DISSERTATION
submitted to the
Combined Faculties of Natural Sciences and Mathematics
of Heidelberg University, Germany
for the degree of
DOCTOR OF NATURAL SCIENCES

Put forward by
JENS PETERSEN
born in Hamburg, Germany

Oral examination: December 9, 2020

LEARNING DISTRIBUTIONS OF FUNCTIONS ON A CONTINUOUS TIME DOMAIN

—

CONTRIBUTIONS IN THE CONTEXT OF IMAGE-BASED GLIOMA GROWTH ANALYSIS

Referees: Prof. Dr. Dr. Jürgen Debus
Prof. Dr. Klaus H. Maier-Hein

ABSTRACT

This work presents several contributions on the topic of learning representations of function spaces, as well as on learning the dynamics of glioma growth as a particular instance thereof. We begin with two preparatory efforts, showing how expert knowledge can be leveraged efficiently in an interactive segmentation context, and presenting a proof of concept for inferring non-deterministic glioma growth patterns purely from data. The remainder of our work builds upon the framework of Neural Processes. We show how these models represent function spaces and discover that they can implicitly decompose the space into different frequency components, not unlike a Fourier transform. In this context we derive an upper bound on the maximum signal frequency Neural Processes can represent and show how to control the learned representations to only contain certain frequencies. We continue with an improvement of a more recent addition to the Neural Process family called ConvCNP, which we combine with a Gaussian Process to make it non-deterministic and to improve generalization. Finally, we show how to perform segmentation in the Neural Process framework by extending a typical segmentation architecture with spatio-temporal attention. The resulting model can interpolate complex spatial variations of segmentations over time and, applied to glioma growth, it is able to represent multiple temporally consistent growth trajectories, exhibiting realistic and diverse spatial growth patterns.

ZUSAMMENFASSUNG

Diese Arbeit präsentiert mehrere Beiträge zum Thema lernbarer Repräsentationen von Funktionsräumen, sowie zum Lernen der Wachstumsdynamiken von Gliomen als ein Anwendungsbeispiel davon. Wir präsentieren zunächst zwei vorbereitende Leistungen und zeigen dabei, wie Expertenwissen im Kontext interaktiver Segmentierung möglichst effizient genutzt werden kann. Zudem präsentieren wir als Proof of Concept, wie Wachstumsmuster des Glioms ausschließlich von Daten inferiert werden können. Der Rest unserer Arbeit baut auf so genannten Neural Processes auf. Wir zeigen wie diese Modelle Funktionsräume repräsentieren und entdecken, dass sie implizit eine Frequenzzerlegung vornehmen können, ähnlich einer Fourier-Transformation. In diesem Kontext leiten wir eine obere Grenze für Frequenzen her, die dargestellt werden können, und zeigen wie die gelernten Repräsentationen kontrolliert werden können, sodass nur bestimmte Frequenzen darin enthalten sind. Weiterhin präsentieren wir eine Erweiterung zu einem neueren Mitglied der Neural Process-Familie, ConvCNP genannt, welches wir mit einem Gaussian Process kombinieren, um es nicht-deterministisch zu machen und seine Generalisierung zu verbessern. Zuletzt zeigen wir, wie man Segmentierung im Rahmen von Neural Processes abbilden kann, indem wir eine typische Segmentierungsarchitektur mit einem zeitlich-räumlichen Attention-Mechanismus ausstatten. Das resultierende Modell kann komplexe räumliche Variationen einer Segmentierung über die Zeit interpolieren und kann, in der Anwendung auf Gliom-Wachstum, mehrere zeitlich konsistente Wachstums-Trajektorien darstellen, wobei letztere sowohl divers als auch realistisch sind.

PRIMARY PUBLICATIONS

Petersen, Jens, Paul F. Jäger, Fabian Isensee, Simon A. A. Kohl, Ulf Neuberger, Wolfgang Wick, Jürgen Debus, Sabine Heiland, Martin Bendszus, Philipp Kickingereder, and Klaus H. Maier-Hein (2019). "Deep Probabilistic Modeling of Glioma Growth". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 806–814.

Petersen, Jens, Sabine Heiland, Martin Bendszus, Jürgen Debus, Marco Nolden, Caspar J. Goch, and Klaus H. Maier-Hein (2018). "Leveraging Open Source Software to Close Translational Gaps in Medical Image Computing (Abstract)". In: *Bildverarbeitung für die Medizin*, pp. 22–22.

Petersen, Jens, Martin Bendszus, Jürgen Debus, Sabine Heiland, and Klaus H. Maier-Hein (2017a). "Effective User Interaction in Online Interactive Semantic Segmentation of Glioblastoma Magnetic Resonance Imaging". In: *Journal of Medical Imaging* 4.3, p. 034001.

Petersen, Jens, Martin Bendszus, Jürgen Debus, Sabine Heiland, and Klaus H. Maier-Hein (2017b). "Effective User Guidance in Online Interactive Semantic Segmentation". In: *SPIE Medical Imaging*.

Petersen, Jens, Sabine Heiland, Martin Bendszus, Jürgen Debus, and Klaus H. Maier-Hein (2017c). "Quantification of Guidance Strategies in Online Interactive Semantic Segmentation of Glioblastoma MRI". In: *Bildverarbeitung für die Medizin*, pp. 231–236.

Petersen, Jens, Martin Bendszus, Jürgen Debus, Sabine Heiland, and Klaus H. Maier-Hein (2016). "A Software Application for Interactive Medical Image Segmentation with Active User Guidance". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI) – Interactive Medical Image Computing Workshop*.

UNDER REVIEW

Petersen, Jens, Paul Jäger, Gregor Köhler, David Zimmerer, Fabian Isensee, and Klaus H. Maier-Hein (2020a). "Frequency Decomposition in Neural Processes". In: *International Conference on Learning Representations (under review)*.

Petersen, Jens, Gregor Köhler, David Zimmerer, Fabian Isensee, Paul Jäger, and Klaus H. Maier-Hein (2020b). "GP-ConvCNP: Improving Generalization in Convolutional Conditional Neural Processes". In: *AAAI Conference on Artificial Intelligence (under review)*.

OTHER PUBLICATIONS

Isensee, Fabian, Paul F. Jäger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein (2020). "Automated Design of Deep Learning Methods for Biomedical Image Segmentation". In: *Nature Methods (accepted)*.

Brugnara, Gianluca, Fabian Isensee, Ulf Neuberger, David Bonekamp, Jens Petersen, Ricarda Diem, Brigitte Wildemann, Sabine Heiland, Wolfgang Wick, Martin Bendszus, Klaus Maier-Hein, and Philipp Kickingereder (2020). "Automated volumetric assessment with artificial neural networks might enable a more accurate assessment of disease burden in patients with multiple sclerosis". In: *European Radiology* 30.4, pp. 2356–2364.

Zimmerer, David, Fabian Isensee, Jens Petersen, Simon Kohl, and Klaus Maier-Hein (2019a). "Unsupervised Anomaly Localization using Variational Auto-Encoders". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 289–297.

Kickingereder, Philipp, Fabian Isensee, Irada Tursunova, Jens Petersen, Ulf Neuberger, David Bonekamp, Gianluca Brugnara, Marianne Schell, Tobias Kessler, Martha Foltyn, Inga Harting, Felix Sahm, Marcel Prager, Martha Nowosielski, Antje Wick, Marco Nolden, Alexander Radbruch, Jürgen Debus, Heinz-Peter Schlemmer, Sabine Heiland, Michael Platten, Andreas von Deimling, Martin J van den Bent, Thierry Gorlia, Wolfgang Wick, Martin Bendszus, and Klaus H Maier-Hein (2019). "Automated quantitative tumour response assessment of MRI in neuro-oncology with artificial neural networks: a multicentre, retrospective study". In: *The Lancet Oncology* 20.5, pp. 728–740.

Zimmerer, David, Simon A. A. Kohl, Jens Petersen, Fabian Isensee, and Klaus Maier-Hein (2019b). "Context-encoding Variational Autoencoder for Unsupervised Anomaly Detection". In: *International Conference on Medical Imaging with Deep Learning (MIDL – Extended Abstract Track*.

Zimmerer, David, Jens Petersen, Simon A. A. Kohl, and Klaus H Maier-Hein (2018). "A Case for the Score: Identifying Image Anomalies using Variational Autoencoder Gradients". In: *Medical Imaging meets NeurIPS*.

Isensee, Fabian, Jens Petersen, Andre Klein, David Zimmerer, Paul F. Jaeger, Simon Kohl, Jakob Wasserthal, Gregor Koehler, Tobias Norajitra, Sebastian Wirkert, and Klaus H. Maier-Hein (2018). "nnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation". In: *arXiv:1809.10486 [cs]*. Winning entry of the Medical Segmentation Decathlon.

Goch, Caspar J., Jasmin Metzger, Martin Hettich, André Klein, Tobias Norajitra, Michael Götz, Jens Petersen, Klaus H. Maier-Hein, and Marco Nolden (2018). "Automated Containerized Medical Image Processing Based on MITK and Python". In: *Bildverarbeitung für die Medizin*, pp. 315–315.

Kleesiek, Jens, Jens Petersen, Markus Döring, Klaus Maier-Hein, Ullrich Köthe, Wolfgang Wick, Fred A. Hamprecht, Martin Bendszus, and Armin Biller (2016). "Virtual Raters for Reproducible and Objective Assessments in Radiology". In: *Scientific Reports* 6, p. 25007.

ACKNOWLEDGEMENTS

Ich möchte mich zunächst bei Prof. Klaus Maier-Hein für die Betreuung meiner Arbeit bedanken; für das inspirierende Arbeitsumfeld sowie für das Vertrauen und die Freiheit, Themen zu erkunden, die manchmal gar nicht mehr so viel mit medizinischer Bildverarbeitung zu tun haben. Ich bin zutiefst beeindruckt von der Entwicklung, die Deine Arbeitsgruppe—und mittlerweile Abteilung—während meiner Promotion gemacht hat, und bin stolz, Dich als Mentor bezeichnen zu dürfen.

Prof. Jürgen Debus möchte ich für das Vertrauen danken, mich als externen Doktoranden vor der Fakultät zu vertreten, ohne dass Sie je Einfluss auf meine Arbeit nehmen wollten.

Weiterhin danke ich meinen Kollegen und Partnern an der Uniklinik Heidelberg, ohne die große Teile dieser Arbeit nicht möglich gewesen wären, allen voran Philipp Vollmuth, Prof. Martin Bendszus, Prof. Sabine Heiland und Marcel Prager.

Ein besonderer Dank gilt der gesamten Abteilung für Medizinische Bildverarbeitung, aber auch der Abteilung für Computer-assistierte Medizinische Interventionen. Es ist ein Privileg, mit so vielen talentierten, lustigen, herzlichen Menschen zusammenzuarbeiten und zusammengearbeitet zu haben. Ich kann mir kein Arbeitsumfeld vorstellen, das in gleicher Weise Motivation und Leichtigkeit verbindet. Einen großen Anteil daran tragen natürlich meine Bürokollegen (und regelmäßige Besucher) der *deep fridge*: Fabian, Paul, Gregor, Peter, Seb, Simon, Tobi, Jakob und Dasha. Ich bin froh, in Euch nicht nur Kollegen, sondern Freunde gefunden zu haben. Paul und Simon möchte außerdem für heidelberg.ai danken, die Events entfachen meine Leidenschaft für unser Feld regelmäßig neu!

Ich möchte mich bei meinen Freunden bedanken, die mein Leben auf so vielfältige Weise bereichern: Philipp und Franzi, Markus und Maja, Felix, Katha und Flo, Janni, Josef; ich sehe Euch nicht ansatzweise so oft, wie ich gern würde, aber jeder Besuch bleibt mir dafür umso länger in Erinnerung. Gleiches gilt nun leider auch für Sebi—ich freue mich schon, Dich bei meiner Prüfung zu sehen! Ines, Melli, Marius, Chris, Mauch; Ihr alle habt meine Zeit in Heidelberg unvergesslich gemacht, und tut das auch weiterhin. Es heißt ja, dass man selbst der Durchschnitt derjenigen Menschen ist, mit denen man sich umgibt. Ihr alle lasst mich hoffen, dass das wirklich der Fall ist.

Nadine, Dir möchte ich für die unzähligen schönen Momente in den letzten Jahren danken, sowohl für die Urlaube als auch die kleinen Momente im Alltag. Ich kenne niemanden, der so viel Tatendrang hat wie Du, und ich bin immer wieder überrascht, was man mit Dir alles

unternehmen kann. Das motiviert und inspiriert mich, und ich schätze mich sehr glücklich, dich in meinem Leben zu haben. Außerdem gebührt Dir einiges an Dank für Deine Geduld insbesondere in den letzten Monaten!

Zu guter Letzt möchte ich meinen Eltern danken. Es ist schwer, in Worte zu fassen, was ich Euch zu verdanken habe. Ihr seid noch immer meine größten Vorbilder, und ich hätte mir keine schönere Kindheit vorstellen können. Nun könnte man sagen, dass es sich gelohnt hat, die endlosen Warum-Fragen damals zu beantworten, aber ich weiß, dass Ihr genauso stolz auf mich währt, wenn ich einen ganz anderen Weg eingeschlagen hätte. Möglicherweise zeichnet Euch das mehr aus als alles andere, und so bin auch ich stolz, meine Werte von Euch gelernt zu haben.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 MOTIVATION & RESEARCH OBJECTIVE

Much has been written about the great successes deep learning has enjoyed in the last few years, often in reference to a more grandiose umbrella term: Artificial Intelligence (AI). The World Economic Forum considers AI a central element of the "Fourth Industrial Revolution" (Schwab, 2016), while the European Commission has already published a policy paper on "a European approach to artificial intelligence" (European Commission, 2018). And perhaps with good reason. While experts in the field often caution against use of the term "intelligence", there is no denying that the advances deep learning has made and enabled are nothing short of spectacular, beating professional players in complex games (Silver et al., 2018) or generating long text passages that are indistinguishable from human-penned paragraphs (Brown et al., 2020; Palenzuela, 2020). In the medical domain, multiple studies report deep learning models with performance that is comparable to human experts (Gulshan et al., 2016; Esteva et al., 2017; Fauw et al., 2018; McKinney et al., 2020). At the same time, voices that pronounce the limitations of current research become more numerous (Lipton, 2018; Lipton and Steinhardt, 2018), proposing the possibility of a new "AI Winter" (Floridi, 2020), or attributing advancements to increased compute capabilities instead of progress in the methods themselves (Sutton, 2020).

Discussions about AI and its long-term potential, both positive and negative, are certainly useful and even necessary, but we choose not to engage in them for this thesis. Instead, we will treat deep learning and neural networks as what they undoubtedly are: powerful tools to approximate functions. More precisely, we will explore their capacity in a scenario that receives relatively little attention, the learning of distributions of functions (or function spaces) on continuous domains. Many architectures implicitly assume a certain grid-structure, like for example convolutional neural networks that are often used for image processing. While it might be unlikely that we need to work with images that don't have a pixel-grid structure, the assumption is more easily violated when we look at more generic sequences of observations. These are often modeled with recurrent neural networks (Lipton et al., 2015), which rely on the same premise. In other words, these models are missing a concept of *distance* between input elements or they are implicitly assuming equidistant inputs. In this thesis we will investigate how we can learn function representations that

do incorporate such a concept explicitly, meaning functions that are defined on a continuous domain. Moreover, rather than approximating single functions, we instead seek to learn representations of multiple functions at once. Our research objective is best summarized by the following scenario:

We have collected a number of observations over time and at arbitrary times—this could be a measurement of some property of interest, multiple potentially correlated measurements or even images. We would like to be able to estimate what the measurements would have looked like between those times we observed, i.e. interpolate between them, and also estimate what they might look like in the future, i.e. extrapolate from them. Whatever process we observed, we have no a priori knowledge about it, meaning we are unable to manually define or describe the underlying process—maybe we are not an expert, maybe the dynamics are entirely unknown. What we do have is a collection of past observations from similar processes. Our goal in this thesis is to use learning-based methods to automatically form a representation of the dynamics that describe these past observations; in other words, we wish to learn a representation of the distribution of functions from which the observations originated, such that, upon seeing the new measurements we just made, the model automatically selects which elements of the learned distribution best describe the current observations. We speak of elements as opposed to only one element, because we wish that approaches be able to handle both deterministic and probabilistic scenarios.

A very relevant example from the medical domain—and one we use repeatedly throughout the thesis—is the estimation of glioma growth, a form of brain cancer. Patients who have been diagnosed with the disease will regularly receive MRI scans to monitor its progression. It is important to know how large the tumor is and which areas of the brain are affected, and an estimate of the future development of the disease could be of tremendous use. In radiation therapy, clinicians must estimate which areas of the brain are likely affected by the cancer but appear normal in imaging. This is typically done by expanding the visible tumor region isotropically (Paulsson et al., 2014; Mann et al., 2018), and the clinician might adjust the result using personal experience. A better understanding and estimate of the growth dynamics could assist in the process by identifying regions of high and low risk of tumor infiltration, and could thus spare healthy tissue from damaging radiation. Likewise, treatment for glioma patients is often changed or adjusted when a progression, meaning a marked increase in size or the number of lesions, is diagnosed (Nam and Groot, 2017), because it can indicate that the treatment is not working. Earlier knowledge of this could thus improve a patient's prognosis. Finally, the example is useful for our purposes because glioma growth is not deterministic, at least not with respect to the observations we

will look at in this work. Our goal in the context of glioma growth analysis will be to establish methods that can learn representations of a distribution of (spatial) growth dynamics from a population, and also predict a distribution of possible growth trajectories for a given set of observations. In Chapter 5 we will discuss the glioma growth example in more detail.

Finally, a note on terminology. When we speak of *distributions* of functions, we are not referring strictly to distributions in the mathematical sense. Recall that a distribution is a map $D : T(\Omega) \to \mathbb{R}$ (or into the complex numbers), where $\Omega \subset \mathbb{R}^N$ and $T(\Omega)$ is the space of so-called *test functions* (see for example Lighthill (1958)). We are in fact only concerned with the learning of representations of *function spaces*, meaning sets of functions with a shared domain and co-domain. Whether these sets actually fulfill the requirements to be called test functions[1], so that a distribution would exist, is of no concern to us. We use the terms *function space* and *distribution* of functions interchangeably.

1.2 OUTLINE

The chapters in this thesis are designed to be mostly self-contained, each following a structure one would typically find in a journal or conference publication. Indeed, some chapters are directly adapted from already published articles. Related work will also be discussed individually instead of in a dedicated chapter.

We will begin by introducing some relevant background in Chapter 2. It presents concepts not directly related to the work we present, but helpful for a better understanding. The individual sections are not designed to be comprehensive, we rather briefly summarize the topics and provide references for further reading.

Chapter 3 presents preparatory contributions. Our later work requires a large annotated dataset, so we set out to find ways that enable experts to efficiently create reference segmentations for imaging data—MRI data from glioblastoma patients in our case—in an interactive setting.

In Chapter 4 we make an initial attempt at purely learned glioma growth modeling. Designed as a proof of concept, the approach therein is constrained to data *on the grid*, meaning observations equally spaced over time, and to a fixed number of inputs.

Chapter 5 will set up the research question we address in this thesis, using glioma growth as a guiding example. It will highlight why it is desirable a) to work on a continuous domain; b) to learn representations of function spaces instead of direct input-output mappings; and c) to learn these representations as opposed to manually choosing and parametrizing them. The chapter will also show how Neural Processes (Garnelo et al., 2018a; Garnelo et al., 2018b) are a promosing approach

1 Those requirements would be smoothness and a compact support

in this context, motivating our choice to focus on and to extend them in the following chapters.

We continue with a deconstruction of Neural Processes in Chapter 6, investigating how they learn finite-dimensional representations of function spaces. We find that they often do this by decomposing signals by frequency content, so we derive a bound the on maximum frequency signals may contain and show how the training data defines which frequencies will and will not be represented in the Neural Process.

In Chapter 7 we will improve upon a recent contribution in the Neural Process family, called *Convolutional Conditional Neural Processes* (ConvCNP) (Gordon et al., 2020). These models perform well on a variety of tasks, but are deterministic. We combine them with a Gaussian Process to recover the possibility to sample from the model and also find that this improves generalization.

While the previous three chapters are all concerned with scalar or low-dimensional observation spaces, we show in Chapter 8 how to translate segmentation architectures into the Neural Process framework. We apply the result to glioma growth modeling, realizing what we initially attempted in Chapter 4 on a continuous time axis.

Finally, we summarize and discuss our findings and contributions in Chapter 9. We also introduce some contributions in the context of clinical translation that are not directly related to this thesis. We close with an outlook of possible future research directions.

1.3 CONTRIBUTIONS

In Chapter 3 we are concerned with the question of how to place annotations most efficiently in a Random Forest-based interactive segmentation process. We show the following:

- Assuming the annotator is an expert, it is significantly more efficient to correct errors in the classifiers output than to provide additional annotations where the classifier is most uncertain.
- There is no significant difference between corrective annotations and corrective annotations in regions of high classifier uncertainty. We conclude that error regions are usually a subset of the regions with high uncertainty.
- From the above we can conclude that displaying uncertainty information is of little use for an expert user, which is in contrast to large parts of the active learning literature that is concerned with finding measures of a model's lack of "knowledge".

In Chapter 4 we apply probabilistic segmentation to the task of glioma growth modeling, albeit on a discrete time domain, working under the assumption that this growth is not deterministic and that it

is necessary to model distributions of growth trajectories. Our findings and contributions can be summarized as follows:

- We are the first to frame glioma growth modeling as a model-free learning problem, so that all dynamics are inferred directly from data.
- We present evidence that our approach learns a distribution of plausible growth trajectories, conditioned on previous observations of the same tumor.
- We provide an open source implementation of our method[2].

In Chapter 6 we analyze how Neural Processes (Garnelo et al., 2018a; Garnelo et al., 2018b) form finite-dimensional representations of function spaces. Our contributions are:

- We show that both deterministic and variational Neural Processes form representations by associating different dimensions of their representation space with different regions of the input space (i.e. the shared domain of the function space).
- We show that while variation Neural Processes usually partition the input space into separate regions, deterministic Neural Processes associate different dimensions of their representation with different frequencies in the input, thus implicitly performing a frequency decomposition of the signals.
- We derive a theoretical upper bound on the maximum frequency a signal may contain so it can still be represented in a Neural Process with a given representation size. We empirically confirm that this bound holds and show that limiting the representation size can make Neural Processes act like low-pass filters.
- We further show that the learned frequencies strongly depend on the training data, such that frequencies not seen during training will be suppressed when passing signals through a Neural Process at test time. This essentially turns them into programmable band-pass or band-stop filters.

In Chapter 7 we combine a later addition to the Neural Process family, called ConvCNP (Gordon et al., 2020), with a Gaussian Process. In doing so, we achieve the following:

- While ConvCNPs are deterministic, combining the model with a Gaussian Process restores the ability to produce multiple samples, a compelling feature of the original Neural Processes (Garnelo et al., 2018b).

2 https://github.com/jenspetersen/probabilistic-unet

- The integrated Gaussian Process improves the model's ability to generalize. We show that our model (GP-ConvCNP) better extrapolates signals far from the provided context points. It's also more stable to a distribution shift[3] at test time, which we show by applying models trained on simulated data to real world data.

In Chapter 8 we apply Neural Processes to segmentation tasks, distinguishing between scenarios where input images are available at desired target locations and scenarios where they are not available. We show how typical segmentation architectures, namely U-Net-like (Ronneberger et al., 2015) architectures with skip connections, can be leveraged in the Neural Process framework by interpreting the skip connections as additional representation spaces:

- We show that a trivial application of Neural Processes to segmentation architectures, meaning summation of context representations, results in poor performance. We propose a model that combines temporal and spatial attention along the skip connections.
- We show that our model can be used for interactive segmentation by providing a small number of annotated slices from an image volume as context. It can extrapolate information from this context to other slices, which we demonstrate both on a synthetic toy example and for glioma segmentation on MRI data.
- We show that our proposed approach can model complex spatial variations of a target shape over time, which we demonstrate both on a synthetic toy task and on examples of glioma growth.
- We propose a variational version of our model that can produce multiple prediction samples for a given set of context observations. We apply it to the modeling of glioma growth and show that our model can predict diverse growth trajectories, where each sample is consistent over time.

Beyond the contributions outlined here and throughout the thesis, its author has also contributed to a number of other publications related to this work in a broader sense:

- Most importantly, the author has developed a system for automated processing of imaging data within a clinical IT infrastructure. This will be discussed in Section 9.2, but in essence the system allows the automatic deployment of deep learning algorithms in clinical routine. At a later stage, this will allow us to apply our glioma growth model to unseen patients for further

3 *Distribution shift* refers to a general dissimilarity between training data and test data.

validation. The system was described in Petersen et al. (2018) and Kickingereder et al. (2019) and has enabled data analysis for Kickingereder et al. (2019) and Brugnara et al. (2020).

- The author has developed a comprehensive evaluation framework for segmentation models and was also involved in model development for Isensee et al. (2018) and Isensee et al. (2020), where Isensee et al. (2018) was the winning contribution in the 2018 Medical Segmentation Decathlon (Simpson et al., 2019). Isensee et al. (2020) extends this work, has won or is among the top-performing approaches on a large number additional biomedical segmentation challenges, and can thus be considered the current state-of-the-art in biomedical image segmentation.

- The author further contributed in various ways to a number of publications, e.g. in an advisory role for Zimmerer et al. (2018), Zimmerer et al. (2019b) and Zimmerer et al. (2019a); or by performing a number of experiments as in Kleesiek et al. (2016).

2 BACKGROUND

We will use this chapter to introduce a few concepts that are not directly relevant to the work we undertake in this thesis, but might nevertheless be of interest. Everything we introduce will inevitably only be a brief summary, but we will provide the reader with references for a more thorough exploration of the respective topic.

A few things will not be included here. For one, all the methods directly relevant to our work will be presented in the respective chapters. More importantly, we will not introduce the basic concepts of machine learning and deep learning, which we assume the reader is familiar with. Should that not be the case, Hastie et al. (2016) provide a rather extensive resource on machine learning, including neural networks. For a more practical perspective on deep learning in particular see for example Goodfellow et al. (2016).

2.1 MAGNETIC RESONANCE IMAGING

This section will give a quick overview of magnetic resonance imaging (MRI) and introduce its main variants that are used in radiological diagnosis of glioma patients. As resources for general further reading we recommend Haacke et al. (1999), Liang and Lauterbur (2000) or Vlaardingerbroek and Boer (2003). For more details on pulse sequence design and associated imaging characteristics the reader is referred to Bernstein et al. (2004). Abragam (1983) can serve as a resource on the underlying physics of nuclear magnetic resonance, while Reiser et al. (2008) discusses MRI from a more medically inclined viewpoint. The following is presented with varying degrees of detail in all of those sources.

2.1.1 *Physics*

Magnetic resonance imaging leverages the fact that some nuclei possess finite nuclear spin and thus quantized angular momentum and an associated magnetic moment $\boldsymbol{\mu}$. In MRI, the latter interacts with an external (homogeneous) magnetic field $\mathbf{B_0} = B_0 \mathbf{e}_z$ in a way that only quantized states of the angular momentum along the magnetic field are observed, with different energies as dictated by the Zeeman effect. For hydrogen nuclei—the ones most relevant for clinical imaging—there are only two of those states, and their energy difference is given by $\Delta E = \hbar\omega_L = \hbar\gamma B_0$. γ is called the *gyromagnetic ratio* and ω_L is referred to as the *Larmor frequency*. The occupation probabilities (in

case of hydrogen nuclei) can be described by a Fermi-Dirac statistic, and at room temperature and a typical magnetic field strength (1.5 T to 3 T) the excess ratio of spins in the energetically favorable state is in the order of 1×10^{-5}. In other words, the magnetic field only causes a small difference between occupation probabilities, which explains why MRI requires extremely strong magnets. The occupation difference also results in a finite expected value of the macroscopic magnetization $\mathbf{M}$ in direction of the magnetic field at thermal equilibrium, which we call M_0. A manifestation of the *Ehrenfest theorem*, the macroscopic magnetization can be described using classical mechanics, and its temporal evolution is often approximated using the *Bloch equations*:

$$\partial_t \mathbf{M}(t) = \gamma(\mathbf{M} \times \mathbf{B}) - \left(\frac{M_x}{T_2}, \frac{M_y}{T_2}, \frac{M_z - M_0}{T_1} \right)^T \tag{2.1}$$

Switching to a rotating frame of reference, these admit solutions:

$$M_x + iM_y = M_\perp(t) = M_\perp(0)e^{-t/T_2} \tag{2.2}$$

$$M_z = M_\parallel(t) = M_0 - (M_0 - M_\parallel(0))e^{-t/T_1} \tag{2.3}$$

We often speak of T_1-weighted and T_2-weighted imaging, referring to techniques that create an image *contrast* proportional to these *relaxation times*, but what we actually measure is $\partial_t M_\perp$ via induced voltages. To elicit such signals, we employ an additional magnetic field $\mathbf{B_1} = B_1 \boldsymbol{e_x}$ perpendicular to $\mathbf{B_0}$ that oscillates with the Larmor frequency. $\mathbf{M}$ can be imagined as a spinning top that precesses around $\mathbf{B_0}$ with that frequency, and the oscillating $\mathbf{B_1}$ can be used to drive that precession, so that after certain times the magnetization will have flipped into the x-y-plane (called a 90°-pulse) or switched orientation (called a 180°-pulse). Such pulses can be combined in various ways to achieve a signal that contains information about T_1 or T_2.

Spatial resolution is achieved by applying so-called *gradient fields* so that the Larmor frequency varies with position. Combining such gradients at excitation as well as before and during signal acquisition, it is possible to fully resolve the three-dimensional position in measurements.

2.1.2 *Contrasts*

As outlined above, images acquired in MRI can be *weighted* differently, depending on which relaxation time is used to provide a contrast. Note that unless additional measures are taken, these images are only proportional to some function of those relaxation times and not quantitative measures thereof. There are also a number of other contrasts one can achieve, and we quickly introduce the ones encountered in this thesis.

