected **G**enetic **A**lgo**R**ithm
GA	**G**enetic **A**lgorithm
DICOM	**D**igital **I**maging and **CO**mmunications in **M**edicine
FNN	**F**eedforward **N**eural **N**etwork
eFNN	**e**nsembled **F**eedforward **N**eural **N**etwork
ExpInt	**Exp**onential search and **Int**erpolation
ReLU	**Re**ctified **L**inear **U**nit
CV	**C**ross **V**alidation
Init	**Init**ialization
IQR	**I**nter-**Q**uartile **R**ange

Chapter 1

Introduction

According to the most recent data from the World Health Organization, cervical cancer was registered as the 4th most commonly diagnosed type of cancer in women[1] and the number of new incidences is estimated to rise by 46 % by 2045[2]. Continuous improvement of current radiation treatment methods is therefore as relevant as ever and extensive efforts in this field are made to improve survival rate and quality of life for diagnosed patients. High-dose rate brachytherapy (HDR-BT) is one of the developed treatment methods in which the radiation source is directly placed in, or close to, the cancerous tissue. In comparison to conventional external beam radiation therapy (EBRT), HDR-BT allows for a more localized radiation dose delivery. In cervical cancer HDR-BT treatment, radiation is delivered using an applicator, needles, or a combination of both by using a hardware device as shown in Figure 1.1.

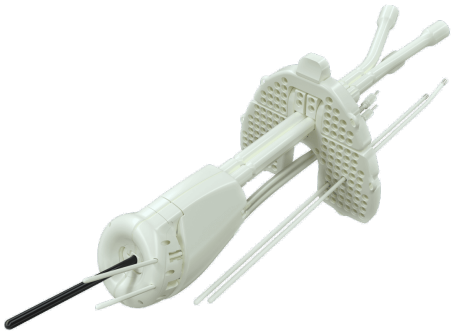


FIGURE 1.1: Oncentra Venezia Elekta - HDR-BT applicator and needles

[1] https://gco.iarc.who.int/today/en/dataviz/pie?mode=cancer&types=0&sexes=0&populations=900

[2] https://gco.iarc.who.int/tomorrow/en/dataviz/tables?mode=cancer&group_populations=1

The applicator consists of a tandem tube that is inserted within the uterine cavity and a pair of ovoids that are placed within the vaginal cavity. Depending on the patient's anatomy, additional interstitial needles are inserted through the applicator to obtain more radiation coverage if necessary. Both the applicator and needles contain so-called dwell positions through which the radiation is delivered. When designing a HDR-BT treatment plan using catheters (applicator and/or interstitial needles), the goal is to optimize the dwell time for each dwell position i.e. how many seconds the radioactive source is held stationary at that particular dwell position. The radioactive source is guided through the catheters and stopped at the dwell position with the use of an afterloader.

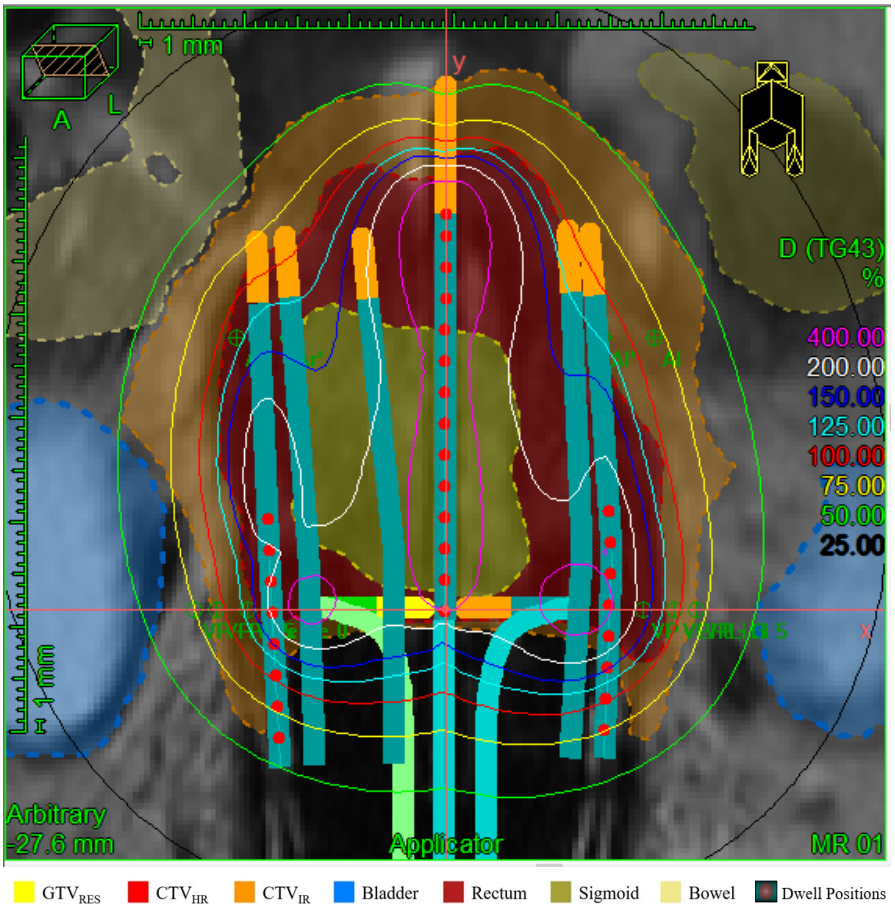


FIGURE 1.2: Sample Visualization of an inserted HDR-BT implant

In clinical practice, a patient approved for cervix HDR-BT treatment will undergo medical imaging (MRI or CT scan) for pre-planning. At this stage, the insertion positions of the applicator and optional needles are determined. After insertion, a second round of medical imaging is performed. The positions of both the catheters are reconstructed, and the Clinical Target Volumes (CTVs) and Organs At Risk (OARs) are delineated manually by the clinician.

The main target volumes for cervical HDR-BT are the high-risk clinical target volume

(CTV_{HR}), intermediate-risk clinical target volume (CTV_{IR}) and the gross tumor volume residue (GTV_{res}). The main OARs in cervical HDR-BT are the bladder, rectum, sigmoid, bowel and ICRU [3] rectovaginal point. For an exemplary figure of the target volume and dwell positions, kindly refer to Figure 1.2.

After delineation, a custom-designed treatment plan can be made. The design of a treatment plan should adhere to a clinical protocol, which describes a set of rules a treatment plan should follow. The global protocol standard for cervical cancer HDR-BT is defined as the EMBRACE-II protocol[4], which can be translated to 2 main objectives: a coverage objective and a sparing objective. The coverage objective aims to maximize the delivered dosage in the CTVs or to maximize the CTV receiving a dosage. The sparing objective aims to minimize the dosage delivered to OAR volumes or to minimize the OAR volume receiving a dosage. These objectives are constructed upon dose volume indicators (DVIs), which describe in more detail the effect of a dwell time configuration on a particular region of interest.

Currently, a treatment plan is made manually by the clinician by iteratively adjusting the dwell times of a plan using interactive software until the plan fits all the desired requirements. Because this manual planning process is time-consuming, inverse planning algorithms have been developed over the years to automate this procedure. One such inverse planning method is BRIGHT (BRachytherapy via artificially Intelligent GOMEA-Heuristic-based Treatment planning). BRIGHT uses the Multi-Objective Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm (MO-RV-GOMEA)[1] to solve a multi-objective problem formulation for HDR-BT. It does so efficiently by utilizing GPU-parallelization [2] and partial evaluations [3]. MO-RV-GOMEA and its integration in BRIGHT have been demonstrated successfully in optimizing HDR-BT prostate treatment plans[4–6], and has led to clinical adoption for prostate cancer treatment [7].

To investigate if this success can also be achieved for different types of cancer, the application of MO-RV-GOMEA optimization to cervical HDR-BT treatment planning is currently being researched. This has resulted in work suggesting that the EMBRACE-II protocol is to be complemented with additional aims to reach more clinically desirable treatment plan solutions [8]. More recently, a formalization of these additional aims represented as a third objective was introduced [9]. Adding additional aims and objectives can improve the quality of a treatment plan but could contribute to a more complex optimization task. This could potentially lead to an increase in runtime required by MO-RV-GOMEA to generate a treatment plan that meets the protocol requirements.

[3] https://www.icru.org/
[4] https://www.embracestudy.dk/Public/Default.aspx?main=1&sub=3&embrace=embrace

In the context of our project, a treatment plan solution is considered to meet the minimal clinical requirements if it is evaluated inside the so-called 'golden corner', which is more precisely formulated in section 2.3. Since HDR-BT treatment planning is to be performed within a limited time frame, a solution is deemed necessary to reduce the optimization time required by MO-RV-GOMEA such that the additional/future aims can be incorporated without violating time constraints.

One component of MO-RV-GOMEA for HDR-BT that has not yet been subjected to extensive research is the initialization of candidate treatment plans. Currently, an initialized treatment plan is built on the heuristic of setting dwell times within a range of [0.0, 0.5] seconds in the needles and a range of [0.5,2.0] seconds in the applicator. This results in an initialization of treatment plans that evaluate far outside the golden corner. Since every patient's treatment implies different optimization considerations and the initialization is an essential part of MO-RV-GOMEA, it is desirable to find initial treatment plans that are more tailored to fit both the patient and MO-RV-GOMEA.

Current advances in Machine Learning (ML) for treatment planning have addressed the relevance of using historical treatment data for future planning purposes. Moreover, machine learning predictions at inference only take limited time and therefore have the potential to dramatically reduce treatment planning time.
Since the implementation of MO-RV-GOMEA for cervical brachytherapy, a range of retrospectively optimized treatment plans has been collected. This data offers the possibility to use a (supervised) ML approach towards modeling the treatment plans. Instead of starting from a heuristic, there is now a possibility to 'learn' from previously generated treatment plans. Therefore, in this thesis, we aim to design such a learned initialization for MO-RV-GOMEA that reduces the runtime required to reach a clinically viable plan for cervical cancer brachytherapy treatment. We refer to this newly designed initialization as a 'warm-start' initialization. Clinically viable is defined as reaching the golden corner. Specifically, we aim to answer the following research question:

Does warm-start initialization reduce the optimization time required for MO-RV-GOMEA to reach a clinically viable treatment plan solution in cervical cancer HDR-BT treatment planning?

The first objective in determining a robust warm-start initialization is defining the input of our predictive model. We will use historical treatment data and brachytherapy literature from the cervical HDR-BT domain to identify important features for prediction. By using sequential backward selection and data analysis, a set of features is determined. The second objective is to define a feedforward neural network architecture via grid search that can output a complete treatment plan for a patient fraction by predicting

one dwell time at a time. Subsequently, our third objective is to correct the eFNN model predictions by using exponential search and interpolation such that MO-RV-GOMEA is provided with a consistent starting point. Finally, to answer our main research question, multiple warm-start initialization schemes are designed and bench-marked against the currently implemented cold-start initialization for MO-RV-GOMEA.

This thesis is structured as follows. Chapter 2 contains a literature review that includes an overview of the current optimization methods used for HDR brachytherapy and key aspects of MO-RV-GOMEA and EA initialization. Subsequently, ML-based optimization for radiotherapy planning is discussed and brought into scope with the proposed research project. Chapter 3 contains an overview of the methods implemented, including a complete view of the pipeline designed for this project. Chapter 4 is split by experimental setup and reports on the main results. Chapter 5 provides a discussion on the methods and results and chapter 6 includes the conclusion and future work suggestions.

Chapter 2

Literature review

Over the past decade, major efforts have been made to optimize clinical treatment plans for (cervical) cancer. Solely within the scope of optimizing radiation HDR-BT plans a wide variety of optimization methods are proposed. We aim to provide a compact overview of the currently published HDR-BT model approaches and discuss the potential and challenges of combining MO-RV-GOMEA with machine learning initialization. Section 2.1 contains an overview of different objective models and section 2.2 provides an analysis of current machine learning approaches for radiation therapy treatment planning. This is followed up by information on MO-RV-GOMEA and its application to HDR-BT in section 2.3. Section 2.4 elaborates on current EA initialization paradigms and the cold start initialization currently implemented for MO-RV-GOMEA cervical HDR-BT treatment planning. In section 2.5 a brief summary is given of the current work and challenges our application entails.

2.1 Current objective formulations and model types

According to review material from the past decade [10, 11], quantitative models applied to HDR-BT problems have been researched since the beginning of the 90's [12]. Concerning their objectives, an important distinction to make is whether the model is optimizing dwell times or catheter placement, or a combination of both. Even though catheter placement is a crucial component in HDR-BT, we will only focus on dwell time optimization and distinguish between applied methods where the catheter position is given as-is. For maintaining structure in this section, we distinguish between different model paradigms as proposed in [11].

A general framework for optimizing treatment plans consists of the objective function(s) subjected to additional constraints. The objective function(s) is/are based on a desired

outcome of coverage and sparing indices backed up by a clinical protocol. Although radiology treatment planning will always contain a multi-objective trade-off between coverage and sparing, the problem is regularly simplified to a single objective function. One way to model this is by using a linear penalty model (LPM), first introduced in [13], in which a penalty is calculated for a dose point if the dosage is outside a desired range. The summed penalty for a certain structure, e.g. a CTV, is then calculated with the aim to minimize the cumulated penalty. Common optimization methods used in LPMs are LP-solvers [14, 15] or simulated annealing [13, 16]. Closely related to LPMs, are quadratic penalty models (QPMs). Unlike LPMs, QPMs have a quadratic penalty objective function. This results in quantitatively larger penalty values for outlying dwell time setups. Example solvers used for dwell time optimization in QPMs are the Broyden–Fletcher–Goldfarb–Shanno algorithm [17, 18] (note, only unconstrained optimization) or interior-point solvers [19, 20]. Although LPMs and QPMs allow for real-value optimization of dwell times, dwell time solutions were proven to be heterogeneous on occasion [21]. This poses the potential problem of hotspots, [15] where some dwell positions in the catheter have substantially larger dwell times than neighboring dwell positions. As the catheter is placed in the dynamic anatomy of a patient, there is a possibility that a hotspot might shift outside the target area and potentially do harm to a patient. To overcome this risk, research was done on adding additional constraints that contribute to more homogeneous dwell times [22–24]. Current work often refers to this concept as the dwell time modulation restriction (DTMR)

A second downside of penalty models is that the optimized objectives do not directly represent the clinical viability. The penalty objective might be well optimized, but dose indices for protocol-mentioned OARs and CTVs could still indicate the plan is not suitable for practical application [25]. Dose-volume models (DVM) consider a different optimization objective, more closely related to the clinical protocol indices. In DVMs, the objective formulation is defined as a mixed integer problem where the goal is to maximize the number of dose points obtaining a prescription dosage (or minimize the number of dose points obtaining the maximum dosage). E.g., if 9 out of 10 dose points in a CTV receive the prescribed (100%) dosage, V_{100}^{CTV} would equal 90%. A disadvantage of DVMs lies in the discretization of dosage goals. For example, a single objective DVM might easily reach 100% coverage with a certain dosage, while 100% coverage could also be reached with less dosage (and potentially spare a patient's OAR's better). Example implementation of DVM's use simulated annealing [26, 27] or a linear solver such as CPLEX [28]. A more precise method is the mean-tail-dose model (MTD) for which the goal is to minimize the mean outlier dosages for both lower (coverage) and upper (sparing) bounds. E.g. in a coverage objective situation, if 10% of our dose points have a mean dosage of 4Gy and the protocol prescription states 5Gy, the resulting aim is then

to increase the mean dosage of the lowest 10% dose points. As the objective of MTDs is formulated linearly (no separate penalty function however), current work uses linear solvers such as CPLEX [29] or Gurobi [30].

The aforementioned applications of LPMs, QPMs, DVMs and MTDs all handle problems that are formulated as single objective functions. Adding additional constraints or accumulating terms in a single objective function can result in better plans. However, this will impose further complexity in explaining how the quality of a treatment plan is measured since this is based on a single outcome value. A resolution to increase explainability lies in adhering to the multi-objective nature of HDR-BT. For example, by defining a coverage vs. sparing trade-off by separating the objective function into two parts. For solving such a multi-objective problem formulation, Evolutionary Algorithms (EAs) have proven to be particularly well-suited [4, 31, 32]. Another advantage of EAs is that they provide a range of possible solutions at each iteration, meaning a medical expert in practice can have a choice between plans with different objective trade-offs. In this thesis, we focus on and build upon MO-RV-GOMEA[1], a proven-in-practice EA for bi-objective HDR-BT. For a detailed description of the implementation, kindly refer to sections 2.3-2.4 and chapter 3.

2.2 Machine learning for radiotherapy treatment planning

An extensive range of modeling approaches for brachytherapy treatment optimization exist. However, methods using machine learning have not until recent been adopted. Few scientific contributions have discussed machine learning applications to (cervical) HDR-BT treatment planning. We therefore include external beam radiation therapy (EBRT) in our overview as it contains insightful model choices relative to our approach.

Supervised ML-driven treatment planning needs historical (or synthetic) data to learn an output representation for a treatment plan. One way to do so is by predicting dose volume histograms (DVH) using segmented images of CTVs and OARs. This method makes use of a U-net[33], which is a convolutional neural network originally designed for image segmentation. It was shown that a U-net can also be applied in dose-volume prediction for EBRT applications (DoseNet [34]). However, U-net applications for HDR-BT pose a more complex problem in comparison to EBRT applications, since the catheters are also to be segmented. During training, a U-net model inputs the historical treatment DVH, segmented catheters and segmented OARs/CTVs. At inference, a U-net model is inputted with only the segmented catheters and OARs/CTVs. The U-net output is a 2D or 3D prediction of the DVH. An example of a U-net applied to HDR-BT treatment exists[35], where an additional transformation is purposed to handle the high gradient

properties of radiation dose near the applicator. These high gradients can dominate the loss function during training, potentially leading to the neglection of important DVIs further away from the applicator. This insight is also adopted in the choice of our proposed (Huber)-loss function in section 3, e.g. by assuring that few high-peak dwell times will not dominate the loss performance for a (still relatively good) treatment plan.

Another example of U-net DVH predictions for HDR-BT is RapidBrachyDL [36]. RapidBrachyDL is a deep CNN model that complements a U-net with 2 components. One component is a 2-layer CNN pre-trained on Monte Carlo-simulated data, proposed as a smart initialization that handles dwell point information. The second component is a FiLM generator [37] that conditions the U-net to predict a DVH that adheres to the patient geometry input. In contrast to the U-net architecture that uses only a generator loss, a generative adversarial network (GAN) as proposed in [38] uses an additional discriminator loss to train on EBRT plans. Closely related is TP-GAN [39], which was recently introduced to reduce the overall treatment planning time in low-dose-rate brachytherapy for prostate cancer. TP-GAN is a concrete example of an ML-based solution that explicitly mentions the reduction of planning time. However, it uses a fixed-needle template, which is not directly applicable to the dynamic catheter setting in our cervical HDR-BT approach.

DVH predictions can also be obtained by using an input structure that uses informative features instead of (segmented) images [40]. For this method, a regression model is proposed to predict a DVH kernel according to 3D distance features. The regression model is based on proven ERBT implementations and only contains features regarding the CTV's and OAR's. Though this method offers robustness as it can handle different types of catheters, the results are only shown for intracavitary cervical HDR-BT treatment with 1 catheter.

Despite DVH prediction shows clinical potential, it introduces a practical problem. Its representation is still to be converted to a treatment planning structure that is interpretable by the radiation equipment. More specifically, dwell times are still to be set at dwell point locations such that they are in line with a DVH prediction. This disadvantage is not present in machine learning that directly optimizes dwell times. One model approach that closely mimics the manual planning procedure can be achieved using deep reinforcement learning as proposed in [41]. This approach (referred to as ITPN) aims to maximize its objective value function by adjusting dwell time states by either increasing, decreasing or keeping the dwell times the same. Although no explicit speed-ups in planning time were reported, ITPN showed an improvement in multiple sparing objectives.

2.3 MO-RV-GOMEA for HDR-BT

In HDR-BT treatment planning, the goal is to generate a treatment plan that meets the protocol aims and is generated within a limited time frame. Although protocol aims can be encapsulated into one objective function, the optimization problem of HDR-BT is inherently multi-objective due to the sparing/coverage trade-off involved. The Multi-Objective Real-Valued Gene-Pool Optimal Mixing Evolutionary Algorithm (MO-RV-GOMEA)[1] is particularly well at handling such an optimization problem. A key characteristic of dwell time optimization for HDR-BT is the strong underlying dependencies between dwell positions. Dwell positions close to each other have a higher dependency than two distanced dwell positions. MO-RV-GOMEA uses a linkage tree (LT) [42] to model these dependencies. The linkage tree is created using bottom-up hierarchical clustering, starting with a univariate set of individual variables (in our case, dwell positions) and ends up with a joint set of all variables. All the nodes and leaves in the LT form a family of subsets (FOS). For each subset in the FOS with a size > 5, a multivariate Gaussian distribution is created which is used to sample from when applying the Gene-pool Optimal Mixing operator (GOM)[43]. If partial variation results in a better solution (in our case, a treatment plan), the mixing operation is performed otherwise the current solution is kept. This process is repeated for several generations, where in each generation a set of the best solutions is updated (elitist archive). To account for the large number of evaluations needed by the GOM-operator, partial evaluations [3] are used to efficiently re-evaluate solutions.

Objective DVIs are calculated based on dose calculation (DC) points, of which the locations are sampled within regions of interest and set before EA optimization. When changes are made to the dwell times of a treatment plan, only the changed dwell times cause a change in the objective DVI result. Thus, only partial evaluations of the complete treatment plan solution are necessary to update the complete coverage/sparing objective value. In recent years, MO-RV-GOMEA runtime has been further accelerated using program parallelization and GPUs [44, 45]. In addition to the GOM-operator and partial evaluations, 3 algorithms are implemented within MO-RV-GOMEA that further improve the output solutions. These operators are all implemented at cluster-level, at which each cluster represents a sub-range of treatment plan solutions based on their sparing/coverage trade-off. For each cluster, adaptive variance scaling is applied during optimization to account for the variance-diminishing effect of (EA) selection operations. Additionally, adaptive mean shifting is applied to steer the optimization more toward the direction of greatest improvements. Finally, a forced improvement procedure is applied. This procedure is used to steer solutions, that have not yielded a non-dominated

Volume	Use	Coverage Criteria	Sparing Criteria	Added Aims
CTV_{HR}	Target	$D_{90\%} > 7.8Gy$ $(> 7.1Gy)$	$D_{90\%} < 8.3Gy$	$V_{100\%} > 99.9\%$
		$D_{98\%} > 5.8Gy$	-	-
CTV_{IR}	Target	$D_{98\%} > 3.5Gy$	-	$V_{50\%} > 99.9\%$
GTV_{RES}	Target	$D_{98\%} > 8.3Gy$ $(> 7.8Gy)$	-	-
Bladder	OAR	-	$D_{2cm^3} < 5.5Gy$ $(< 6.3Gy)$	-
Rectum	OAR	-	$D_{2cm^3} < 4.0Gy$ $(< 6.3Gy)$	-
Sigmoid	OAR	-	$D_{2cm^3} < 4.5Gy$ $(< 6.3Gy)$	-
Bowel	OAR	-	$D_{2cm^3} < 4.5Gy$ $(< 6.3Gy)$	-
Recto-vaginal point	OAR	-	$D_{point} < 4.0Gy$ $(< 6.3Gy)$	-
Mid-CTV_{IR}	Target	-	-	$V_{100\%} < 25\%$
Mid-normal-tissue	OAR	-	-	$V_{100\%} < 0.1\%$
Top-normal-tissue	OAR	-	-	$V_{100\%} < 0.2\%$
Core-CTV_{HR}	Target	-	-	$V_{200\%} > 99.5\%$

TABLE 2.1: Complete set of HDR-BT aims for cervical cancer treatment (radiation dose limits are given between brackets)

solution for a pre-determined number of generations, more toward current elitist solutions. In BRIGHT (and in this thesis project), the forced improvement procedure is not applied.

The objective functions for cervical HDR-BT, presented in this thesis, are based on the EMBRACE-II protocol[46], additional aims that are proposed in [9] and one newly introduced Core-CTV_{HR} that is purposed for additional coverage in the center of the CTV_{HR}. Since this combination of aims is currently under research at Leiden University Medical Center (LUMC), we refer to it as the LUMC protocol. DVI's in the LUMC protocol are stated in three different formulations. The first formulation states the minimum dosage received by the most irradiated subvolume, i.e. $D_{90\%} > 5.6Gy$ for CTV_{HR} states that 90% of the most irradiated subvolume of the CTV_{HR} should at least receive 5.6 grays. A second formulation is $D_{point} < ..Gy$, where one reference point is used instead of a subvolume. The final formulation denotes how much dosage a subvolume of a CTV or OAR should receive. E.g. $V_{100\%}^{CTV_{HR}} > 90\%$ states that 90% of the CTV_{HR} volume should at least receive 100% of the prescribed dosage. All percentages describe the fraction of the planned dosage, which is set at 7 Gy for cervical cancer HDR-BT. Kindly refer to Table 2.1 for an overview of the LUMC protocol for cervical cancer HDR-BT.

The coverage, sparing and additional objectives for a treatment plan t are formulated respectively as the Least Coverage Index ($LCI_w(t)$), Least Sparing Index ($LSI_w(t)$) and Least Added Index ($LAI_w(t)$) denoted in equations 2.1, 2.2, 2.3.

$$LCI_w(t) = w_c(\delta_c(D_{98\%}^{CTV_{HR}})) + w_c(\delta_c(D_{90\%}^{CTV_{HR}})) + w_c(\delta_c(D_{98\%}^{CTV_{IR}})) + w_c(\delta_c(D_{98\%}^{GTV_{RES}})) \tag{2.1}$$

$$LSI_w(t) = w_s(\delta_s(D_{90\%}^{CTV_{HR}})) + w_s(\delta_s(D_{2cm^3}^{\text{Bladder}})) + w_s(\delta_s(D_{2cm^3}^{\text{Rectum}})) + w_s(\delta_s(D_{point}^{\text{ICRU-Rectovaginal}})) + w_s(\delta_s(D_{2cm^3}^{\text{Sigmoid}})) + w_s(\delta_s(D_{2cm^3}^{\text{Bowel}})) \tag{2.2}$$

$$LAI_w(t) = w_a(\delta_a(V_{100\%}^{CTV_{HR}})) + w_a(\delta_a(V_{50\%}^{CTV_{IR}})) + w_a(\delta_a(V_{100\%}^{Mid-CTV_{IR}})) + w_a(\delta_a(V_{100\%}^{\text{Mid-normal-tissue}})) + w_a(\delta_a(V_{100\%}^{\text{Top-normal-tissue}})) + w_a(\delta_a(V_{100\%}^{\text{Core-}CTV_{HR}})) \tag{2.3}$$

Key components in this formulation are as follows. δ_c (coverage), δ_s (sparing) and δ_a (added) denote functions for the normalized differences between the volume/dose index and their corresponding aims. w_c (coverage), w_s (sparing) and w_a (added) are weights that are set to provide order in how much attention an aim should get. The least violated aim obtains the smallest weight w, the second to least violated aim receives an exponentially larger weight and so forth. Using the weighted scheme, the most violated aim receives the most attention. If the most violated aim is not improved further, optimization can continue on less violated aims that have smaller weights. For a detailed description of the objective functions and how they are calculated, we refer to [9] and the included supplementary material.

Before the weighted scheme was introduced, the LCI(t) and LSI(t) performance were described as the most violated DVI [5]. For example, by defining a least sparing index with n DVI's: $LSI(t) = min\{\delta(DVI_1), \delta(DVI_2), ..\delta(DVI_n)\}$. A similar example can be applied to the LCI and LAI. To determine whether a plan adheres to the clinical protocol, the unweighted version is considered sufficient. More precisely, all protocol aims are reached if a solution satisfies $LCI(t) > 0$, $LSI(t) > 0$ and $LAI(t) > 0$.
In this thesis, we primarily evaluate plans on the LCI(t) and LSI(t) conditions as they are still the most used method to describe the quality of a treatment plan. When these conditions are met, we define a treatment plan solution to be located in the so-called 2-D 'Golden Corner' (GC). Treatment plans in- or outside the golden corner are commonly visualized as part of an approximation front of treatment plans as exemplified in Figure 2.1.

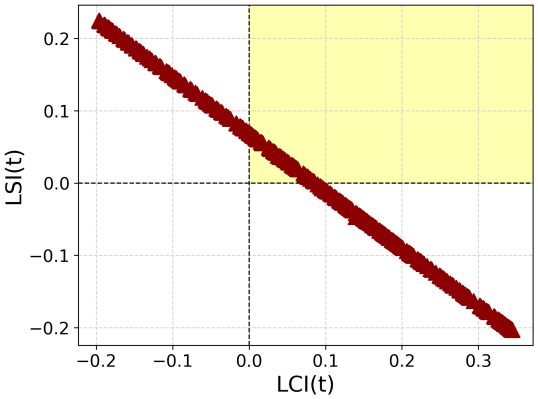


FIGURE 2.1: Sample Approximation Front

Besides the main objectives, 2 additional constraints are implemented to adhere to more clinically desirable treatment plan compositions. One is the DTMR, which regulates the difference in dwell time between most-closely neighbored dwell points (in terms of Euclidian distance). This is set to be maximal $m(t) = \frac{2}{5+t}$ for a dwell time t. A second constraint describes that only 40% of the total given dwell time should originate from the needle dwell points.

2.4 (MO-RV-GOM)EA initialization

How EAs are initialized has an arguable effect on their optimization process. For EA initialization we can distinguish between whether we model the initialization application-specific or apply a generic approach. If a generic approach is applied, the main objective is to generate a population of candidate solutions that provide an unbiased, uniform starting point. More specifically, this means we do not make any assumptions about the initial configuration regarding its decision variables beforehand. In this case, one can either use stochastic or deterministic methods to generate an initial population[47]. Widely used stochastic initializations are Pseudo-Random-Number-Generators (PRNGs) such as the Mersenne Twister [48] or Chaotic-Number-Generators (CNGs) that apply a chaotic map iteratively to generate values. Furthermore, deterministic methods such as Halton [49], Sobol [50] or Orthogonal sampling [51] are also widely adopted for EA initialization purposes.

Although extensive research is done on the effect of initializations in [52–54], there is no single method that outperforms other methods in general. Some EAs are more sensitive to initialization schemes whereas other EA's might not be so much. Yet most importantly, the aforementioned comparative studies address that well-performing initializations contain some dependency between the initialization and the applied problem.

Hence, it suggests that exploiting domain-specific knowledge of the problem is highly relevant. One way of incorporating that knowledge is by using heuristics. For example, [55] discusses initialization strategies for a job scheduling problem such as ordering jobs by the expected number of operations a job has.

Another way of incorporating domain-specific knowledge is by using a machine learning (ML) model. One example of ML initialization for genetic algorithms (GA) is the Case-Injected Genetic AlgoRithm (CIGAR [56]). CIGAR keeps track of past problem-solving attempts and periodically injects itself with intermediate solutions that are similar to previously solved problems. Another example is the hybrid genetic algorithm proposed in [57], applied to a power economic dispatching problem. A Hopfield neural network, trained on previous optimization solutions, was used to generate a part of the initial population. As ML-initialization for EAs shows potential for solving real-life problems more efficiently, we will further extend this concept for the initialization of HDR-BT treatment plans.

Algorithm 1: Cold Start Initialization

// initial population $\mathcal{P}_{init}$ size n; treatment plan $\boldsymbol{x}$ size m; LowerBound Init Applicator LB_{app}; UpperBound Init Applicator UB_{app}; LowerBound Init Needles $LB_{needles}$; UpperBound Init Needles $UB_{needles}$;

$\mathcal{P}_{init} = \{\}$;
$LB_{app} \leftarrow 0.5$;
$UB_{app} \leftarrow 2.0$;
$LB_{needles} \leftarrow 0.0$;
$UB_{needles} \leftarrow 0.5$;
for $i \in \{0, 1, .., n-1\}$ **do**
    $\mathbf{x} = \{\}$;
    for $j \in \{0, 1, .., m-1\}$ **do**
        $dwellpoint \leftarrow GetDwellPointInformation()$;
        $z \sim \mathcal{U}(0, 1)$;
        if *dwellpoint is needle dwellpoint* **then**
            $dwelltime \leftarrow LB_{needles} + (UB_{needles} - LB_{needles}) * z$;
        else
            $dwelltime \leftarrow LB_{app} + (UB_{app} - LB_{app}) * z$;
        $\mathbf{x} \leftarrow \mathbf{x} \cup \{dwelltime\}$;
    $\mathcal{P}_{init} \leftarrow \mathcal{P}_{init} \cup \{\mathbf{x}\}$;

MO-RV-GOMEA is well suited for HDR-BT treatment plan optimization. However, its initialization of candidate treatment solutions is still based upon the heuristic of setting the dwell times within a pre-defined range multiplied with uniform noise. More specifically, the current initialization is denoted in Algorithm 1. We will refer to this initialization as the 'cold start' initialization throughout this thesis. For cold start

initialization, lower and upper bound ranges for the dwell times are set to be [0.5, 2.0] in the applicators and [0.0, 0.5] in the interstitial needles.

2.5 Summary & challenges

Our overview shows that extensive efforts are made to define a good objective function for HDR-BT inverse treatment planning. Various optimization methods exist of which GA's and specifically MO-RV-GOMEA proved well for solving the problem multi-objectively. ML applications have not been adopted widely yet for HDR-BT, but their use has the potential to decrease runtime dramatically. Thus, investigating a synergy between MO-RV-GOMEA and an ML-based treatment plan initialization is deemed relevant.

The cold start initialization is currently evaluated as a high-sparing plan with small uniformly sampled variations between candidate solutions. This is not a problem in terms of effectivity, since MO-RV-GOMEA is well-equipped to exploit the small variations that result in better performance. It can however pose problems in terms of efficiency. Starting from small dwell times with little variation could require MO-RV-GOMEA to perform many steps before converging to a clinically desirable treatment solution. An improved initialization should therefore aim to create candidate solutions that are closer to the the target solutions and provide some variation which MO-RV-GOMEA can exploit efficiently.

Our approach uses solely retrospectively optimized treatment plan data generated by MO-RV-GOMEA. One reason to use this particular data, instead of using the manually designed treatment plan data from the clinic, is that we aim to create a plan that is specifically designed as input to MO-RV-GOMEA. A second reason is the effortless navigation that is possible through generated MO-RV-GOMEA plans. Instead of building upon a single, manually designed, treatment plan with a given sparing and coverage performance, we can choose from a range of treatment plans and pick the plan with the most desirable coverage/sparing trade-off to train our model on. Thirdly, manually created/adjusted treatment plans can involve some bias, e.g. plans produced by different radiation oncologists can differ slightly, which one would ideally avoid when training a predictive model.

Using automatically generated data poses challenges however, since MO-RV-GOMEA treatment plans do not directly encompass the intuition of a medical expert in contrast to historical manually-created data. For example, a radiation oncologist can have strong argumentation on why certain adjustments in the treatment plan are necessary, which could be translated in features to train the model on. For MO-RV-GOMEA plans,

however, translating the reason behind changes in the plan is a different task. E.g. after recombination, a plan can be improved in quality. Extracting information on such a change and then deducing a feature from it is seemingly less trivial. Additionally, the HDR-BT treatment planning optimization problem is multi-modal. There exist multiple solutions for a clinically acceptable plan, which makes identifying any pattern in the generated treatment plans less straightforward.

Chapter 3

Methods

3.1 Pre-processing & Feature Engineering

The data used in this project originates from 2016 to 2020 and contains a total of 119 patients with multiple fractions per patient. Patient-fraction information is extracted from DICOM[1]-formatted RT-struct and RT-plan files. The RT-struct files represent primarily OAR and CTV contours, whereas the RT-plan files define information about the applicator and needles. Target treatment plans are generated by using the BRIGHT[7] software solution which incorporates MO-RV-GOMEA for brachytherapy. The target treatment data was generated using the LUMC protocol excluding the Core-CTV_{HR} dose indicator. Since BRIGHT outputs an approximation front of non-dominated treatment plan solutions, we select the most balanced plan in terms of coverage and sparing, the so-called 'L-plan', for modeling. Selecting the treatment L-plan is formalized as:

$$\text{L-Plan} = \underset{t \in \, treatment\ plans}{argmax} \{min\{LCI(t), LSI(t)\}\} \tag{3.1}$$

Each selected treatment plan consists of dwell times that are corrected for the radioactive source strength at the date and time of the treatment. To account for the varying radioactive source strength, we inverse-transform the dwell times to dwell weights using a conversion factor.

Features are extracted from the DICOM data and are constructed on a dwell point level. For a visualization of the data extraction, preprocessing and eFNN training pipeline, kindly refer to Figure 3.1. In practice, rt-plan and rt-struct files encompass more target and risk markers than described by the protocol. Nevertheless, since we aim for a generalizable approach we will omit any markers that are not consistently given for each

[1] https://www.dicomstandard.org/

patient. If either the DICOM data or BRIGHT data is incomplete for a patient fraction, we remove the patient fraction from the dataset. For an overview of the designed features refer to Table 3.1. Refer to A for an exploratory analysis of the patient data.

Feature Category	**Feature**	**Feature configuration for x**	**Feature Type**
A	Catheter Type	-	$\mathbb{Z}$
A	No. Neighbors In Radius x	$x \in \{5.0, 7.5, 10.0, 12.5\}$	$\mathbb{Z}$
A	Centroid Distance x	$x \in \{CTV_{HR}, CTV_{IR}, GTV_{RES}\}$	$\mathbb{R}$
A	Closest Contour Distance x	$x \in \{CTV_{HR}, CTV_{IR}, GTV_{RES}, Bladder, Rectum, Bowel, Sigmoid\}$	$\mathbb{R}$
A	Furthest Contour Distance x	$x \in \{CTV_{HR}, CTV_{IR}, GTV_{RES}\}$	$\mathbb{R}$
A	Is Inside ROI Samples Count x	$x \in \{CTV_{HR}, CTV_{IR}, GTV_{RES}\}$	$\mathbb{Z}$
A	Superior Contour Distance x	$x \in \{CTV_{HR}, CTV_{IR}, GTV_{RES}\}$	$\mathbb{R}$
A	Inferior Contour Distance x	$x \in \{CTV_{HR}, CTV_{IR}, GTV_{RES}\}$	$\mathbb{R}$
A	Lateral Left Contour Distance x	$x \in \{CTV_{HR}, CTV_{IR}, GTV_{RES}\}$	$\mathbb{R}$
A	Lateral Right Contour Distance x	$x \in \{CTV_{HR}, CTV_{IR}, GTV_{RES}\}$	$\mathbb{R}$
A	Anterior Contour Distance x	$x \in \{CTV_{HR}, CTV_{IR}, GTV_{RES}\}$	$\mathbb{R}$
A	Posterior Contour Distance x	$x \in \{CTV_{HR}, CTV_{IR}, GTV_{RES}\}$	$\mathbb{R}$
B	Right Neighbor1 x	$x \in$ {Features Category A}	Feature Category A Type
B	Right Neighbor2 x	$x \in$ {Features Category A}	Feature Category A Type
B	Right Neighbor3 x	$x \in$ {Features Category A}	Feature Category A Type
B	Left Neighbor1 x	$x \in$ {Features Category A}	Feature Category A Type
B	Left Neighbor2 x	$x \in$ {Features Category A}	Feature Category A Type
B	Left Neighbor3 x	$x \in$ {Features Category A}	Feature Category A Type

TABLE 3.1: Dwell-point level features

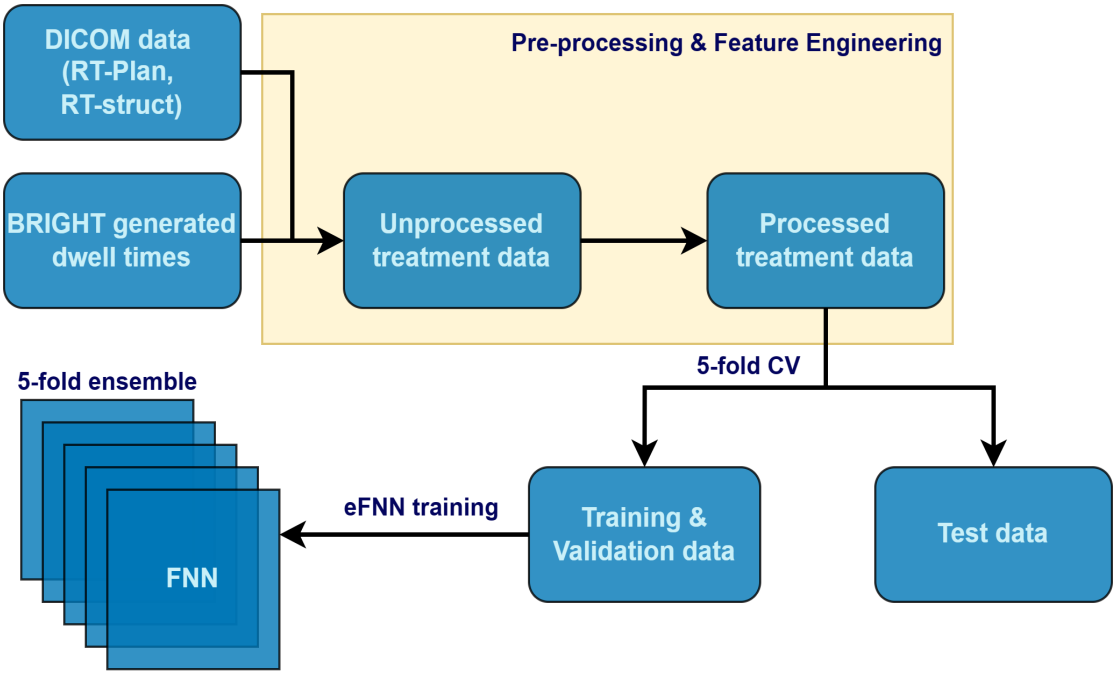


FIGURE 3.1: Data pre-processing and training pipeline

We define 3 catheter types to distinguish in which hardware module the dwell position is located. That is, either in the tandem (value = 1), ovoids(value = 2) or needles (value

= 3). The cold start initialization is based on a clinical heuristic to avoid major use of the needles if possible, therefore a feature to distinguish the catheter type is assumed to be relevant. Modelling the catheter surroundings of a dwell point is done by the No. Neighbors In Radius feature. The 3-dimensional Euclidean distances in millimeters between one dwell point and all other dwell points are calculated, followed by a count of the number of neighboring dwell points that are within the specified radius. Using different radii, we can map dwell point surroundings on different scales.

All the distance features are based on the medical intuition behind manual treatment planning and the workings of the radiotherapy hardware. For example, a dwell point close to the centroid of a target volume is assumed to have a higher dwell weight than a dwell point that is further away from a target volume. Similarly, a dwell point close to the contour of a target volume is assumed to need a higher dwell weight than a dwell point further away from a target volume. In the case of sparing OARs, a dwell point closer to the OARs should have a lower dwell weight than dwell points further away.

Additionally, we add 6 anatomical direction distances to the CTV contours to create a more precise picture of the dwell point positions inside or outside of the CTV. To determine whether a dwell point is inside (1) or outside (0) a CTV or OAR, we apply a bounding box check and the ray-casting algorithm [58]. The bounding box is created based on the minimal and maximal contour bounds of the OAR/CTV contours. If a dwell point is outside the 3-D bounding box, we can save time and return a zero. If the dwell point is inside the bounding box, we apply the ray casting algorithm for each 2-D contour slice and return either a 0 if the point is outside or a 1 if the point is inside the OAR/CTV. The Is Inside CTV Samples Count feature makes use of this by sampling 50 reference points within a specified radius and returning the sum of samples that are inside the CTV. Since the CTVs generally differ in size, different radii parameters are set. We apply a radius of 2.5 mm for the GTV_{RES}, 7.5 mm for HR_{CTV} and 10 mm for IR_{CTV}. The radii parameters are the result of manual tuning, which was done by looking at the resulting Is Inside CTV Samples Count feature for different radii settings. Since the GTV_{RES}, IR_{CTV} and HR_{CTV} are of different sizes, a different radius per target volume was considered more informative for the Is Inside CTV Samples Count feature.

To include whether a dwell point is inside or outside a CTV, we multiply the distance features by -1 if the dwell point is outside and +1 if the dwell point is inside the specified CTV. By using this method, we can exclude an Is Inside CTV feature and include this information directly in the distance features.

The aforementioned reasoning only considers single dwell points. However, multiple dwell points have a cumulative effect on, for instance, an OAR. To provide our model

with some spatial awareness of this effect, neighbor features (type B features) are considered, in addition to the features of the current dwell point, as model input. In other words, this results in 6 additional sets of type A features. The Next and Prev Neighbor features are determined on a catheter level, which means that for each unique catheter ID (e.g. tandem 1 or ovoid 2) only the neighbors for that catheter ID are considered. If the dwell point is positioned on the far left or far right of the specified catheter, it will have only neighbors in either the right or left direction. A dwell point in the center of the catheter will have both right and left-direction neighbors. Non-existing neighbor features are set to zero and existing neighbor features are set according to the feature-A output of that particular neighbor.

As the final step for pre-processing, min-max normalization per patient fraction is applied to all features except the catheter type.

Not all features in Table 3.1 are deemed necessary as input to our model in the final framework. Many combinations of the proposed features are possible and each combination can yield a different eFNN performance. Therefore, redundant features are identified by performing a sequential backward selection experiment in Appendix B. As a result, we exclude the following features from the final feature set: the GTV_{res} and HR_{CTV} anatomical distances, the Sigmoid Closest Contour Distance, the neighbor features for the Is Inside Samples Count GTV_{res} and the IR_{CTV} Centroid Distance.

3.2 FNN Model

We aim to solve the regression task of predicting individual real-valued dwell weights. A feedforward neural network (FNN) is particularly suitable for this purpose. Since FNNs can be designed in all sorts of ways, a systematic analysis of FNN architectures and hyperparameters ought to be important. We accomplish this by performing a grid search experiment which can be referred to in Appendix C.

From the grid search experiment, we deduct a FNN architecture (Model # 2 in Appendix C) as denoted in Figure 3.2. In the backpropagation process, we apply AdamW with a learning rate of $1e^{-5}$ and a weight decay of 0.01. Other parameters for AdamW include the β-range set at (0.9, 0.999) and a stability ϵ of $1e^{-8}$. After the final output neuron, the predicted dwell weight is processed through a Softplus function of which we set the β parameter to 1. The Softplus activation is a smooth approximation of the ReLU function and constrains the output dwell weight to be positive.

Prediction performance is evaluated using mean Huberloss with an input parameter δ set to 1. Huberloss combines the absolute difference with the mean squared error. This

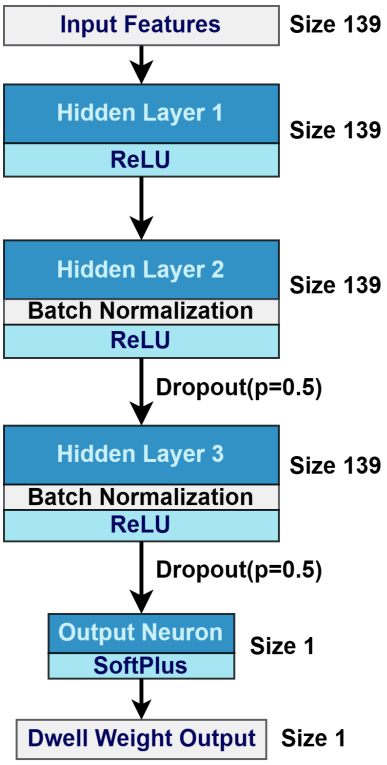


FIGURE 3.2: FNN architecture

combination is particularly useful if we do not want the outliers in our data to dominate the loss, which can result in unstable training given our imbalanced CV-splitted datasets. Errors on outliers, of which the absolute difference is larger than δ, are penalized linearly. Whereas errors within an absolute difference lower than δ are penalized quadratically.

By making use of a 5-fold cross-validation principle, we essentially create an ensemble of 5 FNN models (eFNN), each trained and validated on different sets of patient fractions. Since many different treatment plans exists with a similar coverage and sparing trade-off, an ensemble of 5 models aligns better to the multimodal nature of the problem compared to using only one model prediction.

3.3 Exponential Search & Interpolation Correction

After the eFNN outputs 5 treatment plans, an indication of the plan quality is not available yet. Although a low Huberloss can indicate a close resemblance to the target plan, it is not guaranteed that the plan quality will be close or equal to the target solution. Therefore, it is deemed necessary to evaluate how good the plan actually is and potentially adjust plans that are still far away from the desired plan quality. We aim to accomplish this by performing iterative BRIGHT re-evaluations through Exponential search and Interpolation (ExpInt) correction.

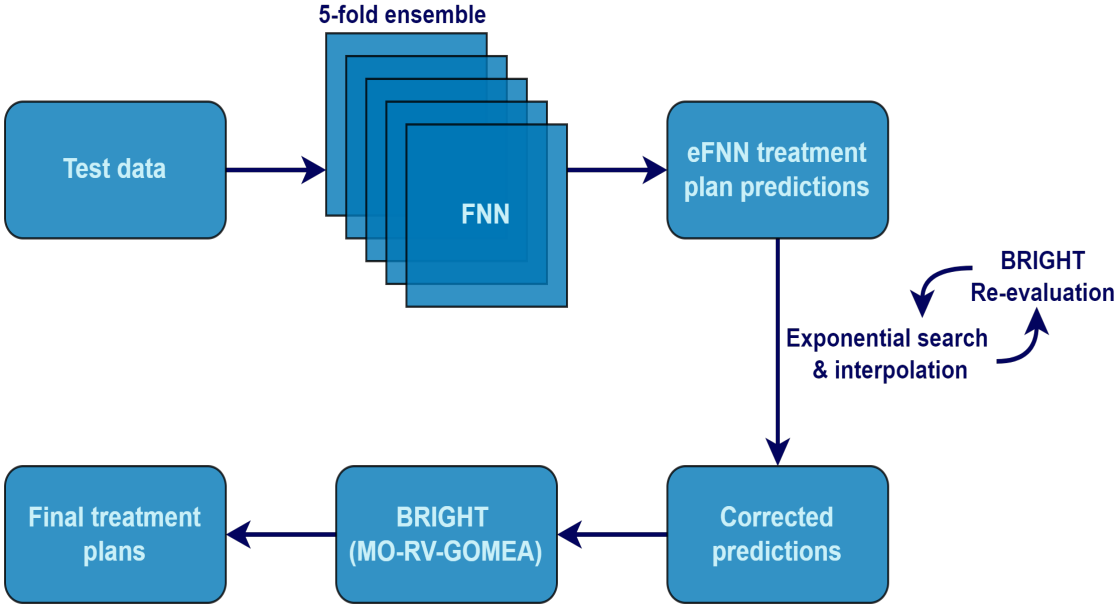


FIGURE 3.3: Pipeline at-inference

To provide a general framework description, we denote the number of models (and the number of original, eFNN-produced treatment plans) as n. n equals the number of cross-validation splits, which in our case equals 5. At inference, an ensemble of n FNNs predicts n HDR-BT treatment plans as denoted in Figure 3.3. To identify the corresponding $LCI(t)$ and $LSI(t)$ for each of the treatment plans, a re-evaluation step is performed using the BRIGHT software solution. Once evaluated, a treatment plan is located in one of the three sections as shown in Figure 3.4. A treatment plan in area A has stronger sparing properties whereas a treatment plan in area C has stronger coverage properties. If a treatment plan is located in area B, it has a more balanced sparing/coverage trade-off. For a warm-start solution, we aim to input MO-RV-GOMEA with a range of balanced treatment plans that all lie between the sparing and coverage line (ideally, inside the golden corner). This is done by performing a (bi)-directional exponential search conditioned on the starting quality of the eFNN-predicted treatment plans. For an overview of the ExpInt procedure, refer to Algorithm 2.

If a treatment plan is located in area A, we perform one search in the direction of Area C crossing first the sparing line, followed by the coverage line. If a treatment plan is located in area C, we perform one search in the direction of Area A crossing first the coverage line, followed by the sparing line. If a treatment plan is located in area B, we perform two searches in opposite directions. The exponential searches return lower and upper bounds, to which we apply interpolation to get a range of treatment plans between the lower (sparing) and upper (coverage) bounds. How many treatment plans are interpolated between the two bounds is defined by an interpolation parameter, which we set to 25 (including the bounds). Subsequently, all resulting treatment plans that

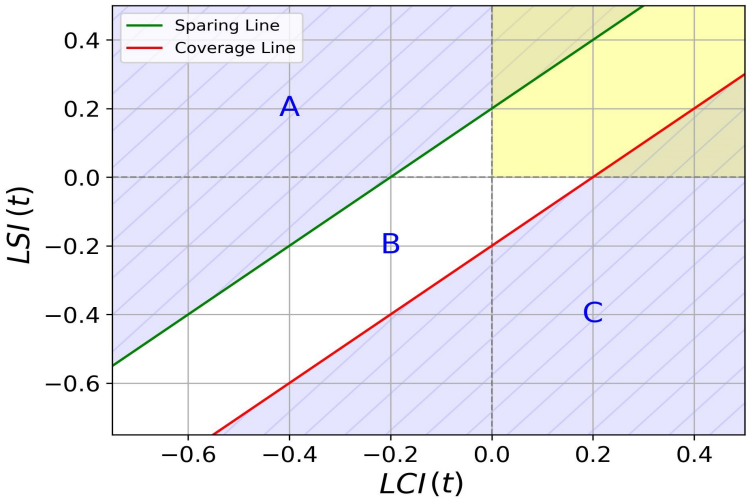


FIGURE 3.4: Evaluation categories treatment plan

Algorithm 2: Exponential Search & Interpolation Correction

// Warm start input population $\mathcal{P}_{warmstart}$, original treatment plan x_j, number of original treatment plans n, direction factor α_j, set of lower bounds LB, set of upper bounds UB

$\mathcal{P}_{warmstart} \leftarrow \{\}$;
for $j \in \{0, 1, .., n-1\}$ **do**
  if x_j ***in*** *Area A* **then**
    $\alpha_j \leftarrow 1$;
    $LineToCross_1 \leftarrow SparingLine$;
    $LineToCross_2 \leftarrow CoverageLine$;
    $LB, UB \leftarrow DoubleboundExpSearch(x_j, \alpha_j, LineToCross_1, LineToCross_2)$;
  else if x_j ***in*** *Area B* **then**
    $\alpha_j \leftarrow -1$;
    $LB \leftarrow SingleboundExpSearch(x_j, \alpha_j, SparingLine)$;
    $\alpha_j \leftarrow 1$;
    $UB \leftarrow SingleboundExpSearch(x_j, \alpha_j, CoverageLine)$;
  else
    $\alpha_j \leftarrow -1$;
    $LineToCross_1 \leftarrow CoverageLine$;
    $LineToCross_2 \leftarrow SparingLine$;
    $LB, UB \leftarrow DoubleboundExpSearch(x_j, \alpha_j, LineToCross_1, LineToCross_2)$;
  $\mathcal{P}_{warmstart} \leftarrow \mathcal{P}_{warmstart} \cup \{GetInterpolatedSolutions(LB, UB)\}$;
  $\mathcal{P}_{warmstart} \leftarrow GetNonDominated(\mathcal{P}_{warmstart})$;
return $\mathcal{P}_{warmstart}$;

are dominated or do not lie between the sparing and coverage lines are filtered out. The final treatment plans are then given as input to MO-RV-GOMEA. Specifically, the plans are inserted in the elitist archive in generation 0. The elitist archive is then filtered by removing treatment plans that violate any of the constraints. The first-generation

population includes the filtered elitist archive complemented by additional cold-start treatment plans up to the determined population size.