T_1-WEIGHTED T_1-weighted images are typically hyperintense when T_1 is small. As a result, tissue appears brighter with increasing fat content, such as for example white matter compared to grey matter.

T_2-WEIGHTED T_2-weighted images are typically hyperintense when T_2 is large. Tissue with a higher free water content will appear brighter in them, like grey matter compared to white matter, but also edema in glioma patients.

CONTRAST AGENTS Contrast agents, a common choice being Gadolinium, typically work by drastically reducing the T_1 of nuclei in their proximity. As a result, they appear very pronounced on T_1-weighted images. In glioblastoma patients, they are used to visualize where the tumor has destroyed the blood-brain barrier, a region often referred to as *enhancing tumor*.

FLAIR Inversion recovery can be used to suppress the signal from regions with a specific T_1. If this is done for fluids, for example the CSF, we speak of fluid-attenuated inversion recovery, or FLAIR. In glioma patients, this is used as a T_2-weighted sequence to better visualize the edema.

2.2 GLIOMA GROWTH MODELING

2.2.1 *Glioma*

Around 80 % of all malignant brain tumors are glioma, a type of tumor that originates in the glial cells of the central nervous system (Goodenberger and Jenkins, 2012). There are several subtypes, like astrocytoma or oligodendroglioma, distinguished by the type of glial cell they share histological properties with. The World Health Organization suggests a classification into four grades of severity (Louis et al., 2016), and the highest grade is typically referred to as *glioblastoma multiforme*. Glioblastoma is itself the most common glioma and holds extraordinarily poor prognosis for patients with a median survival time of a little over a year, depending on treatment, or only a few months without treatment (Ohgaki and Kleihues, 2005; Johnson and O'Neill, 2012). Considering the aggressiveness of the tumor, it is not surprising that treatment typically consists of multiple combined approaches, beginning with resection, if possible, and followed by variations of radiation and chemotherapy (Nam and Groot, 2017).

The disease is monitored in more or less regular intervals using MRI to ascertain the extent of the tumor and to discover potential new lesions. The change in tumor size and the presence of new lesions will decide if a tumor is classified as progressive, stable or responding (Wen et al., 2010) and will thus decide on the indication of a treatment change. The growth of glioblastoma is often very irregular, with large

variations in growth rate both among patients and over time (Stensjøen et al., 2015). There is a large number of environmental factors that can have an influence on cell proliferation, both on the patient-level and microscopically, rendering it essentially stochastic (Thomas et al., 2018). While it has long been suggested that tumor growth should also be viewed macroscopically as a (partly) stochastic process (Hanson and Tier, 1982), we find that spatial growth models seldom follow this assumption, likely because stochastic models require more data for parameter determination.

2.2.2 *Reaction-Diffusion Models*

As outlined above, one of the defining characteristics of glioma growth, especially of high grade ones known as glioblastoma multiforme (GBM), are the highly irregular growth patterns they exhibit, involving multiple tissue types for which the change in composition is notoriously difficult to predict. Most existing approaches that model the growth of glioma do this using variants of the reaction-diffusion equation, i.e.

$$\frac{\partial c(\mathbf{x}, t)}{\partial t} = \nabla \left(D \nabla c \right) + R(c, t) \tag{2.4}$$

where $c(\mathbf{x}, t)$ is the concentration of tumor cells, $\mathbf{D}$ is a diffusion tensor that is itself possibly a function of space and time, and $R(c, t)$ is the so-called *reaction* term that dictates proliferation, i.e. the overall increase (but also decrease) of c. R is typically modeled without a spatial dependence. Some common choices are exponential growth, meaning $R \propto c$ (also referred to as a linear *model*), or self-limiting models like logistic growth with $R \propto c(1 - c)$ (also called *Fisher-Kolmogorov*) or Gompertz growth with $R \propto c \log(1/c)$. We will attempt to give a rather comprehensive list of these works, but we would also like to point to two review articles that summarize the work before 2007, and 2011 respectively, in greater detail, namely Harpold et al. (2007) and Menze et al. (2011b).

The first applications of a reaction-diffusion model to glioma growth were introduced by Cruywagen et al. (1995) and Tracqui et al. (1995), both using a linear model and a scalar diffusion tensor. They also add the treatment effect as a negative linear term in Equation (2.4), encoded as a step function over time, presenting results for a single patient. Woodward et al. (1996) and Burgess et al. (1997) and Burgess et al. (1997) do the same, but instead simulate patients with their model analyze different outcomes (under hypothetical treatment). The latter two also extend their simulations to 3D.

There is an abundance of literature that suggest glioma cells infiltrate preferentially in white matter and along fiber tracts therein (Kuroiwa et al., 1994; Kelly and Hunt, 1994; Chicoine and Silbergeld,

1995; Giese et al., 1996; Silbergeld and Chicoine, 1997; Price et al., 2003; Giese et al., 2003; Esmaeili et al., 2018). The first to incorporate this into a reaction-diffusion model were Swanson et al. (2000) and Swanson et al. (2002). They also assume a scalar D, but allow different values for grey and white matter regions, showing results on simulated data. Although introduced much later, Yuan et al. (2013) present a direct extension of this by allowing tissue to belong to the two categories by varying degrees, showing results on data from 12 different patients.

While the reaction-diffusion model represents two mechanisms of tumor growth, namely infiltration and proliferation, a third mechanism is the so-called *mass effect*, i.e. the displacement of healthy tissue by the fast-growing cancer. This was first modeled for glioma by Wasserman et al. (1996) and Kyriacou et al. (1999), assuming linear and neo-Hookean elasticity, respectively, in the healthy tissue. Mohamed and Davatzikos (2005) extend this by differentiating between solid tumor mass and the surrounding edema. The first to combine a reaction-diffusion model with a simulation for the mass effect were Clatz et al. (2005). They further work with an anisotropic—meaning non-scalar—diffusion tensor that they derive from an atlas of DTI data from multiple healthy patients, while they present results for a single GBM patient. The coupling between their reaction-diffusion model and the mass effect model is unidirectional, the latter does not inform the reaction-diffusion model e.g. by changing the diffusion tensor map over time. This is remedied in Hogea et al. (2007), Bondiau et al. (2008), and Hogea et al. (2008), who show that the mass-effect can be represented as an additional advection term in Equation (2.4), i.e. the first-order derivative of a *velocity field*. The evolution of this velocity field is in turn described by another set of differential equations coupled with the original model. Branco et al. (2014) later model the brain as a visco-elastic material, leading to a stress diffusion term in Equation (2.4) with the the evolution of the stress tensor again dictated by a coupled differential equation. Viscous stress is also introduced in Yuan and Liu (2016), but they don't work with an anisotropic diffusion tensor, instead choosing an approach based on their earlier work (Yuan et al., 2013).

How to map the water diffusion tensor acquired in diffusion tensor imaging to a diffusion tensor describing the motility of the cancerous cells is non-trivial. Jbabdi et al. (2005) show empirically for data from one patient that the anisotropy of the cell diffusion tensor should be greater than of that of a measured water diffusion tensor. Painter and Hillen (2013) derive a connection between a microscopic formulation of cell transport in an anisotropic environment and the macroscopic formulation in Equation (2.4). They arrive at a slight variation of this equation where the first term becomes $\Delta(Dc)$, thus giving rise to additional advective terms. Engwer et al. (2015) expand on this by also including adhesion effects in their microscopic model formula-

tion. Stretton et al. (2013) evaluate the overall importance of using anisotropic diffusion and find a large difference in performance between a scalar D and one based on an atlas of DTI data. Another small increase is afforded by DTI data from the same patient, which is typically not available. They evaluate this on multiple time points from three different patients. These results are also confirmed by Swan et al. (2018).

Another difficulty in the application of reaction-diffusion models is the question of how to relate the parameters of the model to the—often scarce—observations. Konukoglu et al. (2007) argue that it is hard or impossible to derive the full cell distribution $c(\mathbf{x}, T_i)$ from an MRI at time t_i and that one can only really observe how a contour of the tumor evolves over time. As a consequence, they translate Equation (2.4) into an eikonal equation that describes travelling wavefronts. While they initially present simulated results, they later apply this model to real patient data as well (Konukoglu et al., 2010a). These works also switched to a logistic growth model compared to the earlier work above, and virtually all later work maintains this approach. Menze et al. (2011a) are the first to explicitly model uncertainty in their work. They combine a deterministic reaction-diffusion model with a Bayesian image observation model that allows for multi-modal observations. This enables them to obtain a posterior estimate of model parameters conditioned on patient MRIs via Monte-Carlo integration. Results are demonstrated both on synthetic and two real datasets. Perhaps one of the most difficult parameters to estimate is the initial tumor cell density $c(\mathbf{x}, t_0)$, especially when it is not assumed to be directly represented in one of the available observations. A common choice is to assume a Gaussian density in the center of mass of the visible tumor, but Rekik et al. (2013) show on data from 4 low-grade glioma patients that the true source location can deviate significantly from this assumption.

In a clinical context, growth models promise improvements for radiation therapy, where clinicians must estimate a probable infiltration margin around the visible tumor that will be irradiated as well. The visible tumor is referred to as the *gross tumor volume* (GTV) while inclusion of the aforementioned safety margin leads to the *clinical target volume* (CTV) (Burnet et al., 2004). Current clinical practice defines this CTV with an isotropic margin of 2 cm to 3 cm, although the extent can vary (Paulsson et al., 2014; Mann et al., 2018). A more accurate understanding of the tumor growth dynamics would allow for a more refined definition of the CTV margin, identifying regions of higher or lower risk of recurrence. Konukoglu et al. (2010b) propose using their travelling wave formulation (Konukoglu et al., 2007; Konukoglu et al., 2010a) to define a target volume margin based on growth estimates. Mosayebi et al. (2012) suggest replacing the Euclidean isotropic expansion with a geodesic expansion on a manifold representing the

white matter fibers. They show that this results in more realistic extrapolations on a comparatively large dataset of 11 patients. Unkelbach et al. (2014b) discuss the implications of radiotherapy planning with a reaction-diffusion for target volume definition as well as for dose distribution (Unkelbach et al., 2014a). A proof of concept for an automatic treatment plan—from MRI to dose distribution plan—based on a reaction-diffusion model is presented by Lê et al. (2017). They evaluate two different scenarios, specifically the availability of two or only one MRI for parameter estimation, and also incorporate a possible uncertainty stemming from multiple segmentations from different readers. Results are shown for two patients. The most recent work in this line was put forward by Lipková et al. (2019), who cast the efforts by Menze et al. (2011a) in a radiation therapy planning framework. Crucially, they show for 8 patients that their approach can maintain efficiency with a smaller CTV, where efficiency is defined as the relative volume of the recurrence region contained within the CTV.

2.2.3 *Machine Learning Approaches*

The number of works that apply machine learning to the prediction of glioma growth are quite limited. To the best of our knowledge, the first to do so were Morris et al. (2006), who mimic a growth model by starting out with a tumor segmentation and iteratively predict a probability for boundary voxels to be transformed into tumor tissue. The voxel label is then updated with the predicted probability, not unlike a random walk procedure. On data from 17 patients, they evaluate different classifiers for this task (they show results for logistic regression, but link to results for Naive Bayes and SVM), incorporating both image-based features and other information like the age of the patients. Akbari et al. (2016) use an SVM classifier to predict a tumor infiltration map on pre-operative MRI of GBM patients, using the common structural contrasts as well as parameters derived from diffusion-weighted and susceptibility-weighted imaging. Training the classifier on 31 patients and evaluating on 34, they find a statistically significant correlation between regions of recurrence after resection and their predicted infiltration. Not working with glioma but to the best of our knowledge the first to apply deep learning to tumor growth, Zhang et al. (2018) use a CNN to predict pancreatic tumor growth. In a relatively straightforward fashion, their network predicts which voxels will "become tumor" after a predefined time. Along with several image inputs from the current time step, the network also has access to an optical flow map computed between the last two available time steps, which according to the authors allows the network to estimate the mass effect. More recently Gaw et al. (2019) tried to combine a reaction-diffusion model with a learning-based approach.

In addition to several MRI channels as well as image-localized biopsy maps, they feed the prediction from the reaction-diffusion model to a classifier as an additional input. They find that this improves predictive performance compared to both the reaction-diffusion model as well as a machine learning model without the reaction-diffusion input.

2.3 MEDICAL IMAGE SEGMENTATION

The segmentation of medical images, compared to natural images, is historically subject to a few unique challenges. Datasets in the domain are typically orders of magnitude smaller than datasets of natural images. As an example, the BraTS (Brain Tumor Segmentation) challenge, one of the largest and most renowned challenges in the field of medical image segmentation, provides a few hundred data points for training in its latest iteration (Bakas et al., 2019). In contrast to that, the Cityscapes dataset, a common segmentation benchmark that contains street scenes, provides several thousand annotated data points (Cordts et al., 2016). At the same time, these two examples highlight another difference that often distinguishes medical images from natural images. BraTS provides annotated image *volumes*, while Cityscapes contains 2D images. As such, the individual items in the BraTS data are much larger and hence more challenging to process.

The difference in data availability can be explained both by the overall ease of access to un-annotated data—medical data is often governed by stringent privacy protections—and by the difficulty of generating annotations, which requires expert knowledge only trained professionals possess. Associated with the latter is a general ambiguity in the interpretation of imaging data. In most cases, the outline of a car will be rather well-defined, while even experts with many years of experience can disagree with respect to the precise contour of a lesion—or indeed whether or not some input contains a lesion in the first place. We observe that many of the more seasoned approaches to medical image segmentation, which we summarize below, are very specific to their task, while more recent work based on convolutional neural networks doesn't necessarily distinguish between medical and other input modalities. To be sure, there are still many publications that modify CNN architectures for a particular task like the segmentation of a certain organ, but we will not discuss those.

2.3.1 *Classical Approaches*

Methods that don't use convolutional neural networks or deep learning in general can be grouped roughly into three categories: shape models, atlas-based methods and conventional classifiers. Shape models and atlas-based methods are similar in that they aggregate the

statistics of some population to describe a prior for the segmentation. In shape models, this description is typically one that uses landmarks to define a shape, and at test time the task reduces to identifying the most likely configuration of landmark positions using both the prior and information from the underlying input image. For an overview of statistical shape models see for example Heimann and Meinzer (2009). In atlas-based methods, the prior is formed by combining both the input images, e.g. by registration to a common frame of reference, and the segmentations from multiple cases. A new case is then segmented by comparing it to this atlas, which usually involves registration to the same frame of reference. Alternatively, the individual reference cases are *not* combined, but instead treated as individual atlases, and only the predicted segmentations on the test case are averaged in a suitable way. Iglesias and Sabuncu (2015) give a review of atlas-based segmentation. Both shape models and atlases work well for cases that are close to the reference or prior, for example healthy patients, but generally struggle with pathologies such as tumors.

Medical images can of course also be segmented by framing the problem as a pixel classification task. In doing so, one can apply essentially any conventional classifier, popular choices of which are Random Forests (Breiman, 2001) and Support Vector Machines (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995). A key challenge lies in defining the input features for the classifier, as the input image intensities are generally insufficient. Common choices are edge filters, texture filters or blurring, often applied with various scale parameters, so that a larger information vector is constructed for each pixel to be used in the classifier. Random Forests, for example, construct multiple decision trees from random subsets of the available data, building each tree by repeatedly looking at a random subset of the above features and splitting the data along the feature dimension that maximizes or minimizes a certain criterion. The choice of this split criterion, along with the number of trees, the depth of the trees and other parameters, is generally done manually. Prediction on a test case is performed by aggregating the votes from the individual trees. SVMs work by implicitly lifting the data to a higher-dimensional feature space and finding a hyperplane that best separates the classes in the training data. As each hyperplane can only separate the data into two half-spaces, multi-class classification is possible by simply converting the problem into multiple binary decisions (e.g. one-vs-one or one-vs-rest) and training an appropriate number of classifiers. One advantage of these approaches is that they generally require little training data, so that they can be used for example in interactive segmentation, as we do in Chapter 3.

2.3.2 *Convolutional Neural Networks*

Earlier segmentation architectures were essentially image classification CNNs like AlexNet (Krizhevsky et al., 2012) or VGG (Simonyan and Zisserman, 2015) modified to be fully convolutional (Long et al., 2015; Chen et al., 2015) so that one can essentially "slide" them across an image and successively classify pixels. The first encoder-decoder structure, to the best of our knowledge, was presented in Badrinarayanan et al. (2015) and Badrinarayanan et al. (2017). There the authors saved the indices of max-pooling operations in the encoder for upsampling in the decoder. Such encoder-decoder structures allow predictions for the entire input image at once and don't require padding of the input to be able to make predictions at the borders of the image. The first encoder-decoder structure with skip connections, and arguably the most well known, is the U-Net (Ronneberger et al., 2015), and we typically refer to any architecture that resembles this setup as *U-Net-like*. Such skip connections were also shown to enable training of very deep architectures (He et al., 2016) (called ResNets[1]) by alleviating the so-called *vanishing gradient* problem (Hochreiter et al., 2001), but in encoder-decoder structures they primarily serve another purpose: they give the decoder access to information at a higher resolution, which allows it to produce fine detail in its prediction. U-Nets have since been extended and modified in numerous ways; some of the more well-known ones are those to three-dimensional inputs (Çiçek et al., 2016; Milletari et al., 2016), which is especially relevant in the medical domain, or those that attempt to operate on multiple scales of the input (Zhou et al., 2020; Tao et al., 2020). It is worth noting that not all segmentation architecture use an encoder-decoder pattern, some work almost entirely on the full scale of the input (Jégou et al., 2017; Wang et al., 2020a). For an overview of recent segmentation architectures, not limited to medical imaging, see for example Minaee et al. (2020).

2.3.3 *Evaluation*

As we mentioned above, medical images are often difficult to interpret. Considering their ambiguity, it makes a lot of sense to not assign hard class labels to image pixels, but to instead speak of probabilities with which pixels belong to a certain class. And this is exactly what most classifiers or deep learning models do: they output so-called pseudo-probabilities for the individual classes that sum to one. Ideally we would use these probabilities when evaluating model outputs, but quite to the contrary it has become the de-facto standard in the medical imaging community to only evaluate the resulting hard labels

1 ResNets actually use summation, while U-Nets use concatenation, but the effect is the same.

using set overlap measures. By far the most common of these is the Dice coefficient (Soerensen, 1948; Dice, 1945), given by:

$$\text{Dice}(S_1, S_2) = 2 \cdot \frac{S_1 \cap S_2}{|S_1| + |S_2|} \tag{2.5}$$

for two segmentations S_1, S_2, interpreted as sets, such that $|\cdot|$ refers to the cardinality. This is only a binary measure, while segmentations usually contain multiple labels. Measures that take multiple labels into account simultaneously exist (Crum et al., 2006), but it is common practice to just treat individual labels separately or to average their scores.

The Dice coefficient further considers each pixel identically, while segmentations usually vary most at the their borders. Some metrics have been proposed that put more focus on the *surface* of a segmentation, such as the *Surface Dice* (Nikolov et al., 2018) or the *Hausdorff Distance* (which itself isn't new, see for example Rockafellar and Wets (1998), but has recently seen some use, for example in Karimi and Salcudean (2020)). Notwithstanding the above, the Dice score remains by far the most commonly used measure to evaluate segmentations, and as a result it will also be our main tool for the task.

3

EFFICIENT EXPERT ANNOTATIONS IN INTERACTIVE SEGMENTATION

The impressive performance of deep learning-based approaches to image processing, specifically convolutional neural networks (CNNs), hinges on the availability of large bodies of annotated data. Creating such annotations is a time-consuming process, especially in more challenging domains like medical imaging, where many years of experience and expertise are required to confidently make the decisions associated with the annotation process. We were faced with this very problem in the context of our work, as we sought to create reference segmentations for a large dataset of magnetic resonance images of glioblastoma patients, which serves as the basis of many of the analyses in this thesis.

Interactive segmentation methods are a compelling tool for this task, as they are typically much faster than fully manual annotation procedures (e.g. manually painting in pixels in an image to create a segmentation) but still offer full expert supervision that is not given when using automatic methods. To aid our clinical partners in their work, we implemented an interactive segmentation method in the open-source toolkit MITK (Wolf et al., 2004; Nolden et al., 2013) and investigated how to best perform annotations in this framework. Results presented in this chapter were in part published in the following works:

Petersen, Jens, Martin Bendszus, Jürgen Debus, Sabine Heiland, and Klaus H. Maier-Hein (2017a). "Effective User Interaction in Online Interactive Semantic Segmentation of Glioblastoma Magnetic Resonance Imaging". In: *Journal of Medical Imaging* 4.3, p. 034001.

Petersen, Jens, Martin Bendszus, Jürgen Debus, Sabine Heiland, and Klaus H. Maier-Hein (2017b). "Effective User Guidance in Online Interactive Semantic Segmentation". In: *SPIE Medical Imaging*.

Petersen, Jens, Sabine Heiland, Martin Bendszus, Jürgen Debus, and Klaus H. Maier-Hein (2017c). "Quantification of Guidance Strategies in Online Interactive Semantic Segmentation of Glioblastoma MRI". In: *Bildverarbeitung für die Medizin*, pp. 231–236.

Petersen, Jens, Martin Bendszus, Jürgen Debus, Sabine Heiland, and Klaus H. Maier-Hein (2016). "A Software Application for Interactive Medical Image Segmentation with Active User Guidance". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI) – Interactive Medical Image Computing Workshop*.

The contents of this chapter closely match those in the publication, but have been expanded where necessary. In particular, we append

Section 3.6, which gives an overview of interactive deep learning-based segmentation methods introduced since our work was conducted.

3.1 INTRODUCTION

The purpose of the experiments presented in the following sections was to establish methods that enable expert users to generate high quality segmentations of medical image data with only a small number of input annotations. The interactive segmentation technique we based our work on trains a classifier on a small number of labeled pixels and predicts labels for all remaining pixels. The expert user can see intermediate results and the underlying data to perform additional interactions in order to improve the resulting segmentation in an iterative fashion. As outlined above, we focus on glioblastoma segmentation, a task that has been quite extensively studied in the context of the *Brain Tumor Segmentation (BraTS)* challenge (Menze et al., 2015) and that is considered one of the most challenging multi-class segmentation problem in medical image analysis. We based our interactive segmentation process on a Random Forest classifier, which has proven to be the best overall choice of classifier on a wide range of tasks (Fernandez-Delgado et al., 2014) and has also achieved very good results for the specific task of glioblastoma segmentation (Zikic et al., 2012; Bauer et al., 2013; Kleesiek et al., 2014; Goetz et al., 2016). Note that the study by Fernandez-Delgado et al. (2014) did not include any deep learning approaches. We give an overview of deep learning-based interactive segmentation in Section 3.6.

An implementation of an interactive segmentation scheme using Random Forests is offered by the open-source *ilastik* framework (Sommer et al., 2011). We implemented a comparable technique in the MITK framework (Petersen et al., 2016): the classifier predicts labels for all pixels in an image based on a few manually annotated pixels. In an iterative process the user is asked to provide additional annotations to improve the segmentation. Normally the choice of where to annotate next in the process is left entirely to the user. Our contribution is the proposal and evaluation of five interaction methods to ensure optimal usage of user inputs that use 1. the classifier uncertainty, 2. an expert user's knowledge of the correct segmentation and 3. also a combination of the two. To the best of our knowledge, we were the first to evaluate how useful uncertainty information is compared to correctness information in this setting.

We compare 5 different methods of placing annotations in an online interactive segmentation task. In an iterative process a user annotates a small part of an image (i.e. a small number of pixels) and a classifier is trained on these inputs to predict labels for all image pixels. The result is displayed back to the user so they can input additional annotations to refine the result until satisfied. In this setting we

compare interaction modes (i.e. guidelines which parts of the image the user should annotate) based on the classifier uncertainty as well as the user's knowledge of the correct segmentation.

3.2 RELATED WORK

The question of what would be the optimal next annotation pertains to the domain of active learning (Settles, 2010), where the premise is that the algorithm can query an *oracle* (the user in this case) for the correct label of some data point to improve its prediction, but only at great cost (for example time), hence the need to keep the number of queries minimal. Semantic segmentation is non-trivial in this context, because the instances (pixels) are strongly correlated and users will rarely perceive them as separate entities. Triebel et al. (2014) present work that uses scribble annotations and Gaussian processes to segment individual pixels, but they only work with 2D images of everyday objects. Our problem is innately three-dimensional, which makes many computations infeasible. The use of superpixels can reduce the complexity of the problem, but previous studies that employ them focus more on generic computer vision tasks (Vijayanarasimhan and Grauman, 2009; Vezhnevets et al., 2012). There the difficulty is not so much the correct delineation of an object—most have pretty clear boundaries—, but instead assigning the correct label out of a large number of possible categories. The challenge in medical images is often not the large number of classes, but instead to correctly identify entities that exhibit no clear boundary and an appearance similar to their surroundings.

Notable work in the medical domain was put forward by Top et al. (2010) and Top et al. (2011), who rely on an active contour for segmentation and construct a measure of uncertainty that is then used to identify a plane (which can be oblique) of maximal uncertainty in which the user is asked to provide additional inputs. While their measure of uncertainty should generalize to various segmentation techniques, the authors use a contour based approach, which is suboptimal for problems like ours, where there are no clear boundaries. Additionally radiologists often work in the predefined orthogonal orientations, so that oblique planes might be more confusing than helpful. We chose to restrict ourselves only to axial, sagittal and coronal planes for annotation.

Konyushkova et al. (2015) work with superpixels and specifically incorporate correlations with neighbouring superpixels into a measure of *geometric uncertainty* that is then used to identify a plane for optimal annotation, which will again be generally oblique. Interestingly, the authors evaluate their approach on MRI data of glioblastoma patients, the same task we worked on. We will compare our results with theirs in Section 3.5. Note however that their mode of interaction is binary,

meaning the user must only decide on a boundary between inside and outside of a target object. Our segmentation approach naturally incorporates multi-class segmentation.

Maiora et al. (2014) and Chyzhyk et al. (2015) both combine a Random Forest classifier with active learning to segment Abdominal Aortic Aneurysm and stroke lesions respectively. Both also employ pixel-level annotations and an interactive workflow, but require users to annotate with single-pixel accuracy. The segmentation problems they tackle are binary and their query measure is the standard deviation of the class labels, which, if at all, makes sense only for binary categorization (using 0 and 1 as numerical values).

3.3 METHODS

3.3.1 *Data & Preprocessing*

The experiments in this chapter were designed with the express purpose of finding an efficient way to annotate the larger dataset used in the subsequent parts of this work. As a consequence, they had to be conducted on a different dataset, and we selected the 2013 BraTS challenge (Menze et al., 2015), which comprises magnetic resonance imaging (MRI) scans of glioblastoma patients similar to our target dataset. The full BraTS dataset consists of both real (acquired at field strengths of 1.5T or 3T) and synthetic MRI data for both low grade and high grade glioma patients. In our experiments we intentionally left out the synthetic data, because they "are less variable in intensity and less artifact-loaded than the real images" (Menze et al., 2015), as well as the low grade glioma data, because the high grade cases are much more difficult to annotate and thus require significantly more expert labor. We could have chosen data from later iterations of the BraTS challenge, but 2013 was the last year that had fully manual annotations. In the following years, annotations were generated by taking predictions from the best performing algorithms from previous years and then manually correcting these predictions. The process has been critized as introducing bias (Menze et al., 2015).

The remaining data we conducted our experiments with comprised 20 individual subjects, for each of which there were four three-dimensional image volumes of different MR contrasts available: native T1-weighted (T1), contrast-enhanced (Gadolinium) T1 (T1ce), native T2 (T2) and native T2-weighted FLAIR[1] (FLAIR). An example of what these contrasts look like is given in Figure 3.2.

The available data for each patient were already co-registered, resampled to 1mm isotropic resolution and skull-stripped by the challenge authors, we further applied the following pre-processing:

1 Fluid-attenuated inversion recovery

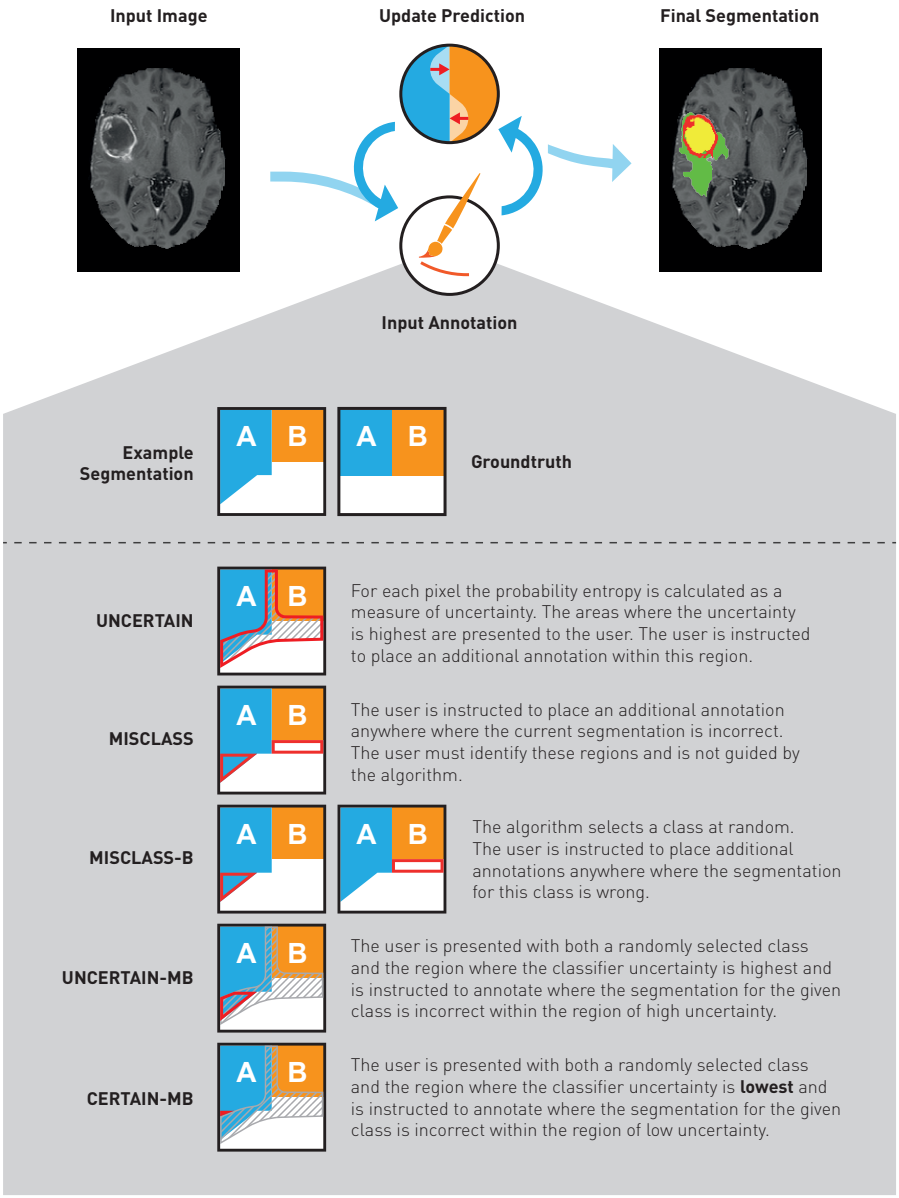

Figure 3.1: Visualization of the interactive annotation process. For each annotation mode an exemplary region is shown in which the simulated user will annotate, based on an abstract ground truth and corresponding example segmentation. The update process is repeated N times and yields the final segmentation. Note that for demonstration purposes there is only low and high uncertainty, and no intermediate region, hence the regions for UNCERTAIN-MB and CERTAIN-MB share a border.