Algorithm 3: Double-bound Exponential Search
// *original treatment plan* x_j*, step precision value* p*, direction factor* α_j, modified treatment plan y_i, number of original treatment plans n, lower bound factor $lb_i \in LB$, upper bound factor $ub_i \in UB$ //

$iteration \leftarrow 0$;
$LB \leftarrow \{\}$;
$UB \leftarrow \{\}$;
while $iteration = 0$ ***or*** $\exists y_i$ *s.t.* $SearchDone(y_i)$ *is False* **do**
  $i = 0$;
  for $k \in \{0, 1, .., n-1\}$ **do**
    if $j \neq k$ **then**
      if $iteration = 0$ **then**
        $Stepsize_i \leftarrow 0$;
        $lb_i \leftarrow 0$;
        $ub_i \leftarrow 2^{Stepsize_i}$;
        $SearchDone(y_i) \leftarrow$ False;
      else
        if $LineEvaluation(y_i, LineToCross_1)$ *is False* **then**
          $Stepsize_i \leftarrow Stepsize_i + 1$;
          $lb_i \leftarrow ub_i$;
          $ub_i \leftarrow 2^{Stepsize_i}$;
        else if $LineEvaluation(y_i, LineToCross_1)$ *is True* ***and*** $LineEvaluation(y_i, LineToCross_2)$ *is False* **then**
          $Stepsize_i \leftarrow Stepsize_i + 1$;
          $ub_i \leftarrow 2^{Stepsize_i}$;
        else
          if $SearchDone(y_i)$ *is False* **then**
            $LB \leftarrow LB \cup \{lb_i\}$;
            $UB \leftarrow UB \cup \{ub_i\}$;
            $SearchDone(y_i) \leftarrow True$;
        $y_i \leftarrow x_j + \alpha_j * p * ub_i\,|x_j - x_k|$;
        $i \leftarrow i + 1$;
  $iteration \leftarrow iteration + 1$;
return LB, UB;

Importantly, the aforementioned way of moving treatment plans through the $LCI(t)$ and $LSI(t)$ plane is not directly controllable. In other words, we do not change the $LCI(t)$ and $LSI(t)$ values directly but change dwell time sequences and then evaluate what the resulting $LCI(t)$ and $LSI(t)$ values are. More formally, we pick a treatment plan x_j and add or subtract the dwell time differences between that treatment plan x_j and another treatment plan x_k. To consistently accumulate/deplete all dwell times instead

of a single dwell time, we take the absolute difference between the two treatment plans. Whether addition or subtraction is applied depends on the direction of travel α_j, which is based on the initial position of a treatment plan in the $LCI(t)$, $LSI(t)$ plane and the line to cross. A parameter p is introduced to add additional precision if steps of base 2 are posed too large in the first few iterations of the search. In our setup, we define $p = 0.1$. If x_j is located in Area A or C, we apply the Double-Bounded Exponential

Algorithm 4: Single-bound Exponential Search
// *original treatment plan* x_j*, step precision value* p*, direction factor* α_j, modified treatment plan y_i, number of original treatment plans n, upper bound factor $ub_i \in UB$ //

```
1  iteration ← 0;
2  UB ← {};
3  while iteration = 0 or ∃y_i s.t. SearchDone(y_i) is False do
4      i = 0;
5      for k ∈ {0, 1, .., n − 1} do
6          if j ≠ k then
7              if iteration = 0 then
8                  Stepsize_i ← 0;
9                  ub_i ← 2^Stepsize_i;
10                 SearchDone(y_i) ← False;
11             else
12                 if LineEvaluation(y_i, LineToCross) is False then
13                     Stepsize_i ← Stepsize_i + 1;
14                     ub_i ← 2^Stepsize_i;
15                 else
16                     if SearchDone(y_i) is False then
17                         UB ← UB ∪ {ub_i};
18                         SearchDone(y_i) ← True;
19             y_i ← x_j + α_j * p * ub_i |x_j − x_k|;
20             i ← i + 1;
21     iteration ← iteration + 1;
22 return UB;
```

Search (Algorithm 3). In the double bounded search, if a perturbed treatment plan y_i has crossed the first line in terms of the corresponding $LCI(t)$ and $LSI(t)$ values, the lower bound is identified. When the second line to cross is surpassed, the upper bound is identified. If all the double-bounded searches for a treatment plan x_j are completed, a set of the lower and upper bound values is returned. If x_j is located in Area B, we apply the Single-Bound Exponential Search (Algorithm 4) twice. Once the search has surpassed the pre-determined line to cross, the upper bound value is identified. When all the single bounded searches for a treatment plan x_j are completed, a set of the upper bound values is returned.

3.4 Experimental Setup

To measure the individual effect of the eFNN model and the corrections by exponential search and interpolation, we derive the following experimental setup. We analyze 4 initializations including the cold start benchmark initialization originally implemented. The second initialization makes sole use of the eFNN predictions and the third initialization is a combination of the eFNN predictions with applied corrections by Exponential search and Interpolation (referred to as eFNN+ExpInt).

The fourth initialization strategy includes 5 treatment plans that are based on a recently developed heuristic to improve upon the original cold start. After applying exponential search and interpolation, we refer to this initialization as Heuristic+ExpInt. This recently developed heuristic initialization determines the lower and upper bounds based on the weight-To-Time conversion factor. The main motivation for using this recently developed heuristic is that it provides a more balanced starting point in comparison to the cold start. Moreover, it allows us to compare the exponential search and interpolation method with an eFNN starting point and the best-effort heuristic starting point so far developed. Kindly refer to Algorithm 5 for a formulation of the heuristic start used for the Heuristic+ExpInt initialization Method.

For the sake of clarity in reference, we consider the eFNN, eFNN+ExpInt and Heuristic+ExpInt initialization all to be 'warm-start' initializations in contrary to the original initialization which is referred to as the 'cold-start'.

Algorithm 5: Heuristic Start Initialization
// initial population $\mathcal{P}_{init}$ size n; treatment plan $\boldsymbol{x}$ size m; weightToTime-factor wTT; LowerBound initialization LB_{init}; UpperBound initialization UB_{init};

$\mathcal{P}_{init} = \{\}$;
$LB_{init} \leftarrow 0.85 \cdot wTT$;
$UB_{init} \leftarrow 0.95 \cdot wTT$;
for $i \in \{0, 1, .., n-1\}$ **do**
    $\mathbf{x} = \{\}$;
    for $j \in \{0, 1, .., m-1\}$ **do**
        $z \sim \mathcal{U}(0, 1)$;
        $dwelltime \leftarrow LB_{init} + (UB_{init} - LB_{init}) * z$;
        $\mathbf{x} \leftarrow \mathbf{x} \cup \{dwelltime\}$;
    $\mathcal{P}_{init} \leftarrow \mathcal{P}_{init} \cup \{\mathbf{x}\}$;

The most notable parameter configurations for MO-RV-GOMEA are a population size of 288 and a maximum runtime of 5 minutes per patient fraction. We use 200,000 dose calculation points during optimization for (re-)evaluation and 1,000,000 dose reference points in the final evaluation.

We identify two reference criteria to measure the convergence speed of MO-RV-GOMEA for the given initializations. The first reference criterium is reached when a treatment plan reaches the golden corner ($LCI(t) > 0.0$ and $LSI(t) > 0.0$). Since MO-RV-GOMEA optimization does not always return a plan that is evaluated inside the golden corner, we include a second reference criterium conditioned at ($LCI(t) > -0.2$ and $LSI(t) > -0.2$). For the eFNN training, ExpInt procedure, and MO-RV-GOMEA optimization, we make use of scripts parallelized over 1 Nvidia Titan XP and 3 Nvidia Titan X GPUs. The codebase for this project can be accessed by navigating to this private GitHub Repository. Patient-fraction naming conventions in the GitHub repository and thesis are anonymized and randomly numbered.

Chapter 4

Results

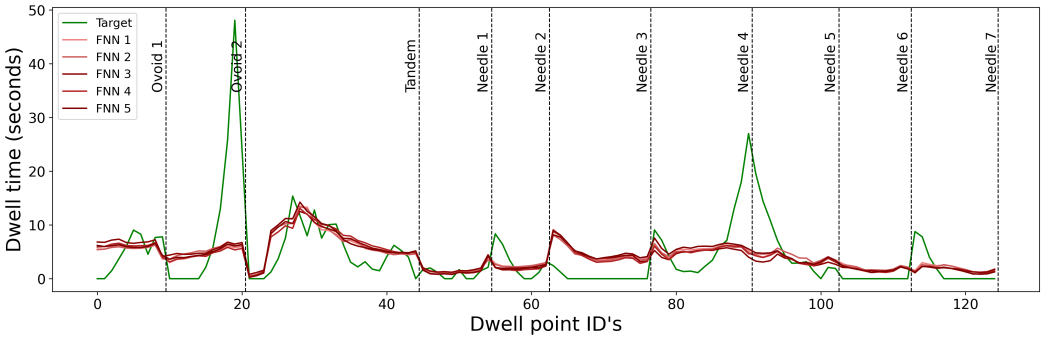
4.1 eFNN Predictions

The final performance of the eFNN model is determined by calculating the average Huberloss per dwell point using a test set that contains 35 patient fractions. Refer to Table 4.1 for a summary of the performance of each FNN model trained and evaluated on a different cross-validation split. In comparison to the average validation set performance (see Table C.2), the test set Huberloss is approximately 4-5% higher. Note that the Huberloss is the Huberloss of the target dwell weight (dwell time corrected by radioactive source).

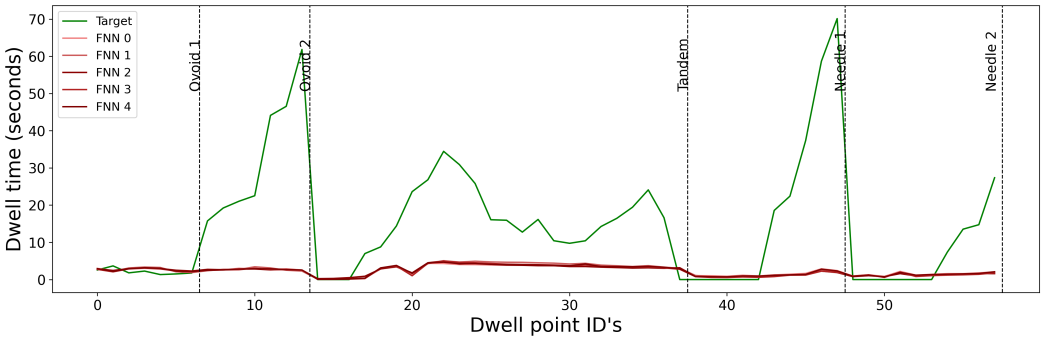
	Huberloss	Absolute Dwell time error (seconds)
FNN Model CV Split #1	0.730	4.660
FNN Model CV Split #2	0.737	4.720
FNN Model CV Split #3	0.728	4.624
FNN Model CV Split #4	0.729	4.672
FNN Model CV Split #5	0.732	4.609

TABLE 4.1: Test Set Results

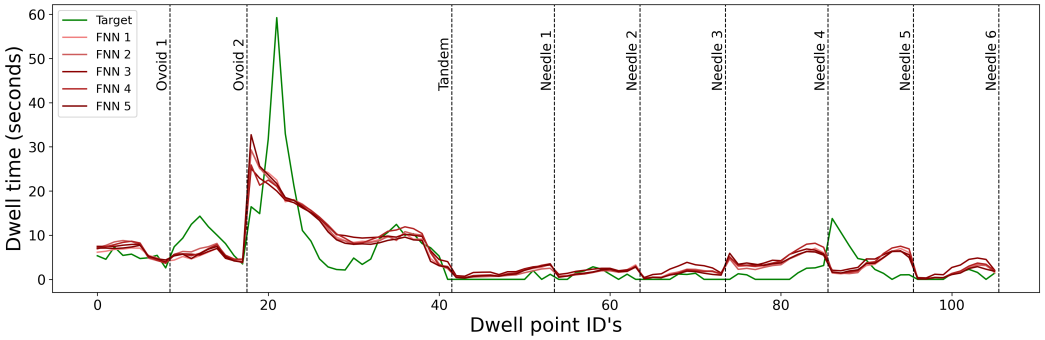
To illustrate how the eFNN predicted treatment plans are composed, 4 sample patient-fractions are denoted in Figure 4.1. We present the predicted treatment as a sequence of dwell times, divided into sections to distinguish between needles, ovoids and the tandem. Importantly, the figures denote dwell times and not dwell weights. One should therefore keep in mind that the actual radiation delivered is also dependent on the radiation source and not solely indicated by the dwell time.



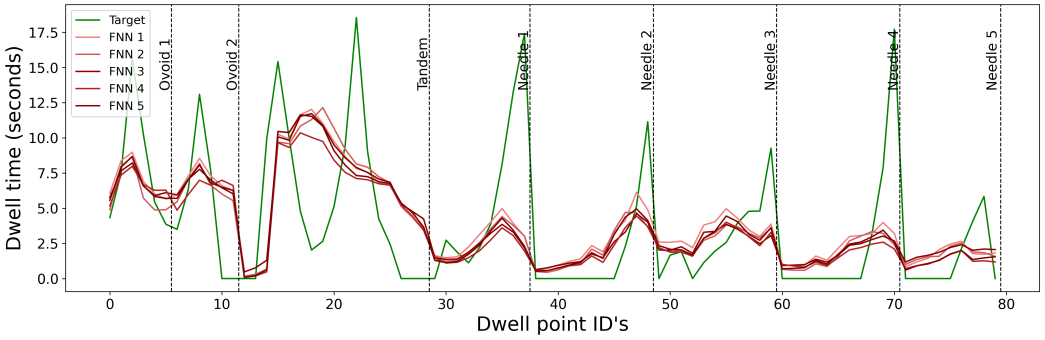
(A) Patient 1 Fraction b



(B) Patient 3 Fraction a



(C) Patient 8 Fraction b



(D) Patient 16 Fraction b

FIGURE 4.1: eFNN dwell time prediction vs. target dwell time.

For each of the eFNN predictions (numbered by CV split), the treatment plan predictions show to be considerably similar per patient fraction. In Figure 4.1a, the eFNN prediction

captures a slight downward pattern in the tandem which is also observable in Figure 4.1c and 4.1d. The order of the dwell point ID's should be read as analogous to how the implant is inserted. E.g. for the tandem, the sequence from left to right represents the tip of the tandem up to the dwell point in the tandem closest to the applicator base. In 4.1d, the eFNN predictions show to capture the spiking pattern in the target plan to some extent. However, especially in the needles, the predictions show to be off by 15 seconds for some of these time peaks. In Figure 4.1b, the eFNN predictions are underfitting the target treatment plan. The target treatment plan for this patient fraction has considerably high dwell time peaks, which are not well captured by the eFNN model. The eFNN model generally tends to predict shorter dwell times, as can be seen in Figure 4.2 by the lower spread of outlying points under the diagonal.

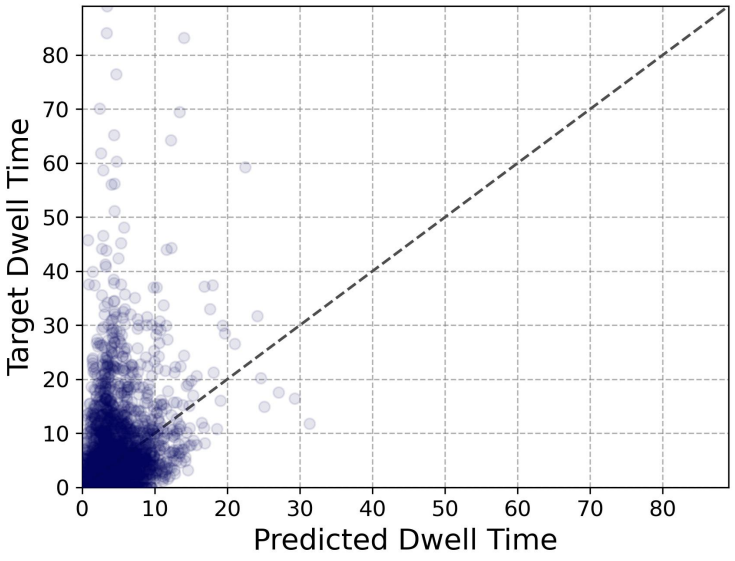
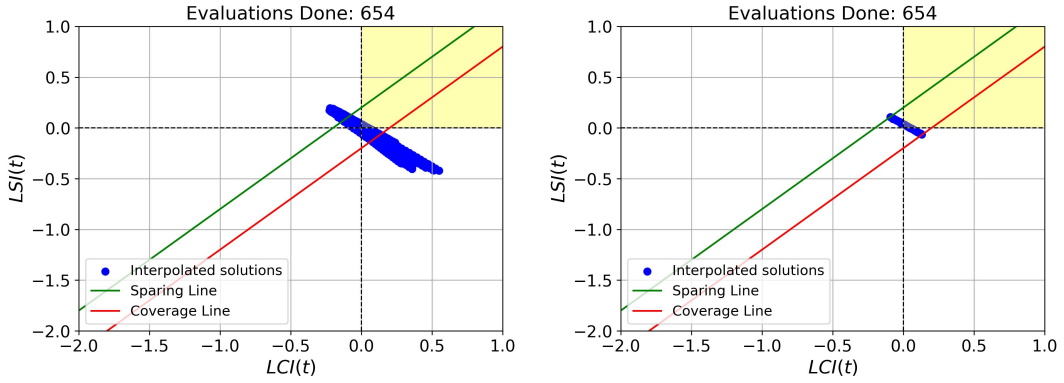


FIGURE 4.2: Target vs. Predicted Dwell Times - FNN Model CV Split #1

4.2 Exponential Search & Interpolation

A complete set of visualizations for the exponential search and interpolation method is provided in this SURFdrive repository.

In Figure 4.3 an interpolation example is given for Patient 12a. The exponential search for this patient originally started left from the sparing boundary and extended substantially beyond the coverage line. A total of 500 points are then interpolated, shown by the blue area in Figure 4.3a. As a result of picking only the non-dominated solutions between the sparing and coverage line, a fraction of the 500 points remain as



(A) Interpolated Solutions

(B) Interpolated Non-Dominated Solutions Between Lines

FIGURE 4.3: Sample Result Interpolation Patient 12a

input for MO-RV-GOMEA optimization (Figure 4.3b). To both the eFNN and heuristic start initialization, the Exponential search and Interpolation method is applied. A brief overview of the average ± (standard deviation) required evaluations is shown in Table 4.2 (complete results shown in Appendix E.1). On average, eFNN+ExpInt required less exponential search evaluations than Heuristic+ExpInt. As no underlying sample distribution is assumed, we apply a two-sided Mann Whitney-U test ($\alpha = 0.01$) to statistically quantify this difference. With a resulting p-value of 0.002, we state that the eFNN+ExpInt method required significantly fewer exponential search evaluations than the Heuristic+ExpInt method.

	# Evaluations Exponential Search	**# Evaluations Interpolation**
Heuristic+ExpInt	193.17 ± (31.65)	500 ± (0.0)
eFNN+ExpInt	165.8 ± (31.82)	500 ± (0.0)

TABLE 4.2: Evaluations Exponential Search & Interpolation (n=35)

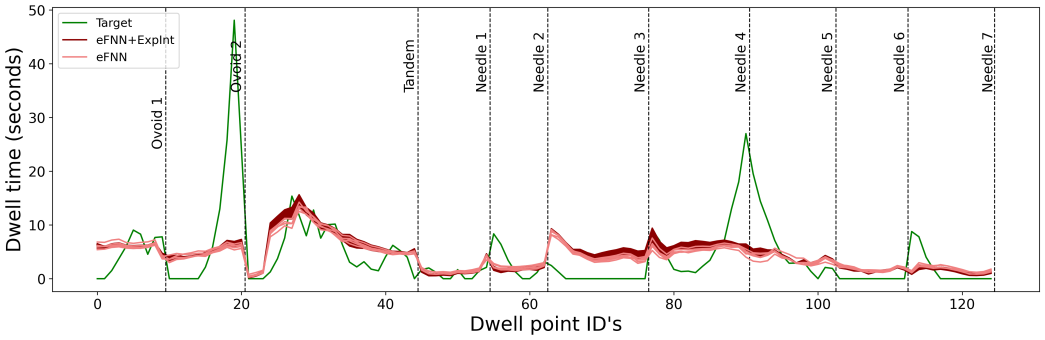
Since we always interpolate 25 points between each of the 20 lower -and upper bound pairs, the number of evaluations for interpolation is constant at 500. The maximum amount of exponential search evaluations required was 259 for a Heuristic+ExpInt patient-fraction run and 233 for an eFNN start patient-fraction run. At minimum, 145 evaluations were required for a Heuristic+ExpInt patient-fraction run in comparison to 123 for an eFNN+ExpInt patient-fraction run.

Similar to the eFNN prediction visualizations, we highlight 4 patient-fraction treatment plan configurations in Figure 4.4 including the output plans derived by exponential search and interpolation. The patient fraction in Figure 4.4b shows the largest change in dwell time sequences compared to the eFNN predictions. Given that an eFNN prediction needs large adjustments is not the only indicator for a high number of required

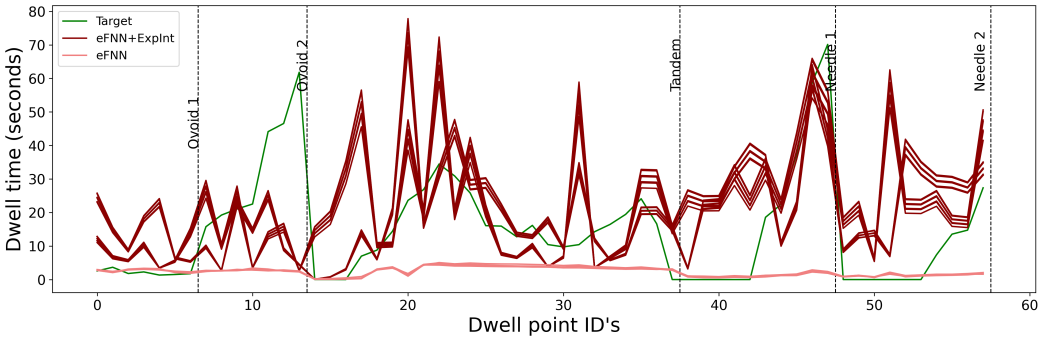
exponential search evaluations. As noted in Figure 4.4d, the number of evaluations exceeds that of Figure 4.4b, even though the eFNN+ExpInt dwell time sequence does not appear to have changed as much in comparison to the eFNN sequence. This can be due to a potentially very small difference between the two FNN treatment plans, resulting in more evaluations to accumulate/deplete dwell times. Another cause can be the location of the initial eFNN predictions in the LCI/LSI plane. If a treatment plan is evaluated in area B at the start of exponential search, two single exponential walks have to be done in order to find the bounds for interpolation. Thus, in comparison to e.g. a combined search from Area A close to the sparing line, more evaluations could be required. This is however not always the case given the below-average number of evaluations shown in Figure 4.4c.

The results from the Heuristic+ExpInt procedure show a considerably different dwell time pattern in Figure 4.5. As indicated by the light-blue horizontal line, the heuristic start sequences appear to be almost constant around a non-zero dwell time. This results in a treatment plan that already incorporates a substantial amount of coverage, in contrast to the original cold start which uses pre-defined ranges of dwell times that are generally lower. The effect of having a substantial amount of coverage, before running the exponential search and interpolation, can be observed in Figure 4.5a, 4.5c and 4.5d. For these cases, the Heuristic+ExpInt dwell times are generally lower than the Heuristic dwell times. In comparison to the original cold start, dwell time depletion would generally not be necessary since the uniform sequence of dwell times in only the tandem and ovoids generates a more sparing-oriented treatment plan.

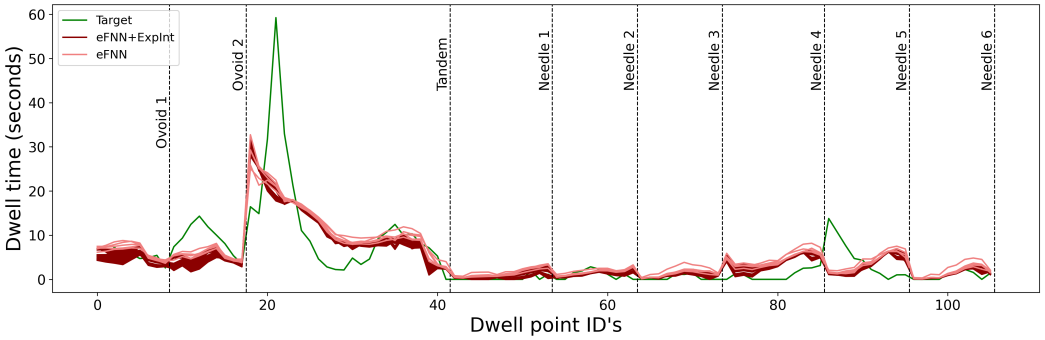
Figure 4.5b shows various Heuristic+ExpInt treatment plan patterns that are all non-dominated by each other in terms of their corresponding $LCI(t)$ and $LSI(t)$ values. Although the complete set of non-dominated solutions appears noisy, it is composed of several treatment plan sets that follow roughly the same pattern. Figure 4.4b exemplifies how 2 of such sets of plans occur: following roughly the same pattern, but with gaps between the lines (best visible at flat peaks). The gaps are a direct result from the interpolation interval which is set at a fixed stepsize distance. Additionally, one should note that the highly fluctuating peaks in the Exp+Int treatment plans for patient 3a could encompass a high DTMR constraint value.