- Compute the T1ce – T1 difference map as a 5th channel as proposed in Ellingson et al. (2014)
- Perform N3 bias-field correction for T1, T1ce, T2, but not FLAIR, because edema signatures can look similar to field inhomogeneties in this contrast.
- Apply Histogram-Matching using 3D-Slicer's (Fedorov et al., 2012) *HistogramMatching* routine, excluding voxels below mean intensity.
- Normalize intensities by mean CSF[2] value (which is obtained by automatic segmentation), except for the FLAIR, where the CSF already has a value of zero.

For a more detailed description of the effect of these processing steps see Kleesiek et al. (2014). We then computed the following feature maps for use with the classifier presented in Section 3.3.2:

- Gaussian Smoothing ($\sigma = [0.7; 1.6]$)
- Gaussian Gradient Magnitude ($\sigma = [0.7; 1.6]$)
- Laplacian of Gaussian ($\sigma = [0.7; 1.6]$)
- Hessian of Gaussian Eigenvalues (3 feature maps, $\sigma = [0.7; 1.6]$)
- Structure Tensor Eigenvalues (3 feature maps, $\sigma = [0.7; 1.6]$)

This results in a feature vector length of 95. Most image volumes have a size of $176 \times 216 \times 176$ (some patients differ slightly), so that a patient is described by a $176 \times 216 \times 176 \times 95$ matrix.

Our choice of features was motivated by their success in earlier work (Kleesiek et al., 2014; Kleesiek et al., 2016). We did not perform any feature selection on the above set of features, which might have shown some redundancy among them and would have allowed us to select an equally performant subset.

For each of the 20 patients there was a ground truth segmentation obtained by merging manually created annotations from four different raters. The segmentations describe five different tissue categories:

0. Healthy tissue / background
1. Necrosis
2. Edema
3. Non-enhancing abnormalities
4. Enhancing Tumor

2 Cerebral Spinal Fluid

We additionally define the whole tumor region as the union of all four non-background classes. Figure 3.2 shows an example of a ground truth segmentation and the corresponding MR contrasts. The enhancing tumor is best identified from hyperintensities in the T1ce image within the whole tumor region. The necrosis and the non-enhancing tumor regions typically exhibit very similar signatures—they are hypointense in T1-weighted images and hyperintense in T2-weighted images with heterogeneous texture—and are often hard to distinguish. In fact, they are so similar that in later years the BraTS challenge changed their annotation guidelines to no longer distinguish between the two. In the subsequent chapters of this work, we also merge them. The edema can be identified from hyperintense signatures in T2-weighted images, especially FLAIR, that do not belong to the other tumor regions. On average, this class makes up the majority of the tumor region in terms of volume.

3.3.2 *Classifier*

The classifier we employed was a Random Forest (Breiman, 2001), an ensemble classifier that builds multiple decision trees from randomly bootstrapped samples of the training data. A brief introduction is also given in Section 2.3.1. Random Forests have proven to be the best generic choice of classifier among a large variety of conventional (i.e. non-deep learning) methods (Fernandez-Delgado et al., 2014). They are often used very successfully for glioblastoma segmentation (Zikic et al., 2012; Bauer et al., 2013; Kleesiek et al., 2014; Goetz et al., 2016). Our decision for Random Forests was further supported by the availability of toolkits for interactive segmentation that also employ Random Forests and scribble annotations (Sommer et al., 2011; Petersen et al., 2016).

The classifier works on a per-pixel basis, meaning that each pixel, represented by a 95-dimensional feature vector (see previous section), is treated as a separate instance. The decision trees are built from the annotated pixels, which constitute the training set in each case. A prediction is then made (for all unannotated pixels) by letting each decision tree vote for a class. The relative number of votes a label/class receives is treated as its probability and the label with the highest number of votes is assigned to the tested instance. Note that this construction means the classifier only uses information from the current test case for its prediction. In contrast, deep learning methods usually rely on statistics inferred from a larger population.

The measure of uncertainty we use is the probability entropy:

$$H(x) = -\sum_{i \in C} p(y_i|x) \log\left(p(y_i|x)\right) \tag{3.1}$$

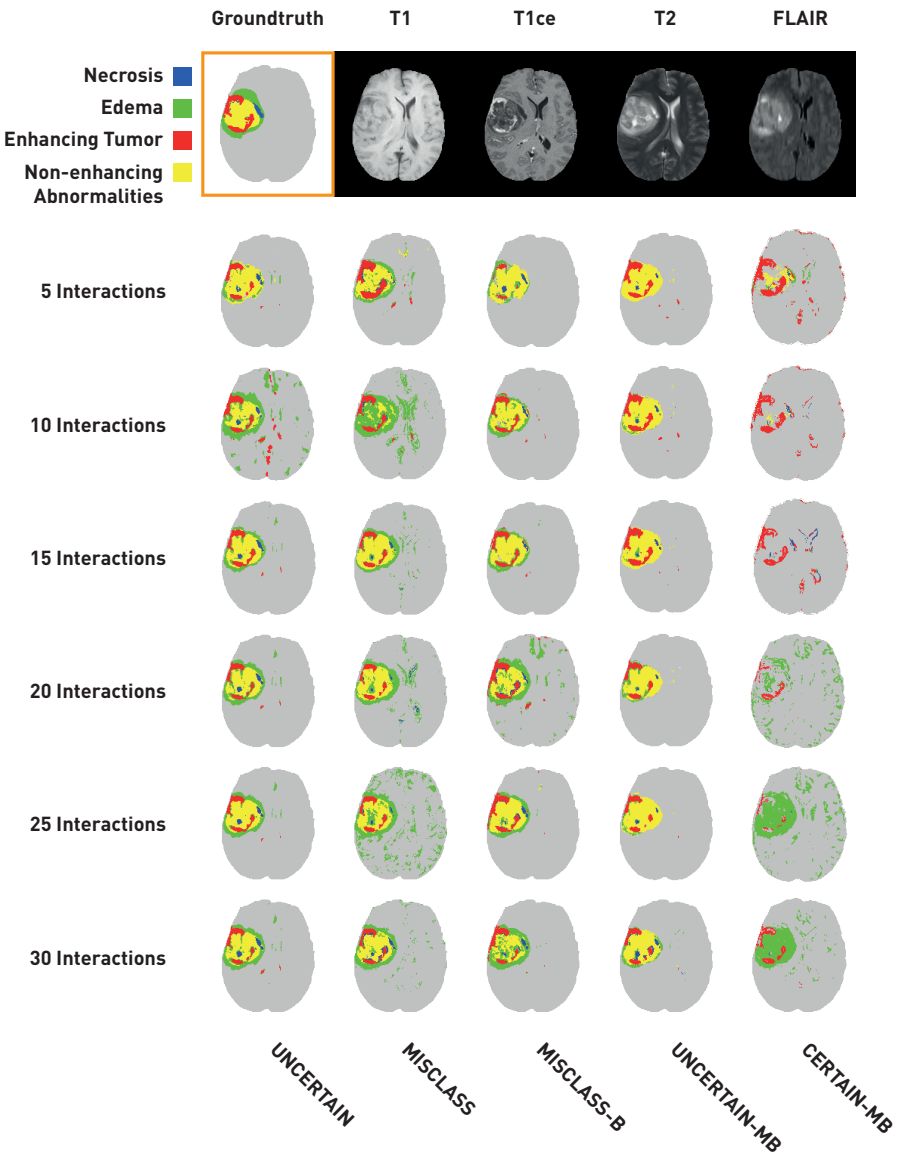

Figure 3.2: Exemplary segmentation results after 5 – 30 interactions for each annotation mode for a random patient along with the corresponding ground truth segmentation and the four base channels (slice selected for illustrative purposes). The results are not representative of the overall segmentation quality for a given method, but show that in general the algorithm needs little data to roughly approximate the solution and that most annotations only refine the result. In almost all cases, there remain very small falsely classified regions throughout the healthy brain, indicating that our results would benefit from post-processing. We refrain from post-processing in this work, as it is more appropriate at the end of the interactive routine, not in every step.

where $p(y_i|x)$ is the probability that pixel x belongs to class i. We also tried confidence and probability margin (Settles, 2010) as uncertainty measures, but the results were not meaningfully different.

The number of trees in the Random Forest as well as their depth was determined with a coarse grid search, the results of which are visualized in Figure A.1; we work with 50 trees and a maximum depth of 10. The split criterion we use is the Gini impurity, and splits are performed by looking at random subset of $\sqrt{95} \simeq 10$ features in each split.

3.3.3 *User Interactions*

Our goal was to simulate experts in an interactive annotation and segmentation process. We wanted to establish how the expert should interact with the algorithm and whether uncertainty information from the algorithm can be used to guide the user in the process. Because we assumed the user to be an expert radiologist, we also assumed they are able to see and interact with all three orientations (axial, coronal, sagittal) simultaneously, and more importantly, we assumed they possess knowledge of the correct segmentation that they wish to transfer onto the image. The basic concept of the iterative segmentation process is that in each step the user sees the current output of the algorithm, ideally as an overlay, to compare it with the underlying data, and then interacts with the algorithm by providing additional training instances.

The interaction process is based on scribble annotations. That means that the user can impaint pixels in the image to label them as belonging to a certain class (note that we use the terms *class* and *label* interchangeably), similar to a paintbrush tool found in almost any image editing application. Theoretically this would allow the user to paint in any way they desire (single disconnected pixels, large round blobs, etc.), but the most common and intuitive way to annotate in such a scenario is by painting lines, or scribbles. The classifier is trained on the labeled pixels and the resulting segmentation is presented to the user so they can add a new input to improve the output.

We deliberately chose to over-sample the interaction process by allowing only very short scribbles of 10 connected pixels and by updating the prediction after each scribble annotation. Experienced users usually make multiple annotation scribbles before updating the prediction, especially in the beginning, where the algorithm requires at least some data from all classes for a somewhat reasonable prediction. In our simulation we skipped this initial step by initializing the algorithm with 50 randomly drawn pixels, weighted by class occurrence (so that each class received at least one training instance). We found 50 to be

the lowest number of initial training points to achieve a reasonable initialization. We discuss this further in Section 3.4.

Having established the general process of how users place their annotations, i.e. by painting a small number of connected pixels with the correct label, the interesting question is *where* user inputs should be given to create a high quality segmentation with the least amount of interactions. To this end we defined five interaction modes, each characterized by a region in which the user will place their annotation randomly in each step, based on classifier uncertainty (information the algorithm possesses) or on correctness (information the user possesses). These regions will of course change in each step. Abstract examples for all methods are given in Figure 3.1. While the simulated users generally annotated randomly within a specified region, we placed some further constraints on the inputs to make them more realistic: the scribbles, which we fixed at a length of 10 pixels, must be connected. They must further be in one of three main planes (axial, sagittal, coronal). They must lie entirely within the specified region and finally the must not cross classes. We defined the following annotation modes:

1. UNCERTAIN: *Place annotations randomly in regions of high classifier uncertainty.*
   To simulate this we first divided the uncertainty into 5 quantiles and kept only the regions belonging the highest quantile. We then find the largest connected region of those and annotated randomly within this region. The dividing into 5 quantiles might seem somewhat arbitrary, but we empirically found that using the top 20% of the uncertainty still resulted in large enough regions that one could comfortably annotate, while smaller numbers would often yield very small and thin regions that require pixel accuracy annotations.

2. MISCLASS: *The user identifies falsely classified regions in the segmentation overlay from the previous step and then randomly annotates anywhere in the entire error region.*
   This implicitly weights classes by occurrence. This method depends on our assumption that the user has knowledge of the correct segmentation and is able to identify falsely classified regions in the segmentation overlay. It does not use uncertainty information.

3. MISCLASS-B: *The algorithm chooses a class at random and the user identifies and annotates in falsely classified regions (both false positive and false negative) for that particular class in the segmentation overlay.*
   This weights classes equally. The method also depends on the assumption that the user has knowledge of the correct segmentation and is able to identify falsely classified regions. It does not use uncertainty information.

4. UNCERTAIN-MB: *A combination of UNCERTAIN and MISCLASS-B.* Annotations are placed where the region identified by UNCERTAIN and the error region for a randomly chosen class intersect. Should there be no intersection, ignore uncertainty region, i.e. fall back to MISCLASS-B. This method utilizes both uncertainty and correctness information.

5. CERTAIN-MB: *Essentially the same as UNCERTAIN-MB, but using the most certain region.*
   We now identify the region where the classifier is *most* certain, meaning the lowest of 5 quantiles of the uncertainty. This might seem counterintuitive, but we hypothesize that if the classifier is very certain about an error, the corrective annotation should have a much stronger effect. Again, if there is no intersection, fall back to MISCLASS-B.

3.3.4 *Evaluation*

Our goal was to evaluate the quality of the segmentation over time, i.e. as a function of the number of interactions. The de facto standard for segmentation assessment in the medical domain is the Soerensen-Dice coefficient introduced in Section 2.3.3, a binary measure we computed for all classes separately.

For each patient and for each interaction method we evaluated the Dice scores over the course of 50 interactions. After each interaction step the classifier was trained on all pixels that were annotated in the current and the previous interactions. We used no training data from other patients or from earlier assessments of the same patient. A prediction was always made on the entire three-dimensional image volume for the current patient that was then compared with the corresponding ground truth segmentation. We repeated the process five times for each patient and averaged the results to suppress random variations, treating the 5-run average as a single measurement. For our first analysis we then also averaged the scores for all patients and compared the different interaction methods by means of the Dice score as a function of the number of interactions.

For our second analysis we did not average scores across patients. We performed a statistical comparison of the methods after 20 interactions. The findings are not very dependent on the evaluation point and after roughly 20 interactions the benefit of additional annotations became rather small, as seen in Figure 3.3. For each pair of methods and each region we used a Wilcoxon signed-rank test (Wilcoxon, 1945) to find the probability p that the two sets of measurements (a set of measurements meaning the scores for the 20 different patients of a given method) originate from the same distribution (not all our measurements were normally distributed). We chose a base significance threshold of $p < 0.05$ and applied Bonferroni correction for 50 indi-

vidual tests (5 regions times 10 comparisons), resulting in an adjusted threshold of $p < 0.001$. Note that the results are not independent, so the correction was likely stronger than necessary.

3.4 RESULTS

Figure 3.2 shows exemplary segmentation results for a single patient and 30 interactions in steps of 5. For comparison the ground truth segmentation and the four base channels (features) are displayed. These results are of course only a single sample from a stochastic process and are not necessarily representative of the overall performance of the approaches. However, a few things can be seen that were similar in the majority of cases. In general, very little training is necessary to get a rough estimate of the desired result (with the exception of CERTAIN-MB in this case). Most later inputs introduce rather small changes and only refine the segmentation. The segmentation doesn't necessarily improve in each step, in most cases this is due to considerable changes in the edema region. Lastly, there are almost always a number of very small false positive regions dispersed throughout the healthy part of the brain. We will discuss these findings in detail in Section 3.5.

To obtain quantitative results, we simulated the interactive segmentation process 5 times for each of the 20 different patients. Note that the classifier used only live input annotations for the current subject and did not incorporate knowledge from other patients or earlier assessments. In each step, the training set consisted of all pixels annotated by the simulated user and the test set was the remainder of unannotated pixels.

Figure 3.3 shows the Dice score over time for all methods and tumor classes including the 1σ standard deviation of the 5-run patient means for the overall best performing method MISCLASS-B to illustrate how scores varied across different patients. Other methods' standard deviations are comparable. Figure 3.4 represents a cross section of Figure 3.3 and shows mean Dice scores after 20 interaction cycles for all methods and classes. Highlighted are all pairs of methods with a significant ($p < 0.001$) performance difference. A full overview of test scores (p-values, test statistics and difference of medians) is given in Table 3.1. In the following we first present results for the comparison of annotations in uncertain regions (UNCERTAIN) and corrective annotations (MISCLASS & MISCLASS-B). We then show the comparison of the best of those methods (MISCLASS-B) with corrective annotations in very uncertain (UNCERTAIN-MB) and very certain (CERTAIN-MB) regions.

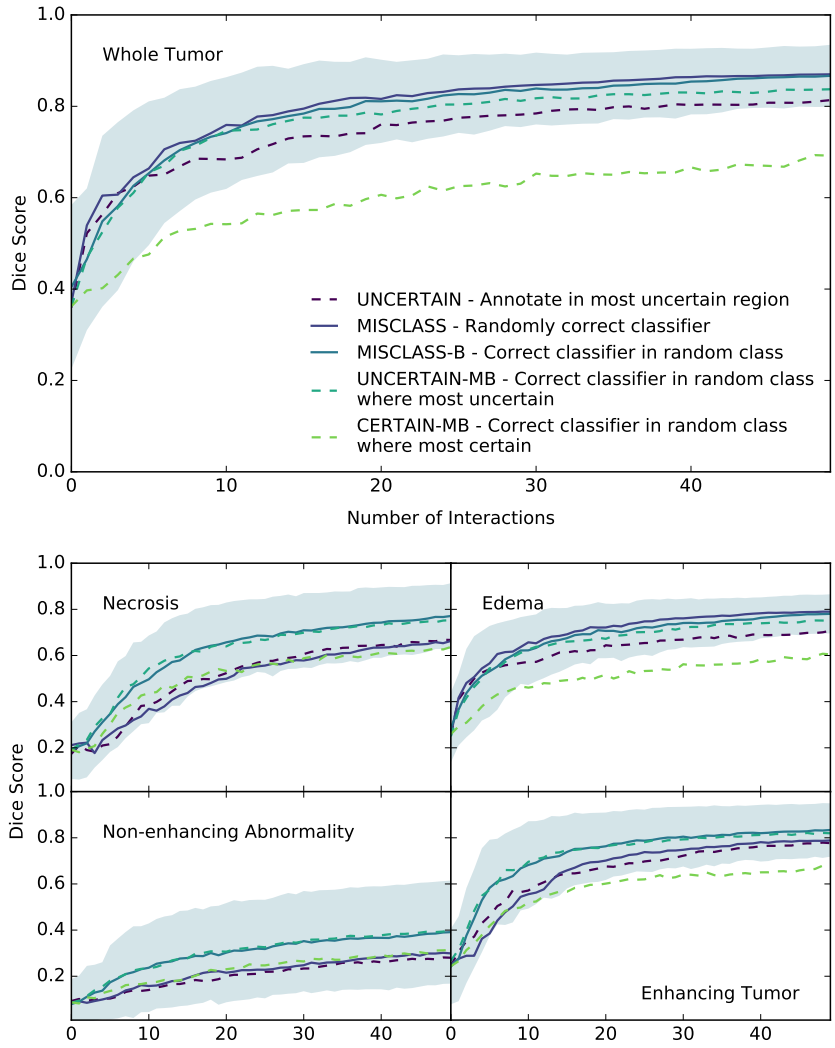

Figure 3.3: Dice score as a function of the number of interactions for different tumor regions. Dashed lines indicate that the method relies on the prediction uncertainty. Filled area shows 1σ standard deviation of patient means for MISCLASS-B to give an estimate of the spread of scores across patients. Standard deviations for other methods are comparable. MISCLASS-B and UNCERTAIN-MB show the overall best performance in all regions. In the larger regions edema and whole tumor, MISCLASS performs similarly, in smaller regions (necrotic core, enhancing and non-enhancing tumor) MISCLASS and UNCERTAIN perform comparably. CERTAIN-MB is always among the poorest performing methods.

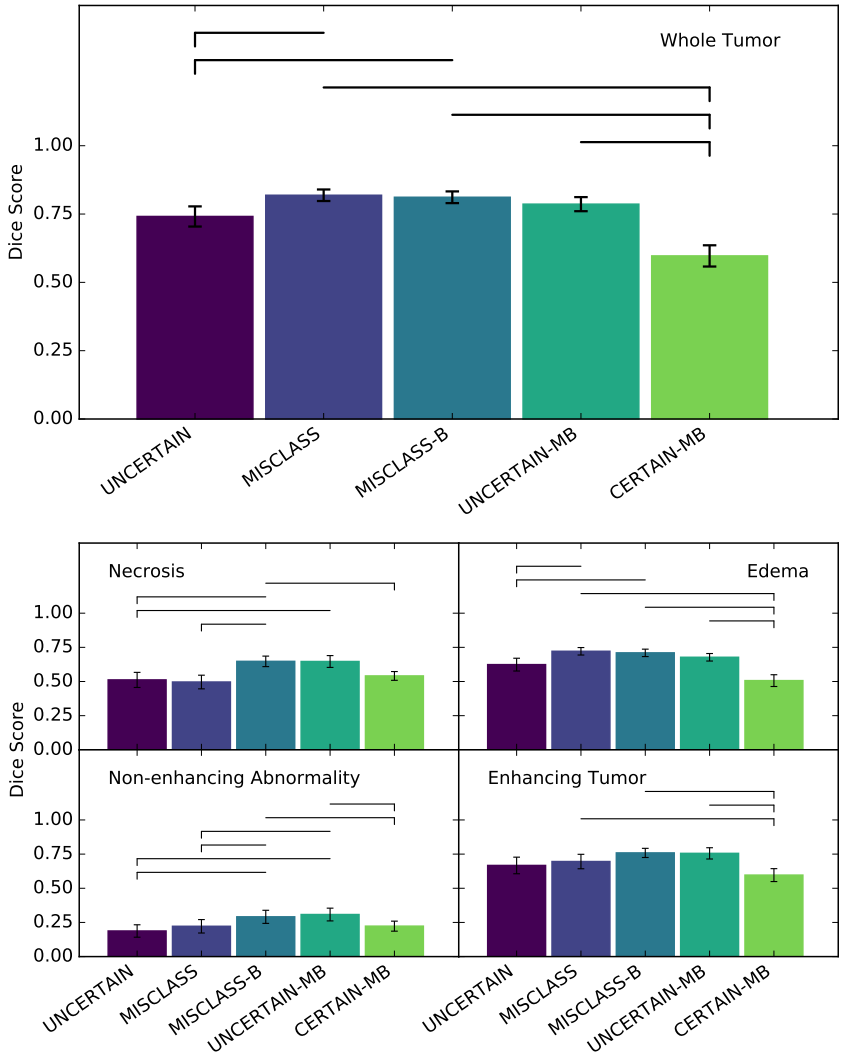

Figure 3.4: Dice score after 20 interaction cycles for different tumor regions. Errors show standard error of the mean. Horizontal bars indicate that $p < 0.001$ for the Wilcoxon signed-rank test of the two methods, with a dash indicating the method with poorer performance. This is essentially a cross section of Figure 3.3 after 20 interactions. Again, in the larger regions (edema and whole tumor) MISCLASS, MISCLASS-B and UNCERTAIN-MB perform comparably while in the other region MISCLASS-B and UNCERTAIN-MB dominate and UNCERTAIN and MISCLASS perform similarly. CERTAIN-MB performs poorly in all regions.

Table 3.1: Pairwise comparison of all methods for each tumor region after 20 iterations, using a Wilcoxon signed-rank test and 5-run averages for each patient. Displayed are test statistic and p-value results as well as the difference of the medians for each comparison. Highlighted in bold are comparisons where $p < 0.001$. This threshold is the result of a base significance level of $p < 0.05$, Bonferroni corrected by 50 individual comparisons. See Table A.1 for Edema and non-enhancing regions.

Methods	Whole Tumor			Necrosis			Enhancing Tumor		
	Statistic	p	ΔMedian	Statistic	p	ΔMedian	Statistic	p	ΔMedian
UNCERTAIN v MISCLASS	**11**	**<0.001**	**−0.073**	101	0.881	0.049	81	0.370	−0.014
UNCERTAIN v MISCLASS-B	**12**	**<0.001**	**−0.041**	**12**	**<0.001**	**−0.126**	45	0.025	−0.058
UNCERTAIN v UNCERTAIN-MB	45	0.025	−0.026	**4**	**<0.001**	**−0.156**	32	0.006	−0.060
UNCERTAIN v CERTAIN-MB	19	0.001	0.150	99	0.823	−0.002	57	0.073	0.114
MISCLASS v MISCLASS-B	65	0.135	0.031	**15**	**<0.001**	**−0.175**	40	0.015	−0.044
MISCLASS v UNCERTAIN-MB	32	0.006	0.047	19	0.001	−0.205	38	0.012	−0.046
MISCLASS v CERTAIN-MB	**0**	**<0.001**	**0.222**	78	0.313	−0.051	**12**	**<0.001**	**0.129**
MISCLASS-B v UNCERTAIN-MB	30	0.005	0.016	100	0.852	−0.030	100	0.852	−0.002
MISCLASS-B v CERTAIN-MB	**0**	**<0.001**	**0.191**	**7**	**<0.001**	**0.124**	**0**	**<0.001**	**0.173**
UNCERTAIN-MB v CERTAIN-MB	**2**	**<0.001**	**0.176**	18	0.001	0.154	**0**	**<0.001**	**0.175**

3.4.1 *Annotating Uncertain Regions vs Classifier Correction*

In both Figure 3.3 and Figure 3.4 it can be seen that annotations in uncertain regions (UNCERTAIN) performed worse than class-balanced classifier corrections (MISCLASS-B) across all categories and over time, and the difference after 20 interactions was significant in all regions but the enhancing tumor with $-0.024 \leqslant \Delta\text{Median} \leqslant -0.126$.

Annotations in uncertain regions (UNCERTAIN) also performed worse than random corrective annotations (MISCLASS) in the whole tumor region and the edema. The difference after 20 interactions was significant for both the whole tumor ($\Delta\text{Median} = -0.073$) and the edema ($\Delta\text{Median} = -0.038$). Performances were roughly on par in the smaller necrosis, enhancing and non-enhancing regions.

Random classifier corrections (MISCLASS) performed significantly worse than class-balanced corrections (MISCLASS-B) in the necrotic core regions ($\Delta\text{Median} = -0.175$) and the non-enhancing regions ($\Delta\text{Median} = -0.086$). They also performed worse in the enhancing tumor region, but without a significant difference after 20 interactions. In the larger whole tumor and edema regions the two were roughly on par.

3.4.2 *Combination of Uncertainty-based Annotations and Classifier Correction*

As outlined above, MISCLASS-B was among the top performing approaches in all tumor regions. We now compare it with UNCERTAIN-MB and CERTAIN-MB; both methods are designed to work like MISCLASS-B and to additionally incorporate uncertainty information to improve segmentation results. We intentionally leave out the comparison of UNCERTAIN-MB and CERTAIN-MB with MISCLASS and UNCERTAIN.

Figure 3.3 & Figure 3.4 show that CERTAIN-MB, the class-balanced corrective annotations in the most certain regions, was always among the poorest performing approaches and performed significantly worse than balanced corrections (MISCLASS-B) as well as balanced corrections in uncertain regions (UNCERTAIN-MB) after 20 interactions across all classes except for the necrotic core, where $p = 0.001$ (but not $p < 0.001$) for the comparison between CERTAIN-MB and UNCERTAIN-MB.

Balanced classifier corrections (MISCLASS-B) and the combination of that approach with annotations in uncertain regions (UNCERTAIN-MB) performed similarly well across class with a slight advantage for the former over the latter in the whole tumor and edema regions. However, the difference after 20 interactions was not significant in any of the tissue classes.

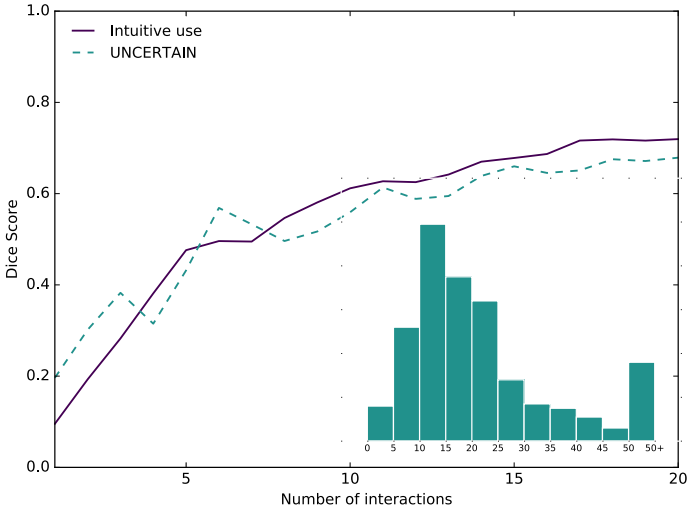

Figure 3.5: Interactions by real human users with between 1 and 4 years experience working with glioblastomas. Users were first asked to annotate without instructions, then used the UNCERTAIN method. Intuitive annotations perform better than annotations in uncertain regions, because the majority of annotations users provide are corrective. Overall scores are lower than what was achieved in the simulations, likely because users do not fully satisfy our assumption that they possess knowledge of the correct segmentation. Inlay: The distribution of scribble lengths.

We mentioned in Section 3.3.3 that we initialized the algorithm with 50 random training instances, distributed among the classes by their relative occurrence. The purpose of this was to skip the initial steps where the classifier has too little information about a given class and essentially stays constant at 0 or a very low score. This can be seen in Figure 3.3 in the necrotic region. The score appears to be at a constant low before increasing quite sharply and then following the common pattern. Without initialization this effect would be much more pronounced and visible in all classes. The best final Dice scores out of all methods after 50 interactions for each class were 0.870 for the whole tumor, 0.771 for necrosis, 0.789 for the edema, 0.400 for the non-enhancing abnormalities and 0.833 for the enhancing tumor.

To get an idea of how users would annotate intuitively, we let four users (1 to 4 years experience working with glioblastoma) annotate a subset of randomly selected patients. In total we recorded 4 separate assessments for each of four different patients, where each rater performed 2 assessments on a given patient, first with no instructions and then following the UNCERTAIN approach as a comparative baseline. The result for the whole tumor region is shown in Figure 3.5

3.5 DISCUSSION

We compared five different methods of providing annotations in the context of online interactive segmentation based on a pixel classifier that receives inputs in the form of annotation scribbles. We compared them with respect to their ability to evoke inputs that let the classifier make faithful predictions with minimal interactive effort. Our analysis is based on a Random Forest classifier, applied to the task of segmenting multiple glioblastoma tissue classes in magnetic resonance images. The efficient annotation of a large dataset of this kind was a prerequisite for the work shown in the following chapters.

The methods we proposed use uncertainty information from the classifier and correctness information from the user. Interactive segmentation based on scribbles is not a new concept and neither is the usage of uncertainty information in interactive segmentation (Triebel et al., 2014; Maiora et al., 2014; Chyzhyk et al., 2015). But to the best of our knowledge we were the first to compare in this context the merits of inputs in regions of high uncertainty and inputs that correct the classifier. Consequently we know of no prior work that attempts to combine both in this context.

We recognize that our approach is quite specific in the sense that we tested methods that could be applied to almost any choice of classifier only on a single one, namely a Random Forest, however this was motivated by this classifier's remarkable performance across a multitude of challenges (see Section 3.3.2). And indeed the final Dice scores we obtain after 50 interactions compare favorably to previously reported results on the same dataset (Menze et al., 2015), even those from early work with CNNs (Pereira et al., 2016). It is important to recognize that this work does not compete with automated segmentation methods, but rather enables them. Since the publication of our results, glioblastoma segmentation as well as the field of medical image segmentation in general has progressed remarkably. This is not least due to the increased availability of annotated data in the context of challenges, for example the Medical Segmentation Decathlon (Simpson et al., 2019), a combination of ten different segmentation tasks with an average of 263 annotated training items per task. Our contribution (Isensee et al., 2018; Isensee et al., 2020) won this challenge and represents the state-of-the-art in automatic medical image segmentation.

We observed that in general our algorithm needs relatively little training data to get a rough estimate of the correct segmentation (especially the whole tumor region) and that most later inputs only refine the segmentation. Interestingly, while the results on average improve over time, this is not necessarily the case for any single experiment. Especially decisions with unclear boundaries, for example the transition from the edema to healthy tissue, can change quite drastically with small changes in the training set. This behaviour

can be seen in the exemplary results in Figure 3.2 for MISCLASS between steps 20 and 25. Note that the classifier finds edema regions throughout the brain. That is a common limitation of many classical approaches, including ours, that perform predictions on a per-pixel basis. Correlation between neighbouring pixels is only introduced in the feature maps. The problem can be mitigated by introducing post-processing, for example regularization with Conditional Random Fields (Schroff et al., 2008). The computation requirements render such processing infeasible for an interactive setting, which is why we intentionally excluded it from our experiments.

Another interesting observation one can make in the examples we present is that the algorithm picks up a patch of necrosis in the center of the tumor, whereas the ground truth segmentation classifies it as non-enhancing tumor. The two classes have very similar imaging signatures and judging from the MRI channels it would be an entirely reasonable decision to classify the patch as necrosis. Evidently the interactive segmentation workflow can serve to give the user feedback on their perceived correct segmentation. In a comparable setting this was shown to reduce inter- and intra-rater variability (Kleesiek et al., 2016). Do note that later iterations of the BraTS challenge recognized this problem and altered their annotation protocol (Bakas et al., 2019).

The methods we compared were annotations where the classifier uncertainty is highest (UNCERTAIN), annotations that randomly correct the classifier (MISCLASS), annotations that correct the classifier, but with equal distribution of inputs among classes (MISCLASS-B) as well as balanced corrections in regions of high uncertainty (UNCERTAIN-MB) as well as regions of low uncertainty (CERTAIN-MB). Note that we first compared UNCERTAIN, MISCLASS and MISCLASS-B, where MISCLASS-B emerged as the best performing approach, and then compared only MISCLASS-B with UNCERTAIN-MB and CERTAIN-MB, neglecting the remaining comparisons.

3.5.1 *Annotating Uncertain Regions vs Classifier Correction*

In the comparison of annotations in regions of high classifier uncertainty (UNCERTAIN), random corrective annotations (MISCLASS) and class-balanced corrective annotations (MISCLASS-B) we found that it is generally preferable to let users annotate falsely classified regions, assuming the user has complete knowledge of the correct segmentation, because MISCLASS-B performed better than UNCERTAIN in all regions and significantly so in all but the enhancing tumor region, while MISCLASS performed significantly better than UNCERTAIN in the whole tumor and edema regions. The difference between MISCLASS and MISCLASS-B can be attributed to the fact that the problem is one with a large class imbalance. The edema and whole tumor regions are generally large or not much smaller than the background,

and hence are automatically balanced with respect to the background, in which case there is no functional difference between MISCLASS and MISCLASS-B. This is reflected in the results where the two exhibited very similar performance in those two classes. In the smaller regions on the other hand, MISCLASS-B performed better (significantly so in the necrotic and the non-enhancing region), because purely random annotations are more likely to miss those regions, resulting in fewer training data from which the classifier can learn to discern them. This will likely hold true for most scenarios with a strong class-imbalance. Note that our findings also suggest that classifier uncertainty and classification error are generally not congruent.

3.5.2 *Combination of Uncertainty-based Annotations and Classifier Correction*

Because MISCLASS-B, the class-balanced corrective annotations, proved to be such a successful approach, we were curious if it could be combined with knowledge about the classifier uncertainty. We had two opposing hypotheses in this regard: Either that performing the corrective annotations in the most uncertain regions could boost the performance or, to the contrary, that doing so in the most certain regions could improve the performance, because the corrective effect should be stronger in the latter case. The second idea is clearly refuted by our results as CERTAIN-MB was among the poorest performing approaches for all tissue classes and performed significantly worse than MISCLASS-B and UNCERTAIN-MB across all classes. UNCERTAIN-MB on the other hand performed about as well as MISCLASS-B, but did not improve upon the performance of MISCLASS-B, so that both of our hypotheses can be dismissed. Because of the additional computational cost of computing the uncertainty it is beneficial to prefer MISCLASS-B over UNCERTAIN-MB.

Out of the publications mentioned in the beginning of this chapter, Konyushkova et al. (2015) are the only ones who report the Dice score as a function of the number of interactions for a comparable task and we will compare our results with theirs in some detail. The authors apply their geometric uncertainty sampling to the 2012 BraTS challenge data while we use data from the following year. It is not immediately clear what their segmentation objective is, as the 2012 BraTS challenge specifies two tumor categories. We assume the authors just segmented both tumor classes as a union, like the Whole Tumor category we evaluate. From a qualitative perspective the curves Konyushkova et al. (2015) obtain exhibit the same characteristics as ours, with a steep incline in the beginning that gradually becomes smaller. Interestingly, the methods they compare perform virtually the same for the first 10 interactions up to a Dice score of 0.4, which is exactly the range of scores we skip by providing initial training samples. The authors

compare four different query methods, one of which is very similar to UNCERTAIN, as it always selects the most uncertain superpixel for annotation. Not including the first 10 interactions, their method achieves scores of (0.4, 0.5, 0.6, 0.65) in the first steps of 10 interactions while ours achieves (0.4, 0.65, 0.7, 0.75) in the same interval (rounded to 0.05 accuracy). Their method seems to asymptotically approach a score of 0.75, ours tends towards 0.8. Their best performing method stays just below 0.8, while ours is again at an advantage of about 0.05 points. Do note that this comparison is based on a visual assessment of their figures. One interesting finding they report is that random sampling results in an almost constant Dice score. We could confirm this, but chose to omit the result, as it is in no way representative of a realistic user.

To get an estimate of how a real user would approach the problem, we let four different users with varying experience (1 to 4 years in the relevant domain) annotate four randomly selected patients, first without instructions (see Figure 3.5). Our hypothesis was that users would intuitively tend to a corrective annotation style, but because the individuals can not all be considered experts, we let them annotate using the UNCERTAIN method as a baseline, as it does not rely on correctness information. Users did indeed annotate in a corrective manner, but less pronounced than expected. In total 79/197 annotations were fully corrective (compared to 54/212 for UNCERTAIN) and 146/197 majority corrective (compared to 116/212 for UNCERTAIN), but of course we cannot assess whether non-corrective annotations were intended to be that way. Intuitive use only showed a slight margin over uncertainty-guided annotations, however the data we collected were too few in number to support this finding in a statistically significant manner. The scores obtained for the UNCERTAIN approach were lower than in our simulation, which is not surprising, as not all our users can be considered experts. The strokes users used were mostly between 10 and 25 pixels in length, so that our simulations are in fact quite realistic in this regard, but this could be due to demonstration bias.

Overall, our main finding was that correcting the classifier was significantly more efficient than providing inputs where it is uncertain. This is not too surprising, as it is easy to imagine that corrections will on average effect stronger change in the model. More surprising was the fact that a combination of the two yielded no additional benefit. We assume that corrections will on average happen automatically at points where the classifier is uncertain about its output, which would result in MISCLASS-B and UNCERTAIN-MB performing similarly, which is what we observed. At the same time we found a significant difference between UNCERTAIN and MISCLASS-B. This would then imply that on average the error regions are a subset of the high uncertainty regions. In the cases we inspected visually, we found most

of the error regions to overlap with the uncertainty regions to a large extent, but not entirely.

Our findings suggest that uncertainty information, at least the probability entropy, is virtually useless to query inputs from a user, which is in contrast to existing literature in the active learning domain, much of which is concerned with finding variations of measures that describe the *model's* knowledge or lack thereof. In our scenario the *user's* knowledge was clearly more important. Of course, this relies strongly on the assumption that the user possesses knowledge of the correct segmentation. If we were to omit this constraint and compare our UNCERTAIN method with completely random annotations, it would fare much better. We tested this, and completely random annotations performed even worse than CERTAIN-MB, obviously because small regions will almost never be annotated. However, as elaborated on above, it is in no way reasonable to assume that a real user would just place random annotations, which is why we did not include these results. We chose the probability entropy as a measure of the model's uncertainty mainly because it is very easy to compute. The question remains whether there are other, maybe more complex, measures of uncertainty or ways to query inputs from the user at certain points that would achieve even better results. This is of course the key objective in active learning, where numerous methods to tackling this have been proposed. Settles (2010) gives a good introduction to the different groups of approaches; the ones that are conceptually more similar to our corrective annotations, like expected model change and expected error reduction, are unfortunately also among the most computationally expensive and hardly usable in a truly interactive setting.

Potential improvements over what we have shown could lie in methods that exploit committees such as the Random Forest classifier we used. It might be worth exploring ways to intelligently reweight individual trees based on how well they agree with new inputs and criteria to reject existing and to build new trees. As an additional benefit, this could also speed up the training and prediction steps. It should also be noted that our methodology is such that it does not translate easily to larger data (like larger volumes or multiple observations over time), because it requires the user to view as much as possible of the given instance at a given time. We further make the assumption that the user knows the correct segmentation and is able to identify falsely classified regions, which is not a trivial assumption for complicated tasks like tumor segmentation. There is also the possibility that our choice of features was not ideal. For example, features that better capture long-range correlations (Kontschieder et al., 2013) could yield an improvement. Finally it should be noted that what we define as uncertainty, i.e. the probability entropy, is not an uncertainty in the Bayesian sense but only a measure of how

confident the classifier is in its prediction. In other words, this type of uncertainty tells us how certain the prediction is, given our model, but not how certain we can be that our model is correct.

To summarize, we found that interactive semantic segmentation of glioblastoma MRI based on a pixel-wise Random Forest classifier should be performed such that the user annotations correct the classifier with a roughly equal number if inputs for all tissue classes. This finding will be relevant for any similar problem with a large class imbalance. For problems with a balanced class distribution it will still be advantageous to prefer corrective annotations over ones where the classifier exhibits high uncertainty. Applications that benefit from these findings are those that seek to create segmentations fast but with a reliability that renders automatic methods inapplicable. The creation of high quality training data for the latter is one such example that also motivated this work. We close with an overview of deep learning-based interactive segmentation, an outlook with hindsight of sorts, as the vast majority of these were published later than our work.

3.6 INTERACTIVE SEGMENTATION WITH DEEP LEARNING

The first studies on deep learning-based interactive segmentation were published while we conducted the experiments presented here. One of the first and most influential was done by Xu et al. (2016), inspiring several follow-up works. They convert binary (foreground/background) user clicks into Euclidean distance maps and feed those as two additional input channels, along with the input image, into a conventional CNN. Lin et al. (2016) perform scribble-supervised learning on superpixels of an input image. The annotated pixels and the segmentation output of a neural network are combined in the unary term of a graphical model, propagating the information via the pairwise terms. They then alternating solve the graphical model (via graph cuts) and update the network parameters during training. Liew et al. (2017) note that the user annotations in the work of Xu et al. (2016) often have relatively little influence on the network prediction. They try to remedy this by separating the CNN into a local and a global branch. The local branch takes higher resolution inputs around the user inputs and produces predictions only for these localized areas that are then fused with the global predictions from the other branch. While Xu et al. (2016) evaluate their network with a simulated user that corrects errors iteratively, this behaviour is not reflected in the training procedure. Mahadevan et al. (2018) modify the click sampling strategy to also simulate an error-correcting user during training. The work by Le et al. (2018a) is very similar to that of Xu et al. (2016), but instead of segmenting regions, their network predicts object boundaries from user clicks. Maninis et al. (2018) encode user clicks not as distance maps but as simple Gaussian blobs. They further let users annotate

extreme points of objects instead of points within object regions and show that this improves performance compared to the work of Xu et al. (2016) and Liew et al. (2017). Hu et al. (2019) postulate that the influence of the user interaction input on the network output becomes stronger when it is introduced later in the network. Consequently, they have two paths in their network—one to process the input image, one to process the user inputs—that are joined just before the final output. The also employ an additional refinement network, as the first one operates on a coarser scale.

A slightly different approach was chosen by Li et al. (2018), who employ two separate networks. The first is trained to produce diverse segmentations—the network has multiple output paths—consistent with the user annotations, while the second network must select one of those as the final prediction. The authors find that this leads to more coherent object boundaries in the individual outputs of the first network. In a sense, their model finds a *latent space* of segmentations, not unlike probabilistic segmentation models (Kohl et al., 2018; Kohl et al., 2019; Baumgartner et al., 2019), including our own presented in Chapter 4 (Petersen et al., 2019). While these were not designed for interactive segmentation, their latent space can be explored manually and allows a user to adjust the prediction to their liking.

Wang et al. (2019a) introduced the first deep interactive segmentation approach in the medical domain. They employ two networks: one produces an initial prediction, the second produces the final prediction from the first and user scribbles that are supposed to correct errors in the first prediction. They also employ a CRF on the output to enforce consistency in the segmentation with the user inputs. Wang et al. (2018) expands on the previous work[3] and evaluates performance on unseen organs with similar appearance to those in the training set. Sakinis et al. (2019) choose an approach very similar to Xu et al. (2016) but encode clicks as Gaussian blobs instead of distance maps. They also focus on performance on unseen but visually similar structures in medical images.

Jang and Kim (2019) observe that in prior art it is not guaranteed that the network outputs match the user input. They introduce a technique called *backpropagating refinement scheme* that ensures this via an iterative optimization procedure. At test time, they define a loss that measures the difference between the network prediction and the user inputs at the annotated points. The loss is backpropagated onto the user input map (a distance map like in Xu et al. (2016)) which is updated while the network weights remain fixed. The authors show that this test time optimization results in significantly improved performance. Sofiiuk et al. (2020) expand upon this work by recognizing that the iterative forward and backward passes come at significant computational cost.

3 Wang et al. (2019a) was published in TMI in 2019, but was already available on arXiv from 2017.

They introduce auxiliary inputs later in the network that are updated in the same way as the distance map input in Jang and Kim (2019). As consequence, the iterative forward and backward passes have to be performed only on a smaller section of the network.

All the works mentioned above are concerned with binary segmentation. Lenczner et al. (2020) extend the work of Xu et al. (2016) to multi-class segmentation by simply having distance map inputs for more than just foreground and background classes, but also try to combine the inputs into a single input channel. Agustsson et al. (2019) choose a rather unique approach to interactive segmentation. They start with a Mask-RCNN (He et al., 2017) but remove the region proposal network and get bounding boxes from user-annotated extreme points like in Maninis et al. (2018). Each of the extracted regions is processed separately to produce a segmentation, but the architecture is modified to also take into account optional user-provided scribbles at this point. As a consequence, their approach can handle both multi-class and instance segmentation in an interactive setting.

4

DISCRETE TUMOR GROWTH MODELING WITH PROBABILISTIC SEGMENTATION

We will be using glioma growth as a motivating example for large parts of this thesis. In the previous chapter we investigated how to efficiently annotate large datasets of MRI scans from glioma patients. The goal in this chapter is to leverage those findings and to try to learn glioma growth dynamics from a large annotated dataset. Existing approaches to modeling these dynamics, an overview of which is given in Section 2.2, employ biologically inspired models of cell diffusion, using image data to estimate associated parameters. These models generally require the explicit specification of the dynamics in the form of an additional term in the diffusion equation. We propose an alternative approach, based on recent advances in probabilistic segmentation and representation learning, that implicitly learns growth dynamics directly from data without an underlying explicit model. We will show that our approach is able to learn a distribution of plausible future tumor appearances conditioned on past observations of the same tumor.

It is important to emphasize that in this chapter we don't concern ourselves with the *prediction* of glioma growth, but instead try to *model* it. We no longer ask "How much will the tumor grow (or shrink)?", but instead ask "If the tumor were to grow (or shrink), what would it look like?". We will explore growth prediction a little more in the following chapter, here the goal is to establish a proof of concept for purely data-driven growth modeling. We further restrict ourselves to growth modeling in *discrete* time steps, even though the main theme of this thesis revolves around learning functions on a *continuous* domain. We will translate our findings to continuous time modeling in Chapter 8.

The findings in this chapter have been partly published in the following manuscript:

Petersen, Jens, Paul F. Jäger, Fabian Isensee, Simon A. A. Kohl, Ulf Neuberger, Wolfgang Wick, Jürgen Debus, Sabine Heiland, Martin Bendszus, Philipp Kickingereder, and Klaus H. Maier-Hein (2019). "Deep Probabilistic Modeling of Glioma Growth". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 806–814.

The chapter follows this publication closely and reproduces sections and figures where appropriate, but we show updated results, as those in the published work were produced on a small subset of the data that was available to us earlier. With the above publication, we also

released the entire source code required to reproduce the experiments, available at https://github.com/jenspetersen/probabilistic-unet.

4.1 INTRODUCTION & RELATED WORK

Our goal in this work is to learn the dynamics of glioma growth from annotated MR image data, without specifying an explicit model. An introduction to glial cell tumors is given in Section 2.2; as we outline there, cell proliferation is a stochastic process on the microscopic scale. To what extent that translates to randomness on the macroscopic scale may be debated, but it is clear that many parameters beyond imaging have an influence on growth behaviour: above all the administered treatment (and changes thereof), but also a patient's age and other health-related factors. We treat these as additional sources of ambiguity and work under the assumption that the growth process is not deterministic and that the observed trajectory for a given patient is only one realization of any number of possibilities.

Virtually all existing glioma growth models are deterministic, the vast majority of which employ a variant of the reaction-diffusion equation. An overview is given in Section 2.2. Of those works, only Menze et al. (2011a), Lê et al. (2017), and Lipková et al. (2019) address uncertainty in one way or the other, but the underlying growth models are still deterministic. The same is true for the machine learning-based approaches presented in the same section. To the best of our knowledge, our work is the only one that tries to represent a distribution over possible growth trajectories. The second major difference of our work compared to reaction-diffusion models is that we attempt to learn growth patterns directly from data, only leveraging the statistics of a large dataset. One might be inclined to think that existing learning-based approaches do the same, but they only do so on a *pixel scale* (see Section 2.2.3). In contrast, our approach learns a distribution on a *global scale*, where each sample represents a differently realized trajectory.

The approach we present is based on the *Probabilistic U-Net* (Kohl et al., 2018), which is an extension of the widely known and used U-Net architecture (Ronneberger et al., 2015) with a so-called *latent space*. In fact, the model can be interpreted as a conditional Variational Autoencoder (Sohn et al., 2015; Jimenez Rezende et al., 2014; Kingma and Welling, 2014) where the decoder is now a U-Net. We interpret consecutive MRI scans for a given patient as different input channels to the model, which limits our approach to data on a grid, i.e. with roughly equal time differences between scans. The Probabilistic U-Net has been introduced, as the name suggests, in the context of probabilistic segmentation: the authors show that a model trained with annotations from multiple raters is able to represent these raters as different positions in the learned latent space. Since publication

of our work this field has seen some progress, with extension of the Probabilistic U-Net to hierarchical latent models (Kohl et al., 2019; Baumgartner et al., 2019) but also other approaches, e.g. by Monteiro et al. (2020) who model correlations between pixels in the output space. Of course, the Probabilistic U-Net is not the only method that allows one to model a continuous distribution of segmentations, in fact much of the work we discuss in the following chapters in the context of learning distributions on continuous domains (see e.g. Section 6.3) can trivially be applied to data on a grid. Kohl et al. (2018) provide an extensive comparison with prior art and show that the Probabilistic U-Net is superior, a) because it models global distributions as opposed to distributions over pixel outputs and b) because it has very well calibrated likelihoods, which means the network captures relative frequencies of class occurrences well. As a consequence, we choose not to re-evaluate these works in our context.

Models that take in multiple consecutive inputs over time are often implemented as *recurrent neural networks* (RNNs)—for an introduction see for example Lipton et al. (2015)—meaning networks that are evaluated multiple time and that allow connections from any node at one time step to any node in the next, including to the same one. One of the motivations for RNNs is that the network doesn't have to learn the same concept multiple times (e.g. one encoder for the first input, another encoder for the second input, etc.), thus saving capacity. With only few inputs, network capacity is typically not an issue. We further pass inputs together through a shared architecture, so that ideally the network can pick up on differences between them at each point in the forward pass. It is however possible that our approach could have just as well been implemented using RNNs.

4.2 METHODS

The underlying hypothesis of our approach is that tumor growth is at least in part stochastic, so that it's not possible to predict a single correct growth trajectory in time from image data alone. Hence, our aim is to *model a distribution of possible changes* of a tumor given the current and in our case one previous observation[1]. We achieve this by training a model to reproduce true samples of observed growth trajectories—with shape and extent of the tumor being represented as multi-class segmentation maps—and using variational inference to allow the model to automatically recognize and account for ambiguity in the task.

1 We also repeated our experiments with three and four input time steps, but found no significant difference.

4.2.1 *Data*

We work with a dataset containing MRI scans from 488 glioma patients, with between 2 and 13 consecutive visits to the hospital. Some more characteristics, like the distribution of time differences between scans or the distribution of changes in tumor size, will be presented in the following chapter. For the published work, we only had a subset of the data available, which contained a total of 199 longitudinal MRI scans from 38 patients suffering from both low grade glioma (15 patients) and glioblastoma (23 patients), with a median of 96 days between scans and 5 scans per patient. Patients have undergone different forms of treatment, a fact that we deliberately neglect by declaring it an additional source of ambiguity in the dataset. Each scan consists of 4 contrasts: native T1 (T1n), postcontrast T1 (T1ce), T2 (T2) and fluid-attenuated inversion recovery (FLAIR), which we also introduce in Section 2.1. All contrasts and time steps for a given patient are skull-stripped, registered to T1 space and resampled to isotropic 1 mm resolution (Jenkinson et al., 2012). For intensity normalization, we only employ basic z-score normalization. Ground truth segmentations of edema, enhancing tumor and necrosis were created semi-automatically by an expert radiologist. We show results both on the published subset and the full dataset available for this thesis.

4.2.2 *Model*

Our model along with the training procedure, based on a probabilistic segmentation approach (Kohl et al., 2018), are visualized in Figure 4.1. The architecture comprises three components: 1. A U-Net (Ronneberger et al., 2015) to map scans from present and past to future tumor appearance. 2. A fully convolutional encoder that maps scans from present and past—presented to the model as different input channels—to an N-dimensional diagonal Gaussian (the *prior*; we choose $N = 3$). 3. An encoder with the same architecture that maps scans from present and past as well as the ground truth segmentation from the future to another diagonal Gaussian (the *posterior*; $N = 3$). The prior and posterior encoders employ global average pooling at the end to remove any spatial resolution. During training we sample from the posterior and concatenate the sample to the activations of the last decoder block in the U-Net, so as to condition the softmax predictions on the sample. We employ multi-class cross entropy as the segmentation loss and use the Kullback-Leiber divergence to force prior and posterior towards each other, so that at test time—when a ground truth segmentation is no longer available—the predicted prior is as close as possible to the unknown posterior. This objective is the

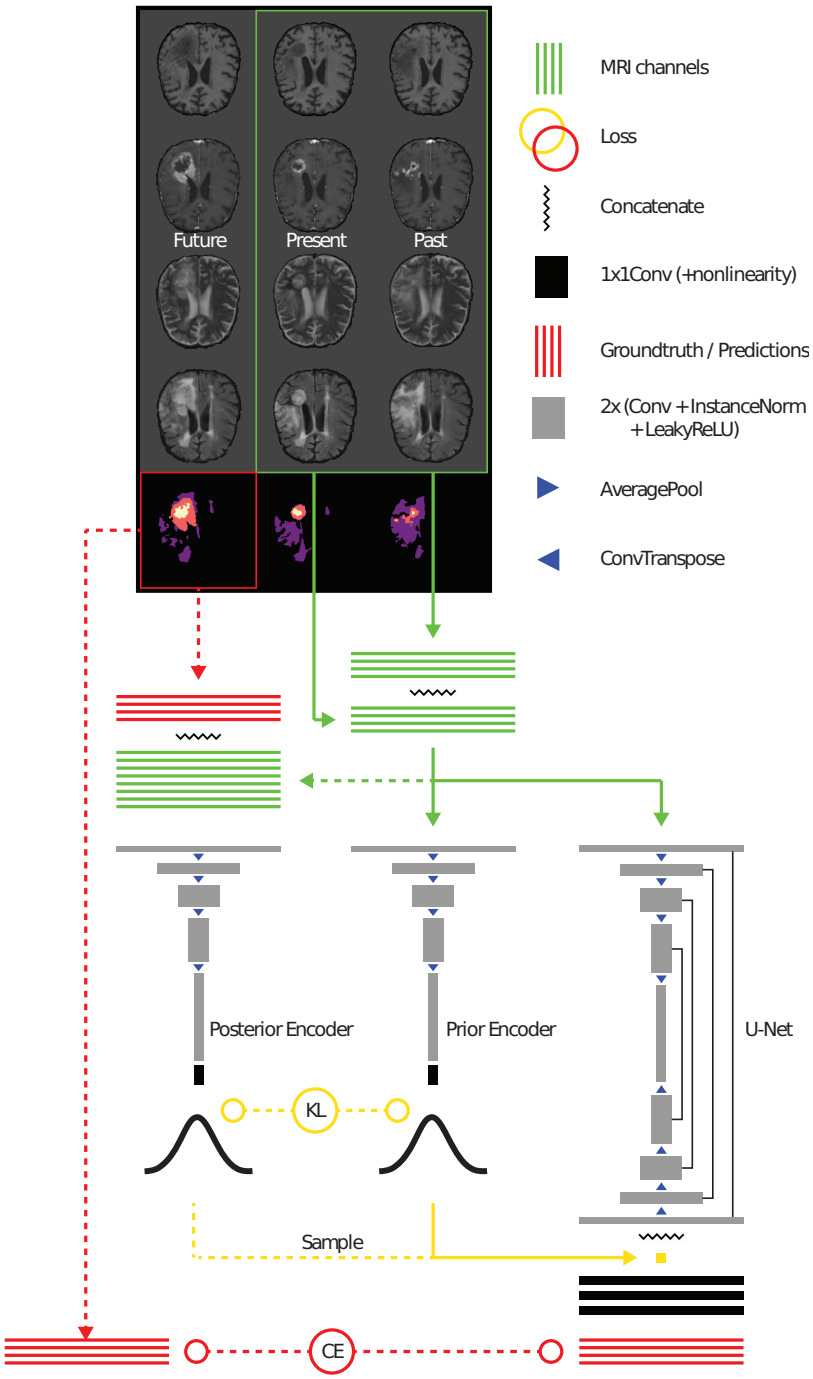

Figure 4.1: The architecture employed in our work. Following the approach in Kohl et al. (2018), a U-Net (Ronneberger et al., 2015) is augmented with two additional encoders, one for the prior and one for the posterior. The prior encoder maps the inputs of present and past scans to an N-dimensional diagonal Gaussian while the posterior does the same with additional access to the ground truth segmentation from the future. During training, a sample from the posterior is injected into the U-Net, during testing samples can only be drawn from the prior. Dashed lines indicate paths that only apply during training.

well known *evidence lower bound* used in variational inference (Bishop, 2006):

$$\max_{\theta} \mathop{\mathbb{E}}_{z \sim q_\theta(Z|X,Y)} \Big[\log P(Y|S_\theta(X,z)) \Big] - D_{KL}\left(q_\theta(z|X,Y) || p_\theta(z|X)\right) \quad (4.1)$$

Here S is the segmentation network, q_θ is the distribution predicted by the posterior encoder and p_θ the distribution predicted by the prior encoder. P is a pixel-wise categorical distribution, which translates to a cross entropy loss.