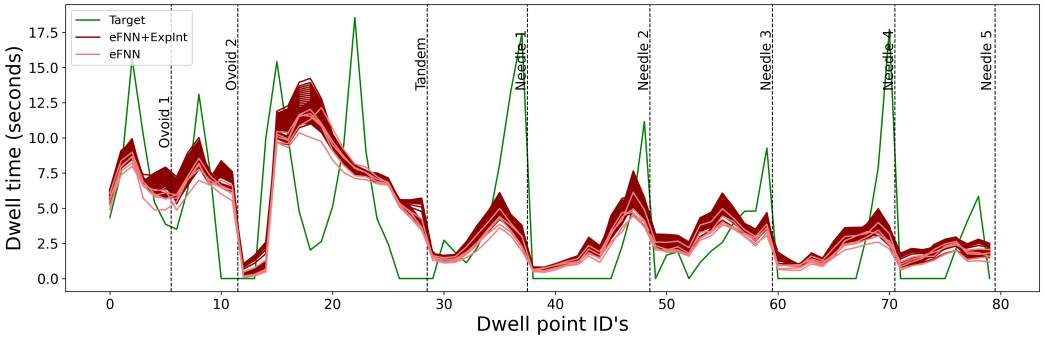
(A) Patient Fraction Patient 1b: 689 Exponential search and Interpolation evaluations



(B) Patient Fraction Patient 3a: 733 Exponential search and Interpolation evaluations

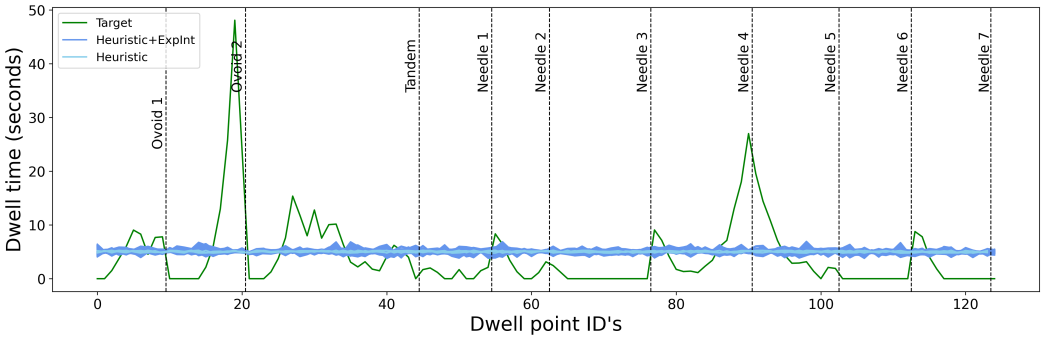


(C) Patient Fraction Patient 8b: 633 Exponential search and Interpolation evaluations

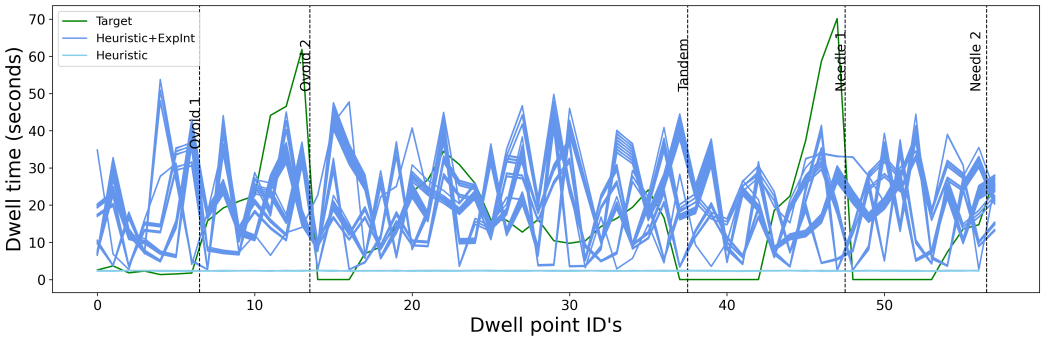


(D) Patient Fraction Patient 16b: 740 Exponential search and Interpolation evaluations

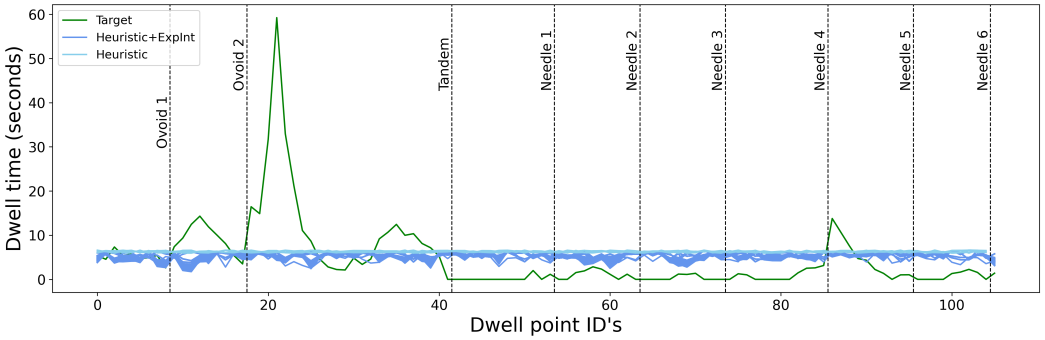
FIGURE 4.4: eFNN dwell times with Exponential search & Interpolation



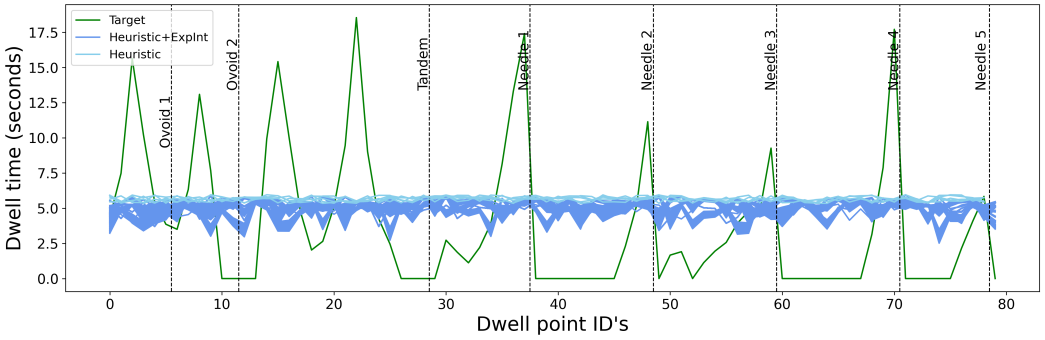
(A) Patient Fraction Patient 1b: 665 Exponential search and Interpolation evaluations



(B) Patient Fraction Patient 3a: 759 Exponential search and Interpolation evaluations



(C) Patient Fraction Patient 8b: 647 Exponential search and Interpolation evaluations



(D) Patient Fraction Patient 16b: 653 Exponential search and Interpolation evaluations

FIGURE 4.5: Cold Start dwell times with Exponential search & Interpolation

4.3 Initialization Quality Performance

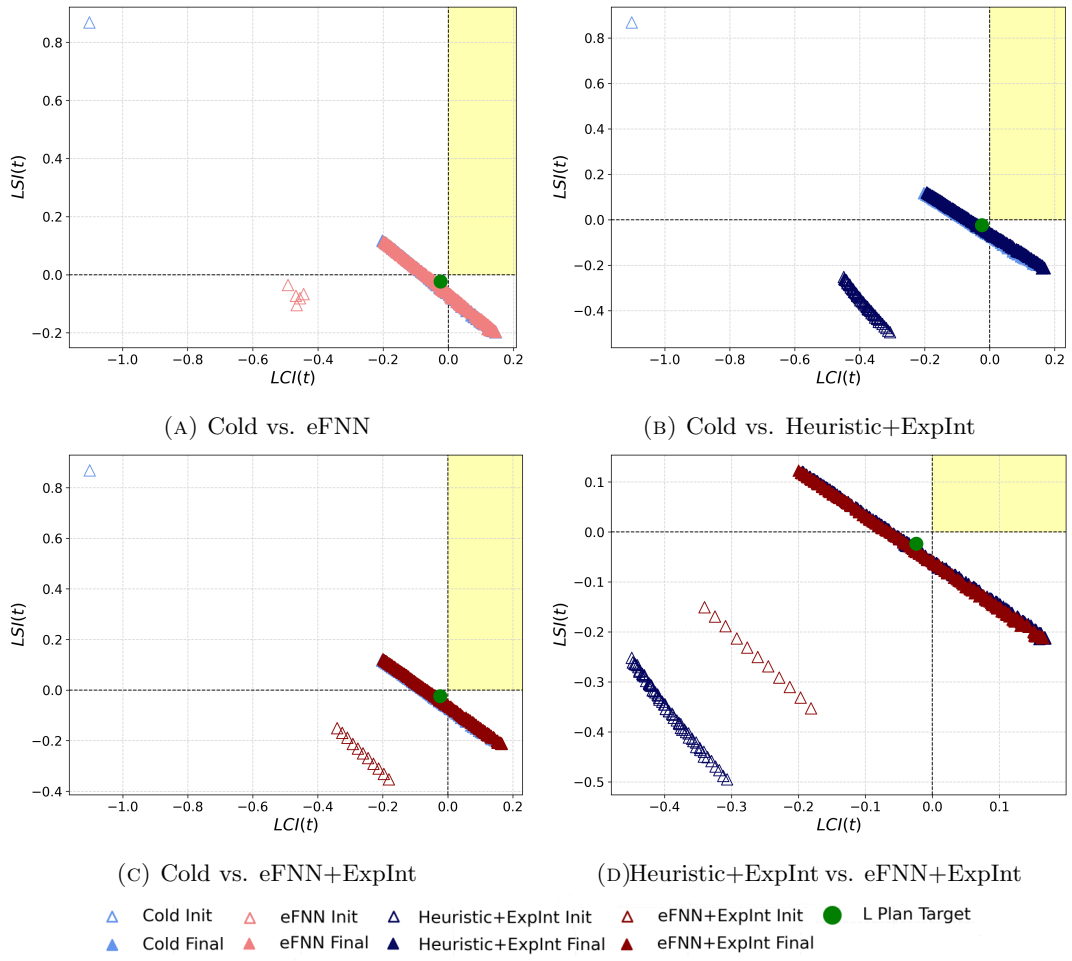
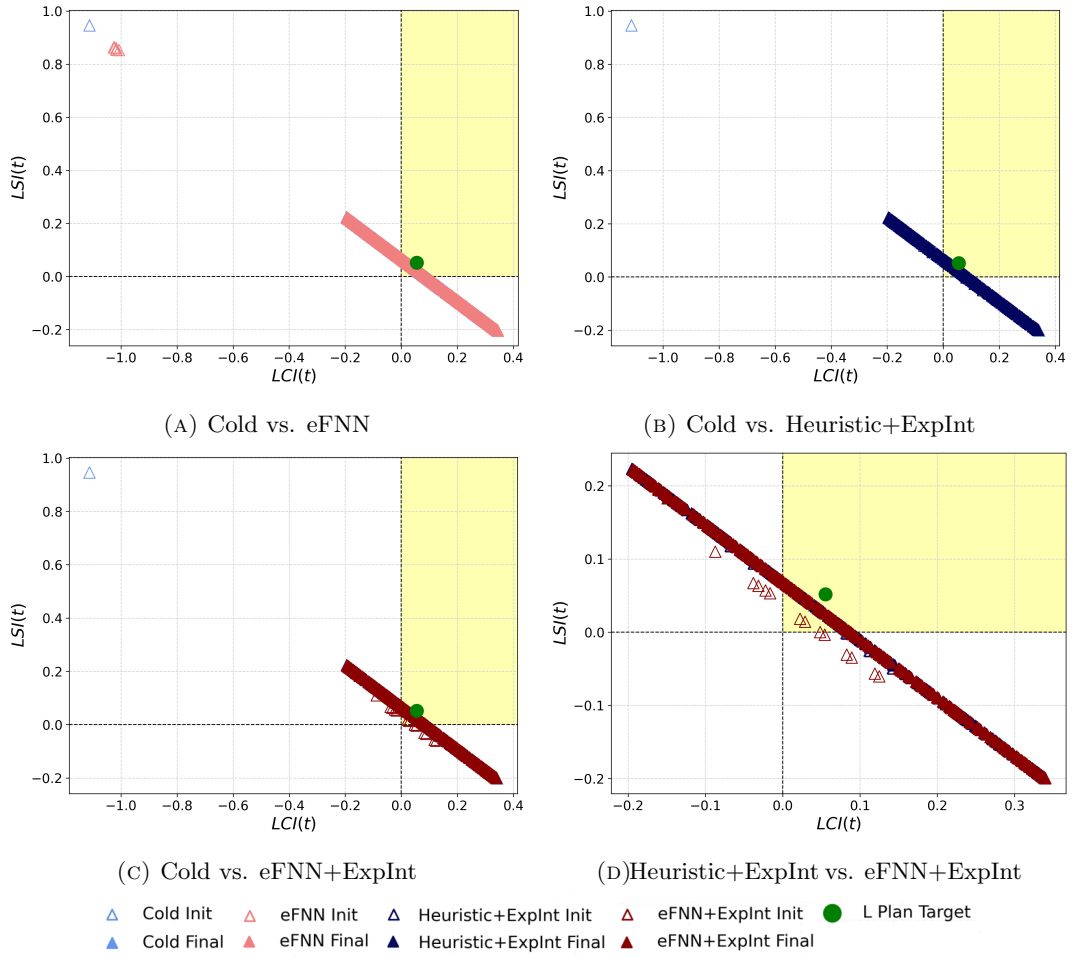


FIGURE 4.6: Warm Start Initializations Patient 1b

Figure 4.6 shows an approximation front comparison between 4 initializations for patient 1b, including the corresponding final output of 5 minutes MO-RV-GOMEA optimization. In terms of distance to the golden corner, the cold start is outperformed by all other initializations for this patient-fraction. The eFNN+ExpInt initialization shows to approximate a treatment plan closest to the golden corner, closely followed by the eFNN and Heuristic+ExpInt initialization. Apart from some small deviations, the front of final treatment plans all show to be considerably similar regarding $LCI(t)$ and $LSI(t)$ performance, close to the original L plan target but not yet reaching the golden corner.

A second approximation front example is shown in Figure 4.7. For Patient 3a, the cold start and eFNN initialization were evaluated as a high-sparing plan. After applying exponential search and interpolation, some plans are evaluated inside the golden corner and the Heuristic+ExpInt Init even spans a front that was not visibly improved after 5 minutes of runtime (plotted behind the Heuristic+ExpInt Final plans).



(A) Cold vs. eFNN

(B) Cold vs. Heuristic+ExpInt

(C) Cold vs. eFNN+ExpInt

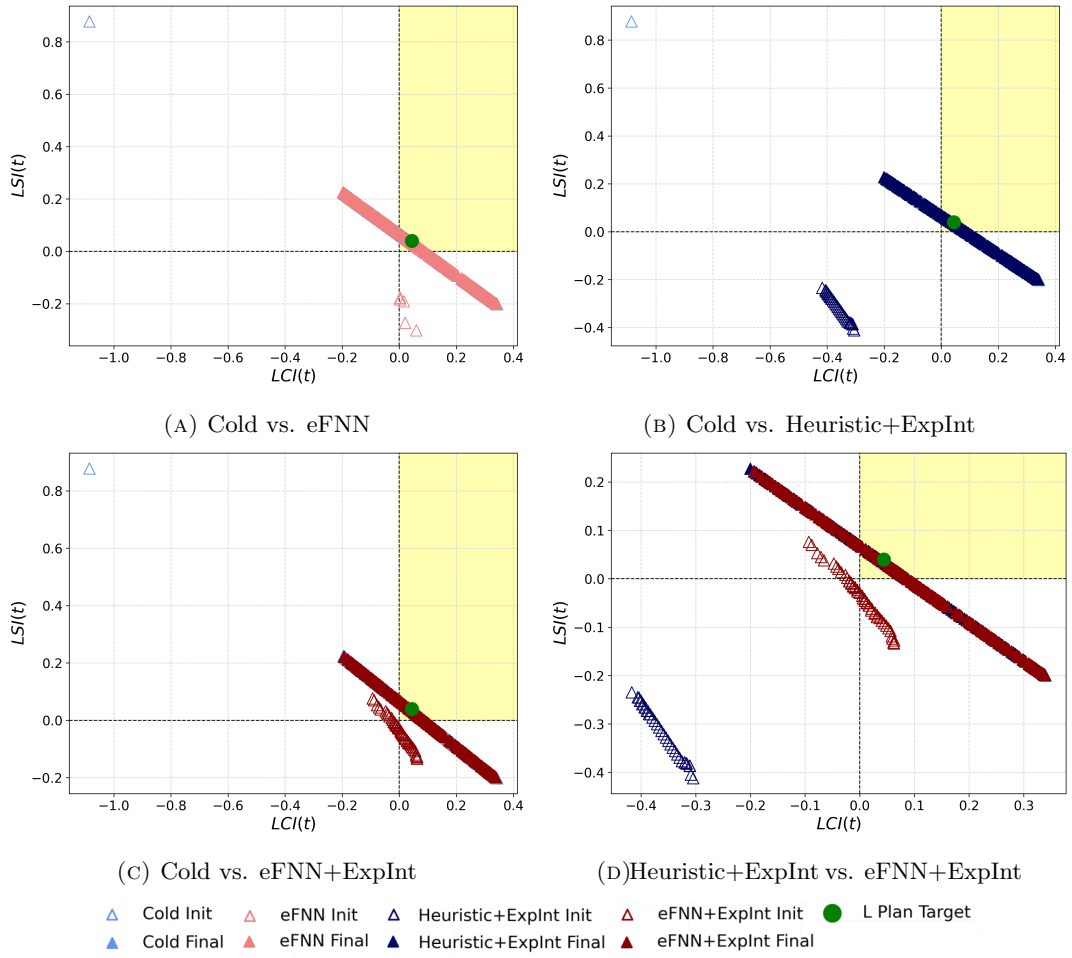
(D)Heuristic+ExpInt vs. eFNN+ExpInt

FIGURE 4.7: Warm Start Initializations Patient 3a

An example of where the eFNN+ExpInt Init outperforms all other initializations is shown by the example fronts in Figure 4.8. In this example, the eFNN+ExpInt Init front closely reaches the golden corner whereas the Heuristic+ExpInt Init front is positioned considerably further away from the golden corner. Please note the mere visibility of some of the final fronts for this patient fraction (for example the Cold Final front in 4.8b and the eFNN+ExpInt Final front in 4.8c). The final fronts are in this case mostly overlapping.

In Figure 4.9, we exemplify a second Heuristic+ExpInt initialization that outperforms the eFNN+ExpInt initialization. Particularly, Figure 4.9d shows that the Heuristic+ExpInt Init produced a range of treatment plans that are evaluated inside the golden corner. However, patient 16b could also be a considerably easy patient to optimize since the eFNN+ExpInt Init and eFNN Init are evaluated close to the golden corner as well.

Given the initializations from the aforementioned 4 patient fractions, one can observe that the initializations with ExpInt provide a more consistent and LCI/LSI balanced configuration in comparison to the eFNN Init. The ExpInt initializations are guaranteed



(A) Cold vs. eFNN

(B) Cold vs. Heuristic+ExpInt

(C) Cold vs. eFNN+ExpInt

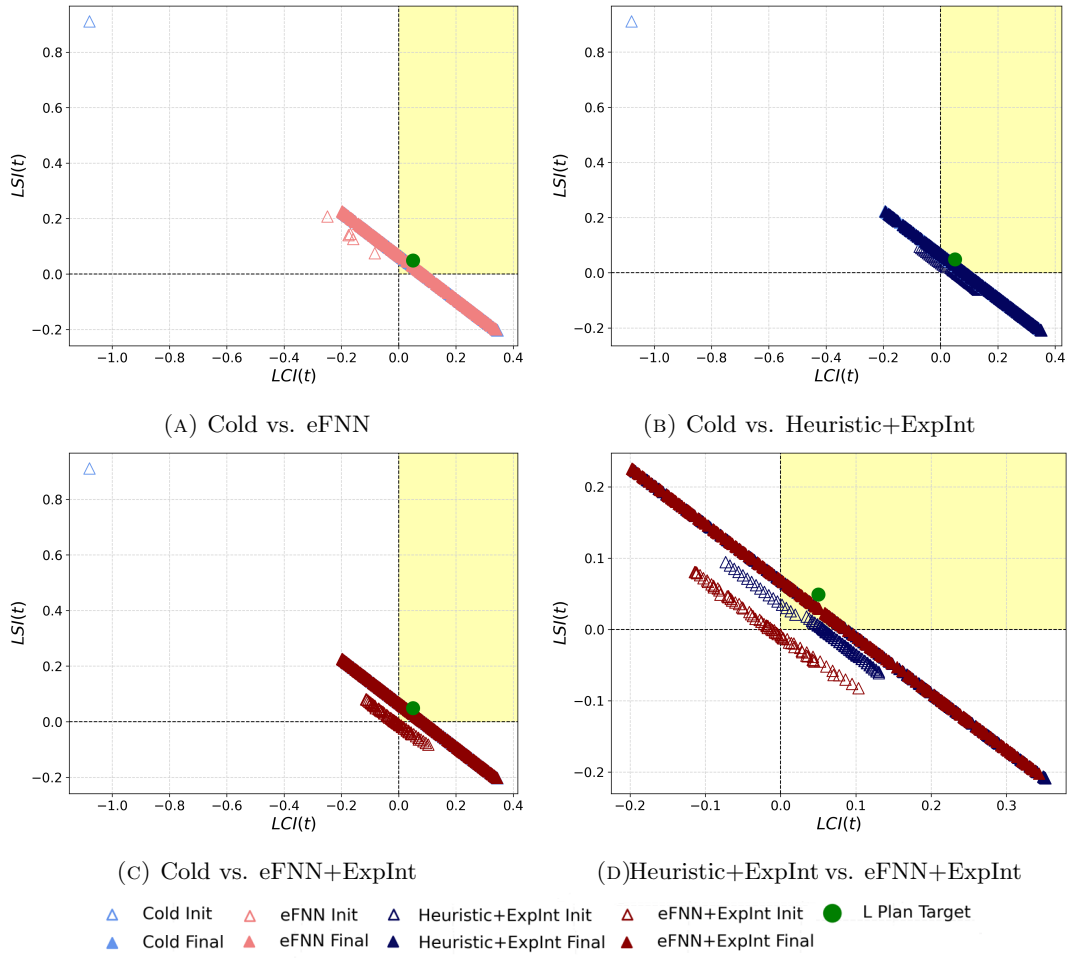
(D)Heuristic+ExpInt vs. eFNN+ExpInt

FIGURE 4.8: Warm Start Initializations Patient 8b

to start between the previously defined coverage and sparing line thresholds, whereas an eFNN Init plan can occasionally be far positioned from this threshold area (e.g. in Figure 4.7a). For all 4 sampled patients, the warm start initializations outperformed the original cold start initialization in terms of distance to the golden corner in the LCI/LSI plane.

4.4 Initialization Runtime Performance Golden Corner

We depict the MO-RV-GOMEA runtimes that followed from the different initialization approaches as boxplots in Figures 4.10 and 4.11. The left box-plot whisker indicates the first quartile minus 1.5 times the inter-quartile range (IQR) and the right boxplot whisker indicates the third quartile plus 1.5 times the IQR. Outliers are visualized as black dots and for scale reference a vertical dotted line (at 60 seconds runtime) is included.



(A) Cold vs. eFNN

(B) Cold vs. Heuristic+ExpInt

(C) Cold vs. eFNN+ExpInt

(D)Heuristic+ExpInt vs. eFNN+ExpInt

FIGURE 4.9: Warm Start Initializations Patient 16b

Considering all boxplots in Figures 4.10 and 4.11, it is clear that the optimization time required by MO-RV-GOMEA to reach the GC differs dramatically between patient fractions. Runtimes differ substantially within each patient fraction as well, shown by the wide-spanned boxplots of e.g. patient 4a, 5b and 16a. For this set of sample patients, the cold start initialization is outperformed by at least one of the alternatively proposed initializations for the majority of the treatment plans generated. A clear exception is patient 5b, indicating no distinct preferable initialization in terms of runtime. This is potentially due to the position of the maximally achievable LCI and LSI for this patient-fraction, which is just over the GC threshold. Patient-fractions of which the GC threshold is easier to reach, such as patient 3b, 12b or 13b, indicate no clear visual difference between initializations as well.

Patient 4a exemplifies that all the alternatively proposed initializations made a large impact on the required runtime. More specifically, the ExpInt initializations for this patient show that a more robust (lower spread in runtime) and efficient (lower average runtime) optimization process was achieved. However, this does not hold for every

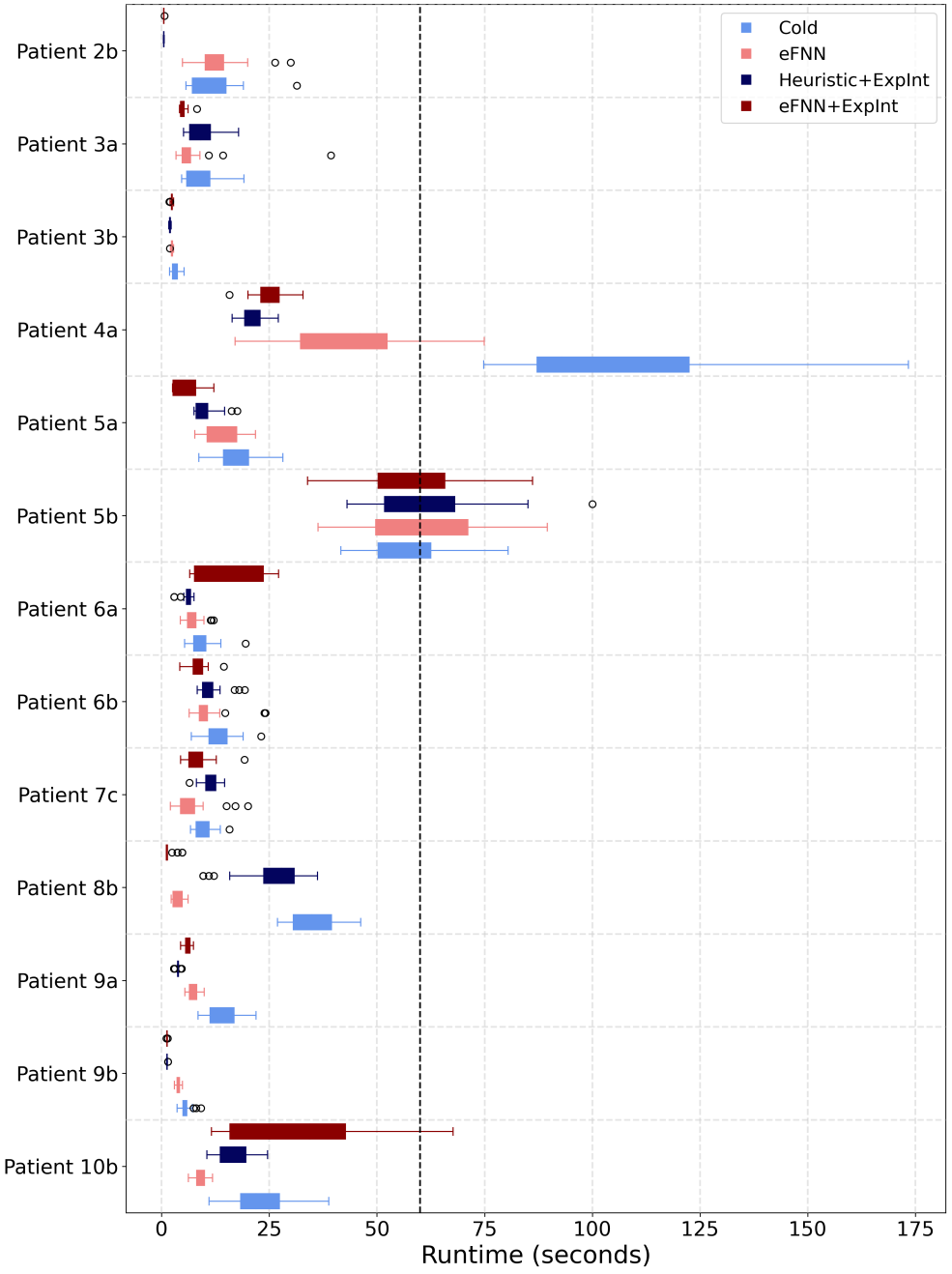


FIGURE 4.10: Part I: MO-RV-GOMEA Runtime to reach the Golden Corner (0.0, 0.0), n=30

patient as is illustrated by patient 10b, for which the eFNN+ExpInt runtime varied substantially and occasionally took longer than the cold start. Another example is patient 16a, where the Heuristic+ExpInt initialization led to a longer required runtime in comparison to the cold start.