The described training scheme will give rise to the following desirable properties: 1) The model will learn to represent the task's intrinsic ambiguity in the Gaussian latent space, in our case different plausible future tumor shapes and sizes, as we show in Section 4.3. 2) At test time we can sample multiple consistent hypotheses from the latent space (as seen in Figure 4.2), and select those that match desired criteria (e.g. tumor volume increases by 20%).

We train with data augmentation[2] on patches of size 112^3, but evaluate on full sized scans of 192^3. We train for a total of 50 000 batches of batch size 2, using the Adam optimizer (Kingma and Ba, 2015). The initial learning rate of 0.0001 is repeatedly decayed with a factor of $\gamma = 0.985$ after 200 batches. There are obviously vastly more parameter configurations than we could hope to try, so our main goal was to find one that resulted in reliably stable training.

4.2.3 *Experiments & Evaluation*

As outlined in Section 4.1, existing biological growth models, even those that incorporate uncertainty, are deterministic, while we seek to learn a distribution of possible growth trajectories, so these models can't serve as a baseline to judge the quality of the distributions learned by our model. Our goal is to show that our approach learns meaningful future tumor appearances instead of just segmentation variants of the present input. For this reason we construct a baseline that is restricted to learning the latter.

Let A denote past, B present and C future. Our model is trained and evaluated for triples $AB \rightarrow C$ (that we will refer to as cases), as shown in Figure 4.1. An upper bound on performance is given by a regular probabilistic segmentation model that is trained and evaluated with tuples $C \rightarrow C$. This is essentially a model that has complete knowledge of the future and just needs to segment it. At the same time, a model trained on $B \rightarrow B$ but evaluated on $B \rightarrow C$ can serve as a lower bound to our model trained on $AB \rightarrow C$. If our performance matches that of the lower bound, we have learned to produce plausible segmentations for the current time step, but not the future.

2 https://github.com/MIC-DKFZ/batchgenerators

We split our subjects randomly into 5 groups and perform 5-fold cross validation, i.e. we train on 4 subsets and predict the remaining one. For many triples, the real change between time steps is small, which makes it hard to show that our approach actually learns meaningful change. As a consequence we define two groups to report results for:

1. **Large Change**: The 10% of cases with the most pronounced change in terms of whole tumor Dice overlap, resulting in a threshold of 0.48 and 13 cases for the published subset and a threshold of 0.42 and 55 cases for the full dataset.

2. **Moderate Change** The cases with larger than mean change (0.70 for subset, 0.68 for full dataset), but not in top 10%, resulting in 31 cases and 261 cases respectively.

We are not interested in predictive capabilities, so it makes little sense to look at the overlap of the prior mean predictions with the future ground truth (our approach performs not much better than the lower bound here). We report metrics that are representative of our model's desired capabilities, 1) a clinically relevant question, i.e. what the tumor will look like for a given expected size, and 2) how well the model is able to represent large changes in its latent space:

1. **Query Volume Dice**: We take samples from a grid around the prior mean (-3σ to $+3\sigma$ in steps of 1σ) and select the segmentation for which the whole tumor volume (i.e. all tumor classes contribute) best matches that of the ground truth. If our approach is able to model future appearances, it should perform better than the lower bound with increasing real change.

2. **Surprise**: This is the KL divergence the model assigns for a given combination of past & present scans and future ground truth. A lower KL divergence between prior and posterior means the model deems the combination more realistic, i.e. it is less *surprised*.

4.3 RESULTS

4.3.1 *Qualitative Results*

We first present several qualitative examples, selected to illustrate the types of changes our approach is able to represent.

Figure 4.2 a) shows three cases with outlines for the prior mean (solid purple) prediction as well as the sample from the prior (dotted purple) that best matches the volume of the real future (red). The similarity of the latter two in the first two columns indicates that our

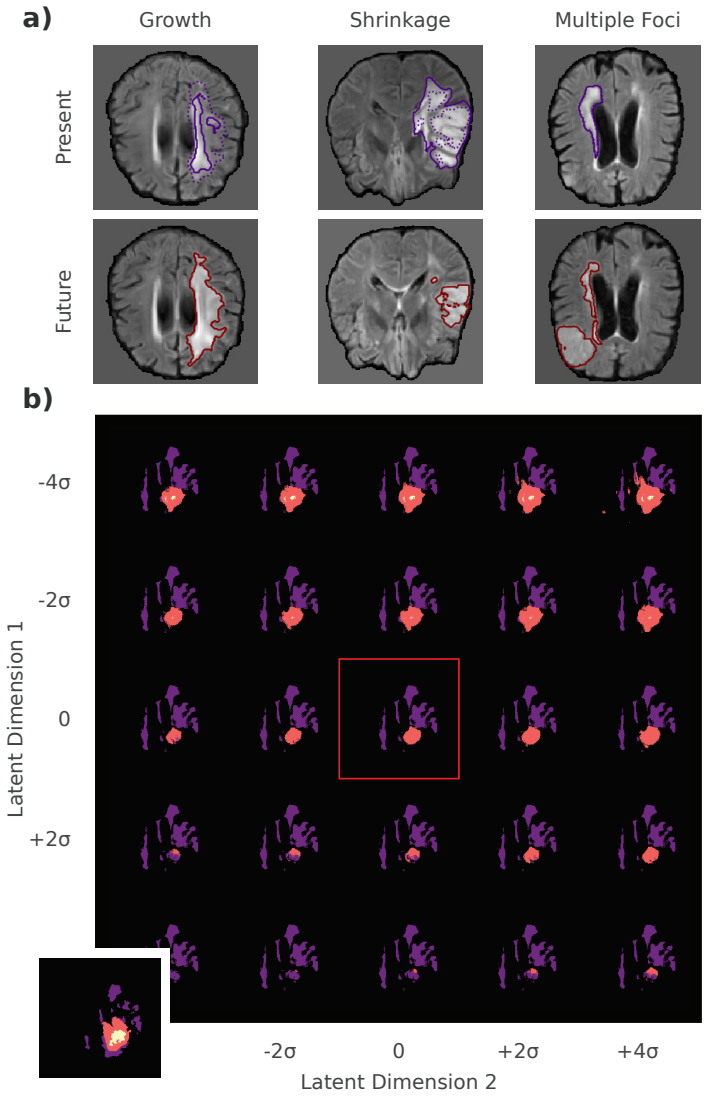

Figure 4.2: Qualitative Examples: (a) Prior mean prediction (solid purple) and sample with best volume match (dashed purple) as well as future ground truth (red) overlaid on FLAIR. The approach is able to model growth or shrinkage, but is unable to represent tumors with both growth and shrinkage in different locations (for multiple foci, dotted and solid overlap). (b) Regular grid samples from prior, with mean highlighted in red and ground truth inlay in bottom left corner (unrelated to (a)). The learned latent space separates class contributions, dimension 1 seems to encode tumor core size (enhancing tumor and necrosis) while dimension 2 encodes enhancing tumor size (note how necrosis is virtually constant in the top row). The third latent dimension, not shown here, captures small variations in edema size. Purple – Edema, Orange – Enhancing Tumor, Yellow – Necrosis

model is able to represent both strong growth and strong reduction in size well. It can also be seen that the mean prediction closely matches the current state of the tumor, which is unsurprising, because small changes occur most frequently. The third column is illustrative of a general limitation of our model: encoding into the latent space removes all spatial resolution, so tumors that both shrink and grow in different locations (e.g. with multiple foci) are not represented in the current setup.

Figure 4.2 b) illustrates how the learned latent space represents semantically meaningful continuous variations: Dimension 2 changes the size of just the enhancing tumor while dimension 1 changes the size of the tumor core (enhancing tumor and necrosis combined). The third axis that is not shown encodes variation in the size of the edema, meaning that the model automatically learned to separate the contributions of the different tumor regions. Most importantly, all variations seem plausible. Note that while a reduction in necrosis is biologically implausible in a treatment-naive context, it might very well occur under treatment like in our dataset.

4.3.2 *Quantitative Results*

In this section we compare our approach with an upper bound and a lower bound. These are given by a regular probabilistic U-Net Kohl et al. (2018) trained for segmentation with (upper bound) and without (lower bound) knowledge of the future and both evaluated with respect to future ground truth.

Figure 4.3 shows median results for two different metrics and both moderate change and large change, evaluated on the published subset of the data. *Query Volume Dice* represents the clinically motivated question of estimating spatial extent for a given change in size (e.g. for radiation therapy). Particularly for cases with large change our approach outperforms the lower bound. At the same time, the *Surprise*, a measure of how close estimated prior and posterior are for a given set of inputs and future ground truth, is on par with the upper bound for cases with moderate change and still much lower than the lower bound's for large change cases. For reference, in VAEs this usually comes at the cost of poor reconstruction, but the reconstruction loss (i.e. segmentation cross entropy, not shown) is also much lower for our approach compared to the lower bound in both cases. We performed a Wilcoxon signed-rank test to see if the difference between lower bound and our proposed method is significant. For the Query Volume Dice we find $p = 0.597$ in the case of moderate changes, but $p = 0.019$ for large changes, which can be considered significant. For the surprise we find $p < 0.001$ and $p = 0.221$, respectively.

In Figure 4.4, we show the same evaluation done on the full dataset available for this thesis. We now find that the difference between our

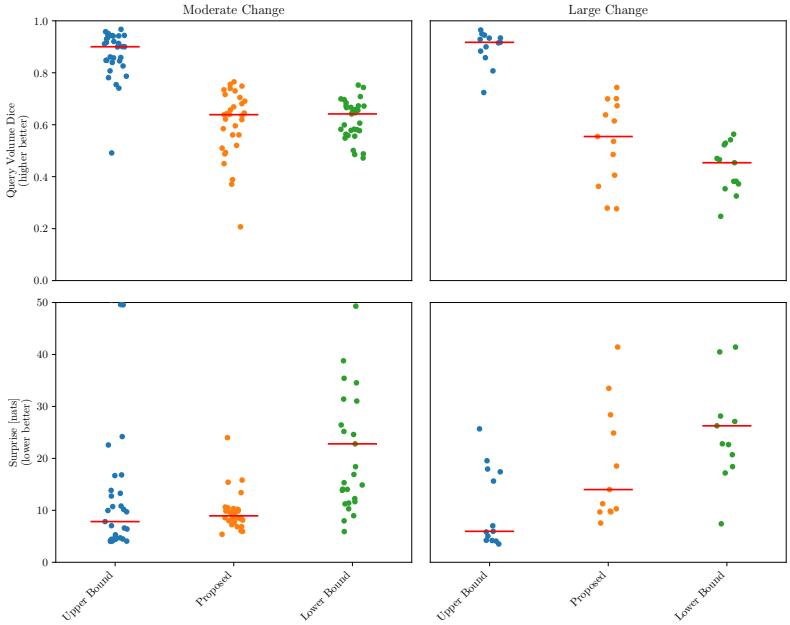

Figure 4.3: Quantitative results for *Query Volume Dice* and *Surprise* on the subset published in Petersen et al. (2019), for groups with moderate and large change and median indicated in red. For large changes, our approach can represent the future much better than the lower bound. The low surprise in our model indicates that our model's learned prior assigns higher likelihood than the lower bound to the real future tumor appearance, leveraging temporal information from previous scans. The difference between our proposed approach and the lower bound can be considered significant in the case of large changes, with $p = 0.019$ for the Query Volume Dice. For the surprise, the difference is significant for moderate changes with $p < 0.001$, but only $p = 0.221$ for large changes. We used a Wilcoxon signed-rank test for the statistical analysis.

proposed method is much less pronounced. For moderate changes, there is virtually no difference between the two, and the Wilcoxon signed-rank test results in $p = 0.873$ for the Query Volume Dice and $p = 0.176$ for the surprise. For large changes however, we still find a difference for the Query Volume Dice that can be considered significant, with $p = 0.019$, which is exactly the same value as above. For the surprise, the corresponding test resulted in $p = 0.073$, an even lower value than found above. We will discuss in the following section what we expect to be the reason for the reduced difference in median values between our proposed method and the lower bound.

4.4 DISCUSSION

In this work we investigated whether glioma growth dynamics can be learned directly from data without an underlying explicit biolog-

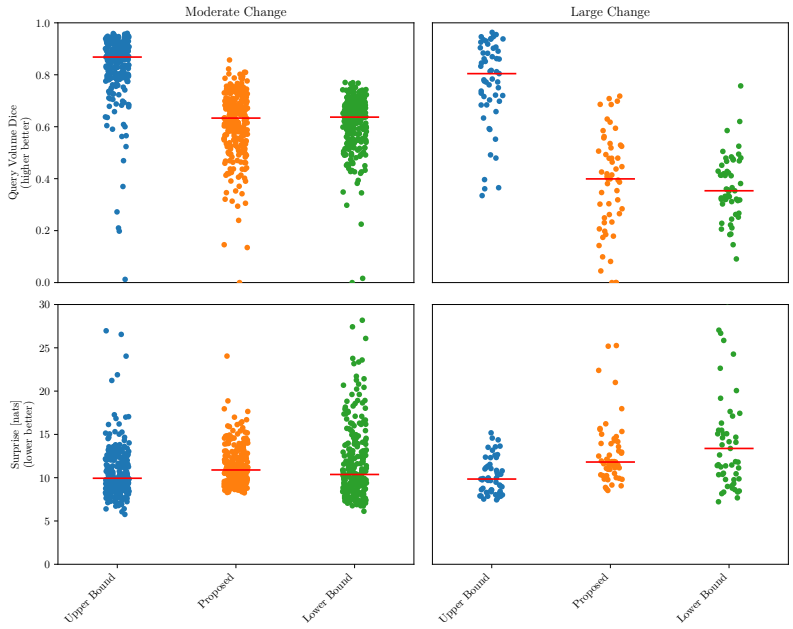

Figure 4.4: Quantitative results for *Query Volume Dice* and *Surprise* on the full dataset. Compared to the results in Figure 4.3, the difference between our proposed approach and the lower bound is much less pronounced, especially for the Query Volume Dice.

ical model, instead relying on probabilistic segmentation to model distributions of future tumor appearances.

Our results indicate that this is indeed possible. We showed quantitatively that our approach can represent large variations in the inferred distributions and that these learned distributions model growth trajectories instead of just segmentation variants for a known input. Qualitative examples show overall realistic growth as well as shrinkage patterns. Compared to existing work, our approach relies on a very different hypothesis, so we elected to present metrics that evaluate our desired goals, but are unfortunately unsuitable for quantitative comparison with classical methods. We also presented two different sets of results, the first being a reproduction of the results published in Petersen et al. (2019), the other an evaluation on a much larger dataset available to us at a later time. While the overall findings were the same, in terms of our method outperforming the lower bound, the difference between the two was much smaller for the evaluation on the larger dataset. This is most likely because the underlying data distribution the models were trained on changed. Specifically, in the subset evaluated for the publication, the *large change* category accounted for 10.6% of all cases. In the complete dataset, only 5.1% of cases fall into the same class. That means the model has seen fewer cases with a very pronounced change, and consequently the learned distribution models smaller changes on average.

In the results presented in Section 4.3, we always trained on two input time steps. We also evaluated the same with three or four input time steps, but found no significant difference. Likewise, when the models are trained with segmentations instead of the MRI scans as input, the results also stay the same. This suggests that there is no "hidden" information in the images, like a certain imaging signature that might indicate infiltration. Instead, the growth patterns are learned purely from the shape and position statistics in the dataset. This will be important for Chapter 8, where we translate the findings in this chapter to a continuous time axis and only work in segmentation space.

We see a number of advantages that our approach offers. The ability to sample consistent hypotheses from the latent space, as opposed to just having pixel-wise probability estimates, lends itself to answering clinically motivated questions, e.g. exploring only samples that correspond to strong growth or those that produce predictions where a certain region is or is not affected by the tumor. We further don't rely on imaging modalities like DTI that are not typically acquired in clinical routine. It would in fact be interesting to explore if our approach can benefit from including the latter. Overall, we think that our work opens up a promising new avenue of approaching glioma growth and tumor growth in general. Our work is entirely complementary with respect to diffusion-based models, and combining them should be exciting to explore, as we discuss further in Section 9.3.2.

At the same time, there are a number of disadvantages that need to be addressed. Our method clearly requires more data than existing ones that are based on explicit biological diffusion models. As we pointed out, our model is also unable (and not designed) to predict a single correct growth trajectory. It is further unable to resolve spatially varying growth for a single tumor, likely because we employ a simple global latent space. Perhaps the most important limitation is that our approach requires an equidistant spacing of observation, i.e. scans need to have been performed at fixed intervals. Neither can the models handle a varying number of input scans at test time. As the focus of this thesis lies on learning distributions of functions on *continuous* domains, both of these shortcomings will be addressed in the following chapters, specifically Chapter 8.

5

A MOTIVATING EXAMPLE: TUMOR VOLUME PREDICTION

We will use this chapter to introduce the overarching theme of this thesis. Our contributions pertain to the area of learning distributions of functions, or function spaces, on continuous domains, where *learning* means inferring some form of representation. We will present an example that should illustrate why this is desirable: the prediction of tumor burden in glioma patients, when we have already monitored their disease for some time. In the previous chapter we presented a proof of concept for modeling glioma growth using an approach that infers growth dynamics entirely from data. There we worked under the hypothesis that glioma growth cannot be predicted accurately and that we should rather try to model distributions of possible growth trajectories. We will see in this chapter that the assumption was justified.

We will begin with some basic analysis of the data we have at our disposal, to equip the reader with an understanding of its key features and the difficulties associated with predicting or modeling glioma growth using learning-based methods. This will be followed by the application of a few simple approaches that one might select in an attempt to tackle the problem. Finally, we will briefly introduce the methods most of the following chapters are based on and see how they fare in comparison. This chapter should not be understood as a research contribution in the same way as the other chapters. Rather, it establishes the research question we address using a concrete example, and provides context for the contributions we make in the following chapters. As a result, the presentation of results in this chapter also follows a somewhat unconventional format.

5.1 A GLIOMA GROWTH DATASET

We will use the modeling and prediction of glioma growth as an example where learning function spaces directly from data is desirable and could offer new insights. An introduction to these glial cell tumors is given in Section 2.2, where we also summarize related work that is concerned with modeling glioma growth mostly from biological principles instead of by learning from data.

At our disposal is a dataset that consists of MRI scans glioma patients typically receive in regular intervals for disease monitoring. The majority of the data stem from a study that compared treatment using a combination of chemotherapy (Lomustine) and an angiogenesis

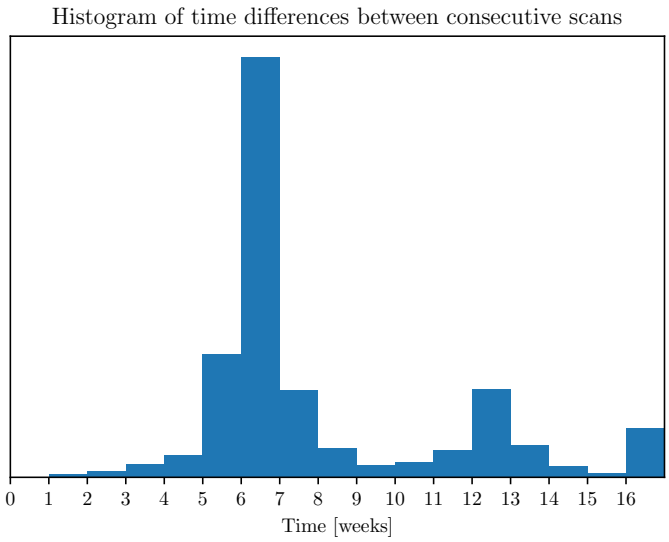

Figure 5.1: Distribution of time differences between consecutive scans in the dataset. We see the largest peak at 6-7 weeks and another at 12-13 weeks, i.e. twice that duration. This is because patients are usually asked to present to the hospital in regular intervals for progression monitoring. It also means that this is in fact not an ideal representation of a *continuous* domain.

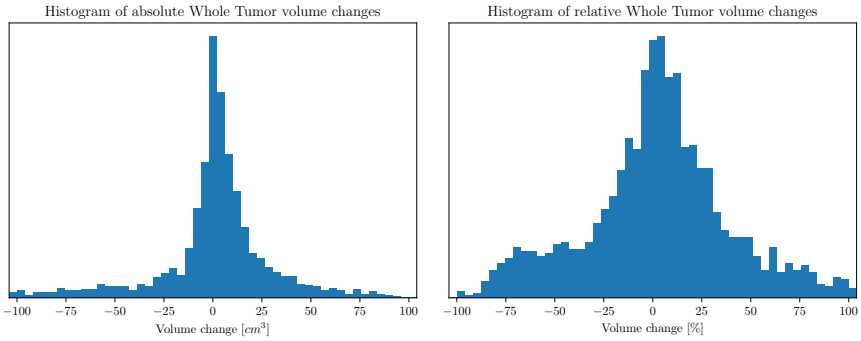

Figure 5.2: Changes in whole tumor size in the dataset that is being used in this work. The outermost bins collect all points outside of the given range. The distribution is almost symmetric around 0 and only slightly shifted to the positive region. This is due to the fact that patients in the dataset receive treatment, otherwise we would expect positive growth almost exclusively. The corresponding histograms for the individual tumor tissues are given in Figure A.2.

inhibitor (Bevacizumab) with treatment using chemotherapy alone (Wick et al., 2017). The study focused on patients with glioblastoma (meaning high grade glioma) after first recurrence of the tumor and found no significant difference in overall survival between the groups. The data comprises scans of 488 different patients from various treatment centers in Europe. An additional 40 patients with glioma of various grades were added to the dataset, whose data was acquired in clinical routine at Heidelberg University Hospital. Treatment information is generally not available in a structured format, so we do not incorporate it in our experiments for this thesis. Each patient was monitored for a minimum of two and a maximum of 13 consecutive times, and each time at least four different MRI contrasts were acquired: native T1 weighted, T1 weighted after administration of a contrast agent (Gadolinium), T2 weighted, fluid-attenuated inversion recovery (FLAIR, also T2-weighted). These contrasts represent the standard of care in glioma progression monitoring, and we describe them in more detail in Section 2.1. Other imaging was occasionally employed as well, like diffusion-weighted or susceptibility-weighted MRI, but too sporadically to be of use for our work. We conduct all our experiments on preprocessed versions of the data, where all contrasts and all scans for a given patient were registered to a common space, skull-stripped and resampled to an isotropic resolution of 1 mm. We further normalize each channel at each time step by its mean and standard deviation. Along with the MRI scans, segmentations of different tumor tissues are available, namely edema, contrast-enhancing tumor and necrosis and non-enhancing tumor (as a joint label). Many of our experiments and evaluations also consider the *whole tumor* region, which es the union of those individual classes. The segmentations were created with various combinations of automatic and semi-automatic methods, but all of them have undergone a final inspection by an experienced radiologist.

While patients should usually present for follow-up visits in regular intervals, this is not always the case in practice. In Figure 5.1 we show the histogram of time differences between consecutive hospital visits in the dataset. There are pronounced peaks at 6 weeks and 12 weeks, but a considerable number of times patients deviated from this pattern. So while this data could certainly be used in contexts that require equidistant inputs—like our first attempt at learned glioma growth modeling in Chapter 4—it already highlights that working on a continuous time domain is generally preferable.

5.2 THE DIFFICULTY OF PREDICTING TUMOR GROWTH

We mentioned in Section 2.2 that changes in tumor size are the primary factor deciding the classification into progression, stable desease or treatment response, and consequently about potential changes in

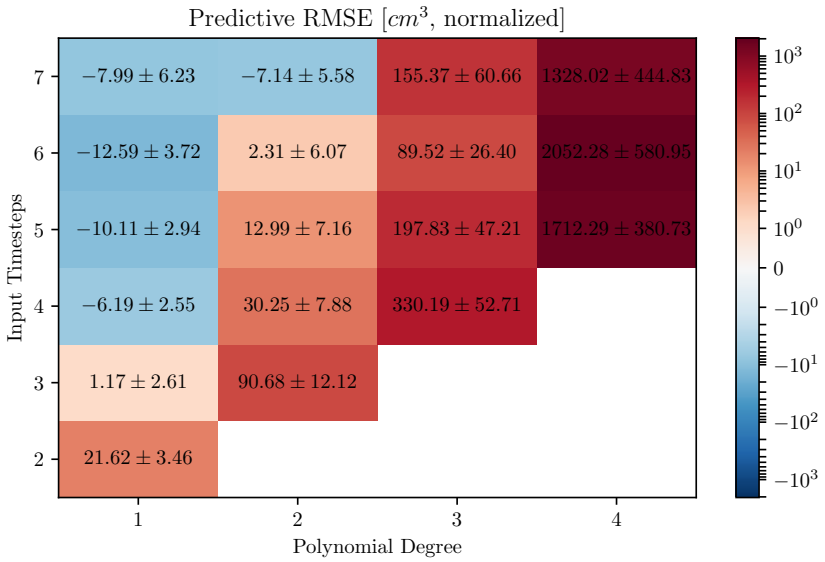

Figure 5.3: Predictive root-mean-square error (RMSE) for polynomial fits of different degrees and varying numbers of input time steps, evaluated for whole tumor volumes. Errors represent one standard deviation. We subtract the baseline of predicting no change at all, and find that performance decreases as the polynomial order increases. For a benefit compared to the baseline, at least 4 consecutive time steps are necessary to achieve an improvement.

treatment. There is no question that an accurate estimate of future tumor size would be tremendously helpful for therapy planning. As an exploratory exercise, we will try to predict tumor growth from volume measurements alone.

Consider the scenario that we monitored a patient for some time, collecting measurements of the tumor volume. In our dataset, we obtain this by simply summing the number of voxels belonging to the desired tumor class (whole tumor in this case) in the available segmentation map. We'd now like to estimate the tumor size at some point in the future, or rather the change compared to the last available observation. What should we expect a priori, when we have only the current size of the tumor at our disposal? Figure 5.2 shows the histograms of absolute and relative changes between consecutive scans in our datasets. Rather surprisingly, it is almost centered around zero, meaning the treatments patients receive appear to be working on average. For an untreated tumor, we would expect almost exclusively positive growth. Qualitatively, the distribution appears to be a Gaussian with a wider tail, but we are not interested in a precise description of the distribution, rather in its main takeaway: when trying to predict tumor growth at an increment of the average time difference in our data, we should not expect any change!

But now we have collected multiple measurements. Can we perhaps discover some sort of trend in them that we can extrapolate into the future? The simplest approach for this would linear regression: simply fitting a straight line to the available measurements. But of course higher order moments might be interesting as well, so we explored polynomials of varying degrees for this task. Figure 5.3 shows the root-mean-squared error for polynomials of degrees one to four, with the number of input time steps between two and seven. The values are normalized by the error we would obtain by not predicting any change. We find that a first order polynomial works best, and can improve upon the trivial baseline at least to some extent. The error we get for the baseline is $41.34\,\mathrm{cm}^3$, meaning the best error obtained for linear regression with 6 input time steps is still $28.75\,\mathrm{cm}^3$, which would be a sizable tumor in itself. There are of course many other possibilities beyond polynomials one could explore, as we outline in Section 2.2. One common choice is logistic growth, for which we show results in Figure 5.7, but for now we continue with linear regression.

It is quite clear that with such large errors it is paramount to have an estimate of uncertainty in the predictions. We can do this by performing the regression in a Bayesian framework, and because of the results in Figure 5.3 we will focus on linear regression. It can be shown that Bayesian linear regression is, under some mild assumptions[1], equivalent to applying a Gaussian Process with a dot-product kernel $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}\mathbf{x}' + \sigma^2$. As a comprehensive source on Gaussian Processes (GP) we recommend Rasmussen and Williams (2006), which includes a derivation of the dot-product kernel from linear regression. We also work a little more with Gaussian Processes in Chapter 7, where we provide further details. As the name suggests, GPs model the joint distribution of available observations and desired target locations as a Gaussian distribution. Examples for some predictions from a dot-product GP can be seen in Figure 5.4, including samples from the predicted distributions. As desired, we now have an uncertainty estimate, which increases moving away from the observations. We also see that the mean predictions are often far from the true observed tumor volume in the future. For example, rows one and four in the *Large Change* column essentially show almost constant tumor volume over the observed time range, but at the target location it has grown by an enormous amount. This highlights one of the challenges that make predicting tumor growth so difficult: the growth patterns are often highly irregular.

Applying a Gaussian Process, we're no longer just interested in the average error of the mean, i.e. the RMSE. Instead, we'd like to know how good the predicted distribution is, especially how good the uncertainty estimate is. To evaluate this, we can measure the

1 The assumptions are a standard normal prior on the slope and a zero-centered Gaussian prior with variance σ^2 on the offset.