A compact overview of all average runtimes is provided in Table 4.3. This overview only contains patient fractions for which all of the 30 performed optimization runs converged to a range of treatment plans that reached the golden corner. The results are

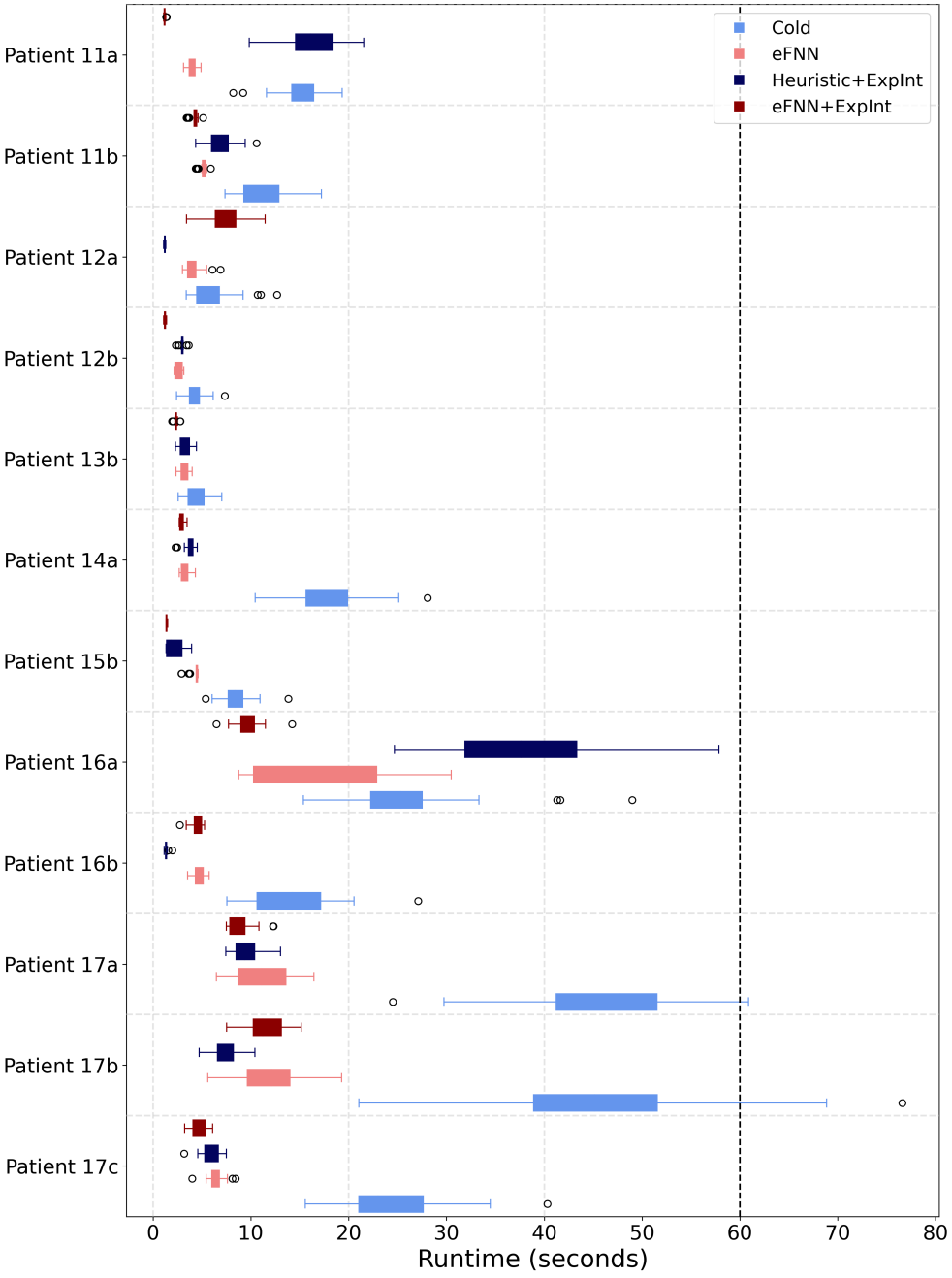


FIGURE 4.11: Part II: MO-RV-GOMEA Runtime to reach the Golden Corner (0.0, 0.0), n=30

statistically supported, a complete overview of the statical test results can be found in Appendix D. In 11 out of 25 patient fractions, the eFNN+ExpInt initialization resulted in a significantly lower runtime required to reach the golden corner in comparison to all other initializations. In 20 out of 25 patient fractions, the eFNN+ExpInt initialization significantly outperformed the original cold start. For 6 patient fractions, the Heuristic+ExpInt initialization significantly outperformed all other initializations.

Patient	eFNN	eFNN+ExpInt	Cold	Heuristic+ExpInt
Patient 2b	13.158 ± (5.297)	0.512 ± (0.044)*	11.913 ± (5.271)	0.503 ± (0.043)*
Patient 3a	7.199 ± (6.367)	5.003 ± (0.847)*	8.931 ± (4.009)	9.265 ± (3.011)
Patient 3b	2.403 ± (0.2)*	2.355 ± (0.226)*	3.187 ± (0.913)	**1.935 ± (0.143)***
Patient 4a	42.369 ± (14.005)*	24.861 ± (3.114)*	104.25 ± (24.782)	**21.128 ± (2.538)***
Patient 5a	14.381 ± (4.356)	**6.02 ± (2.6)***	17.726 ± (4.649)	10.122 ± (2.548)*
Patient 5b	60.455 ± (14.174)	58.974 ± (11.725)	57.289 ± (9.283)	61.566 ± (12.675)
Patient 6a	7.397 ± (1.866)	16.099 ± (7.481)	9.254 ± (2.841)	6.032 ± (1.095)*
Patient 6b	10.789 ± (3.975)*	**8.146 ± (2.059)***	13.613 ± (3.464)	11.309 ± (2.643)
Patient 7c	6.532 ± (4.251)*	8.284 ± (2.858)	9.596 ± (2.064)	11.24 ± (1.946)
Patient 8b	3.722 ± (1.061)*	**1.812 ± (1.254)***	35.6 ± (5.3)	26.133 ± (7.401)*
Patient 9a	7.21 ± (1.061)*	6.141 ± (0.735)*	13.856 ± (3.676)	**3.706 ± (0.494)***
Patient 9b	3.869 ± (0.431)*	1.26 ± (0.059)*	5.542 ± (1.283)	1.287 ± (0.07)*
Patient 10b	**8.92 ± (1.279)***	32.828 ± (16.107)	22.974 ± (6.537)	16.57 ± (3.762)*
Patient 11a	4.048 ± (0.465)*	**1.189 ± (0.076)***	15.01 ± (2.494)	16.167 ± (2.724)
Patient 11b	5.14 ± (0.344)*	**4.225 ± (0.409)***	11.396 ± (2.392)	6.863 ± (1.503)*
Patient 12a	4.045 ± (0.866)*	7.315 ± (2.123)	6.016 ± (2.425)	**1.2 ± (0.059)***
Patient 12b	2.613 ± (0.337)*	**1.224 ± (0.089)***	4.278 ± (0.976)	2.984 ± (0.234)*
Patient 13b	3.195 ± (0.42)*	**2.34 ± (0.181)***	4.437 ± (1.184)	3.291 ± (0.596)*
Patient 14a	3.274 ± (0.394)*	**2.918 ± (0.228)***	18.111 ± (3.911)	3.729 ± (0.536)*
Patient 15b	4.262 ± (0.47)*	**1.367 ± (0.041)***	8.499 ± (1.613)	2.24 ± (0.996)*
Patient 16a	16.815 ± (7.336)*	**9.727 ± (1.389)***	26.33 ± (7.204)	37.375 ± (8.219)
Patient 16b	4.797 ± (0.467)*	4.406 ± (0.533)*	14.209 ± (4.523)	**1.343 ± (0.145)***
Patient 17a	11.13 ± (2.7)*	8.989 ± (1.316)*	45.915 ± (8.92)	9.522 ± (1.433)*
Patient 17b	12.316 ± (3.134)*	11.6 ± (1.998)*	47.448 ± (12.274)	**7.387 ± (1.348)***
Patient 17c	6.5 ± (0.799)*	**4.835 ± (0.773)***	24.543 ± (5.323)	5.993 ± (0.932)*
Total Average	10.662 ± (3.043)	9.297 ± (2.331)	21.597 ± (5.092)	11.156 ± (2.284)

TABLE 4.3: MO-RV-GOMEA Runtime in seconds for the L-plan to reach the Golden Corner unconstrained (mean ± (standard deviation) with n = 30). Bold markings indicate a statistically significant runtime decrease in comparison to all 3 other methods. Asterisk markings indicate a statistically significant runtime decrease in comparison to the cold start initialization.

In 20 cases, the Heuristic+ExpInt initialization outperformed the cold start initialization. For patient 10b, the eFNN initialization outperformed all other initializations which demonstrates that using the ExpInt method for one case did not improve the eFNN predictions in favor of the runtime required. Similar to the eFNN+ExpInt and Heuristic+ExpInt initialization, the eFNN outperformed the cold start significantly in 20 out of 25 patient-fraction cases. Averaged over all patient-fractions in Table 4.3, the eFNN+ExpInt initialization showed the fastest convergence time towards the golden corner with 9.297 seconds on average.

4.5 Initialization Runtime Performance (-0.2, -0.2) Corner

Since golden corner performance is not always achieved for every patient fraction in the test set, we complement the results from the previous section with runtime data that explains when the (-0.2, -0.2) threshold is surpassed in the LCI / LSI plane. Table 4.4 provides an overview of this data. Note that for patients 7a, 7b and 13a the results are

not shown since for these patients the (-0.2, -0.2) threshold was not surpassed in each run. In 12 out of 32 patient fractions, the eFNN+ExpInt required a significantly shorter runtime in comparison to all other initializations.

Patient	**eFNN**	**eFNN+ExpInt**	**Cold**	**Heuristic+ExpInt**
Patient 1a (-)	1.179 ± (0.103)*	1.18 ± (0.072)*	34.842 ± (3.968)	6.382 ± (0.748)*
Patient 1b (-)	7.853 ± (0.873)*	**6.571 ± (0.648)***	22.846 ± (4.182)	9.294 ± (1.022)*
Patient 2a (-)	5.09 ± (1.16)	**0.509 ± (0.047)***	4.322 ± (1.005)	5.095 ± (1.249)
Patient 2b	3.866 ± (0.565)	0.512 ± (0.044)*	3.818 ± (0.747)	0.503 ± (0.043)*
Patient 3a	4.143 ± (3.172)	1.215 ± (0.075)*	3.659 ± (0.442)	1.221 ± (0.1)*
Patient 3b	1.236 ± (0.087)*	1.222 ± (0.078)*	2.065 ± (0.238)	1.207 ± (0.077)*
Patient 4a	2.404 ± (0.973)*	**0.959 ± (0.837)***	17.663 ± (3.824)	3.246 ± (0.529)*
Patient 4b (-)	**3.137 ± (1.289)***	5.034 ± (0.582)*	25.321 ± (5.41)	4.518 ± (0.561)*
Patient 5a	3.249 ± (0.886)*	1.3 ± (0.069)*	8.48 ± (1.901)	**1.219 ± (0.077)***
Patient 5b	6.021 ± (2.353)*	**1.199 ± (0.065)***	8.435 ± (1.171)	8.794 ± (1.355)
Patient 6a	4.574 ± (0.617)*	**0.651 ± (0.05)***	5.677 ± (0.703)	2.309 ± (0.967)*
Patient 6b	7.061 ± (0.819)	**1.282 ± (0.56)***	7.646 ± (0.832)	4.313 ± (0.771)*
Patient 7c	2.634 ± (0.884)*	1.206 ± (0.05)*	5.515 ± (0.801)	1.16 ± (0.053)*
Patient 8a (-)	3.691 ± (1.367)*	**1.122 ± (0.042)***	8.778 ± (1.726)	5.67 ± (0.825)*
Patient 8b	1.129 ± (0.052)*	1.162 ± (0.065)*	8.634 ± (1.563)	6.744 ± (0.805)*
Patient 9a	1.203 ± (0.095)*	1.231 ± (0.097)*	5.56 ± (0.775)	1.215 ± (0.069)*
Patient 9b	2.901 ± (0.334)*	1.26 ± (0.059)*	3.904 ± (0.355)	1.287 ± (0.07)*
Patient 10a (-)	1.229 ± (0.099)*	1.246 ± (0.073)*	11.152 ± (1.971)	7.958 ± (1.036)*
Patient 10b	6.139 ± (0.691)*	1.204 ± (0.074)*	11.197 ± (1.878)	1.261 ± (0.065)*
Patient 11a	3.236 ± (0.308)*	**1.189 ± (0.076)***	4.841 ± (0.562)	4.248 ± (0.81)*
Patient 11b	4.379 ± (0.344)*	1.212 ± (0.083)*	6.559 ± (1.029)	1.562 ± (0.911)*
Patient 12a	2.561 ± (0.173)*	1.157 ± (0.065)*	3.448 ± (0.404)	1.2 ± (0.059)*
Patient 12b	2.283 ± (0.142)*	1.224 ± (0.089)*	2.693 ± (0.315)	1.154 ± (0.06)*
Patient 13b	2.233 ± (0.162)*	1.179 ± (0.059)*	2.602 ± (0.361)	1.183 ± (0.111)*
Patient 14a	2.797 ± (0.084)*	**1.729 ± (0.603)***	5.407 ± (1.049)	2.55 ± (0.182)*
Patient 15a (-)	2.115 ± (0.353)*	**0.595 ± (0.053)***	4.141 ± (0.581)	0.661 ± (0.07)*
Patient 15b	2.647 ± (0.428)*	0.604 ± (0.04)*	5.324 ± (0.721)	0.618 ± (0.043)*
Patient 16a	3.964 ± (1.694)*	**1.326 ± (0.069)***	8.668 ± (1.437)	8.402 ± (3.007)
Patient 16b	1.232 ± (0.08)*	1.281 ± (0.097)*	5.325 ± (0.543)	1.343 ± (0.145)*
Patient 17a	5.082 ± (0.485)*	1.136 ± (0.05)*	11.088 ± (1.724)	1.149 ± (0.04)*
Patient 17b	4.921 ± (0.323)*	**1.122 ± (0.046)***	11.275 ± (2.397)	1.161 ± (0.033)*
Patient 17c	4.312 ± (0.49)*	1.145 ± (0.041)*	6.805 ± (1.186)	1.162 ± (0.033)*
Total Average	3.453 ± (0.671)	1.405 ± (0.155)	8.678 ± (1.431)	3.118 ± (0.498)

TABLE 4.4: MO-RV-GOMEA Runtime in seconds for L-plan to reach the (-0.2, -0.2) Corner unconstrained (mean ± (standard deviation) with n = 30) . Bold markings indicate a statistically significant runtime decrease in comparison to all 3 other methods. Asterisk markings indicate a statistically significant runtime decrease in comparison to the cold start initialization. Treatment plans generated for the patient fraction ID's marked with a (-) did not (consistently) reach the Golden Corner.

In all cases, the eFNN+ExpInt initialization resulted in a significantly shorter required runtime than the cold start initialization. For patient 4b, the eFNN initialization required the shortest runtime, with an average runtime that is approximately 8 times shorter than the cold start initialization. The Heuristic+ExpInt initialization resulted in the shortest runtime on average once and outperformed the cold start initialization in 31 out of 32 cases. Considering the total average of 32 patient fractions, the eFNN+ExpInt initialization resulted in a required runtime that was approximately 6 times shorter than the cold start initialization.

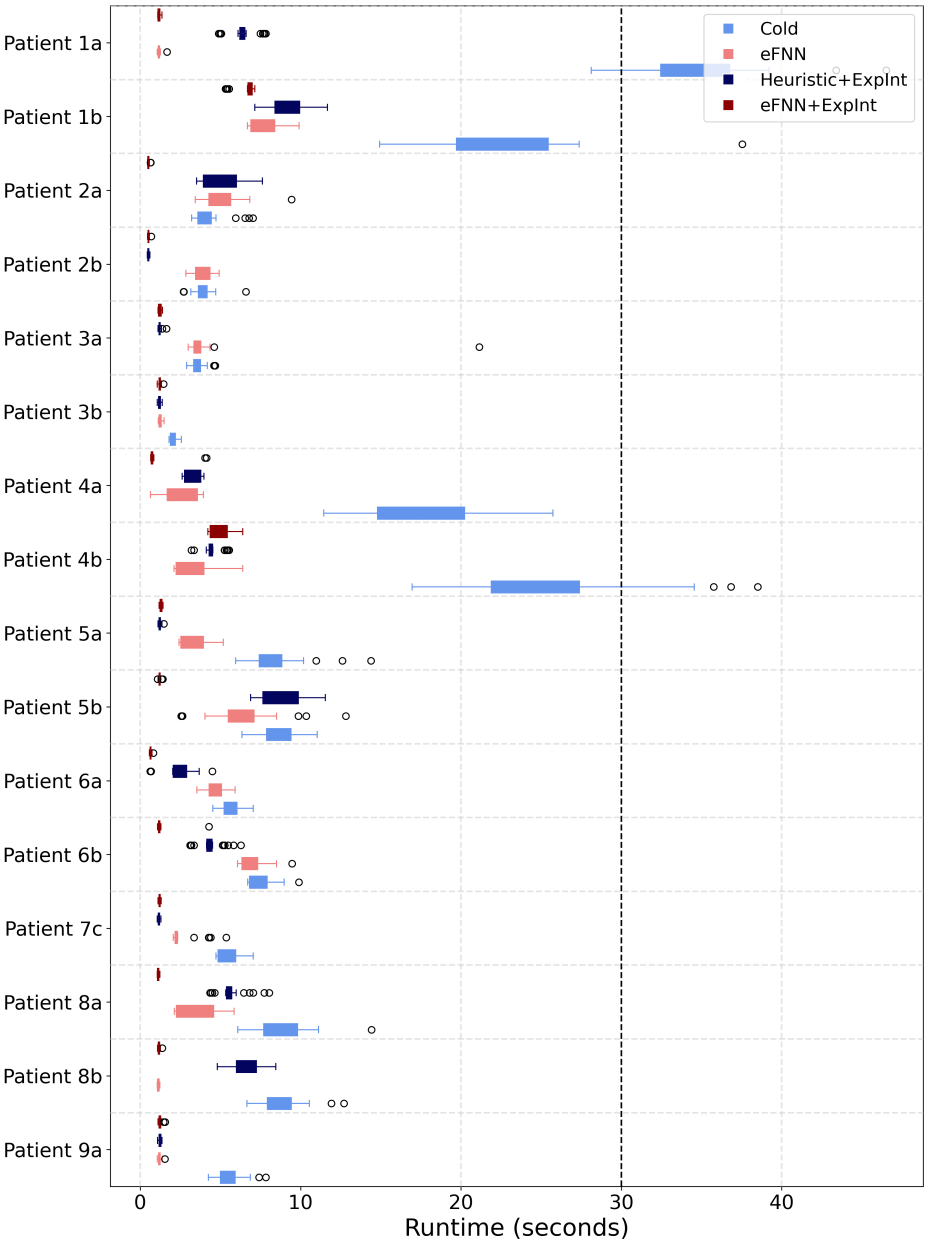


FIGURE 4.12: Part I : Runtime to reach the Golden Corner (-0.2, -0.2), n=30

To provide a visual interpretation of the spread present in the runtimes, we illustrate the runtimes per patient-fraction in Figure 4.12 and 4.13. For ease of visual comparison between plots, a black dotted line is included at the 30 second threshold. Note that in comparison to the golden corner runtime data, the eFNN+ExpInt initialization appears to produce a short required runtime more consistently. Importantly, the eFNN+ExpInt intialized MO-RV-GOMEA optimizations surpasses the (-0.2, -0.2) threshold faster for patient-fractions that could not be optimized to the golden corner in comparison to the cold start. Specifically, this is exemplified by the runtimes for patients 1a, 1b, 2a, 4b, 8a, 10a and 15a. Although the eFNN+Exp initialization required the shortest

required runtimes for most of these patient functions, the eFNN and Heuristic+ExpInt initializations outperformed it in some cases.

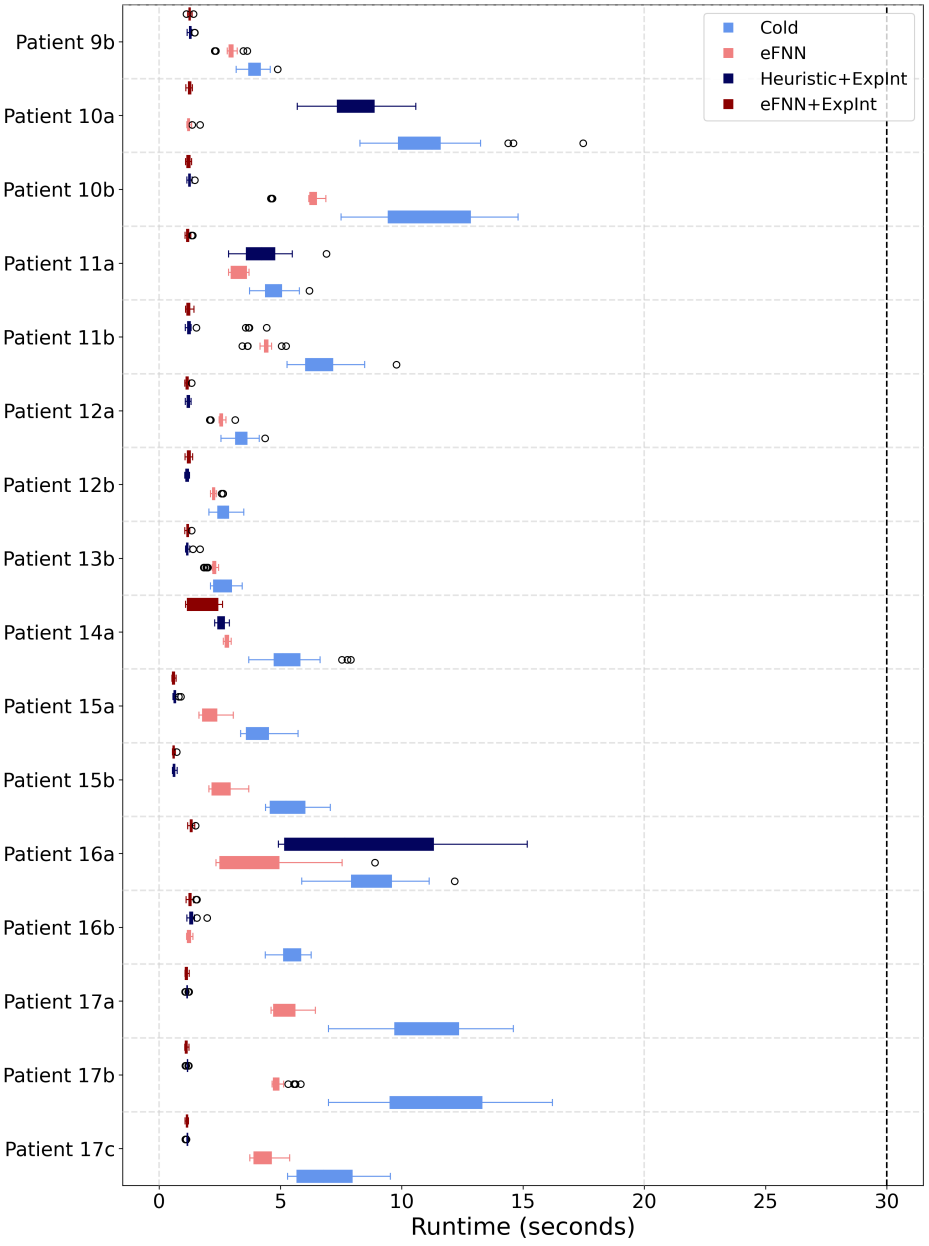


FIGURE 4.13: Part II : Runtime to reach the Golden Corner (-0.2, -0.2), n=30

Chapter 5

Discussion

As visualized by the results in Figure 4.1, the eFNN model showed that it can predict arguably under-fitted treatment plans compared to the target L-plan. Since the treatment plan results can differ considerably between retrospectively optimized MO-RV-GOMEA runs [9], fitting an FNN model to one example does not encompass all the different dwell time sequences possible for creating a good treatment plan. Yet, by using the ensemble of 5 models and applying the ExpInt method subsequentially, we achieved a broader range of promising treatment plans to start with.

Our proposed eFNN model only predicts individual dwell times, instead of predicting a complete treatment plan at once. Ideally, a machine learning model should consider the treatment plan as a whole, as there is an inherent dependency between dwell times. By including the neighbor features, we posed to give the eFNN model some contextual information about surrounding dwell points. By treating the dwell times individually, we have approximately 25,000 training examples which showed to be sufficient for stable eFNN training. Before the eFNN implementation, predicting a whole sequence instead of individual dwell times was considered as well. However, compared to individual dwell times, our dataset only includes around 300-400 sequence instances, which is a relatively small set for training a (deep-learning) sequential model architecture composed of e.g. recurrent, transformer or LSTM modules.

Although the FNN predictions do not fit the dwell time target well for every patient, we consider the LCI(t) and LSI(t) associated with that prediction to be more important. In addition, we do not aim to replace MO-RV-GOMEA with an ML approach. Instead, the eFNN model's purpose is to complement and speed up MO-RV-GOMEA. Since the eFNN predictions showed to be inconsistent, in terms of their position in the LCI(t)/LSI(t) plane, the Exp+Int method was introduced to steer towards a more

consistent starting point for the initialization of MO-RV-GOMEA. A consistent starting point is considered important since we also aim for a consistent MO-RV-GOMEA runtime towards the GC caused by the improved initialization.

For some patient fractions, the required runtime for the eFNN+ExpInt initiation did not result in the desired reduction compared to the cold start required runtime. This is mainly due to the misalignment in evaluation metrics used by the ExpInt method and MO-RV-GOMEA. In the ExpInt correction method, we adapt treatment plans based on their LCI(t) and LSI(t) values. Conversely, the re-evaluations in MO-RV-GOMEA are measured by means of the $LCI_w(t)$, $LSI_w(t)$, $LAI_w(t)$ and constraint values. Although there are slight differences between the weighted and unweighted objective functions, navigating solely on the unweighted objectives LCI(t) and LSI(t) is deemed sufficient as they represent the worst-case coverage and sparing DVIs. However, not considering the constraint value while generating an initial treatment solution has a larger impact on how the warm-start initialization is picked up by MO-RV-GOMEA.

Before MO-RV-GOMEA optimization, a range of warm-start treatment plans are evaluated inside the golden corner. However, the majority of these treatment plans show to violate constraints (e.g. DTMR and needle contribution) and are subsequentially removed from the first generation during optimization. In the elitist archive, only the least violated constraint solution is kept if all other elitist initialization inputs are constrained as well. If multiple solutions are unconstrained, only the non-dominated $LCI_w(t)$, $LSI_w(t)$, $LAI_w(t)$ are kept. How many plans are constrained for each patient-fraction is reported in Table 5.1. As observable, the eFNN initialization provides MO-RV-GOMEA with 5 plans per patient fraction of which the vast majority is constrained (0 out of 5). The eFNN+ExpInt initialization provides MO-RV-GOMEA with more warm-start solutions, although again the vast majority is constrained. Concerning Heuristic+ExpInt initialization, a larger share of the patient-fraction optimizations were shown to be inputted with unconstrained solutions.