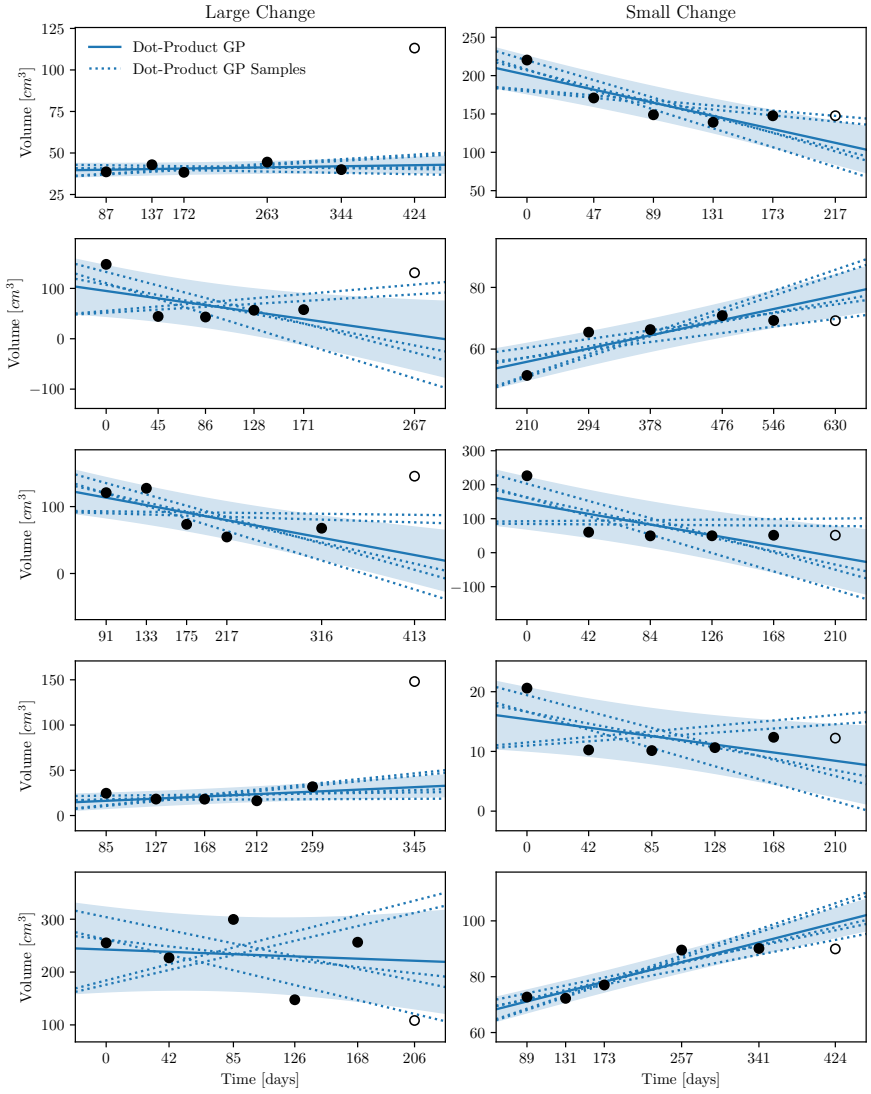

Figure 5.4: Examples from the dataset with predictions from a Gaussian Process with a dot-product kernel. (left) The 5 examples with the *largest* absolute change in tumor volume between the last input and the test point. (right) The 5 examples with the *smallest* absolute change in tumor volume between the last input and the test point. We find that the Gaussian Process can adjust its predictive uncertainty depending on the input data. At the same time, this estimate is often too low, as seen in the top rows. These examples highlight one of the challenges in predicting glioma growth, a sort of *growth explosion*. The inputs in the top row and the fourth row of the large change column look hardly distinguishable from the examples in the small change column, but at the target points the subjects suffer from a much larger tumor. Note that each panel has an individual y-axis and that the predictions from linear regression are almost identical, so we don't show them here.

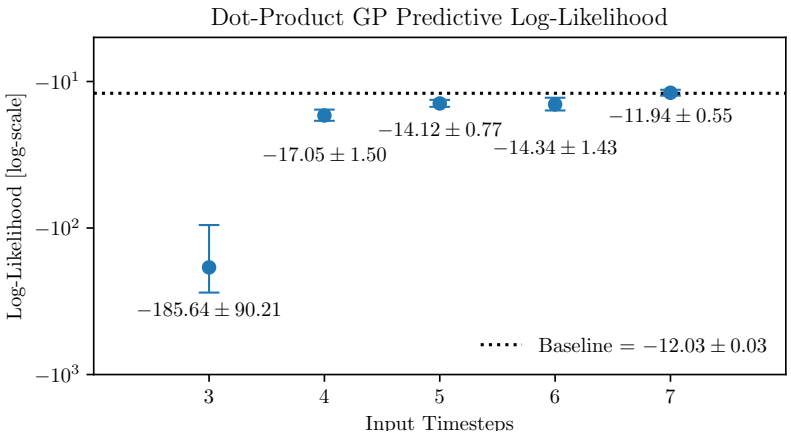

Figure 5.5: Predictive Log-Likelihood (LL) for a Gaussian Process with a dot-product kernel, evaluated for whole tumor volumes. The baseline corresponds to predicting a Gaussian with parameters estimated from the distribution of changes seen in Figure 5.2. With 5 or more input values, the prediction comes close to the baseline, but only matches it when 7 time steps are available. This is likely because the predicted uncertainty is too low, as seen for example in Figure 5.4. A value for only two inputs is not shown, because in that case the predicted Gaussian can become arbitrarily narrow.

(log-)likelihood of the target volumes under the predicted Gaussian distribution. The results are shown in Figure 5.5, again with a trivial baseline that we obtain by simply predicting a Gaussian distribution with mean and variance estimated from the histogram of changes in Figure 5.2. Even though a Gaussian is not an ideal fit for this histogram, the GP never performs better than the baseline, and only matches it with seven input time steps.

This is quite remarkable. On average, the GP extracts no information from the measurements we made and is no better than an informed guess. While glioma growth is certainly not deterministic, as we discuss in Section 2.2, the reader should hopefully be wondering: "Is a dot-product GP really the best choice for growth extrapolation?" And the simple answer is that we don't know, which leads us to the main motivation of our work.

In hopes to find a good set of functions to use for our task, we would almost always try to look at examples and over time try to describe or parametrize an expression that can later be applied to unseen examples. In other words, we try to find a common representation of functions that best describe what we have observed. This is precisely the goal of this thesis: we want to find methods that allow us to automatically learn such a representation from examples, so we don't have to do it manually.

5.3 LEARNED INTERPOLATION WITH NEURAL PROCESSES

The goal of this thesis is to find methods that can automatically learn a representation of some distribution of functions by observing examples. This should happen in a way that we can later interpolate between and extrapolate from a number of observations following the learned pattern. In precisely this context, Neural Processes (Garnelo et al., 2018a; Garnelo et al., 2018b) were proposed as a way to leverage neural networks—arguably the most powerful function approximators currently available—for the task. Our contributions in the following chapters are all based on Neural Processes or follow-up work, and in Chapter 6 specifically we investigate their inner workings in great detail. For now it shall suffice to know that these NPs are deep learning algorithms trained by repeatedly showing example functions, represented as sets of arbitrarily sampled *context* observations and *target* points. By learning to predict the targets from the context, they implicitly form a representation of the underlying function space and can later apply it to new context observations. Here we train the models in a leave-one-out fashion on five random subsets of the data, i.e. four subsets for training, one for testing.

In Figure 5.6 we show the same examples as in Figure 5.4, now applying a so-called *Conditional Neural Process* (Garnelo et al., 2018a), which is a deterministic Neural Process variant, as well as an *Attentive Neural Process* (Kim et al., 2019), which is follow-up work to the former. The CNP has essentially learned to perform linear regression, but as we will see in a moment, its uncertainty estimate is more accurate than that of dot-product GPs. The ANP also sees some curvature in its predictions, depending on the context observations.

In Figure 5.7, we show both the predictive RMSE and the predictive log-likelihood for the ANP as well as for a variational Neural Process (NP) (Garnelo et al., 2018b)—the results for the CNP are virtually identical to the ANP. For comparison, we also show the dot-product GP and the naive baseline in each case. Note that the RMSE for the dot-product GP is almost identical to the linear regression case. We see that the Neural Processes outperform the GP by a large margin, achieving a RMSE of as low as $(12.26 \pm 1.85)\,\mathrm{cm}^3$ and a log-likelihood of -2.20 ± 0.24. This answers our question of whether the dot-product GP was an ideal choice to extrapolate tumor volume measurements quite resoundingly. A learning-based approach, in the form of Neural Processes, evidently yields a large performance margin. Also shown is the RMSE for logistic regression, a common choice in growth modeling. While better than the dot-product GP, it is still bested by the learned estimates. Note that we can't obtain an exact solution for the posterior in Bayesian logistic regression, which is why we don't show the log-likelihood in this approach.

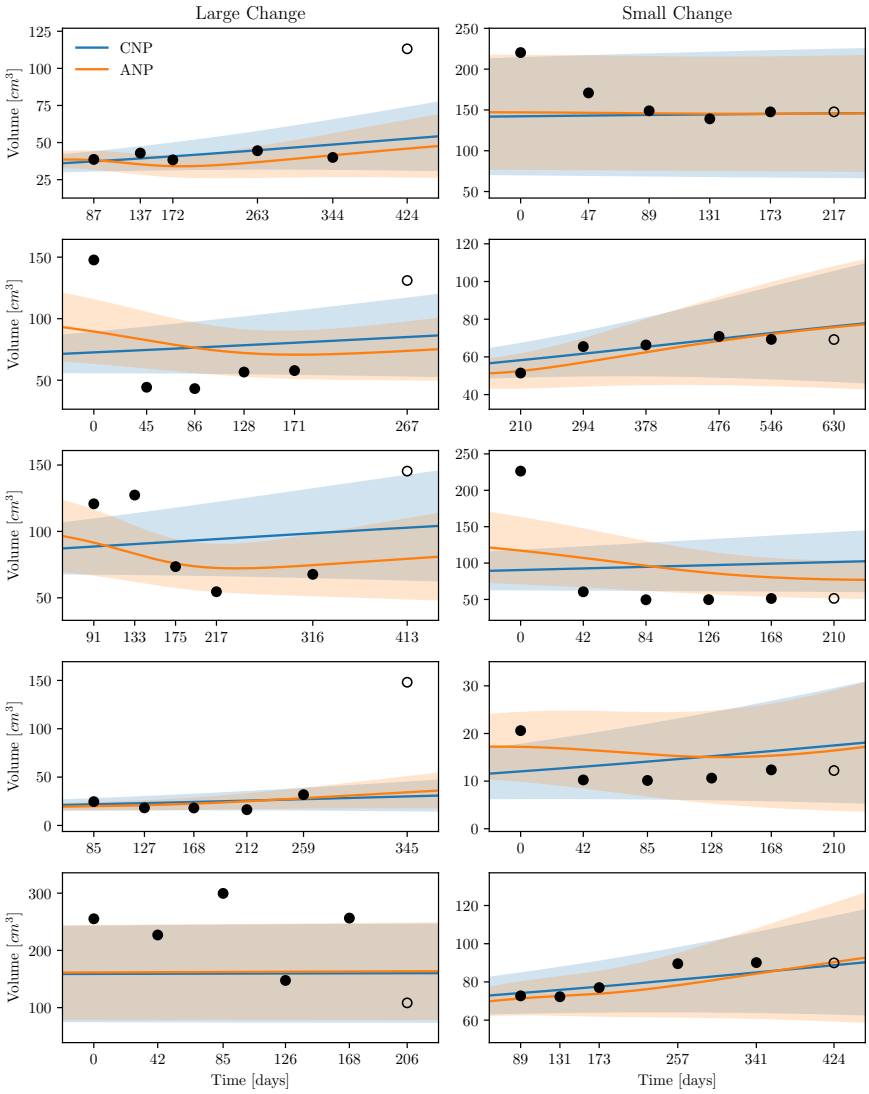

Figure 5.6: Examples from the dataset with predictions from a deterministic Neural Process (CNP) and an Attentive Neural Process (ANP). (left) The 5 examples with the *largest* absolute change in tumor volume between the last input and the test point. (right) The 5 examples with the *smallest* absolute change in tumor volume between the last input and the test point. We find that the CNP essentially learns to perform a form of linear regression, while the ANP predictions exhibit some curvature. Compared to the GP predictions in Figure 5.4, the uncertainty estimates seem more accurate. The models also learned that uncertainty usually increases over time.

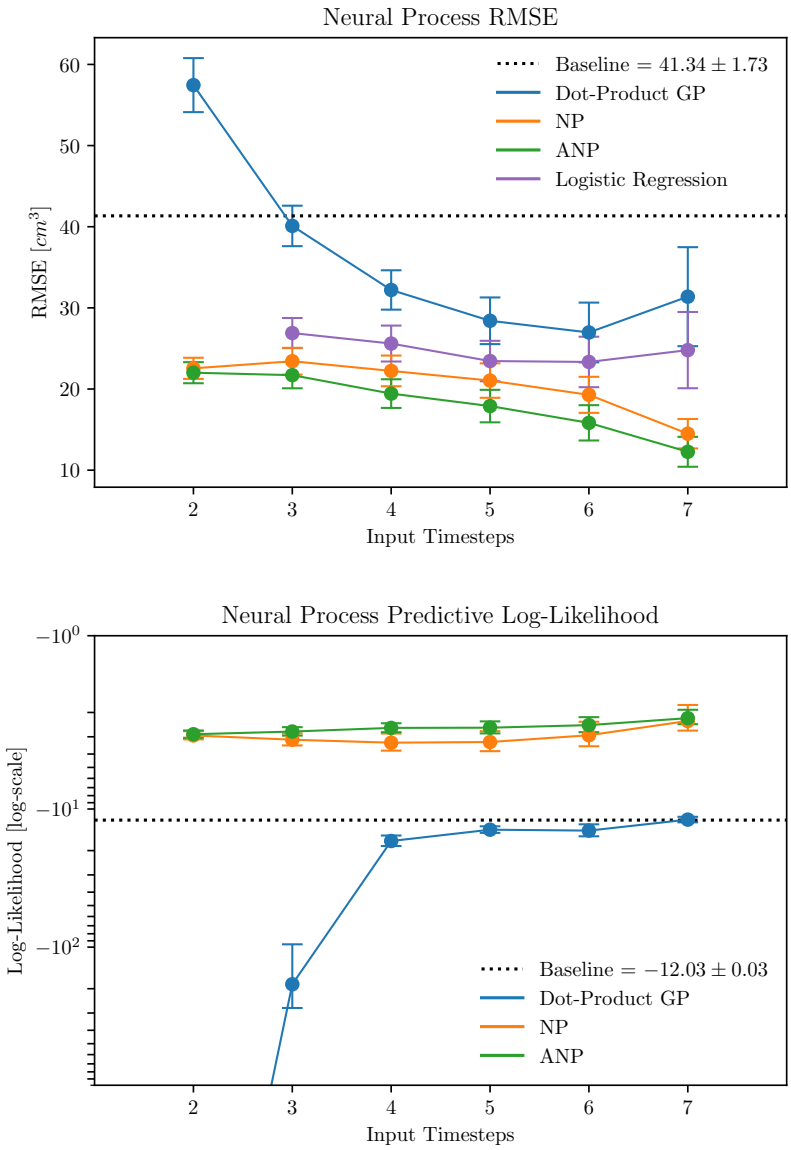

Figure 5.7: RMSE and Log-Likelihood for tumor volume regression with Neural Processes. We show the probabilistic variant (NP) instead of the deterministic variant (CNP), because the performance of CNP and ANP is almost identical. In both metrics, the Neural Processes perform significantly better than the Dot-Product GP. Note that the RMSE for linear regression and the for the dot-product GP are virtually identical. Clearly there is a benefit to using a learned function distribution instead of a manually specified one.

5.4 SUMMARY

Our goal in this chapter was to highlight the need for methods that can learn representations of function distributions purely from data, so as to alleviate the need to manually specify them, which usually requires some expertise in the domain. Using the example of glioma growth prediction (from previously collected volume measurements), we first showcased the overall difficulty of the task, as a reasonably chosen approach performed worse than a trivial baseline when uncertainty estimates were desired. We then introduced Neural Processes as a learning-based alternative and saw that they vastly outperformed both the trivial baseline and our manual best guess in all metrics. At the same time, it should be noted that the predictive error for Neural Processes is still way too high to be considered useful for practical purposes. We assume that the stochasticity associated with glioma growth prohibits more accurate predictions, at least when treatment information is not available.

Note how this echoes the setting in the previous chapter, where we worked under the hypothesis that accurate prediction of glioma growth (in image space as opposed to just volumes) is not possible to a useful degree. As a consequence, one should instead focus on modeling multiple growth trajectories. Similar to this chapter, we presented an approach that learns spatial growth dynamics purely from data. However, this was limited to a fixed number of inputs, sampled at fixed intervals. The Neural Processes introduced here work on a continuous time domain.

Encouraged by the performance of NPs in Section 5.3 we base the contributions in the next chapters on this approach. In the following chapter, we will investigate how Neural Processes form representations of function spaces and what that entails for the properties of functions they can represent. In Chapter 7 we improve upon a newer Neural Process variant by combining it with a Gaussian Process. In Chapter 8 we return to the problem of glioma growth modeling and essentially combine our attempts in the previous chapter with the Neural Process framework to achieve learned spatial growth modeling with continuous time inputs.

6

FREQUENCY DECOMPOSITION IN NEURAL PROCESSES

We have established in the previous chapter that Neural Processes (Garnelo et al., 2018a; Garnelo et al., 2018b) are a promising approach for learning distributions over functions purely from observations. They alleviate the need to manually specify or parametrize them, a task that often requires a prohibitive amount of prior knowledge or an oversimplification of the underlying problem. Our goal in this chapter is to understand how Neural Processes represent the functions they learn, and what implications these representations might have for the types of functions that can be learned. We begin with a detailed description of the Neural Process framework, so as to enable the reader to better follow the analyses conducted afterwards. Therein, we will derive a theoretical upper bound on the frequencies in signals that can be represented, which we subsequently validate empirically. We will find that the finite-dimensional representations learned by Neural Processes will always decompose the input space in a way that different dimensions are used to represent different parts of the input space. In variational Neural Processes (NP) (Garnelo et al., 2018b), this usually happens as a spatial partitioning, while deterministic Neural Processes (CNP for *Conditional Neural Processes*) (Garnelo et al., 2018a) perform a decomposition such that different dimensions represent different frequencies, not unlike a Fourier transform. As a consequence, Neural Processes can act like a low-pass filter when their representation size is insufficient to represent higher signal frequencies, in agreement with the derived bounds. On top of that, we find that Neural Processes with a sufficiently large representation size can be trained to only represent certain frequencies. They essentially become *programmable band-pass or band-stop filters*. The work in this chapter is currently under review:

Petersen, Jens, Paul Jäger, Gregor Köhler, David Zimmerer, Fabian Isensee, and Klaus H. Maier-Hein (2020a). "Frequency Decomposition in Neural Processes". In: *International Conference on Learning Representations (under review)*.

6.1 INTRODUCTION

Our goal is to learn a distribution over functions, or more generally to represent a function space $\mathcal{F} = \{f_i\}, f_i : X \to Y$. Assume that we are given some observations $C = \{(x_c, y_c)\}_{c=1}^{N} =: (\mathbf{x_c}, \mathbf{y_c})$, often called the *context*, and we would like to know the value y_t at some new location

x_t. This is what we did in the previous chapter, where we tried to interpolate and extrapolate tumor volume measurements over time by fitting a polynomial or a Gaussian Process to the context. These are two examples of commonly used distributions, but how can we be sure that the data can be approximated sufficiently by them? Ideally, we would first observe a number of context sets and corresponding target sets $T = \{(x_t, y_t)\}_{t=1}^{M} =: (\mathbf{x_t}, \mathbf{y_t})$, try to infer what the functions $f_i(x) = y$ generally "look like" and then use that information to make a prediction at test time. This is precisely the idea of Neural Processes (Garnelo et al., 2018a; Garnelo et al., 2018b).

6.2 METHODS

6.2.1 *The Neural Process Framework*

Neural Processes are maps $P : C, X \to Y$, where C is a set of tuples $\{(x, f(x))\}$ with arbitrary but positive cardinality, and $f \in \mathcal{F} : X \to Y$. Note that we introduced boldface above as a shorthand for these collections, but it is important to remember that in our context this refers to a *set*, not a *vector*. In this chapter we restrict ourselves to $X = Y = \mathbb{R}$, because it allows us to visualize learned representations. We further only look at the original NPs, namely the deterministic Conditional Neural Processes (CNP) (Garnelo et al., 2018a) and the variational Neural Processes (NP) (Garnelo et al., 2018b), because newer contributions in the field work in ways that preclude them from being analyzed in the same way. We discuss this further in Section 6.3. Note that we use the name Neural Process as well as the abbreviation NP for both the general family of methods and the variational implementation; the distinction will be made explicit where necessary. In CNPs and NPs, the map P is separated into two parts, a so called *encoding* $E : C \to Z$ and a *decoding* or *generating* part $G : Z, X \to Y$. Z is referred to as the *representation* or *latent space*. To allow Neural Processes to approximate arbitrary function spaces $\mathcal{F}$, E and G are powerful learnable approximators, specifically neural networks, as the name suggests.

As stated above, E and by extension the complete Neural Process P act on *set-valued* inputs. This is contrary to the vast majority of machine learning work where inputs are vectors of fixed dimension and ordering. Recall that sets are permutation-invariant, so we must ensure that the same is true for the output of E. Zaheer et al. (2017) show that E is permutation-invariant if and only if it has a *sum-decomposition*, i.e. it can be represented in the form

$$E(\mathbf{x}, \mathbf{y}) = \rho\left(\sum_{i=1}^{N} \phi(x_i, y_i)\right) \tag{6.1}$$

where ρ, ϕ are appropriately chosen functions. Wagstaff et al. (2019) further show that to be able to represent all continuous permutation-invariant functions on sets with a cardinality of at most N, the dimension of the image Z must at least be N.

What these works don't consider are the implications for situations where the elements of the sets are input-output tuples of some function f, like in our case. Switching to signal-processing terminology, we know that to represent—meaning to be able to fully reconstruct—a continuous signal, we need $N > 2|b-a|f_{max}$ samples, where f_{max} is the maximum frequency component of the signal defined on the interval $[a, b]$. This is most commonly known as the Nyquist-Shannon sampling theorem (Whittaker, 1915; Kotelnikov, 1933; Shannon, 1949). Conversely, when the representation has dimension D_r, the maximum cardinality of sets we can represent is also D_r, meaning the maximum frequency the signal can have is $f_{max} < \frac{D_r}{2|b-a|}$. In practice, this might be even smaller, because the Nyquist-Shannon theorem assumes equidistant sampling, while we sample points randomly, as we discuss later. We will investigate empirically how the reconstruction quality in Neural Processes changes when the signals contain increasingly high frequencies.

6.2.2 *Optimization*

Assume we are given a function space $\mathcal{F} = \{f\}$, with each f represented by random samples, which we partition into context set $(\mathbf{x_c}, \mathbf{y_c})$ and target set $(\mathbf{x_t}, \mathbf{y_t})$. We further have encoder E and decoder G of a Neural Process implemented as neural networks, for which we summarize the parameters in θ. In our implementation, both are multilayer perceptrons (MLP), meaning simple fully connected networks. Our goal is then to find the optimal set of parameters θ^* that maximizes the likelihood of $\mathbf{y_t}$, given $\mathbf{x_c}$, $\mathbf{y_c}$ and $\mathbf{x_t}$, over all f:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{f \in \mathcal{F}} \log p_\theta(\mathbf{y_t}|\mathbf{x_t}, \mathbf{x_c}, \mathbf{y_c}) \tag{6.2}$$

where p_θ is a placeholder for some parametrized likelihood function. We introduce the logarithm because we assume the likelihood factorizes across individual f, turning the expression into a sum. So what would this optimization look like in practice? For example, we could minimize the mean squared error between $\mathbf{y_t}$ and the predictions $\hat{\mathbf{y}}_\mathbf{t}$ from our network. This implicitly assumes a Gaussian likelihood with a fixed variance[1]. However, we would like our model to predict a variance, so that it can indicate how uncertain it is about a prediction. We achieve this by implementing G as a network that predicts both

1 The log-likelihood of $\mathbf{y_t}$ under a diagonal Gaussian with mean $\hat{\mathbf{y}}_\mathbf{t}$ is simply $\|\mathbf{y_t} - \hat{\mathbf{y}}_\mathbf{t}\|_2^2$ times some constant.

the mean and the variance of a diagonal Gaussian distribution, and Equation (6.2) becomes:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{f \in \mathcal{F}} \sum_{t} \log \mathcal{N}(y_t; G_\theta^\mu(Z, x_t), G_\theta^\sigma(Z, x_t)) \tag{6.3}$$

In deterministic Neural Processes (CNP), we can directly optimize this with maximum likelihood training. In variational Neural Processes (NP), Z is also parametrized by a Gaussian, meaning just like G, E predicts mean and variance of a Gaussian with D_r dimensions. In this case, we need to rewrite the summands of Equation (6.2):

$$\log p_\theta(\mathbf{y_t}|\mathbf{x_t}, \mathbf{x_c}, \mathbf{y_c}) = \log \mathop{\mathbb{E}}_{z \sim p(Z|\mathbf{x_c}, \mathbf{y_c})} p_\theta(\mathbf{y_t}|\mathbf{x_t}, z) \tag{6.4}$$

Here, $p(Z|\mathbf{x_c}, \mathbf{y_c})$ is *not* the distribution predicted by our encoder, but some *true distribution* we don't have access to. The idea of *variational inference* (see for example Bishop (2006) for an introduction) is to approximate this p by some other distribution q_θ and then to optimize p_θ and q_θ simultaneously. q_θ is what our encoder E predicts, just like p_θ is what our decoder G predicts. Continuing from Equation (6.4):

$$\begin{aligned}
\text{LHS} &= \log \mathop{\mathbb{E}}_{z \sim q_\theta(Z|\mathbf{x_t}, \mathbf{y_t})} p_\theta(\mathbf{y_t}|\mathbf{x_t}, z) \frac{p(z|\mathbf{x_c}, \mathbf{y_c})}{q_\theta(z|\mathbf{x_t}, \mathbf{y_t})} && (6.5)\\
&\geqslant \mathop{\mathbb{E}}_{z \sim q_\theta(Z|\mathbf{x_t}, \mathbf{y_t})} \log \left(p_\theta(\mathbf{y_t}|\mathbf{x_t}, z) \frac{p(z|\mathbf{x_c}, \mathbf{y_c})}{q_\theta(z|\mathbf{x_t}, \mathbf{y_t})} \right) && (6.6)\\
&\approx \mathop{\mathbb{E}}_{z \sim q_\theta(Z|\mathbf{x_t}, \mathbf{y_t})} \log \left(p_\theta(\mathbf{y_t}|\mathbf{x_t}, z) \frac{q_\theta(z|\mathbf{x_c}, \mathbf{y_c})}{q_\theta(z|\mathbf{x_t}, \mathbf{y_t})} \right) && (6.7)\\
&= \mathop{\mathbb{E}}_{z \sim q_\theta(Z|\mathbf{x_t}, \mathbf{y_t})} \log p_\theta(\mathbf{y_t}|\mathbf{x_t}, z) \\
&\quad - D_{KL}(q_\theta(z|\mathbf{x_t}, \mathbf{y_t}) || q_\theta(z|\mathbf{x_c}, \mathbf{y_c})) && (6.8)
\end{aligned}$$

where LHS refers to the left hand side of Equation (6.4). In the first line, we have switched the underlying distribution from the true *prior*—meaning conditioned on the context—to an approximate *posterior*—meaning conditioned on both context and target, but for notational simplicity we only write out the target set. The second line follows from Jensen's inequality while in the third line we have replaced the true prior with the approximate prior. Finally, we have rewritten the right hand side using the Kullback-Leibler (KL) divergence, a measure of distance between two distributions. Because we predict Gaussian distributions, the KL divergence has a closed-form expression. Otherwise it would be impractical to use it in an optimization context. The last line is often called the *evidence lower bound* (ELBO) in variational inference. One might ask why we replace the true prior with an approximate posterior, and not for example an approximate prior. It can in fact be shown that maximizing Equation (6.8)

also minimizes the KL divergence between the *approximate posterior* and the *true posterior* (see e.g. Blei et al. (2017)).

6.2.3 *Implementation*

Let us now put the above in more practical terms. When presented with an example consisting of context and target, we first use the encoder network E to encode each context tuple from the context separately. The encoder is a MLP with two input channels (for X and Y), 6 hidden layers with 128 channels, and a final layer mapping to D_r channels, i.e. to the representation. Between the layers we use tanh-nonlinearities as activation functions. While it could be interesting to explore other architectures or maybe the influence of having fewer or more layers, we adopt the implementation from the original works. A number of configurations were also evaluated in Le et al. (2018b), and our setup corresponds what the authors found to be the best-performing configuration. For the variational case, the final layer maps to $2D_r$ channels, half for the mean and half for the variance of the predicted Gaussian (in practice, we predict the log-variance to allow negative values). The individual representations are then averaged, and in the variational case we call this the prior ($q_\theta(z|\mathbf{x_c}, \mathbf{y_c})$ in Equation (6.8)). For the posterior, we also encode the target pairs and then average over all individual representations, including the context. During training forward passes, we sample once from the posterior and use this sample as the representation for the decoder. Ideally, we should sample many times to integrate the expectation in Equation (6.8), but for stochastic mini-batch training it was found empirically that a single sample suffices (Jimenez Rezende et al., 2014; Kingma and Welling, 2014), which greatly accelerates training. At test time, we sample 100 times from the prior. Note that in general sampling is not a differentiable operation, but for suitable distributions one can employ the so-called *reparametrization trick*: if $z \sim \mathcal{N}(\mu; \sigma^2)$ then $z = \mu + \sigma\epsilon$ with $\epsilon \sim \mathcal{N}(0; 1)$. This way a gradient of z with respect to μ and σ exists. The decoder predicts a Gaussian from the representation and an input x_t. It is implemented symmetrically to the encoder, meaning it's a MLP with $D_r + 1$ input channels, 6 hidden layers with 128 channels, and two output channels for mean and (log-)variance. We use tanh-activations as well. As a loss we directly use the negative log-likelihood, meaning we evaluate the likelihood of a reference point y_t under a Gaussian parametrized by the predicted mean and variance. Finally, we average over all predicted points, which are the target points as well as the context points.

6.3 RELATED WORK

While standard deep learning is mostly concerned with learning single deterministic functions, the learning of distributions of functions has also been addressed in a variety of ways. *Bayesian Neural Networks* (Neal, 1996; Graves, 2011; Hernández-Lobato and Adams, 2015) try to do this by placing distributions on the weights of a network. Because of the usually enormous number of parameters in current deep learning architectures, estimating posteriors for the weights becomes tremendously difficult. Similarly, it is not straightforward to condition the weights on observations given at test time, like in our case. As a result, these works focus more on improving estimates of the predictive uncertainty. It has also been argued that Dropout effectively functions like a Bayesian neural network with Bernoulli distributions over the weights (Gal and Ghahramani, 2016a; Gal and Ghahramani, 2016b). Likewise, Kingma et al. (2015) argue that Gaussian multiplicative noise (called Gaussian dropout) can be used to approximate a posterior over the weights, but this has since been debated (Hron et al., 2017). The most well-known group of non-deep learning methods that address the problem of conditioning predictions on observations given at test time are Gaussian Processes (Rasmussen and Williams, 2006). We work with them in Chapter 7, so we refer to Section 7.5 for related work in that context. In Chapter 5 we saw that the choice of kernel for a GP requires a lot of prior knowledge, which is precisely the problem we seek to alleviate with Neural Processes.

Even though Neural Processes have only been proposed less than three years ago, there have been several follow-up works. Perhaps the most well known addition are Attentive Neural Processes (ANP) (Kim et al., 2019), which we briefly introduced in Chapter 5 and will revisit in Chapter 7. The basic idea is to no longer require the individual representations to be averaged, which forces a given function to be represented as a single point in a D_r-dimensional space. The averaging is replaced by a learned attention mechanism (Vaswani et al., 2017) that summarizes the representations conditioned on the target point y_t. Unsurprisingly, the removal of the bottleneck improves both reconstruction and prediction performance, but it also removes the need to form a global (i.e. independent of target values) representation of the function space. As a result, the analyses in this chapter can't be performed for ANPs. Another recent addition to the field are Convolutional Conditional Neural Processes (ConvCNP) (Gordon et al., 2020), which we also discuss and improve upon in Chapter 7. The idea of the authors is to not map context points to a finite-dimensional space but instead to a function space. A neural network then operates in this space, and predictions are performed by mapping back to the output space. In practice this means that a kernel interpolation is performed with the context points. This is continuous in principle,

but evaluated on a grid, so a CNN can be applied. The output of the CNN is again convolved with a kernel to produce predictions at the desired locations. A visualization of NP, ANP and ConvCNP is shown in Figure 7.1. Similar to ANP, Louizos et al. (2019) propose to not merge observations into a global latent space, but instead learn conditional relationships between them. Singh et al. (2019) and Willi et al. (2019) address the problem of overlapping and changing dynamics in time series data. Generative Query Networks (GQN) (Eslami et al., 2018; Kumar et al., 2018; Rosenbaum et al., 2018) are essentially Neural Processes where observations are not scalar but entire images. Employing vastly more powerful encoders and decoders, they can (re-)construct unseen views in 3D scenes. GQNs are some of the earlier works in the field of 3D scene understanding, an area that has received a lot of attention more recently, e.g. from Sitzmann et al. (2019) and Engelcke et al. (2020).

As we have outlined in Section 6.1, Neural Processes can also be interpreted from the perspective of deep learning on sets, the earliest work in the field being Zaheer et al. (2017). More theoretical contributions were made by Wagstaff et al. (2019), whose work we use to underpin our hypothesis that the representation size in Neural Processes limits the maximum frequency of signals that can be represented. More applied work in the set-learning context has been performed on point-cloud data (Qi et al., 2017b; Qi et al., 2017a; Wu et al., 2019), which can be interpreted much in the same way as our problem, with data represented as (location, value) pairs.

Our findings in this chapter show that Neural Processes sometimes learn to automatically perform a decomposition of signals into different frequency components. It is well known that neural networks, specifically a MLP with at least one hidden layer, can learn the Fourier transform of an input signal (Gallant and White, 1988), which follows directly from the universal approximation theorem (Hornik et al., 1989; Cybenko, 1989). In fact, there have been a multitude of works that exploit this ability in one way or the other, leading to the term *Fourier Neural Networks*. We refer to the recent review by Zhumekenov et al. (2019) for a comprehensive overview. The difference to Neural Processes is that these networks typically learn a single mapping, while NPs represent a function space. Furthermore, NPs are implemented such that the learned mapping is only applied to individual (x,y) pairs, and their representations are summed. Fourier networks typically feed the full signal sequence into a network. Finally, we'd like to point out that our goal is not to show that NPs, or any deep learning approach for that matter, can in principle learn Fourier transforms or similar frequency decompositions. The key takeway of our work is that this happens *automatically*, without any supervisory signal forcing the networks to do so.

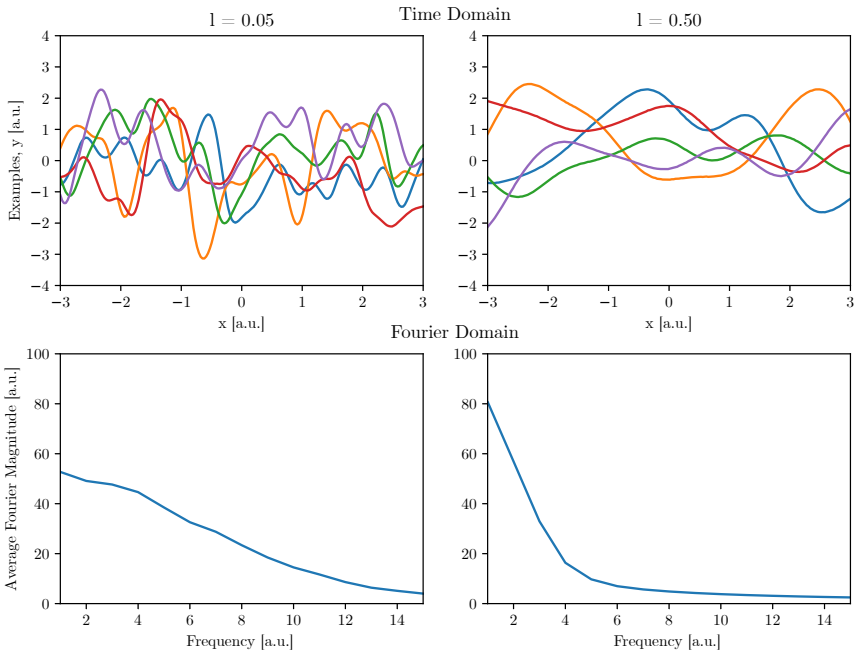

Figure 6.1: Examples from a Gaussian Process prior with an EQ kernel with two different lengthscales, along with the corresponding average magnitude of frequency components.

6.4 EXPERIMENTS & RESULTS

6.4.1 *Does Representation Size Limit Frequency Content?*

Conditional Neural Processes (CNP) (Garnelo et al., 2018a) and variational Neural Processes (NP) (Garnelo et al., 2018b) learn representations of function spaces by mapping pairs of input and output values to a representation space and averaging the individual representations. In doing so, they form global finite-dimensional representations of continuous functions. Our goal is to understand how these representations work and what implications they might have on the types of functions that can be learned or how well they can be learned. CNPs and NPs are the only members of the Neural Process family that form finite-dimensional global representations, and our experiments are not suited for other well-known members introduced in Section 6.3, like ANPs or ConvCNPs. In our experiments, we mostly work with samples from a Gaussian Process prior with an EQ (exponentiated-quadratic) kernel with varying lengthscales l. A few examples can be seen in Figure 6.1, the kernel is given by:

$$k(x_1, x_2) = \exp\left(\frac{\|x_1 - x_2\|_2^2}{2l}\right) \tag{6.9}$$

Also shown in Figure 6.1 is the average magnitude of the frequency components, which decays smoothly to zero. For later experiments, we

require more control over the frequency components, so we conduct those experiments with Fourier series:

$$f(x) = a_0 + \sum_{k=1}^{K} a_k \cos(kx - \phi_k) \,, \quad K = 19 \tag{6.10}$$

Note that in this formulation k is an angular frequency, and in our figures we show Fourier components as angular frequencies as well. The bound we derived in Section 6.1, on the other hand, refers to ordinary frequencies. ϕ_k and a_k, including a_0, are sampled uniformly from $[-1, 1]$.

When we generate a random function, we construct random context and target sets from it as follows: we first sample the number of context points N from $[3, 100)$ and the number of target points M from $[N, 100)$ to ensure it's at least as big as the context. For each point we then sample a random x uniformly from $[-3, 3]$ and evaluate the function at x. As outlined in Section 6.1, we let the networks predict values at both the target and the context locations. This procedure is equivalent to what was done in the original works.

We derived in Section 6.1 an upper bound on the maximum frequency that a signal given on some interval $[a, b]$ may contain so that it can still be represented in a Neural Process. This derivation is based on the Nyquist-Shannon theorem and the work presented by Wagstaff et al. (2019):

$$f_{max} < \frac{D_r}{2|b - a|} \tag{6.11}$$

This limit assumes equidistant sampling, while our input values are sampled uniformly. We further have at most 99 context points, which might pose a similar limit to f_{max}., because the signal must be fully defined by those points. To be more precise, it is unclear whether just the number of context points or the sum of context and target points results in another bound. Our goal in this section is to validate the above empirically.

Figure 6.2 shows the average reconstruction error for CNPs and NPs with different representation sizes on data originating from Gaussian Processes with EQ kernels of varying lengthscale. Evidently, a decreasing representation size results in poorer reconstructions for a given kernel lengthscale, and so does a decreasing kernel lengthscale for a given representation size. Interestingly, around a representation size of 96, there seems to be no more improvement for each l, but an increasing l allows for better reconstructions. Overall, reconstructions for the NP are a bit worse, which is unsurprising, as the variational formulation always introduces some smoothing. This is in fact a problem that still garners attention in the research community, see for example

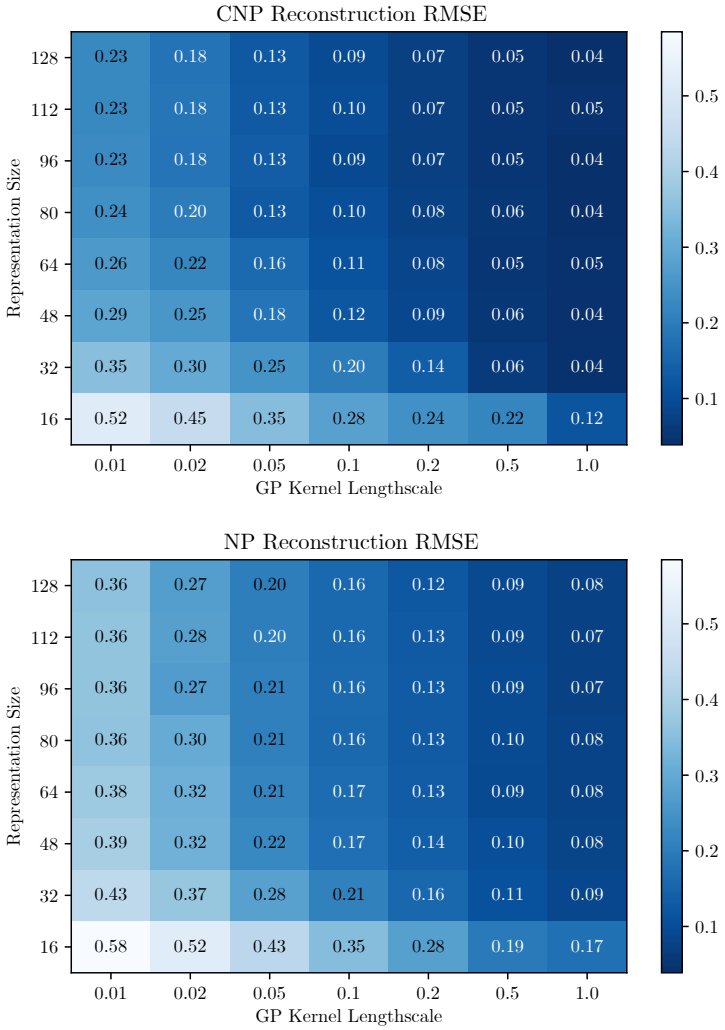

Figure 6.2: Reconstruction performance (RMSE) of CNPs (top) and NPs (bottom) on data generated from a Gaussian Process with a Gaussian (EQ) kernel. We vary the kernel lengthscale—a smaller lengthscale means higher frequency content—and the representation size in the Neural Processes. Evidently, a larger representation allows the models to better represent data with smaller lengthscale. Overall, the RMSE in CNPs is a little lower than in NPs.

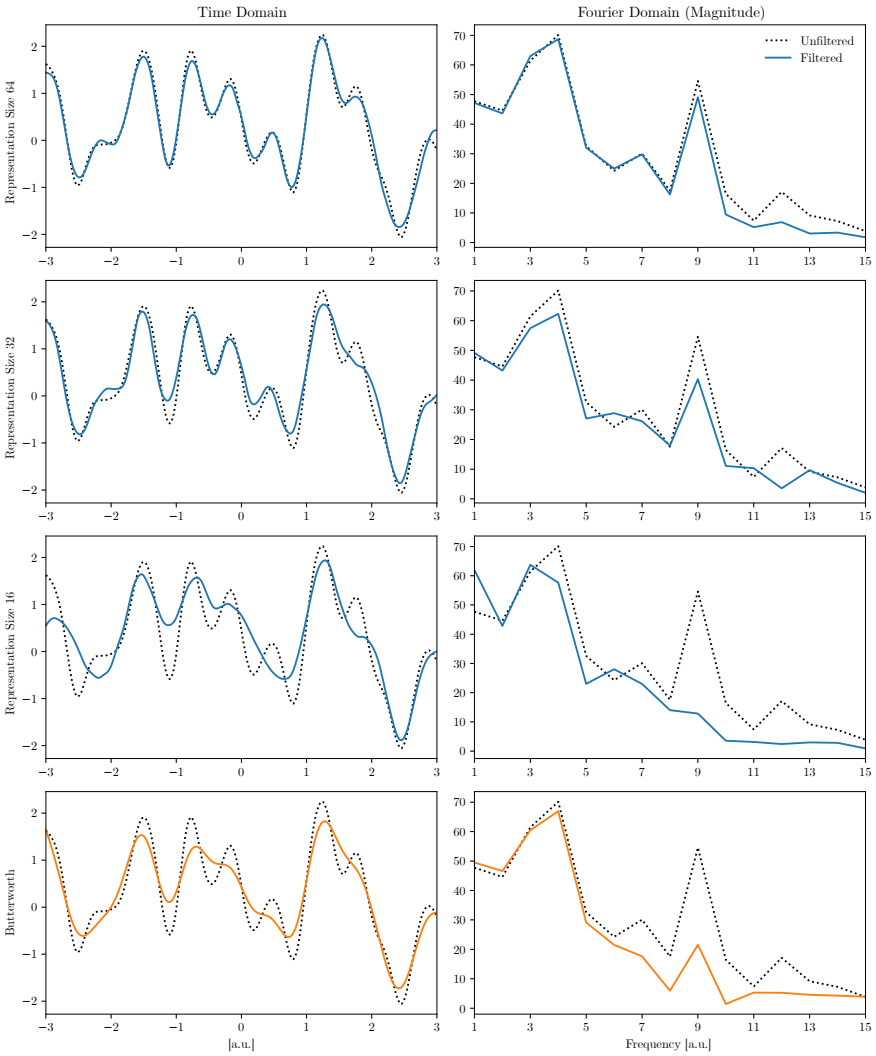

Figure 6.3: Example reconstructions of GP data in CNPs with varying representation sizes. A decrease in representation size leads to the omission of higher frequency components. In other words, the CNP acts like a low-pass filter. For comparison, we also show a simple 3^{rd} order Butterworth filter (Butterworth, 1930) with a cutoff frequency manually selected for visual similarity to the $D_r = 16$ model. The corresponding figure for a NP model is Figure A.5.

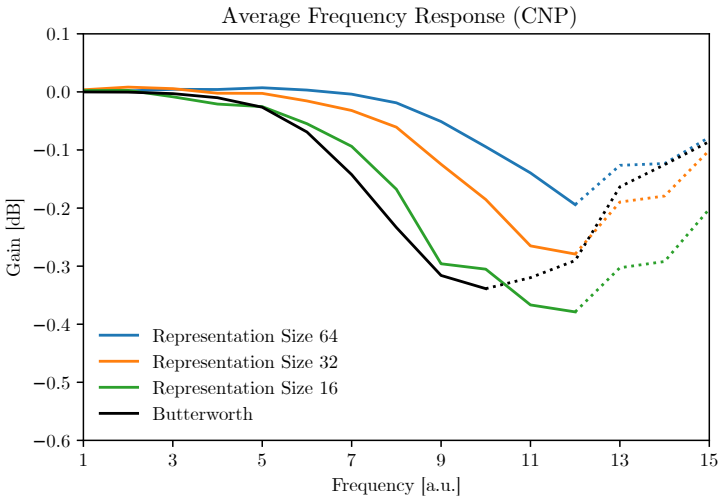

Figure 6.4: Average frequency response on GP data for CNPs with varying representation sizes. Decreasing the representation size leads to the CNP acting like a low-pass filter of increasing strength. For comparison, we also show a simple 3^{rd} order Butterworth filter (Butterworth, 1930). The cutoff frequency was manually selected so that filtered signals look similar to the $D_r = 16$ model in the *time domain*. While a theoretical low-pass filter would have a continuously decreasing gain, in practice there is almost no high-frequency content to suppress, so the response goes back to zero gain.

Razavi et al. (2019). From Equation (6.11) alone, one might perhaps expect a sharp decline of reconstruction performance after some point, but the fact that we sample points randomly along with the smoothly decaying frequency magnitude (see Figure 6.1) renders our findings rather unsurprising. The question remains, however, whether this in fact due to certain frequency components being suppressed. Figure 6.3 shows example reconstructions, along with the corresponding Fourier transform, for CNPs with varying representation size on a sample from a GP with an $l = 0.05$ EQ kernel. We find that a smaller D_r does indeed suppress higher frequency components, so much so that it essentially looks like the signal went through a low-pass filter. For comparison, we show the signal after application of a simple 3^{rd} order Butterworth filter (Butterworth, 1930), for which we configured the cutoff frequency manually for visual similarity to the $D_r = 16$ CNP. While not identical, they do look very similar, not only in the signal domain, but also in terms of dampening behaviour: Figure 6.4 shows the average frequency response of these CNPs when interpreted as low-pass filters. Note that we manually tuned the Butterworth filter to give a simple example of a true low-pass filter, there might be other filters or configurations that are even more similar to our CNPs. The corresponding example reconstructions for NPs can be seen in Figure A.5, with essentially the same result, except that the same

Butterworth filter is more similar to the $D_r = 32$ NP. We can clearly deduce that the representation size in CNPs and NPs can limit the frequency content of representable signals, so that they effectively act like low-pass filters.

Is that always the case? We find that it isn't, because there is another way the limit in Equation (6.11) can be obeyed, which might not be immediately obvious. Figure A.6 is another example, where the CNPs were trained on Fourier series data. Instead of smoothly decaying like in the GP case, the average Fourier magnitude is now uniform. The CNPs reconstruct the example signal rather faithfully, but only on part of the interval if the representation size is too small. Even though we were somewhat surprised by this behaviour, it makes perfect sense from the perspective of Equation (6.11). Rather than limiting f_{max}, the CNPs now limit the interval and thus $|b - a|$. The same example for NPs in Figure A.6 shows that a combination of those behaviours is also possible. From Figure 6.5 we can also very roughly estimate how tight the bound in Equation (6.11) is. With $K = 19$ the data has a maximum frequency of $f_{max} = K/(2\pi) = 3.02$. For $D_r = 32$ this would limit the size of the interval to $|b - a| < 5.29$, for $D_r = 16$ to $|b - a| < 2.65$, which is indeed approximately where the CNPs cut the signal off.

6.4.2 *How Neural Processes Represent Functions*

We saw in the previous section that Neural Processes are indeed subject to a limit of the frequency content in signals they can represent, which suggests that the learned representations contain a notion of frequency. Our goal in this section is to further investigate these representations. Because we are limiting ourselves to scalar input and output spaces, we can visualize how different regions of the signal space influence the individual representations. To this end, we encode (x, y) pairs from the region $x \in [-3, 3], y \in [-3, 3]$ and construct a heatmap for each individual representation channel, meaning $r_i(x, y)$. The result for a CNP with $D_r = 128$ on data from an EQ Gaussian Process with $l = 0.2$ can be seen in Figure 6.6. Because there is no intrinsic ordering in the individual channels, we sort them by taking the cross section at $y = -3$ and $y = 3$ and then forming a weighted average of their Fourier components. It should be noted that this is only one example and in general the learned representations will look different each time, but some general observations can be made that hold across examples and also different function spaces.

First, we find that representations are almost always anti-symmetric across $y = 0$. This is not surprising, as the function spaces we let the models learn are on average symmetric—in the sense that f and $-f$ will occur with the same probability—so the Neural Process learns the same representation, just with a different sign. More importantly,

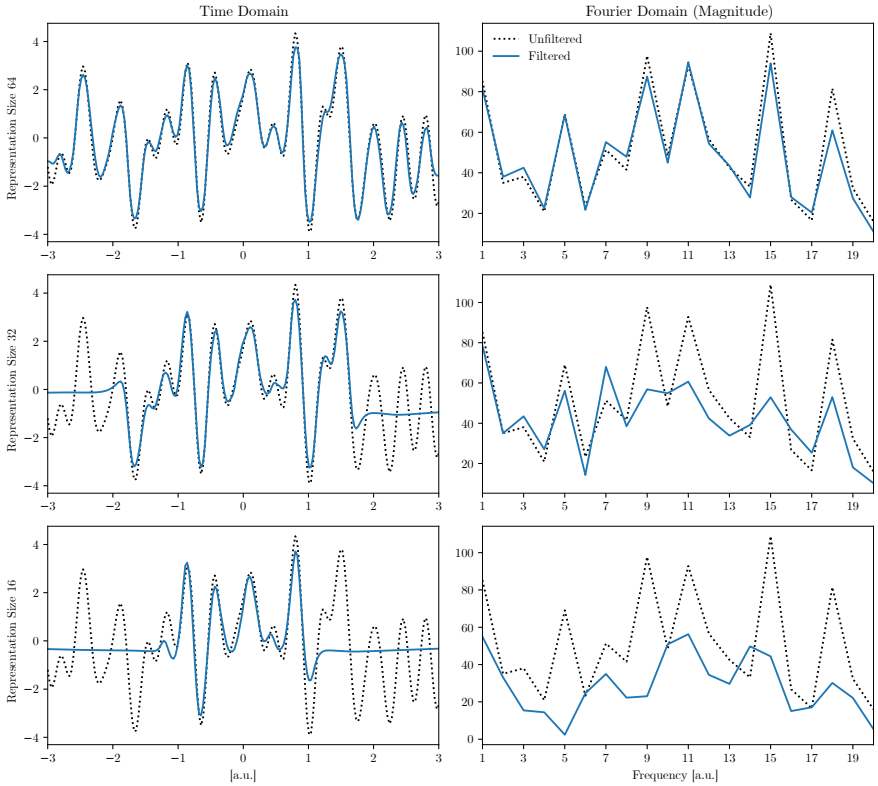

Figure 6.5: Example reconstructions of Fourier data in CNPs with varying representation sizes. A decrease in representation size leads to the model ignoring the outer regions of the input space. Compare this to the reconstructions of GP data in Figure 6.3, where higher frequency components are ignored. The corresponding figure for a NP model is Figure A.6.

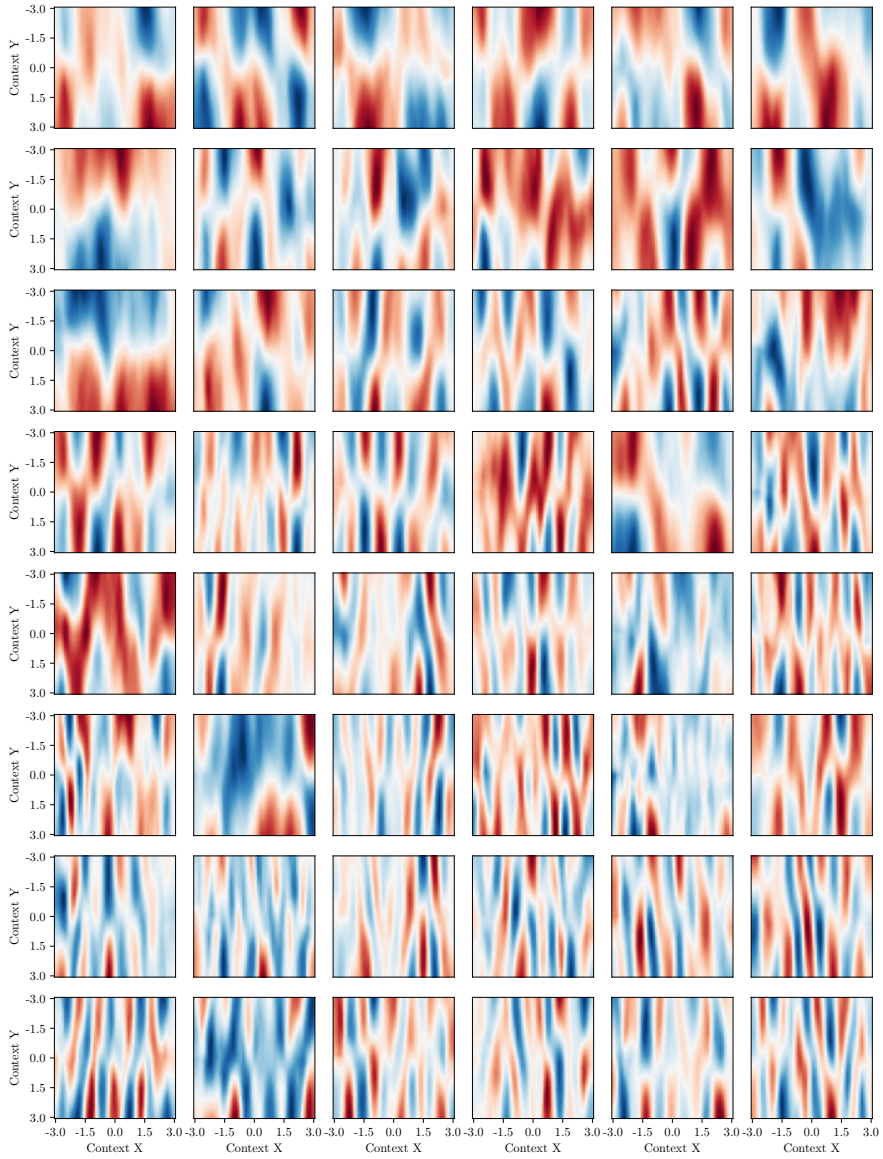

Figure 6.6: Influence of the context on the learned representations in a CNP with $D_r = 128$ and data coming from a Gaussian Process with $l = 0.2$. X refers to the input space (i.e. time), Y to the output space. These are the first 48 representations ordered by their average Fourier components at $y = -3$ and $y = 3$ (left-to-right, top-to-bottom). Note that each panel is normalized separately, so color values are not comparable. Different regions of the signal space write to different representation channels, meaning the representation implicitly learns to spatially resolve the input space. In CNPs, this seems to happen in an oscillating pattern, with different channels representing different frequencies. Compare this to the partitioning learned by a NP in Figure 6.8. We also show a cross-section of the representation channels in Figure 6.7.

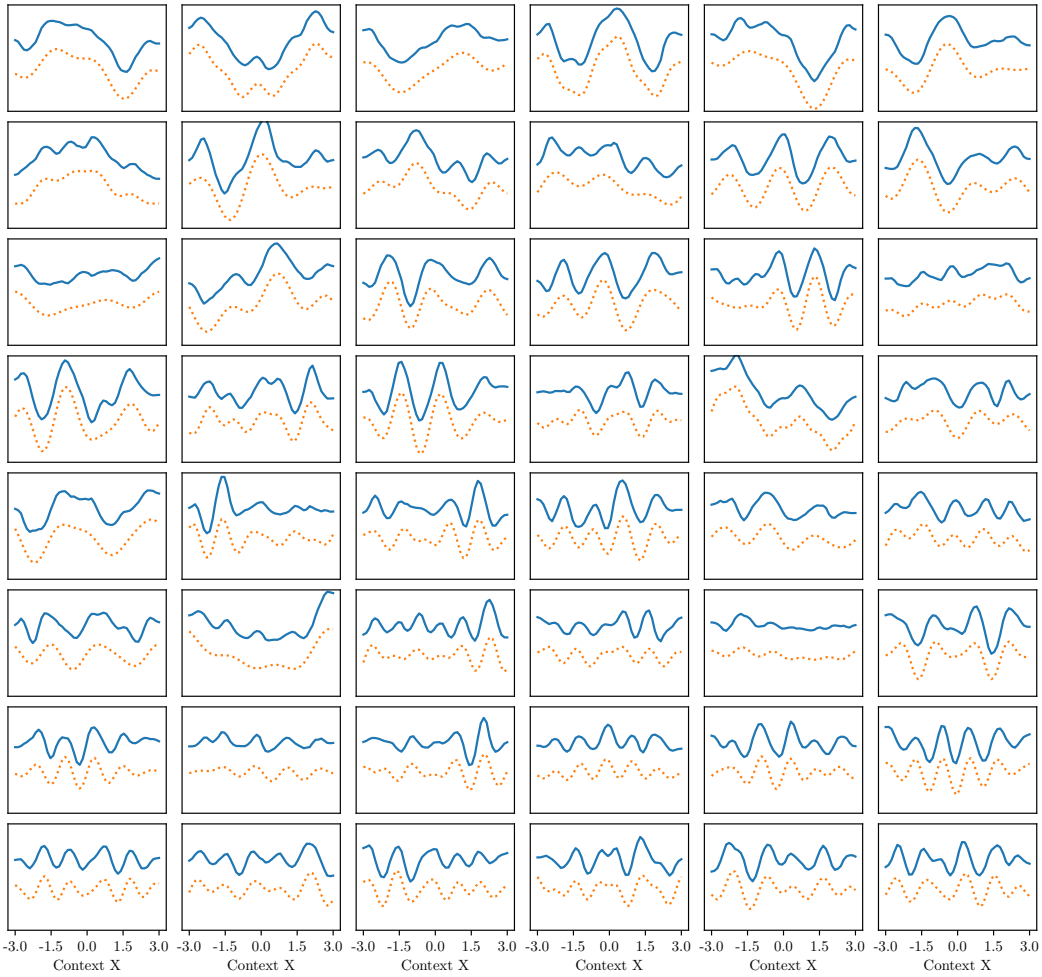

Figure 6.7: Influence of the context on the learned representations in a CNP with $D_r = 128$ and data coming from a Gaussian Process with $l = 0.2$. X refers to the input space (i.e. time), Y to the output space. This shows a cross-section of Figure 6.6, more precisely $1/2 \cdot (r(y = -3) - r(y = 3))$. Shown in orange (dotted and shifted for better visual contrast) is a reconstruction of just the 3 main frequency components of each representation, highlighting again that different channels represent different frequencies of the signal space.