The effect of supplying MO-RV-GOMEA with a set of solutions that are all constrained is that many of those solutions are filtered out during optimization. This can be undesirable, as providing a full range of (unconstrained) warm-start plans could improve our chances of faster MO-RV-GOMEA convergence towards the GC. However, we show that there are some cases in which one constrained warm-start solution can outperform a range of unconstrained warm-start solutions. For example, by looking at the time to reach the (-0.2, -0.2) corner (ref Table 4.4) for patient 1a. For this patient fraction, both the eFNN and eFNN+ExpInt initialization provide only constrained solutions whereas the Heuristic+ExpInt provides 67 unconstrained solutions to MO-RV-GOMEA. Yet, both the eFNN and eFNN+Expint show to outperform the Heuristic+ExpInt in

terms of time to reach the (-0.2, -0.2) corner. An opposite effect is however observable for the time to reach the GC for patient 16b (ref Table 4.3), where 60 unconstrained Heuristic+ExpInt solutions outperform the other initializations.

	eFNN	**eFNN+ExpInt**	**Heuristic+ExpInt**
Patient 1a	0/5	0/58	67/67
Patient 1b	0/5	0/12	100/100
Patient 2a	0/5	8/18	0/24
Patient 2b	3/5	0/20	0/43
Patient 3a	0/5	0/16	0/37
Patient 3b	0/5	0/56	27/27
Patient 4a	0/5	0/58	80/80
Patient 4b	0/5	0/65	34/34
Patient 5a	0/5	0/38	0/20
Patient 5b	2/5	0/13	0/6
Patient 6a	0/5	0/12	0/22
Patient 6b	0/5	0/9	0/20
Patient 7c	0/5	0/34	0/15
Patient 8a	0/5	0/19	38/39
Patient 8b	0/5	28/43	69/69
Patient 9a	0/5	1/23	58/58
Patient 9b	0/5	0/98	47/113
Patient 10a	4/5	25/52	72/72
Patient 10b	0/5	0/15	92/92
Patient 11a	0/5	20/20	0/5
Patient 11b	0/5	0/13	0/18
Patient 12a	0/5	0/30	21/47
Patient 12b	0/5	32/47	0/26
Patient 13a	0/5	0/14	0/18
Patient 13b	4/5	18/31	47/47
Patient 14a	0/5	0/22	0/49
Patient 15a	0/5	0/31	22/38
Patient 15b	0/5	0/50	1/35
Patient 16a	0/5	0/11	0/9
Patient 16b	0/5	0/82	60/60
Patient 17a	0/5	0/20	9/32
Patient 17b	0/5	0/24	0/12
Patient 17c	4/5	0/41	0/37

TABLE 5.1: (# Unconstrained warm-start Solutions / # All warm-start solutions)

Another, supposedly difficult, patient-fraction to optimize was patient 4a with an average cold start runtime of 104.25 seconds. Although the Heuristic+ExpInt initialization for this patient-fraction significantly outperformed the other initializations with 21.128 seconds of average required runtime and 80 unconstrained input solutions, the eFNN+ExpInt initialization closely follows with 24.861 seconds of average runtime and 58 constrained solutions. This exemplifies that the constraint value of warm start solutions does have an impact on the optimization process, although MO-RV-GOMEA

is able to exploit only 1 constrained solution in such a way that it still dramatically improves the required runtime to GC in comparison to the cold start.

To align the work presented in this thesis with the most recently published objectives, the $LAI_w(t)$ aim was optimized in the MO-RV-GOMEA optimization. However, we have not reported on the LAI(t) results since our main objective was to show the viability of alternative initializations in terms of how fast the golden corner is reached. Since the golden corner is still considered as a leading visual indicator for the viability of a treatment plan, we excluded the LAI(t) from our scope.

Similarly to the eFNN+ExpInt initialization, the Heuristic+ExpInt initialization showed to almost halve the required average total runtime in comparison to the cold start initialization. The Heuristic+ExpInt initialized population required approximately 20% longer runtime to reach the golden corner and approximately 30 more exponential search evaluations in comparison to the eFNN+ExpInt initialized population. However, this difference can potentially be negligible when considering the runtime before MO-RV-GOMEA optimization. For an implemented eFNN model, there is some runtime involved at inference for feature engineering and preprocessing the patient data. If this runtime is substantial, the eFNN+ExpInt initialization could turn out to be less favorable (especially for patients where the differences in required runtimes to GC between initializations are not very large). The Heuristic+ExpInt method does not require this hidden runtime cost, and is only dependent on the number of evaluations the ExpInt method needs.

Depending on the patient fraction, MO-RV-GOMEA can perform the equivalence of 1000-2000 complete evaluations (note, not partial evaluations) per second on the earlier specified hardware. To account for the runtime cost of the Exp+Int method, one could argue that the eFNN+ExpInt and Heuristic+ExpInt would add an additional 0.5 to 1.0 seconds to the average total runtime required to reach the GC. We address that this is a mere approximation, no precise indication can be given as the number of evaluations possible per second is highly dependent on the patient fraction, hardware used, and how the MO-RV-GOMEA optimization is going. To put the runtime differences in perspective, a runtime difference of 1-1.5 seconds equals roughly 1 MO-RV-GOMEA generation in our problem context.

We did not report on the actual runtime it took to perform the exponential search and interpolation. This is done on purpose since we make use of a script that externally makes re-evaluation calls to the BRIGHT command line interface. In comparison to a hypothetical fully integrated solution, this makes a big difference in runtime and providing the runtime of the external calls would only give a skewed perception of how efficient the ExpInt correction method is.

We can also question the relevance of using the ExpInt method as an addition to the eFNN model predictions since the total average runtime to reach the GC was only 1.4 seconds faster for the eFNN+ExpInt initialization. On the contrary, this small difference is potentially due to a few outliers, as the eFNN+ExpInt did outperform the eFNN for the majority of the treated patient-fractions.

Another note can be made on the varying number of interpolated treatment plan solutions produced by the ExpInt method. If the distance between the lower and upper bound is large, less interpolated solutions lie between the sparing and coverage line used as the area B thresholds. This results in a non-dominated, located in area B, variable number of warm-start solutions. Additionally, one should note that traversing the LCI (t) / LSI (t) plane by accumulating and depleting dwell times does not follow a linear path. Although the exponential search suits the purpose of our application, we can not always steer a treatment plan solution toward a desirable location on the LCI (t) / LSI (t) plane.

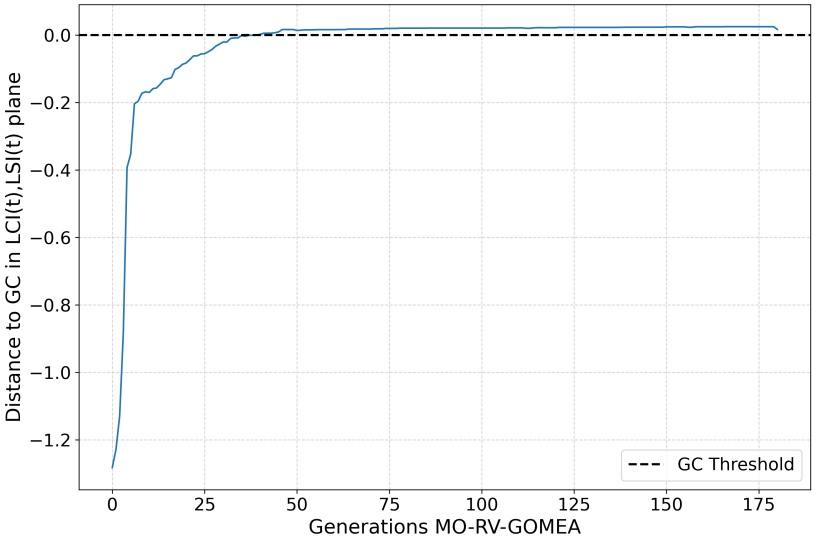


FIGURE 5.1: Patient 5b: Distance from L-plan to the golden corner for 1 cold start sample run

Exemplified by patient 5b in Table 4.3, there are limitations to how consistently we can reach a golden corner solution irrespective of the initialization method used. This is partially due to external factors we cannot correct for during the optimization, such as the catheter placement or how the tumor tissue is distributed. If the maximum achievable LCI(t) and LSI (t) for 5 minutes MO-RV-GOMEA optimization is very close to the origin of the golden corner, we can observe arguably large differences in the required runtime to reach the golden corner within a patient-fraction case. This diminishing returns effect is best visualized by plotting the L-Plan distance to the golden corner in

the LCI(t) / LSI(t) plane (Figure 5.1). In this example, after approximately 40 MO-RV-GOMEA generations, the golden corner is reached and progress toward improved solutions diminishes until it almost completely flattens out. As MO-RV-GOMEA results vary slightly for every run, and optimization progress flattens out very close to the golden corner, a wide spread of runtimes is reported in our results (for each initialization independently). Although the golden corner metric is used as the key indicator for treatment plan viability, one should be aware of the diminishing returns effect when looking at large differences in required runtime between the initialization methods.

Chapter 6

Conclusion and Future Work

In this work, we have presented a method to improve the initialization for MO-RV-GOMEA applied to cervical cancer HDR-BT treatment planning. We showed that multiple model paradigms exist in formalizing the optimization problem for HDR-BT inverse treatment planning. Although single-objective models for this problem are widely adopted, the multi-objective formulation is considered to be aligned best with the inherent multi-objective nature of the problem. EAs are particularly well suited for solving such a problem, of which MO-RV-GOMEA is a prime example.

Using MO-RV-GOMEA optimization, we built on the most recently published multi-objective model for cervical HDR-BT treatment planning. This encompasses a newly introduced third objective, developed to include additional aims deemed relevant by medical specialists at LUMC. By the introduction of a third objective, the problem formulation becomes more complex as more aims are to be satisfied and this can lead to an increase in required runtime to reach a viable treatment plan solution. Since treatment planning time in clinical practice is limited, a solution to reduce the required optimization time is seemingly relevant. The initialization of MO-RV-GOMEA poses an opportunity for this matter, as it has not yet been investigated thoroughly as part of the optimization process. Although generalized initialization schemes for EA's are widely investigated, we aimed for a custom-designed initialization that takes the context of the optimization problem into account.

With the availability of retrospectively optimized treatment plan data, we consider optimization by machine learning as a promising option to reduce the overall treatment planning time. Whereas most recent work on applying machine learning to HDR-BT and EBRT treatment planning concerns the prediction of DVHs, we focused on the direct optimization of dwell times as part of treatment plans that can be directly inserted in MO-RV-GOMEA.

By use of current HDR-BT literature, a set of relevant features was derived that served as input to our eFNN model. Sequential backward selection served as a filter method to remove any redundant features from this set. Subsequently, we investigated a range of architecture and hyper-parameter settings through a grid search experiment. Though the eFNN was optimized to the best of our ability and outputted 5 different treatment plan options, treatment plan predictions on the test set proved to be inconsistent in terms of their corresponding LCI(t) and LSI(t) values. Therefore, the exponential search & interpolation method was introduced to account for this inconsistency. By splitting the LCI(t) / LSI(t) plane into 3 sections by use of a sparing and coverage line, a framework was created for treatment plan adjustments. We showed that through exponential search with dwell time depletion/accumulation, we can controllably traverse the LCI(t) /LSI(t) plane with a limited amount of evaluations required. By interpolation and filtering out dominated solutions, a final set of treatment plans was derived that served as input to the elitist archive of MO-RV-GOMEA. To investigate the independent effect of both the eFNN model as the Exp+Int method on MO-RV-GOMEA runtime, we bench-marked the eFNN+ExpInt initialization against an eFNN and Heuristic+ExpInt initialization.

Based on the results, we show that the runtime towards reaching the golden corner was significantly reduced for the majority of patients by use of one of the warm-start initializations. Averaged over all patient treatment plans that reached the golden corner consistently, the eFNN+ExpInt had the fastest average required runtime (9.292 seconds), followed by the eFNN (10.662 seconds) and Heuristic+ExpInt (11.156 seconds). Averaged over all patient treatment plans that reached the (-0.2, -0.2) corner consistently, the eFNN+ExpInt had the fastest average required runtime (1.405 seconds), followed by the Heuristic+ExpInt (3.118 seconds) and eFNN initialization (3.453 seconds). Reiterating the central question addressed in this thesis, we showed that a warm-start solution reduced the optimization time required for MO-RV-GOMEA to reach a clinically viable treatment plan solution in cervical cancer HDR-BT treatment planning. We conclude this based on one important assumption, which is that the runtime required to create the warm-start initialization is negligible. Therefore, the work presented in this thesis serves purely a proof-of-concept purpose and should not be misunderstood as a faster initialization in clinical practice.

The first step in improving the current warm start initialization is to align the prediction better with the evaluation metrics of MO-RV-GOMEA. That is, most importantly, by explicitly correcting predictions such that they do not violate the constraints. For the DTMR constraints, this could be achieved by applying a smoothening function over the dwell time sequence, for instance by applying a 1-D convolution window. To account for the needle contribution constraint, the exponential search could purposely be set to change dwell times differently for the tandem and ovoid dwell points.

Because the runtime to create the warm-start initialization was not considered as part of the runtime reported in the results, implementing the warm-start initialization in BRIGHT would be the next proposed step in investigating its practicality. On occasion, the average best initialization (eFNN+ExpInt) showed to perform worse than the cold start initialization. Therefore, more research is required to point out why this is occasionally happening. Likewise, the warm start initializations should be validated on a larger set of patient fractions to investigate its robustness better. As there are limitations to how much patient data is available, one could also consider adding synthetic data to the dataset. For instance, by virtually deforming the target volumes or re-positioning the catheter, a wide range of treatment planning situations can be composed. These synthetic patients could then be retrospectively optimized by MO-RV-GOMEA to create a larger training dataset to which larger, more powerful neural network architectures could potentially be applied.

Although our scope was limited to applying a feedforward neural network, investigating the application of other machine learning algorithms is arguably relevant. Since a broad range of retrospectively optimized treatment plans is available, a machine learning solution that can output a large variety of target plans (instead of 1) for the same patient would be desirable. This could be achieved, for example, by applying a larger cross-validation n-fold split if more data is available. Another approach could be the application of a generative model that can be conditioned to output multiple plans based on 1 patient-fraction input.

Since the concept of warm-start initialization showed promising results in the domain of cervical cancer treatment planning, investigating its application to a closely related optimization problem such as prostate cancer treatment planning can be the next step in proving the generalizability of warm-start initialization. Application is however not limited to HDR-BT treatment planning, and other (benchmarking) problems could be investigated. Since the work in this thesis was partially motivated by the introduction of additional objectives and their causing effect on increased required runtime, research at a more fundamental level on the scalability of runtime improvement due to the warm-start initialization can be worth investigating.

Future work can also be focused on improving the exponential search & interpolation method. The interpolation part now takes up a majority of the evaluations, while only a fraction of these evaluations turn out to be useful after filtering out dominated solutions. Removing redundant treatment plans that follow presumably a dominated LCI(t) and LSI(t) combination, could reduce the number of required evaluations further. Additionally, for the exponential search navigation, a more precise objective formulation could be applied. For example, by considering all the weighted MO-RV-GOMEA objectives

instead of the unweighted LCI(t) and LSI(t) values. Furthermore, including information about the LAI(t) could give a better illustration of how good the initial treatment plan is in comparison to solely judging by the golden corner.

Appendix A

Data Analysis

This section provides a brief overview of the exploratory data analysis performed. To prevent any bias in our feature engineering and modelling approach, the data was split into a train, validation and test set before the data analysis. We make solely use of the training and validation data in Appendix A.

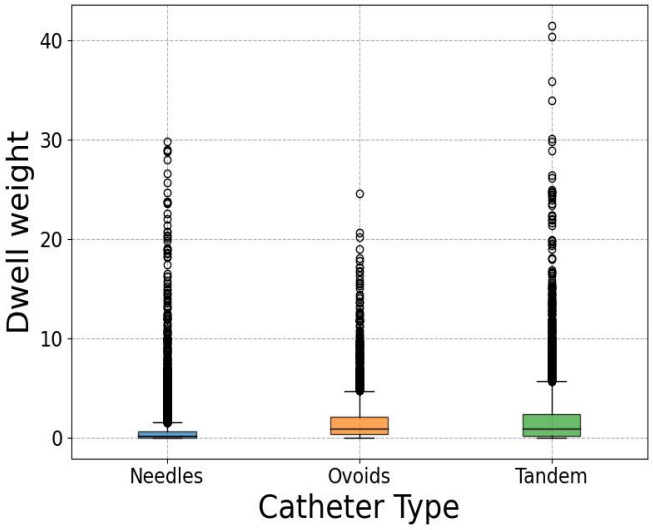


FIGURE A.1: Target Dwell Weights split by Catheter Type

Figure A.1 indicates the target dwell weight distribution, which is considerably right-skewed with a tail of outliers reaching dwell weights up to 40. The bottom boxplot whisker indicates the first quartile minus 1.5 times the inter-quartile range (IQR) and the top boxplot whisker indicates the third quartile plus 1.5 times the IQR. In line with the cold start heuristic, needle dwell weights are distributed closer to zero. Whereas ovoid and tandem dwell weights are slightly higher on average.

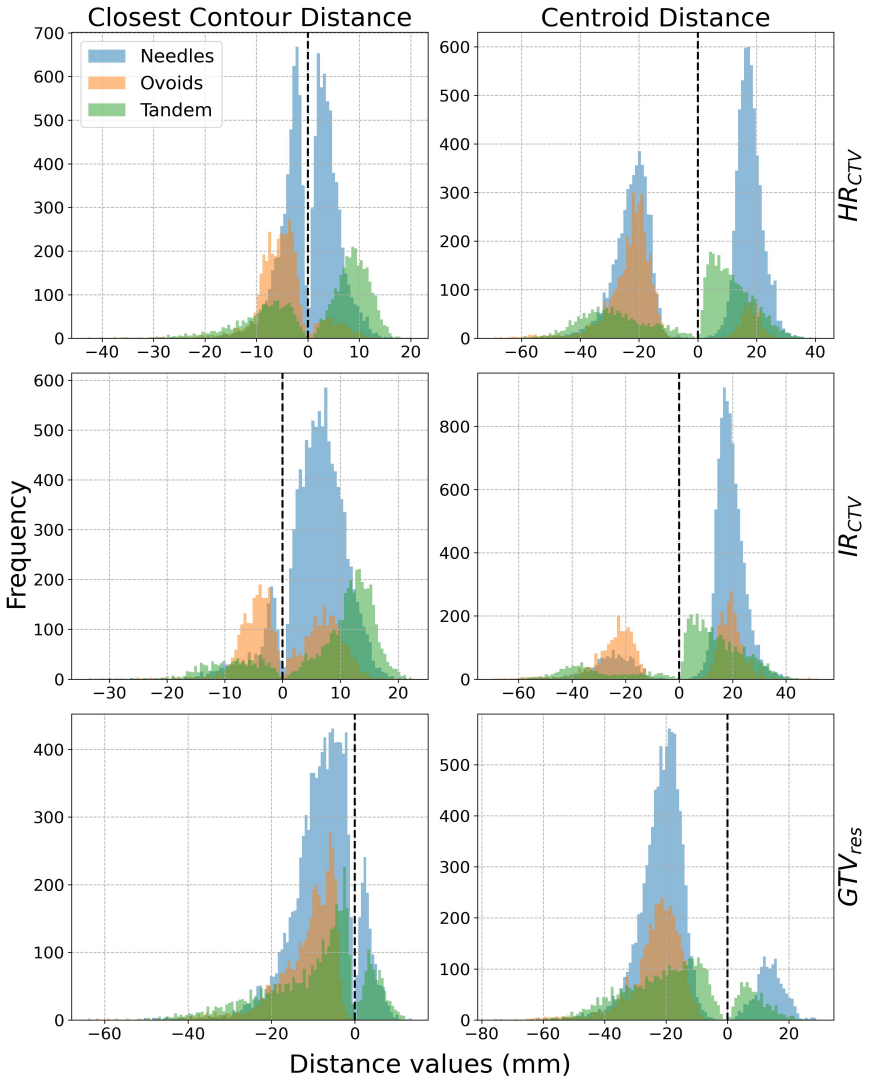


FIGURE A.2: Distance distribution CTV's.

As mentioned in Section 3.1, information on whether a dwell point is in or outside the target volume is incorporated in the distance features. Figure A.2 denotes the distribution of the closest contour distance from a dwell point to the target volume. A black dotted line is provided as a reference for the outside (negative distances) or inside (positive distances) boundary. Ovoid and tandem dwell points outside of the target volumes show to have the largest closest contour distance, which is potentially due to the anatomical constraints. For example, the rearmost dwell points in the tandem might not reach the target volume and the ovoids cannot go beyond the external OS. Furthermore, the majority of the closest contour distances show to be within the high-volume IR_{CTV}. This is in contrast to the low-volume GTV_{res}, where the majority of dwell points are placed outside of its target area.

A final remark should be made on the absence of tandem dwell points inside the GTV_{res}. The absence can be a consequence of the tandem 'nearly missing' the GTV_{res} or due

to the chosen training/validation/test data split. In practice, a clinician would aim for tandem placement inside the GTV_{res}.

Figure A.3 provides an overview of the distance between dwell points and the closest contours of OARs. Generally, dwell points closest to the contours show to be needle dwell points. Comparing this with Figure A.1, one can argue that turning off needles is necessary to spare the close-by organs.

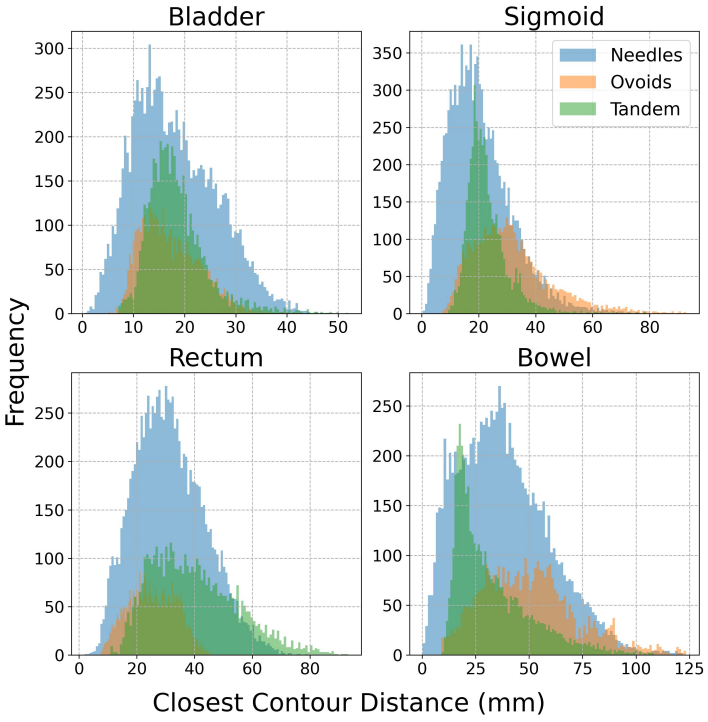


FIGURE A.3: Distance distribution OAR's

To illustrate the relation between dwell weights and how well a dwell point is surrounded by target volume, refer to Figure A.4. Dwell weights that are distanced outside the target volumes have a sample count of zero or close to zero. Radii for sampling are set at 2.5 mm for the GTV_{RES}, 7.5 mm for HR_{CTV} and 10 mm for IR_{CTV} to account for their respectively different volumes. Although the presence of turned-off dwell points for every sample count, one can observe that higher dwell weights appear more frequently when the sample count is higher. Pearson and Spearman correlations in Table A.1 and A.2 provide a respectively linear and non-linear view on the relation between the dwell weights and features. Only correlations lower than -0.1 and higher than 0.1 are reported. The largest positive correlations are present in the Closest Contour Distance and Is Inside ROI Samples Count features for IR_{CTV} and HR_{CTV}. The largest negative correlation is present between the dwell weight and the cathether type. This is mainly due to the numbering order of catheters, where the highest value (3) represents the needles. As a consequence of needles being turned off most frequently of all catheter

Feature	**Pearson Correlation**
Closest Contour Distance IR_{CTV}	0,161
Closest Contour Distance HR_{CTV}	0,160
Is Inside ROI Samples Count HR_{CTV}	0,144
Inferior Contour Distance HR_{CTV}	0,142
Posterior Contour Distance HR_{CTV}	0,129
Closest Contour Distance GTV_{res}	0,122
Centroid Distance GTV_{res}	0,121
Furthest Contour Distance HR_{CTV}	0,120
Centroid Distance HR_{CTV}	0,118
Lateral Left Contour Distance HR_{CTV}	0,116
Furthest Contour Distance GTV_{res}	0,111
Is Inside ROI Samples Count IR_{CTV}	0,105
Neighbors In Radius 12.5	-0,126
Catheter Type	-0,203

TABLE A.1: Pearson Correlations between features and the dwell weight target

types, the negative correlation can be partially explained by this effect. Whereas the correlations provide some information about how dwell target weights are determined, their values remain considerably weak (no correlations above 0.5 or below -0.5).

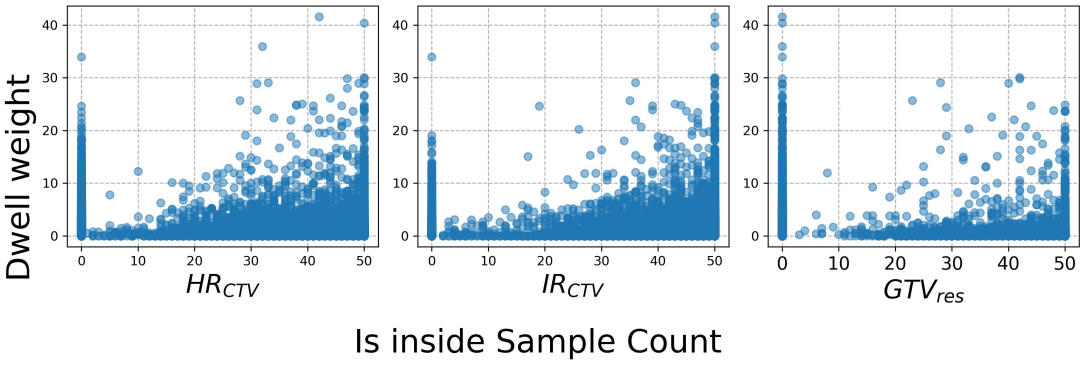


FIGURE A.4: Target Dwell Weights vs. Is Inside Sample Count

Figure A.5 shows the Pearson correlations to provide an interpretation for in-between feature relations. Positive correlations are mostly prevalent between distance features from the target volumes and between the Neighbors In Radius features (indicated by the fairly red squares around the diagonal). High multicollinearities can result in performance decreases, especially when fitting a linear regression model. However, we apply a non-linear eFNN model, which is shown to be less prone to this problem [59]. Secondly, we consider all the features to be informative to some extent for our modelling purpose. Therefore, we do not trim features because of their highly correlating properties.

Feature	**Spearman Correlation**
Closest Contour Distance IR_{CTV}	0,292
Is Inside ROI Samples Count IR_{CTV}	0,245
Closest Contour Distance HR_{CTV}	0,240
Is Inside ROI Samples Count HR_{CTV}	0,222
Inferior Contour Distance HR_{CTV}	0,191
Posterior Contour Distance HR_{CTV}	0,188
Furthest Contour Distance HR_{CTV}	0,185
Furthest Contour Distance GTV_{res}	0,181
Centroid Distance HR_{CTV}	0,177
Centroid Distance GTV_{res}	0,171
Closest Contour Distance GTV_{res}	0,170
Lateral Left Contour Distance HR_{CTV}	0,164
Lateral Right Contour Distance HR_{CTV}	0,143
Neighbors In Radius 5.0	0,137
Anterior Contour Distance GTV_{res}	0,136
Anterior Contour Distance HR_{CTV}	0,115
Inferior Contour Distance GTV_{res}	0,106
Lateral Left Contour Distance GTV_{res}	0,104
Is Inside ROI Samples Count GTV_{res}	0,102
Neighbors In Radius 12.5	-0,145
Catheter Type	-0,377

TABLE A.2: Spearman Correlations between features and the dwell weight target

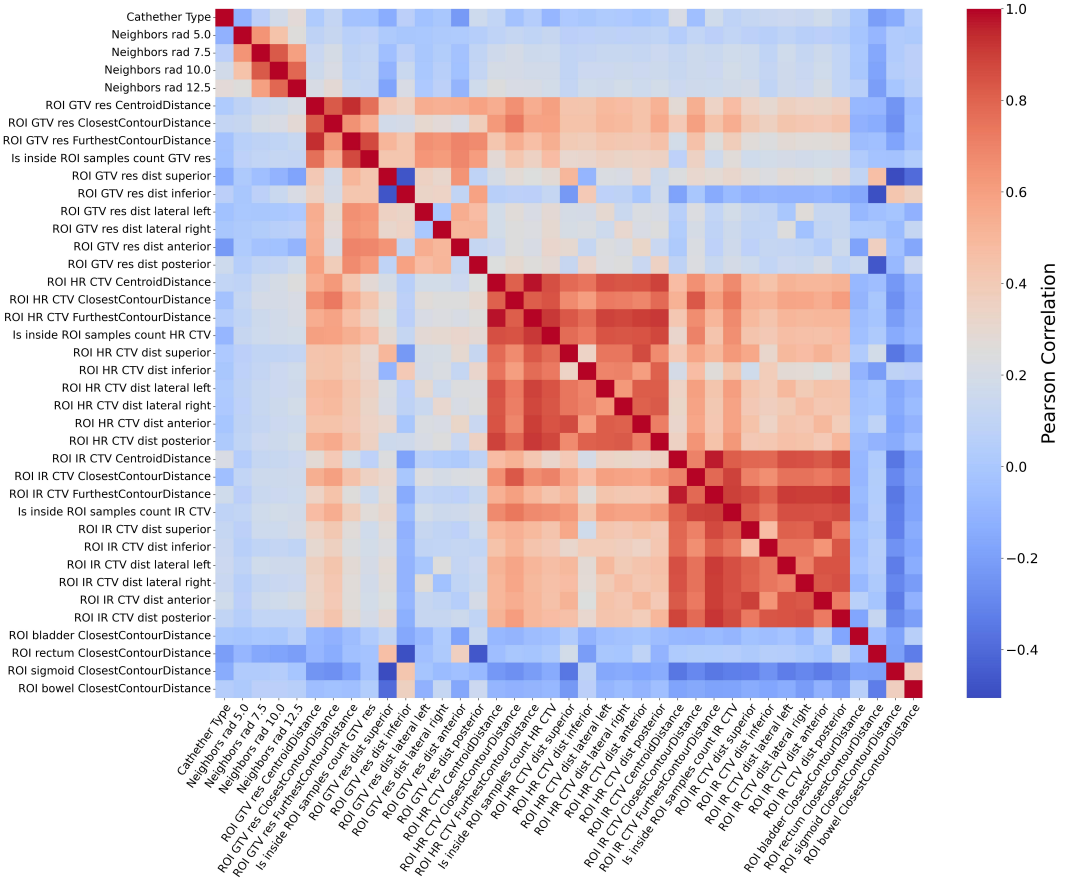


FIGURE A.5: Pearson Correlation Matrix

Appendix B

Feature Selection

Feature selection is applied to a total of 159 features using sequential backward selection (SBS). The minimum number of features is set to 1. This results in a worst-case scenario of $\frac{159(159+1)}{2} - 1 = 12719$ training/evaluation cycles (estimated at 5 minutes per cycle on 4 Nvidia Titan X(P) 68-84 cores GPUs) to complete SBS. To decrease this computational cost, certain features in the iterative removal procedure are grouped. The anatomical distances (AD) superior, inferior, lateral left, lateral right, anterior and posterior are grouped under the AD suffix for each target volume. Additionally, the neighboring features are grouped by the feature type they are applied to. This is indicated by the neighbor groups (NG) suffix. Grouping the features leads to an SBS initial feature set of size 44, which results in a considerably lower number of training/evaluation cycles $\frac{44(44+1)}{2} - 1 = 989$ in the computationally most expensive scenario. The dwell

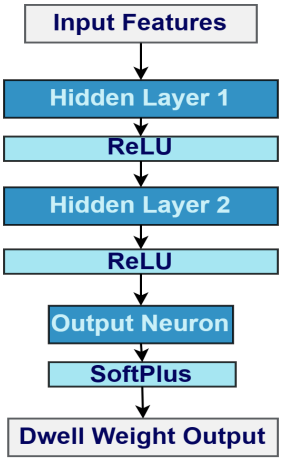


FIGURE B.1: Baseline Architecture for Feature Selection

point features are inputted to a 3-layer FNN as depicted in Figure B.1 with a batch size of 64 and a hidden layer size of 128. A Softplus activation is applied to the output neuron to constrain the dwell weight output to be positive. Dwell weight predictions

are evaluated through Huberloss with a δ of 1. As an optimizer, we apply Adam-W with a learning rate of 0.00003 and a weight decay of 0.001. The β range for Adam-W is set to $(0.9, 0.999)$ and an epsilon of $1e-8$. Training is progressed for a maximum of 500 epochs if the improvement threshold is not reached. The improvement threshold is reached if validation loss performance has not improved for 25 epochs.
By applying a 5-fold cross-validation approach, we can take the mean validation Huberloss of the converged 5 models and use this to assess the performance of a feature set. Ideally, multiple runs of each cross-validation and feature set would be necessary to robustly estimate the performance. Without increasing the number of runs, we aim to estimate the performance by removing any stochasticity in the training procedure. This is done by fixing the seed (333) for each FNN initialization and by deliberate baseline architecture design (e.g. avoiding the use of dropout).

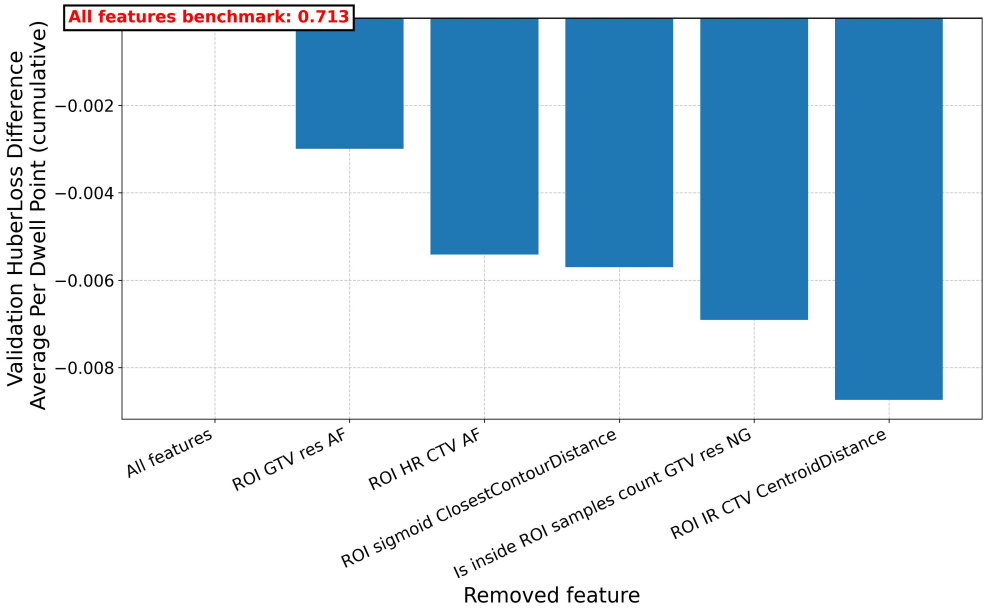


FIGURE B.2: Final SBS results

Figure B.2 shows the cumulative decrease in loss for removing each of the depicted (groups of) features. The removal of the anatomical distances for the GTV_{res} and HR_{CTV} resulted in the largest Huberloss improvement in the first 2 SBS iterations. Removing the sigmoid closest contour distance only resulted in a marginally small improvement. In the 5th and final iteration, the IR_{CTV} centroid distance is deemed unnecessary which can be due to its close resemblance with the HR_{CTV} centroid distance. In total, the average validation huberloss was decreased by approximately 1% by using the final SBS feature set.

To provide a brief comparison of features of which removal did result in a higher Huberloss, the first SBS iteration result is shown in figure B.3. Removing the catheter type feature shows the strongest increase in Huberloss, indicating its relative importance in determining the dwell weight. Removing other features show an impact on the loss performance of less than 1%, varying between an absolute difference of 0.005 and 0.0.

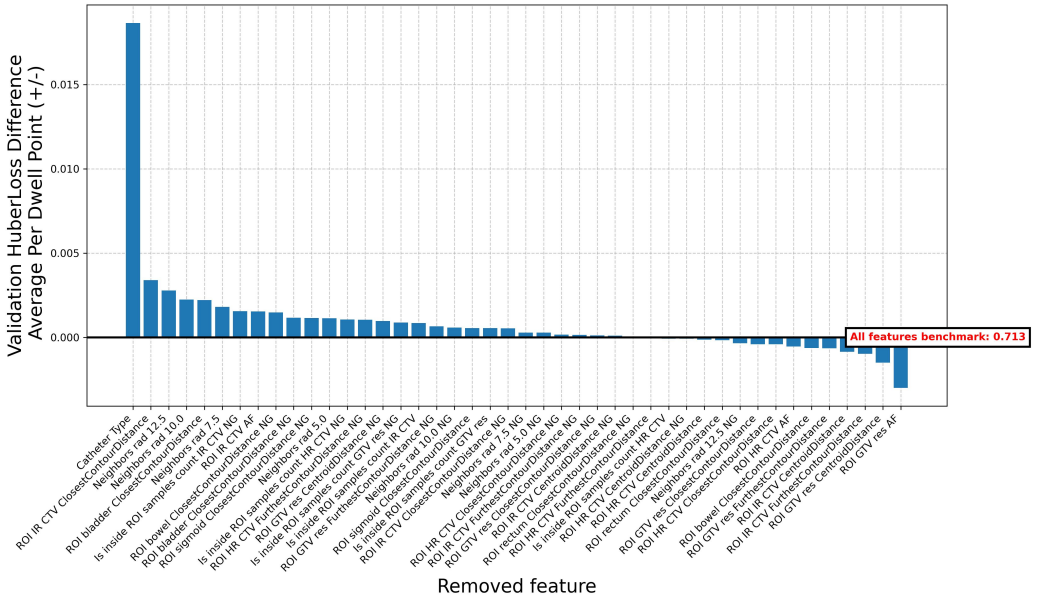


FIGURE B.3: First iteration SBS

Appendix C

Model architecture and hyperparameter grid search

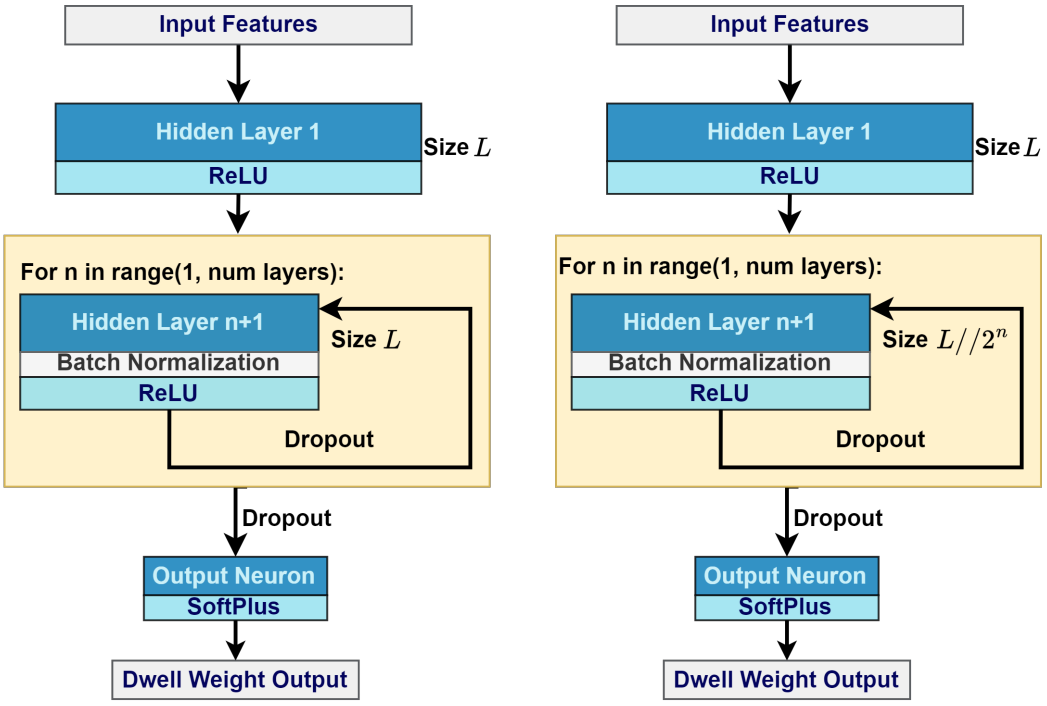
Feedforward neural network architectures can be constructed in many different ways. To identify the most suitable architecture and hyperparameters for our optimization task, a grid search experiment is performed. For a complete overview of the search parameters, refer to Table C.1. The learning rates are implemented in combination with an exponential learning rate scheduler, of which the gamma parameters are determined by manual tuning. For a learning rate of 0.0001, a gamma of 0.99 is applied and for a learning rate of 0.00001, a gamma of 0.95 is applied. Apart from exploring best-practice hidden layer sizes of 256 and 512, a hidden layer size that equals the input size is also considered (139).

Grid Search Parameters	**Values**
Learning rate	$\{0.0001, 0.00001\}$
Weight decay	$\{0.01, 0.001, 0.0001\}$
Hidden Layer Size	$\{139, 256, 512\}$
# Layers	$\{3, 4, 5\}$
Dropout Probability	$\{0.0, 0.25, 0.5\}$
Optimizers	$\{SGD, ADAMW\}$
Narrow Down Architecture	$\{True, False\}$
Nesterov Momentum[1]	$\{0.5, 0.99\}$
LR Scheduler gamma[2]	$\{0.99, 0, 95\}$

Table C.1: Grid Search Space

We evaluate two optimizers, of which stochastic gradient descent (SGD) is implemented with Nesterov momentum to improve convergence towards a global minimum. In both AdamW and SGD, 3 weight decay terms for L2-regularization are evaluated. The β

[1] Only applied to SGD
[2] Dependent on learning rate



(A) Constant Architecture (B) Narrow Down Architecture

FIGURE C.1: Grid Search Architectures

range for Adam-W is set to $(0.9, 0.999)$ and $\epsilon = 1e^{-8}$. Dropouts are performed after each hidden layer, excluding the input and output layers. Finally, so-called narrow-down architectures are evaluated which closely resemble an encoder-like structure towards the final output. When a narrow-down architecture is applied, the hidden layer size halves for every consecutive hidden layer starting from the first hidden layer. A schematic visualization of the narrow-down architecture in comparison to a constant architecture is given in Figure C.1b for a hidden layer size L and n layers.

Similar to the experiment in Appendix B, we perform 5-fold cross-validation with a batch size of 64 dwell points and set the random seed to 333. We use Huberloss with $\delta = 1$ and train for a maximum of 500 epochs if the improvement threshold of 25 epochs is not reached.

	Model #1	**Model #2**	**Model #3**	**Model #4**	**Model #5**
Learning Rate	0.00001	0.00001	0.00001	0.00001	0.00001
Weight Decay	0.01	0.01	0.01	0.0001	0.001
Momentum	-	-	0.99	0.99	-
Hidden Layer Size	139	139	139	256	139
No. Layers	4	4	4	4	4
Dropout Probability	0.25	0.5	0.5	0.5	0.5
Optimizer	ADAMW	ADAMW	SGD	SGD	ADAMW
Narrow Down	True	False	False	True	False
5-fold Average CV Huberloss per Dwell Weight	0.7001	0.7026	0.7030	0.7030	0.7031

TABLE C.2: Top 5 Grid Search Models

	Model #-1	**Model #-2**	**Model #-3**	**Model #-4**	**Model #-5**
Learning Rate	0.00001	0.00001	0.00001	0.00001	0.00001
Weight Decay	0.0001	0.001	0.01	0.001	0.001
Momentum	0.5	0.5	0.5	0.5	0.5
Hidden Layer Size	139	139	256	139	256
No. Layers	5	5	5	5	5
Dropout Probability	0.5	0.5	0.5	0.5	0.5
Optimizer	SGD	SGD	SGD	SGD	SGD
Narrow Down	True	False	True	True	True
5-fold Average CV Huberloss per Dwell Weight	0.7873	0.7815	0.7786	0.7779	0.7739

TABLE C.3: Bottom 5 Grid Search

In total, 972 configurations are assessed. The top 5 configurations are depicted in Table C.2 and the bottom 5 configurations in Table C.3. Each model in the top 5 configurations is trained with a learning rate of $1e^{-5}$ and has 4 layers. Other hyperparameter inputs vary, though the absence of a 0.0 dropout probability and a 512 hidden layer size indicate that fully connected, larger architectures did not result in the best performance for the optimization task at hand.

From the 5 configurations with the highest average Huberloss, a more consistent hyperparameter pattern arises. Each model has a learning rate of $1e^{-5}$, a Nesterov momentum of 0.5, includes 5 layers, has a dropout probability of 0.5 and uses stochastic gradient descent as an optimizer.

Since the grid search experiment only does one 5-fold cross-validation run for each configuration, each of the top 5 configurations are evaluated again by performing 30 consecutive runs. For an overview of the convergence, refer to Figure C.2. For all model/CV split combinations, there is a consistent difference between the validation and training loss. In some cases, the validation loss converges above the training loss, whereas in other cases the validation loss is lower than the training loss. With some exceptions, both the training and validation loss is following a similar path for each cv split. In for example the combination CV split 1 and Model #1, the training and validation loss each follows either one of 2 trajectories. This possibly indicates the presence of two optimization minima for that particular CV split, if one excludes the deviating run in CV split 1, Model #3. The number of iterated epochs towards converges also differs per split and model. For example, CV split 1 converges relatively fast around 100 epochs whereas CV split 5 converges consistently above 300 epochs. This difference is likely due to data imbalance within each CV split. In some splits, the validation data represents the training data more consistently (e.g. CV split 5) whereas in other splits this alignment is less present (e.g. CV split 1). On average, Model #2 performed best with an average validation Huberloss of 0.703. (see Table C.4). Therefore, the Model #2 configuration will be adopted in our final framework.

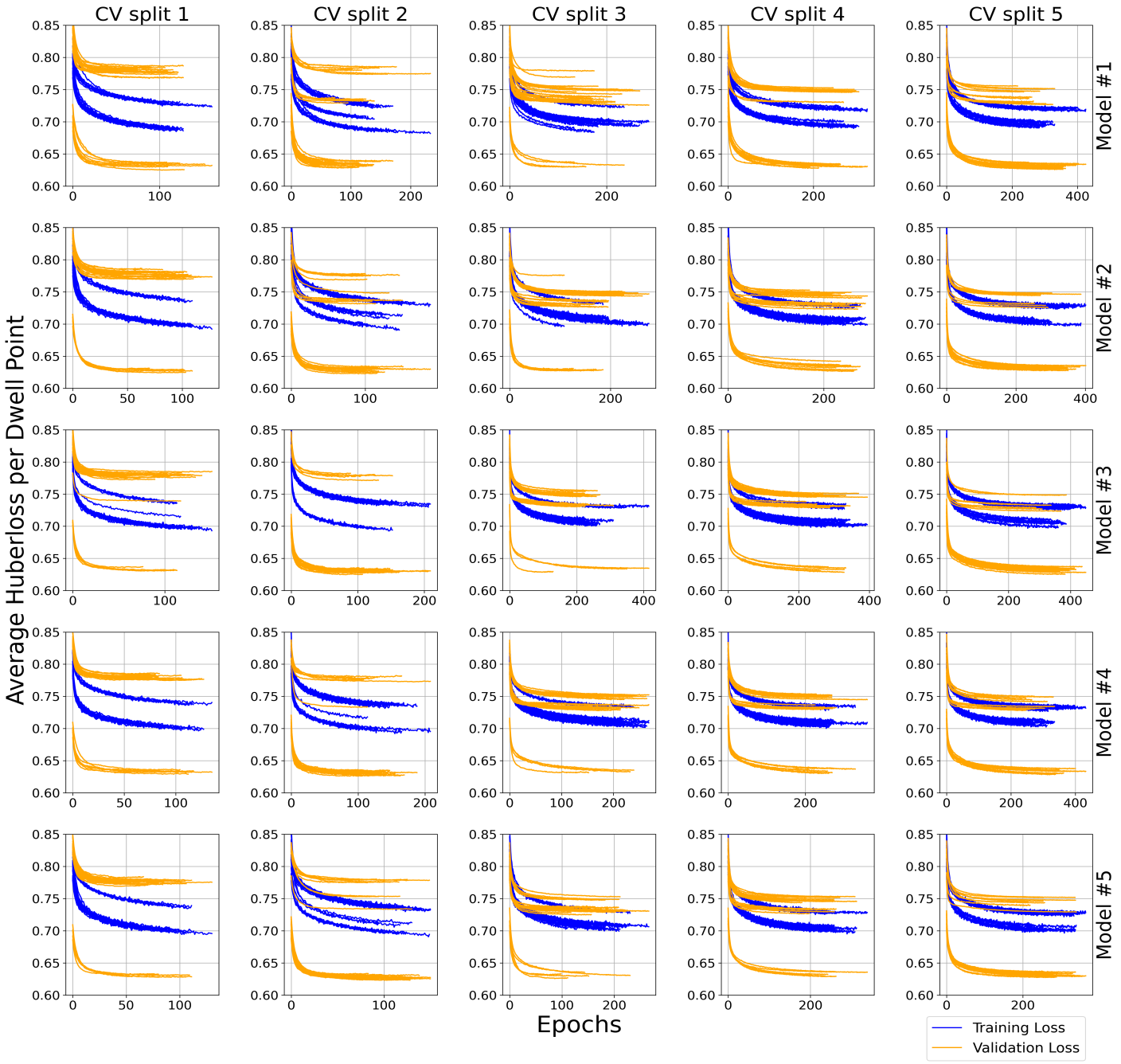


FIGURE C.2: Top 5 grid search models convergence plots, n = 30

CV-Split Model	**Validation Huberloss**
Model #1	0.705 ± 0.062
Model #2	0.703 ± 0.062
Model #3	0.705 ± 0.062
Model #4	0.705 ± 0.061
Model #5	0.704 ± 0.062

TABLE C.4: Top 5 Grid Search Models: Average validation Huberloss per dwell weight (n=30)

Appendix D

Statistical Test Results

		eFNN	eFNN+ExpInt	Cold	Cold+ExpInt
Patient 2b	**eFNN**	-	3.0161e-11	0.4553	3.0161e-11
	eFNN+ExpInt	-	-	3.0161e-11	0.4160
	Cold	-	-	-	3.0161e-11
	Cold+ExpInt	-	-	-	-
Patient 3a	**eFNN**	-	0.0180	0.0023	0.0001
	eFNN+ExpInt	-	-	1.3594e-07	1.6947e-09
	Cold	-	-	-	0.2838
	Cold+ExpInt	-	-	-	-
Patient 3b	**eFNN**	-	0.5249	0.0001	8.8862e-10
	eFNN+ExpInt	-	-	0.0000	3.3520e-08
	Cold	-	-	-	3.4971e-09
	Cold+ExpInt	-	-	-	-
Patient 4a	**eFNN**	-	7.2208e-06	3.3384e-11	3.6459e-08
	eFNN+ExpInt	-	-	3.0199e-11	8.8829e-06
	Cold	-	-	-	3.0199e-11
	Cold+ExpInt	-	-	-	-
Patient 5a	**eFNN**	-	6.1210e-10	0.0163	0.0001
	eFNN+ExpInt	-	-	4.0772e-11	4.8011e-07
	Cold	-	-	-	1.6980e-08
	Cold+ExpInt	-	-	-	-
Patient 5b	**eFNN**	-	0.6952	0.4553	0.7845
	eFNN+ExpInt	-	-	0.4825	0.4825
	Cold	-	-	-	0.1958
	Cold+ExpInt	-	-	-	-
Patient 6a	**eFNN**	-	2.3168e-06	0.0018	0.0067
	eFNN+ExpInt	-	-	0.0021	1.1737e-09
	Cold	-	-	-	2.0152e-08
	Cold+ExpInt	-	-	-	-
Patient 6b	**eFNN**	-	0.0011	0.0002	0.1453
	eFNN+ExpInt	-	-	2.3897e-08	2.6784e-06
	Cold	-	-	-	0.0035
	Cold+ExpInt	-	-	-	-
Patient 7c	**eFNN**	-	0.0037	0.0000	5.5999e-07
	eFNN+ExpInt	-	-	0.0144	5.0912e-06
	Cold	-	-	-	0.0012
	Cold+ExpInt	-	-	-	-
Patient 8b	**eFNN**	-	3.8302e-06	3.0199e-11	3.0199e-11
	eFNN+ExpInt	-	-	3.0123e-11	3.0123e-11
	Cold	-	-	-	1.0277e-06
	Cold+ExpInt	-	-	-	-
Patient 9a	**eFNN**	-	0.0003	6.0621e-11	3.0180e-11
	eFNN+ExpInt	-	-	3.0199e-11	3.6897e-11
	Cold	-	-	-	3.0199e-11
	Cold+ExpInt	-	-	-	-
Patient 9b	**eFNN**	-	3.0180e-11	3.9631e-08	3.0161e-11
	eFNN+ExpInt	-	-	3.0161e-11	0.1038
	Cold	-	-	-	3.0142e-11
	Cold+ExpInt	-	-	-	-

TABLE D.1: (I) Time to Golden Corner Comparison. P-values for a double-sided Mann-Whitney U test with Bonferoni Correction ($\alpha = 0.01/6$ and n = 30).

		eFNN	eFNN+ExpInt	Cold	Cold+ExpInt
Patient 10b	**eFNN**	-	3.6897e-11	3.6874e-11	6.6955e-11
	eFNN+ExpInt	-	-	0.0392	0.0001
	Cold	-	-	-	0.0001
	Cold+ExpInt	-	-	-	-
Patient 11a	**eFNN**	-	3.0199e-11	3.0199e-11	3.0199e-11
	eFNN+ExpInt	-	-	3.0199e-11	3.0199e-11
	Cold	-	-	-	0.1715
	Cold+ExpInt	-	-	-	-
Patient 11b	**eFNN**	-	2.4374e-09	3.0161e-11	0.0000
	eFNN+ExpInt	-	-	3.0180e-11	8.1014e-10
	Cold	-	-	-	2.0328e-09
	Cold+ExpInt	-	-	-	-
Patient 12a	**eFNN**	-	6.5277e-08	0.0000	3.0180e-11
	eFNN+ExpInt	-	-	0.0281	3.0180e-11
	Cold	-	-	-	3.0180e-11
	Cold+ExpInt	-	-	-	-
Patient 12b	**eFNN**	-	3.0161e-11	6.1210e-10	0.0003
	eFNN+ExpInt	-	-	3.0161e-11	3.0142e-11
	Cold	-	-	-	1.6251e-08
	Cold+ExpInt	-	-	-	-
Patient 13b	**eFNN**	-	4.6108e-10	5.0896e-06	0.4688
	eFNN+ExpInt	-	-	4.4960e-11	4.5681e-09
	Cold	-	-	-	0.0001
	Cold+ExpInt	-	-	-	-
Patient 14a	**eFNN**	-	0.0004	3.0199e-11	0.0000
	eFNN+ExpInt	-	-	3.0199e-11	2.0275e-07
	Cold	-	-	-	3.0180e-11
	Cold+ExpInt	-	-	-	-
Patient 15b	**eFNN**	-	3.0161e-11	3.0199e-11	3.1967e-09
	eFNN+ExpInt	-	-	3.0161e-11	0.0000
	Cold	-	-	-	3.0199e-11
	Cold+ExpInt	-	-	-	-
Patient 16a	**eFNN**	-	0.0000	0.0001	5.0695e-10
	eFNN+ExpInt	-	-	3.0199e-11	3.0180e-11
	Cold	-	-	-	1.8602e-06
	Cold+ExpInt	-	-	-	-
Patient 16b	**eFNN**	-	0.0042	3.0199e-11	3.0180e-11
	eFNN+ExpInt	-	-	3.0199e-11	3.0180e-11
	Cold	-	-	-	3.0180e-11
	Cold+ExpInt	-	-	-	-
Patient 17a	**eFNN**	-	0.0021	3.0199e-11	0.0173
	eFNN+ExpInt	-	-	3.0180e-11	0.1334
	Cold	-	-	-	3.0180e-11
	Cold+ExpInt	-	-	-	-
Patient 17b	**eFNN**	-	0.4204	3.0199e-11	9.2603e-09
	eFNN+ExpInt	-	-	3.0199e-11	1.5465e-09
	Cold	-	-	-	3.0199e-11
	Cold+ExpInt	-	-	-	-
Patient 17c	**eFNN**	-	9.7555e-10	3.0199e-11	0.0093
	eFNN+ExpInt	-	-	3.0199e-11	0.0000
	Cold	-	-	-	3.0199e-11
	Cold+ExpInt	-	-	-	-

TABLE D.2: (II) Time to Golden Corner Comparison. P-values for a double-sided Mann-Whitney U test with Bonferoni Correction ($\alpha = 0.01/6$ and n = 30).

		eFNN	eFNN+ExpInt	Cold	Cold+ExpInt
Patient 1a	**eFNN**	-	0.5945	3.0161e-11	3.0161e-11
	eFNN+ExpInt	-	-	3.0199e-11	3.0199e-11
	Cold	-	-	-	3.0199e-11
	Cold+ExpInt	-	-	-	-
Patient 1b	**eFNN**	-	0	3.0199e-11	4.7941e-07
	eFNN+ExpInt	-	-	3.0161e-11	3.3321e-11
	Cold	-	-	-	3.0180e-11
	Cold+ExpInt	-	-	-	-
Patient 2a	**eFNN**	-	3.0161e-11	0.0018	0.8883
	eFNN+ExpInt	-	-	3.0142e-11	3.0161e-11
	Cold	-	-	-	0.0144
	Cold+ExpInt	-	-	-	-
Patient 2b	**eFNN**	-	3.0142e-11	0.4289	3.0142e-11
	eFNN+ExpInt	-	-	3.0123e-11	0.416
	Cold	-	-	-	3.0123e-11
	Cold+ExpInt	-	-	-	-
Patient 3a	**eFNN**	-	3.0066e-11	0.5692	3.0142e-11
	eFNN+ExpInt	-	-	3.0029e-11	0.9941
	Cold	-	-	-	3.0104e-11
	Cold+ExpInt	-	-	-	-
Patient 3b	**eFNN**	-	0.6843	3.0161e-11	0.2549
	eFNN+ExpInt	-	-	3.0123e-11	0.5106
	Cold	-	-	-	3.0180e-11
	Cold+ExpInt	-	-	-	-
Patient 4a	**eFNN**	-	9.0446e-08	3.0161e-11	0.0009
	eFNN+ExpInt	-	-	3.0142e-11	8.4727e-09
	Cold	-	-	-	3.0199e-11
	Cold+ExpInt	-	-	-	-
Patient 4b	**eFNN**	-	1.4913e-06	3.0199e-11	0
	eFNN+ExpInt	-	-	3.0180e-11	0.0147
	Cold	-	-	-	3.0199e-11
	Cold+ExpInt	-	-	-	-
Patient 5a	**eFNN**	-	3.0142e-11	3.0199e-11	3.0180e-11
	eFNN+ExpInt	-	-	3.0142e-11	0.0001
	Cold	-	-	-	3.0180e-11
	Cold+ExpInt	-	-	-	-
Patient 5b	**eFNN**	-	3.0180e-11	7.2208e-06	1.8608e-06
	eFNN+ExpInt	-	-	3.0180e-11	3.0180e-11
	Cold	-	-	-	0.1669
	Cold+ExpInt	-	-	-	-
Patient 6a	**eFNN**	-	3.0180e-11	1.3052e-07	1.9568e-10
	eFNN+ExpInt	-	-	3.0180e-11	4.8806e-08
	Cold	-	-	-	3.0199e-11
	Cold+ExpInt	-	-	-	-
Patient 6b	**eFNN**	-	3.0142e-11	0.0051	4.9752e-11
	eFNN+ExpInt	-	-	3.0123e-11	1.2035e-10
	Cold	-	-	-	3.0180e-11
	Cold+ExpInt	-	-	-	-
Patient 7c	**eFNN**	-	3.0180e-11	1.4626e-10	3.0142e-11
	eFNN+ExpInt	-	-	3.0180e-11	0.0033
	Cold	-	-	-	3.0142e-11
	Cold+ExpInt	-	-	-	-

TABLE D.3: (I) Time to (-0.2, -0.2) Corner Comparison. P-values for a double-sided Mann-Whitney U test with Bonferoni Correction ($\alpha = 0.01/6$ and n = 30).

		eFNN	eFNN+ExpInt	Cold	Cold+ExpInt
Patient 8a	**eFNN**	-	3.0142e-11	3.0180e-11	5.8569e-06
	eFNN+ExpInt	-	-	3.0123e-11	3.0123e-11
	Cold	-	-	-	3.8159e-10
	Cold+ExpInt	-	-	-	-
Patient 8b	**eFNN**	-	0.0537	3.0199e-11	3.0199e-11
	eFNN+ExpInt	-	-	3.0142e-11	3.0142e-11
	Cold	-	-	-	4.4205e-06
	Cold+ExpInt	-	-	-	-
Patient 9a	**eFNN**	-	0.2197	3.0142e-11	0.3365
	eFNN+ExpInt	-	-	3.0123e-11	0.7007
	Cold	-	-	-	3.0161e-11
	Cold+ExpInt	-	-	-	-
Patient 9b	**eFNN**	-	3.0123e-11	5.4773e-11	3.0104e-11
	eFNN+ExpInt	-	-	3.0142e-11	0.1038
	Cold	-	-	-	3.0123e-11
	Cold+ExpInt	-	-	-	-
Patient 10a	**eFNN**	-	0.0877	3.0161e-11	3.0142e-11
	eFNN+ExpInt	-	-	3.0180e-11	3.0161e-11
	Cold	-	-	-	3.1951e-09
	Cold+ExpInt	-	-	-	-
Patient 10b	**eFNN**	-	3.0142e-11	3.0161e-11	3.0161e-11
	eFNN+ExpInt	-	-	3.0142e-11	0.0085
	Cold	-	-	-	3.0161e-11
	Cold+ExpInt	-	-	-	-
Patient 11a	**eFNN**	-	3.0199e-11	3.0180e-11	9.5332e-07
	eFNN+ExpInt	-	-	3.0180e-11	3.0199e-11
	Cold	-	-	-	0.0002
	Cold+ExpInt	-	-	-	-
Patient 11b	**eFNN**	-	3.0142e-11	3.0180e-11	3.4703e-10
	eFNN+ExpInt	-	-	3.0161e-11	0.5058
	Cold	-	-	-	3.0180e-11
	Cold+ExpInt	-	-	-	-
Patient 12a	**eFNN**	-	3.0161e-11	3.8138e-10	3.0142e-11
	eFNN+ExpInt	-	-	3.0180e-11	0.0101
	Cold	-	-	-	3.0161e-11
	Cold+ExpInt	-	-	-	-
Patient 12b	**eFNN**	-	3.0142e-11	3.9613e-08	3.0104e-11
	eFNN+ExpInt	-	-	3.0142e-11	0.0038
	Cold	-	-	-	3.0104e-11
	Cold+ExpInt	-	-	-	-
Patient 13b	**eFNN**	-	3.0161e-11	0.001	3.0180e-11
	eFNN+ExpInt	-	-	3.0123e-11	0.1536
	Cold	-	-	-	3.0142e-11
	Cold+ExpInt	-	-	-	-
Patient 14a	**eFNN**	-	3.0180e-11	3.0180e-11	6.2735e-06
	eFNN+ExpInt	-	-	3.0199e-11	3.7996e-07
	Cold	-	-	-	3.0161e-11
	Cold+ExpInt	-	-	-	-
Patient 15a	**eFNN**	-	3.0180e-11	3.0180e-11	3.0161e-11
	eFNN+ExpInt	-	-	3.0161e-11	0.0004
	Cold	-	-	-	3.0142e-11
	Cold+ExpInt	-	-	-	-
Patient 15b	**eFNN**	-	3.0123e-11	3.0180e-11	3.0104e-11
	eFNN+ExpInt	-	-	3.0142e-11	0.1601
	Cold	-	-	-	3.0123e-11
	Cold+ExpInt	-	-	-	-

TABLE D.4: (II) Time to (-0.2, -0.2) Corner Comparison. P-values for a double-sided Mann-Whitney U test with Bonferoni Correction ($\alpha = 0.01/6$ and n = 30).

		eFNN	**eFNN+ExpInt**	**Cold**	**Cold+ExpInt**
Patient 16a	**eFNN**	-	3.0161e-11	5.0723e-10	2.8314e-08
	eFNN+ExpInt	-	-	3.0161e-11	3.0161e-11
	Cold	-	-	-	0.4553
	Cold+ExpInt	-	-	-	-
Patient 16b	**eFNN**	-	0.0443	3.0180e-11	0.0004
	eFNN+ExpInt	-	-	3.0161e-11	0.0251
	Cold	-	-	-	3.0180e-11
	Cold+ExpInt	-	-	-	-
Patient 17a	**eFNN**	-	3.0066e-11	3.0199e-11	2.9953e-11
	eFNN+ExpInt	-	-	3.0066e-11	0.1645
	Cold	-	-	-	2.9953e-11
	Cold+ExpInt	-	-	-	-
Patient 17b	**eFNN**	-	3.0180e-11	3.0199e-11	2.9729e-11
	eFNN+ExpInt	-	-	3.0180e-11	0.0001
	Cold	-	-	-	2.9729e-11
	Cold+ExpInt	-	-	-	-
Patient 17c	**eFNN**	-	3.0123e-11	8.9934e-11	2.9673e-11
	eFNN+ExpInt	-	-	3.0123e-11	0.0404
	Cold	-	-	-	2.9673e-11
	Cold+ExpInt	-	-	-	-

TABLE D.5: (III) Time to (-0.2, -0.2) Corner Comparison. P-values for a double-sided Mann-Whitney U test with Bonferoni Correction ($\alpha = 0.01/6$ and n = 30).

Appendix E

Exponential Search Evaluations - Complete Table Results

Patient Fraction	Exponential Search # Evaluations - Cold Start	Exponential Search # Evaluations - eFNN
Patient 1a	165	126
Patient 1b	165	189
Patient 2a	225	183
Patient 2b	209	184
Patient 3a	259	233
Patient 3b	165	215
Patient 4a	165	132
Patient 4b	165	128
Patient 5a	170	141
Patient 5b	185	151
Patient 6a	227	205
Patient 6b	225	214
Patient 7a	245	205
Patient 7b	245	205
Patient 7c	187	146
Patient 8a	165	140
Patient 8b	153	134
Patient 9a	145	128
Patient 9b	185	148
Patient 10a	245	186
Patient 10b	234	137
Patient 11a	219	170
Patient 11b	205	185
Patient 12a	180	145
Patient 12b	205	175
Patient 13a	178	148
Patient 13b	149	147
Patient 14a	229	197
Patient 15a	165	138
Patient 15b	166	134
Patient 16a	185	135
Patient 16b	147	123
Patient 17a	188	187
Patient 17b	191	186
Patient 17c	225	203

Table E.1: No. Evaluations per patient-fraction and initialization.

Bibliography

[1] Anton Bouter, Ngoc Hoang Luong, Cees Witteveen, Tanja Alderliesten, and Peter AN Bosman. The multi-objective real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 537–544, 2017.

[2] Anton Bouter, Tanja Alderliesten, Bradley R Pieters, Arjan Bel, Yury Niatsetski, and Peter AN Bosman. Gpu-accelerated bi-objective treatment planning for prostate high-dose-rate brachytherapy. *Medical physics*, 46(9):3776–3787, 2019.

[3] Anton Bouter, Tanja Alderliesten, and Peter AN Bosman. Achieving highly scalable evolutionary real-valued optimization by exploiting partial evaluations. *Evolutionary computation*, 29(1):129–155, 2021.

[4] Ngoc Hoang Luong, Anton Bouter, Marjolein C Van Der Meer, Yury Niatsetski, Cees Witteveen, Arjan Bel, Tanja Alderliesten, and Peter AN Bosman. Efficient, effective, and insightful tackling of the high-dose-rate brachytherapy treatment planning problem for prostate cancer using evolutionary multi-objective optimization algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1372–1379, 2017.

[5] Ngoc Hoang Luong, Tanja Alderliesten, Arjan Bel, Yury Niatsetski, and Peter AN Bosman. Application and benchmarking of multi-objective evolutionary algorithms on high-dose-rate brachytherapy planning for prostate cancer treatment. *Swarm and Evolutionary Computation*, 40:37–52, 2018.

[6] Ngoc Hoang Luong, Tanja Alderliesten, Bradley R Pieters, Arjan Bel, Yury Niatsetski, and Peter AN Bosman. Fast and insightful bi-objective optimization for prostate cancer treatment planning with high-dose-rate brachytherapy. *Applied Soft Computing*, 84:105681, 2019.

[7] Danique LJ Barten, A Bouter, Niek van Wieringen, Bradley R Pieters, Karel A Hinnen, GH Westerveld, Stef C Maree, Marjolein C van der Meer, Tanja Alderliesten, Peter Bosman, et al. Artificial intelligence based planning of hdr prostate brachytherapy: first clinical experience. 2021.

[8] Leah RM Dickhoff, Ellen M Kerkhof, Heloisa H Deuzeman, Carien L Creutzberg, Tanja Alderliesten, and Peter AN Bosman. Adaptive objective configuration in bi-objective evolutionary optimization for cervical cancer brachytherapy treatment planning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1173–1181, 2022.

[9] Leah RM Dickhoff, Renzo J Scholman, Danique LJ Barten, Ellen M Kerkhof, Jelmen J Roorda, Laura A Velema, Lukas JA Stalpers, Bradley R Pieters, Peter AN Bosman, Tanja Alderliesten, and Anton Bouter. Keeping your best options open with ai-based treatment planning in prostate and cervix brachytherapy. *Brachytherapy*, 2024.

[10] Liesje De Boeck, Jeroen Beliën, and Wendy Egyed. Dose optimization in high-dose-rate brachytherapy: A literature review of quantitative models from 1990 to 2010. *Operations Research for Health Care*, 3(2):80–90, 2014.

[11] Björn Morén, Torbjörn Larsson, and Åsa Carlsson Tedgren. Optimization in treatment planning of high dose-rate brachytherapy—review and analysis of mathematical models. *Medical Physics*, 48(5):2057–2082, 2021.

[12] WD Renner, TP O'Connor, and NM Bermudez. An algorithm for generation of implant plans for high-dose-rate irradiators. *Medical physics*, 17(1):35–40, 1990.

[13] Etienne Lessard and Jean Pouliot. Inverse planning anatomy-based dose optimization for hdr-brachytherapy of the prostate using fast simulated annealing algorithm and dedicated objective function. *Medical physics*, 28(5):773–779, 2001.

[14] Ron Alterovitz, Etienne Lessard, Jean Pouliot, I-Chow Joe Hsu, James F O'Brien, and Ken Goldberg. Optimization of hdr brachytherapy dose distributions using linear programming with penalty costs. *Medical physics*, 33(11):4012–4019, 2006.

[15] Åsa Holm, Torbjörn Larsson, and Åsa Carlsson Tedgren. Impact of using linear optimization models in dose planning for hdr brachytherapy. *Medical physics*, 39 (2):1021–1028, 2012.

[16] Etienne Lessard, I-Chow Hsu, and Jean Pouliot. Inverse planning for interstitial gynecologic template brachytherapy: truly anatomy-based planning. *International Journal of Radiation Oncology* Biology* Physics*, 54(4):1243–1251, 2002.

[17] N Milickovic, M Lahanas, M Papagiannopoulou, N Zamboglou, and D Baltas. Multiobjective anatomy-based dose optimization for hdr-brachytherapy with constraint free deterministic algorithms. *Physics in Medicine & Biology*, 47(13):2263, 2002.

[18] M Lahanas, D Baltas, and S Giannouli. Global convergence analysis of fast multiobjective gradient-based dose optimization algorithms for high-dose-rate brachytherapy. *Physics in Medicine & Biology*, 48(5):599, 2003.

[19] Yuhong Zhou, Peter Klages, Jun Tan, Yujie Chi, Strahinja Stojadinovic, Ming Yang, Brian Hrycushko, Paul Medin, Arnold Pompos, Steve Jiang, et al. Automated high-dose rate brachytherapy treatment planning for a single-channel vaginal cylinder applicator. *Physics in Medicine & Biology*, 62(11):4361, 2017.

[20] Sebastiaan Breedveld, Amit BA Bennan, Shafak Aluwini, Dennis R Schaart, Inger-Karine K Kolkman-Deurloo, and Ben JM Heijmen. Fast automated multi-criteria planning for hdr brachytherapy explored for prostate cancer. *Physics in Medicine & Biology*, 64(20):205002, 2019.

[21] Enrique Chajon, Isabelle Dumas, Mahmoud Touleimat, Nicolas Magné, Jérémy Coulot, Rodolfe Verstraet, Dimitri Lefkopoulos, and Christine Haie-Meder. Inverse planning approach for 3-d mri-based pulse-dose rate intracavitary brachytherapy in cervix cancer. *International Journal of Radiation Oncology* Biology* Physics*, 69 (3):955–961, 2007.

[22] D Baltas, Z Katsilieri, V Kefala, S Papaioannou, A Karabis, P Mavroidis, and N Zamboglou. Influence of modulation restriction in inverse optimization with hipo of prostate implants on plan quality: analysis using dosimetric and radiobiological indices. In *World Congress on Medical Physics and Biomedical Engineering, September 7-12, 2009, Munich, Germany: Vol. 25/1 Radiation Oncology*, pages 283–286. Springer, 2009.

[23] Adam Cunha, Timmy Siauw, I-Chow Hsu, and Jean Pouliot. A method for restricting intracatheter dwell time variance in high-dose-rate brachytherapy plan optimization. *Brachytherapy*, 15(2):246–251, 2016.

[24] H Badry, L Oufni, H Ouabi, H Iwase, and L Afraites. A new fast algorithm to achieve the dose uniformity around high dose rate brachytherapy stepping source using tikhonov regularization. *Australasian Physical & Engineering Sciences in Medicine*, 42:757–769, 2019.

[25] Åsa Holm, Torbjörn Larsson, and Åsa Carlsson Tedgren. On the correlation between dvh parameters and linear penalties in optimization of hdr prostate brachytherapy dose plans. *Digitala Vetenskapliga Arkivet*, 2011.

[26] Jeroen Beliën, Jan Colpaert, Liesje De Boeck, and Erik Demeulemeester. A hybrid simulated annealing linear programming approach for treatment planning in hdr brachytherapy with dose volume constraints. In *Proceedings of the 35th International Conference on Operational Research Applied to Health Services (ORAHS)*, 2009.

[27] Timo M Deist and Bram L Gorissen. High-dose-rate prostate brachytherapy inverse planning on dose-volume criteria by simulated annealing. *Physics in Medicine & Biology*, 61(3):1155, 2016.

[28] Bram L Gorissen, Dick Den Hertog, and Aswin L Hoffmann. Mixed integer programming improves comprehensibility and plan quality in inverse optimization of prostate hdr brachytherapy. *Physics in Medicine & Biology*, 58(4):1041, 2013.

[29] Åsa Holm, Torbjörn Larsson, and Åsa Carlsson Tedgren. A linear programming model for optimizing hdr brachytherapy dose distributions with respect to mean dose in the dvh-tail. *Medical physics*, 40(8):081705, 2013.

[30] Christopher L Deufel, Marina A Epelman, Kalyan S Pasupathy, Mustafa Y Sir, Victor W Wu, and Michael G Herman. Pnav: A tool for generating a high-dose-rate brachytherapy treatment plan by navigating the pareto surface guided by the visualization of multidimensional trade-offs. *Brachytherapy*, 19(4):518–531, 2020.

[31] Michael Lahanas, D Baltas, and N Zamboglou. Anatomy-based three-dimensional dose optimization in brachytherapy using multiobjective genetic algorithms. *Medical Physics*, 26(9):1904–1918, 1999.

[32] Marjolein C van der Meer, Arjan Bel, Yury Niatsetski, Tanja Alderliesten, Bradley R Pieters, and Peter AN Bosman. Robust evolutionary bi-objective optimization for prostate cancer treatment with high-dose-rate brachytherapy. In *Parallel Problem Solving from Nature–PPSN XVI: 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part II 16*, pages 441–453. Springer, 2020.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[34] Vasant Kearney, Jason W Chan, Samuel Haaf, Martina Descovich, and Timothy D Solberg. Dosenet: a volumetric dose prediction algorithm using 3d fully-convolutional neural networks. *Physics in Medicine & Biology*, 63(23):235022, 2018.

[35] Katherina G Cortes, Karoline Kallis, Aaron Simon, Jyoti Mayadev, Sandra M Meyers, and Kevin L Moore. Knowledge-based three-dimensional dose prediction for tandem-and-ovoid brachytherapy. *Brachytherapy*, 21(4):532–542, 2022.

[36] Ximeng Mao, Joelle Pineau, Roy Keyes, and Shirin A Enger. Rapidbrachydl: rapid radiation dose calculations in brachytherapy via deep learning. *International Journal of Radiation Oncology* Biology* Physics*, 108(3):802–812, 2020.

[37] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[38] Aaron Babier, Rafid Mahmood, Andrea L McNiven, Adam Diamant, and Timothy CY Chan. Knowledge-based automated planning with three-dimensional generative adversarial networks. *Medical Physics*, 47(2):297–306, 2020.

[39] Tajwar Abrar Aleef, Ingrid T Spadinger, Michael D Peacock, Septimiu E Salcudean, and S Sara Mahdavi. Rapid treatment planning for low-dose-rate prostate brachytherapy with tp-gan. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 581–590. Springer, 2021.

[40] Tahir I Yusufaly, Karoline Kallis, Aaron Simon, Jyoti Mayadev, Catheryn M Yashar, John P Einck, Loren K Mell, Derek Brown, Daniel Scanderbeg, Sebastian J Hild, et al. A knowledge-based organ dose prediction tool for brachytherapy treatment planning of patients with cervical cancer. *Brachytherapy*, 19(5):624–634, 2020.

[41] Gang Pu, Shan Jiang, Zhiyong Yang, Yuanjing Hu, and Ziqi Liu. Deep reinforcement learning for treatment planning in high-dose-rate cervical brachytherapy. *Physica Medica*, 94:1–7, 2022.

[42] Dirk Thierens. The linkage tree genetic algorithm. In *International Conference on Parallel Problem Solving from Nature*, pages 264–273. Springer, 2010.

[43] Dirk Thierens and Peter AN Bosman. Optimal mixing evolutionary algorithms. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 617–624, 2011.

[44] Anton Bouter, Tanja Alderliesten, Arjan Bel, Cees Witteveen, and Peter AN Bosman. Large-scale parallelization of partial evaluations in evolutionary algorithms for real-world problems. In *Proceedings of the genetic and evolutionary computation conference*, pages 1199–1206, 2018.

[45] Anton Bouter and Peter AN Bosman. Gpu-accelerated parallel gene-pool optimal mixing in a gray-box optimization setting. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 675–683, 2022.

[46] Embrace. Embrace study, 2016. URL https://www.embracestudy.dk/Public/Default.aspx?main=1&sub=3&embrace=embrace. Accessed on 6th of March, 2024.

[47] Alaa Tharwat and Wolfram Schenck. Population initialization techniques for evolutionary algorithms for single-objective constrained optimization problems: Deterministic vs. stochastic techniques. *Swarm and Evolutionary Computation*, 67: 100952, 2021.

[48] Makoto Matsumoto and Takuji Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8(1):3–30, 1998.

[49] John H Halton. Algorithm 247: Radical-inverse quasi-random point sequence. *Communications of the ACM*, 7(12):701–702, 1964.

[50] Il'ya Meerovich Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802, 1967.

[51] Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

[52] Eneko Osaba, Roberto Carballedo, Fernando Diaz, Enrique Onieva, P Lopez, and Asier Perallos. On the influence of using initialization functions on genetic algorithms solving combinatorial optimization problems: a first study on the tsp. In *2014 IEEE conference on evolving and adaptive intelligent systems (EAIS)*, pages 1–6. IEEE, 2014.

[53] Qian Li, San-Yang Liu, and Xin-She Yang. Influence of initialization on the performance of metaheuristic optimizers. *Applied Soft Computing*, 91:106193, 2020.

[54] Jeffrey O Agushaka, Absalom E Ezugwu, Laith Abualigah, Samaher Khalaf Alharbi, and Hamiden Abd El-Wahed Khalifa. Efficient initialization methods for population-based metaheuristic algorithms: A comparative study. *Archives of Computational Methods in Engineering*, 30(3):1727–1787, 2023.

[55] Ferdinando Pezzella, Gianluca Morganti, and Giampiero Ciaschetti. A genetic algorithm for the flexible job-shop scheduling problem. *Computers & operations research*, 35(10):3202–3212, 2008.

[56] Sushil J Louis and John McDonnell. Learning with case-injected genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(4):316–328, 2004.

[57] Tankut Yalcinoz and Halis Altun. Power economic dispatch using a hybrid genetic algorithm. *IEEE power engineering review*, 21(3):59–60, 2001.

[58] Scott D Roth. Ray casting for modeling solids. *Computer graphics and image processing*, 18(2):109–144, 1982.

[59] Richard D De Veaux and Lyle H Ungar. Multicollinearity: A tale of two non-parametric regressions. In *Selecting models from data: artificial intelligence and statistics IV*, pages 393–402. Springer, 1994.