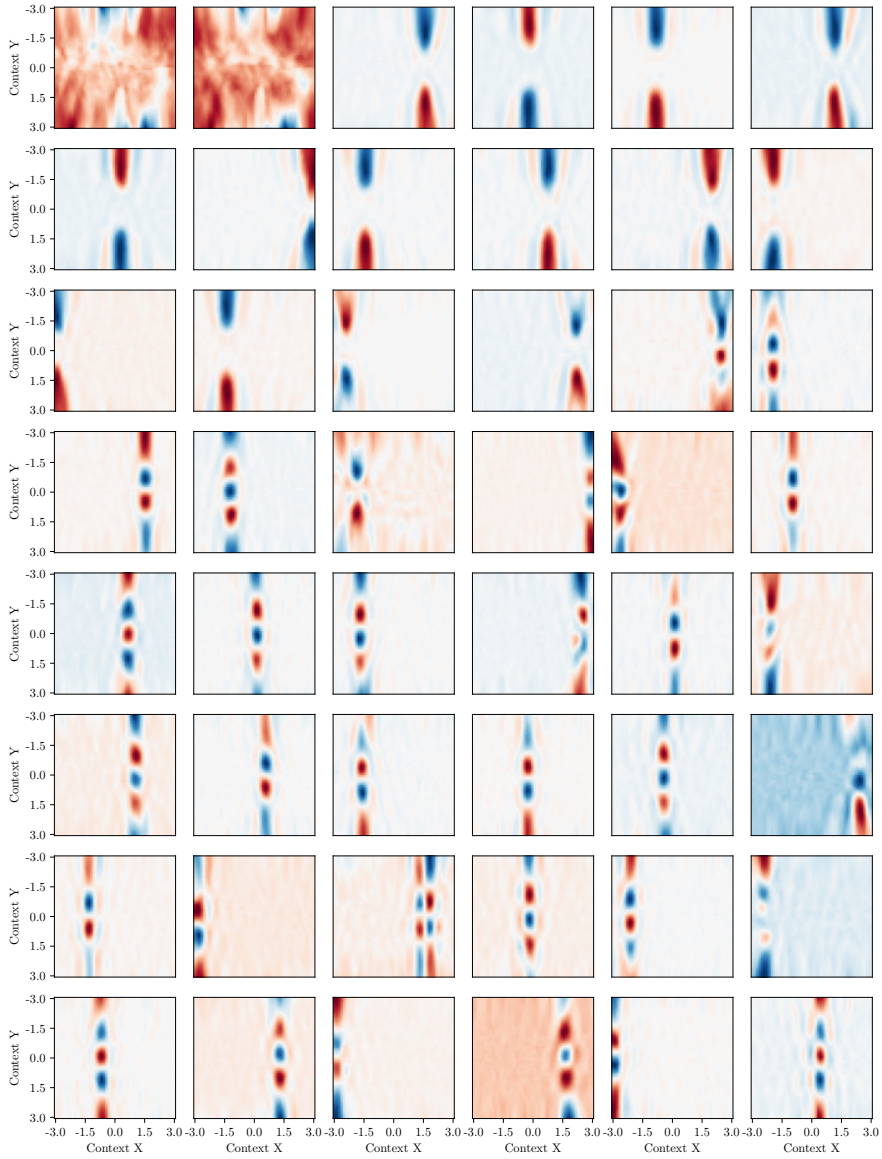

Figure 6.8: Influence of the context on the learned representations in a NP with $D_r = 128$ and data coming from a Gaussian Process with $l = 0.2$. X refers to the input space (i.e. time), Y to the output space. These are the first 48 representations ordered by their average Fourier components at $y = -3$ and $y = 3$ (left-to-right, top-to-bottom). Note that each panel is normalized separately, so color values are not comparable. Different regions of the input space write to different representation channels, meaning the representation implicitly learns to spatially resolve the input space. In NPs, this seems to happen in a way that partitions the signal space, so that a given channels is only used by a narrow section of the signal space. Compare this to the oscillating pattern learned by a CNP in Figure 6.6.

these visualizations reveal how the CNP is able to represent different frequencies in the signals. A given representation channel is utilized by different regions of the input space in way that an oscillating pattern emerges. This pattern has different frequencies for different representation channels. This becomes even clearer when we look at cross sections of these heatmaps. In Figure 6.7, we show the same representation channels as in Figure 6.6, but only their signed average at $y = -3$ and $y = 3$, meaning $1/2 \cdot (r(y = -3) - r(y = 3))$. We can clearly see that different representation channels represent different frequency components on the signals. This is remarkable, as we don't incentivize the model to do this (e.g. with a loss). The behaviour emerges "naturally". Along with the representations, we show their reconstructions from just the three main Fourier components, which is mostly sufficient to reconstruct each representation. This is to further illustrate the decomposition into frequency components. It is important to note that there is no reason for the model to cleanly separate individual frequencies, so each channel is in fact a combination of multiple frequencies. We find the same for representations learned on Fourier series data, just with higher frequencies. The corresponding visualizations can be seen in Figure A.3.

Having learned in the previous section that the representation size limits the frequency in the signals, one might be inclined to assume that this separation into different frequencies is necessary to achieve the former. This is not the case. We visualize the representations learned by a NP, also on Gaussian Process samples with an $l = 0.2$ EQ kernel, in Figure 6.8. Quite surprisingly, these representations look vastly different from those learned by a CNP. Instead of the oscillating behaviour along the x-axis, each representation channel is now written to by a very narrow region of the input space. In other words, the representations *partition* the x-axis, not unlike a simple memorization of the inputs, where a certain x-value "activates" a certain representation channel. This partitioning behaviour is essentially also what happens when the models limit the signal range like in Figure 6.5. It's by no means impossible for a variational Neural Process to learn a frequency decomposition; in Figure A.4 we show NP representations learned on Fourier series data and find that the NP actually combines both behaviours, with some channels representing very narrow input regions and others frequencies like the CNP. We suspect that the smoothing introduced by the variational formulation of NPs compared to a CNP makes it harder for them to learn frequency decompositions. Remember that during training, we sample from the predicted representation. This can also be interpreted as a random perturbation before reconstruction. Variational autoencoders (Jimenez Rezende et al., 2014; Kingma and Welling, 2014), which are very similar to a variational NP, are also thought to partition their latent space in

way that maximally spreads representations of individual data points under the prior distribution (Rezende and Viola, 2018).

6.4.3 *Neural Processes as Frequency Filters*

In Section 6.4.2 we found that Neural Processes are able to perform a decomposition of the function space into different frequencies. In Section 6.4.1 we also saw that limiting the representation size can make Neural Processes act like low-pass filters. At the same time, this behaviour is not reliable, as sometimes the models just ignore part of the signal space. Our goal in this section is to see if we can exert more control over the frequency response. We do this by training Neural Processes as band-pass and band-stop filters. In the bottom row of Figure 6.9, we show the distribution of Fourier component magnitudes for three different configurations of Fourier series. In the first (Reference), all components are allowed; in the second (band-stop), components in the middle of that range are suppressed; in the third (band-pass) only components in the middle of the range are allowed. Models trained on these data were then applied to the reference data, the result of which can be seen in the bottom-right panel of Figure 6.9. The models that are only shown certain frequencies during training will suppress others, meaning they effectively become programmable band-stop or band-pass filters. This is confirmed by the example in Figure 6.9. The only thing one needs to take care of is to adjust the y-range of the data before passing them through the trained filters. When we train a model on data where some frequency components are blocked, the distribution of y-values becomes narrower. That means we have to multiply the y-values of the reference data by $\sigma_{\text{band}}/\sigma_{\text{ref}}$— i.e. the ratio of standard deviations of the relative y-distributions. Ignoring this will result in gain in the non-blocked frequency regions.

Unfortunately, we were only partly able to elicit the same behaviour in variational NPs. While the trained band-stop filter worked exactly like the CNP band-stop, we were not able to train a band-pass filter. The models collapsed during training, meaning the loss plateaued and no meaningful representations were learned. There is no reason why a band-pass shouldn't work when a band-stop does, but we saw in Section 6.4.2 that it is much harder for NPs to learn frequency decompositions. We suspect that our hyperparameter configuration is simply not stable enough and that with some tuning we would be able to train a band-pass as well. However, we elected to work with the fixed hyperparameter settings we adopted from the original publications to ensure comparability.

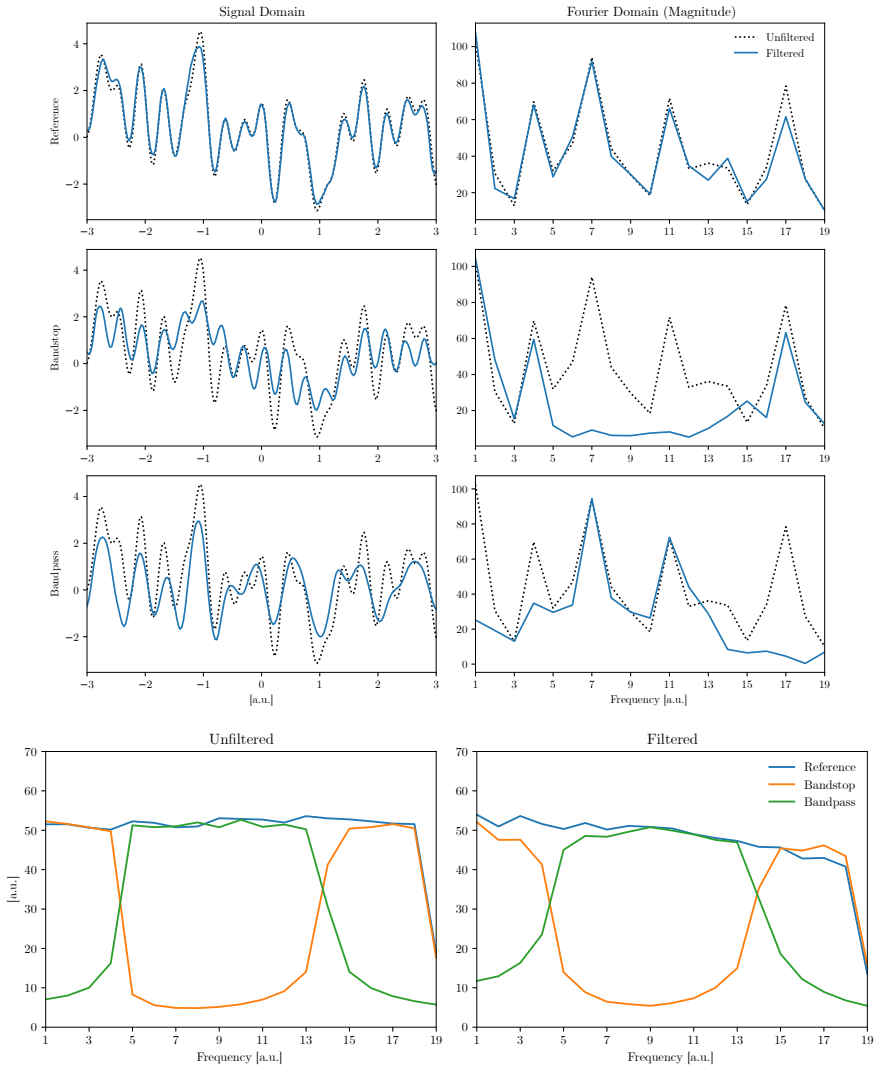

Figure 6.9: CNP trained to be a band-pass and a band-stop filter. We train three CNP models on Fourier series, where we only show certain frequencies during training for the models that should serve as frequency filters. The bottom row shows the distribution of Fourier components in the training data (left) and after applying each model to the reference data (right). The models only learn to represent frequencies seen during training, and as a result act like band-pass and band-stop filters. An example of this is given in the top rows.

6.5 DISCUSSION

In this chapter we investigated how Neural Processes form learned, finite-dimensional representations of function spaces. We first derived a theoretical upper bound on the frequency of signals that can be represented in Neural Processes with a given representation size. We empirically confirmed that the representation size does indeed pose such a limit and that this can result in Neural Processes acting like low-pass filters. Alternatively, models ignore part of the signal to keep higher frequencies. Both behaviours are in agreement with the derived bound. We then visualized learned representations to understand how the models incorporate the concept of frequency into them. In all cases the models formed an implicit representation of the input space, in the sense that different x-values are mapped to different representation channels. For CNPs, an oscillating pattern emerges, such that different representation channels correspond to different frequencies. In contrast to this NPs tend to partition the space into more or less disjunct regions. They are still able to learn a frequency decomposition like CNPs, but we assume that the variational training objective makes it much easier to simply partition the space. Finally, to further test the models' ability to distinguish frequencies and also as an example of possible practical benefits of our findings, we trained CNPs to be band-pass and band-stop filters. This worked extremely well, the Fourier component magnitudes of the training data are essentially "baked" into the models, and any frequency not found therein is subsequently suppressed in reconstructions from the models. An obvious use case would be programmable frequency filters, when perhaps a more complex frequency response is desired, and the value range of the test data is known.

Overall, our work offers exciting new insights into the inner workings of Neural Processes and into the learning of representations of function spaces. Many applications of deep learning are concerned with representation learning in some way, and we hope that our findings inspire further research and forge a better understanding of the methods used in the field. We have also highlighted a possible real-world use case for Neural Processes. While later additions to the family have enjoyed application to a variety of problems, the original Neural Processes (Garnelo et al., 2018a; Garnelo et al., 2018b) have not been studied from a more practical perspective.

7

GP-CONVCNP: IMPROVING GENERALIZATION IN CONVOLUTIONAL CONDITIONAL NEURAL PROCESSES

While the previous chapter focused on the original Neural Processes (Garnelo et al., 2018a; Garnelo et al., 2018b), we use this chapter to present an improvement to a more recent addition to the NP family, called *Convolutional Conditional Neural Process* (ConvCNP) (Gordon et al., 2020). Recall that our choice of Neural Processes was motivated by their ability to learn representations of function spaces. But what does it mean to successfully learn such a representation? We believe that it should be characterized by the following: 1) accurate predictions, meaning predictions should be as close as possible to the true underlying function, 2) good reconstruction of the given observations, 3) generalization, because we assume that there will be some underlying generative process from which the distribution originates and which is valid beyond the finite data we observe, 4) the ability to generate multiple consistent hypotheses, in other words being able to sample from a model. The latter is especially important when only few context observations are given that could be explained by several different functions. ConvCNP improve upon prior art mainly with respect to prediction accuracy and reconstruction ability. Unfortunately, this comes at the cost of the other criteria.

In this chapter, we extend ConvCNP to address all of the above, with a particular focus on the ability to generalize. By combining ConvCNP with a Gaussian Process, we achieve a significant improvement in generalization: the model, which we call GP-ConvCNP, can better extrapolate far from the provided context observations and is more robust to a distribution shift at test time. It further reintroduces the ability to sample from the model, something that ConvCNP is incapable of, showing a better sample distribution than both NP and ANP. Finally, we find that our proposed model often yields a significant improvement in predictive performance on in-distribution data as well. Our evaluation is based on several synthetic datasets and real time series datasets. The findings presented herein are currently under review:

Petersen, Jens, Gregor Köhler, David Zimmerer, Fabian Isensee, Paul Jäger, and Klaus H. Maier-Hein (2020b). "GP-ConvCNP: Improving Generalization in Convolutional Conditional Neural Processes". In: *AAAI Conference on Artificial Intelligence (under review)*.

7.1 METHODS

We introduce the framework of Neural Processes (Garnelo et al., 2018a; Garnelo et al., 2018b) in detail in Section 6.2.1, the following is only a brief summary. We assume that we are given a set of N observations $C = \{(x_c, y_c)\}_{c=1}^{N} =: (\mathbf{x_c}, \mathbf{y_c})$, often called the *context*, where $x_c \in X$ are samples from the input space X and $y_c \in Y$ are samples from the output space Y (commonly $X = \mathbb{R}^{d_X}$ and $Y = \mathbb{R}^{d_Y}$, in this work we restrict ourselves to $X = \mathbb{R}$, because time is scalar). It is assumed that these observations were generated by some function $f : X \to Y$, i.e. $y_c = f(x_c)$, and our goal is to infer f from C so that we may evaluate it at arbitrary new input locations x_t. In reality, this will most likely mean we have collected a number of measurements over time and are interested in an f that lets us interpolate and extrapolate those measurements. Note that when we speak of predictive performance, we refer to both of those cases and don't necessarily mean it in a temporal sense. This problem is ill-posed without placing further assumptions on f, which is why we typically restrict it to some function space $\mathcal{F} = \{f_i\}$: polynomials of some order, a combination of oscillating functions with different frequencies, etc.. However, in many cases it is undesired or even impossible to manually specify $\mathcal{F}$, so Neural Processes propose to use neural networks to learn an approximate representation of $\mathcal{F}$ by observing many examples $f \in \mathcal{F}$. The latter are typically represented as a context set C (the measurements we have) and a *target* set $T = \{(x_t, y_t)\}_{t=1}^{M} =: (\mathbf{x_t}, \mathbf{y_t})$ (the measurements we're interested in). By learning to reconstruct the examples f from a limited number of context points a model should implicitly form a representation of $\mathcal{F}$.

7.1.1 *Optimization*

We have already introduced the optimization target for Neural Processes in Section 6.2.2, so only repeat it here briefly. Our goal is to let a model learn a representation of $\mathcal{F}$. We do this by letting it reconstruct observations $\mathbf{y_t}$ from the context $(\mathbf{x_c}, \mathbf{y_c})$ and inputs $\mathbf{x_t}$, which leads to the following learning objective:

$$\max_{\theta} \sum_{f \in \mathcal{F}} \log p_\theta(\mathbf{y_t} | \mathbf{x_t}, \mathbf{x_c}, \mathbf{y_c}) \tag{7.1}$$

$$= \max_{\theta} \sum_{f \in \mathcal{F}} \sum_{t} \log \mathcal{N}(y_t; G_\theta^\mu(Z, x_t), G_\theta^\sigma(Z, x_t)) \tag{7.2}$$

This objective is common to all approaches we evaluate in our work, and the right hand side formalizes the fact that we choose to always model the output as a diagonal Gaussian, parametrized by mean and variance functions $G_\theta^\mu, G_\theta^\sigma$ that seek to maximize the log-likelihood of the targets $\mathbf{y_t}$. The output variance can also be fixed,

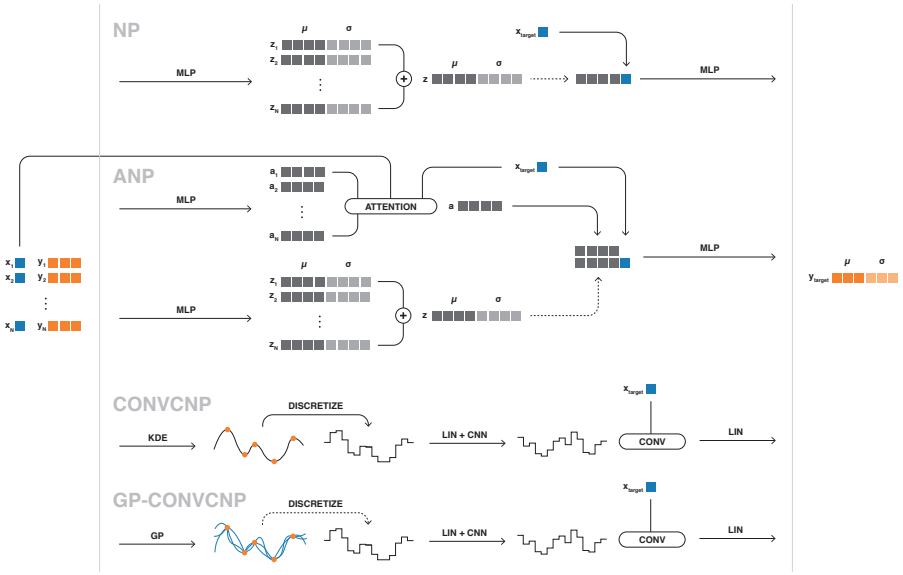

Figure 7.1: Schematic overview of the different methods used in this work. Dotted lines indicate sampling and we use the following acronyms: multilayer perceptron (MLP), kernel density estimate (KDE), Gaussian Process (GP), linear layer (LIN), convolutional neural network (CNN). (*First row*) Neural Processes (NP) encode each context point (x_c, y_c) into a representation z_c. These are then averaged to form a global representation z. A sample from the global representation is concatenated with the target input x_t to predict the target output y_t. (*Second row*) Attentive Neural Processes (ANP) contain a NP, but have a second deterministic path. In this path, the context pairs are also encoded separately into representations $\mathbf{a_c}$. These are then combined via an atttention mechanism that uses x_t as the query, $\mathbf{x_c}$ as the keys and $\mathbf{a_c}$ as the values. The resulting representation a is concatenated with the representation from the NP path and the target input to predict the target output. (*Third row*) ConvCNP performs a kernel density estimate on the context observations $(\mathbf{x_c}, \mathbf{y_c})$, thus mapping to a continuous representation. This representation is evaluated on a grid, i.e. discretized, and a projection and CNN operate on the discretized representation. The result is evaluated at a target input x_t by performing a convolution with the discretized representation and finally projected to predict the target output. (*Fourth row*) GP-ConvCNP works similar to ConvCNP, but instead of a deterministic kernel density estimate a Gaussian Process is applied to the context. We sample from the GP posterior and discretize the result, continuing with the same operations as in ConvCNP. Note that for visual purposes, the KDE and GP outputs are one-dimensional, but in reality the output space can have any number of dimensions.

but Le et al. (2018b) show that a learned output variance is preferable. Z is a representation of the context $(\mathbf{x_c}, \mathbf{y_c})$, i.e. there is a mapping $E : X, Y \to Z$. The implementation of E is where the members of the Neural Process family differ most, as we show in the following sections.

We can rewrite Equation (7.2) as

$$\max_{\theta} \sum_{f \in \mathcal{F}} \log p_\theta(\mathbf{y_t}|\mathbf{x_t}, Z) \tag{7.3}$$

where Z is given by the different E that *encode* the context defined above. For ConvCNP this is deterministic, so we can maximize Equation (7.2) directly. For the other methods we can again rewrite Equation (7.3) as

$$\log p(\mathbf{y_t}|\mathbf{x_t}, \mathbf{x_c}, \mathbf{y_c}) = \mathop{\mathbb{E}}_{z \sim p(z|\mathbf{x_c}, \mathbf{y_c})} \log p(\mathbf{y_t}|\mathbf{x_t}, z) \tag{7.4}$$

where we now distinguish z as an expression of Z. In GP-ConvCNP, $p(z|\mathbf{x_c}, \mathbf{y_c})$ is given by the GP posterior, so for training we would need to integrate over this posterior. In practice, we just draw a single sample, which is common practice in stochastic mini-batch training. Approximating the expectation with this sample, we can also directly maximize the log-likelihood.

In contrast to the above, $p(z|\mathbf{x_c}, \mathbf{y_c})$ is an unknown or intractable mapping in NP and ANP, so we employ variational inference, i.e. we approximate $p(z|\mathbf{x_c}, \mathbf{y_c})$ with a member of some family Q that we can find by optimization. As shown in Section 6.2.2, the log-likelihood then becomes

$$\begin{aligned} \log p(\mathbf{y_t}|\mathbf{x_t}, \mathbf{x_c}, \mathbf{y_c}) \geqslant & \mathop{\mathbb{E}}_{z \sim q(z|\mathbf{x_t}, \mathbf{y_t})} \log p(\mathbf{y_t}|\mathbf{x_t}, z) \\ & - D_{KL}\left(q(z|\mathbf{x_t}, \mathbf{y_t}) || q(z|\mathbf{x_c}, \mathbf{y_c})\right) \end{aligned} \tag{7.5}$$

To maximize the left hand side it is sufficient to maximize the right hand side, and Equation (7.5) is what is being optimized in NP and ANP. q corresponds to what we designated as E above. Like for GP-ConvCNP, we approximate the expectation with a single sample during training.

In our implementation, we use Adam (Kingma and Ba, 2015) with an initial learning rate of 0.001. We train each model for 600 000 batches with a batch size of 256. We repeatedly multiply the learning rate by $\gamma = 0.995$ after training for 1000 batches.

7.1.2 *Neural Processes & Attentive Neural Processes*

We already introduced Neural Process in detail in Chapter 6, so we only summarize them here. We also show an illustration of all methods

used in this work in Figure 7.1. The original Neural Processes (Garnelo et al., 2018b) implement E as a neural network that encodes individual context observations $(\mathbf{x_c}, \mathbf{y_c})$ into a finite-dimensional space. These representations are then averaged to form the global representation Z. Similar to Equation (7.2), Z parametrizes a Gaussian distribution, which enables NP to sample from this latent space and to produce diverse predictions. We do not consider the deterministic NP variant (Garnelo et al., 2018a) in this chapter, because we are interested in models that are able to generate multiple samples. In simpler terms, Neural Processes can be thought of as variational autoencoders that average the representations of multiple inputs, and reconstruct an output from this averaged representation and a target point x_t. Like a VAE, a NP is trained by maximizing a lower bound on Equation (7.2), as shown in Section 7.1.1. In our NP implementation E and $(G_\theta^\mu, G_\theta^\sigma)$ are symmetric 6-layer MLP, with a representation size of 128. Attentive Neural Processes (Kim et al., 2019) are motivated by the observation that NP poorly reconstruct the provided context, i.e. the predictions seem to miss the context points, as seen for example in Figure 7.2. To mitigate this effect, ANP augment NP with an additional deterministic encoder-decoder path. Instead of averaging the individual representations, a learned attention mechanism (Vaswani et al., 2017) combines them, conditioned on a target point x_t. So while NP need to compress representations to a single point in Z, ANP don't have this bottleneck, which likely contributes to their improved performance. In our ANP implementation, the deterministic path mirrors the variational path, with both the representation dimension and the embedding dimension of the attention mechanism being 128. Le et al. (2018b) have evaluated several hyperparameter configurations for NP and ANP and our implementation matches their best performing one.

7.1.3 *From* ConvCNP *to* GP-ConvCNP

With the goal of enabling translation equivariance (i.e. independence of the value range of $\mathbf{x_c}$ and $\mathbf{x_t}$) in Neural Processes, the authors of Convolutional Conditional Neural Processes (ConvCNP) (Gordon et al., 2020) approach their work from the perspective of *learning on sets* (Zaheer et al., 2017). While NP and ANP map the context set into a finite-dimensional representation, ConvCNP map it into an infinite-dimensional function space. This is also visualized in Figure 7.1, where we present an illustration of all methods. The authors show that in this scenario translation equivariance (as well as permutation invariance) can only be achieved if the mapping E can be represented in the form

$$E(\mathbf{x_c}, \mathbf{y_c}) = \rho(E'(\mathbf{x_c}, \mathbf{y_c})) \,, \quad E'(\mathbf{x_c}, \mathbf{y_c}) = \sum_c \phi(y_c)\psi(\cdot - x_c) \quad (7.6)$$

where $\phi : Y \to \mathbb{R}^2$ and $\psi : X \to \mathbb{R}$, so that E' defines a function and ρ operates in function space and must be translation equivariant. The similar naming of E, E' is deliberate, because herein lies a key difference to NP (and also ANP): NP learn a powerful mapping (i.e. neural network) from the context to a representation and then another one from this representation to the output space, whereas ConvCNP employs a very simple mapping to another representation (to function space, because ϕ and ψ are defined with kernels, see below). A powerful approximator is then learned that operates *within* this representation space, as ρ is a CNN operating on a discretization of E'. The mapping back to output space is again a simple one, usually also ψ combined with a linear map. In this sense, both E and E' can be thought of as representations when we make the connection to NP. See also Figure 7.1 for a visualization of these differences. In Gordon et al. (2020), ψ is chosen to be a simple Gaussian kernel, and ϕ such that the resulting E' has two components:

$$E'_0(\mathbf{x_c}, \mathbf{y_c}) = \sum_c k(\cdot, x_c)\,, \quad E'_1(\mathbf{x_c}, \mathbf{y_c}) = \sum_c \frac{y_c k(\cdot, x_c)}{E'_0} \tag{7.7}$$

which is simply the combination of a kernel density estimator and a Nadaraya-Watson estimator (Nadaraya, 1964; Watson, 1964). This estimate is discretized on a suitable grid and a CNN ρ is applied, the result of which is again turned into a continuous function by convolving with ψ. We use the official implementation[1] in our experiments.

In this work, we propose GP-ConvCNP, a model that replaces the deterministic kernel density estimate E' in ConvCNP with a Gaussian Process posterior (for an introduction see for example Rasmussen and Williams (2006)). This posterior is a normal distribution with a mean function $m(\mathbf{x_t})$ conditioned on the context and a covariance function $K(\mathbf{x_t})$ specified by some kernel k:

$$m(\mathbf{x_t}) = k_{tc}^T \left(k_{cc} + \sigma^2 \mathbb{I}\right)^{-1} \mathbf{y_c} \tag{7.8}$$

$$K(\mathbf{x_t}) = k_{tt} + \sigma^2 - k_{tc}^T \left(k_{cc} + \sigma^2 \mathbb{I}\right)^{-1} k_{tc} \tag{7.9}$$

where $k_{tc} = k(\mathbf{x_t}, \mathbf{x_c})$ etc. and σ^2 is a noise parameter that essentially determines how close the prediction will be to the context points. We make this parameter learnable. The first obvious benefit of this model is that we can sample from the GP posterior distribution and thus also from our model, recovering one very compelling property of NP that ConvCNP lacks. Another advantage we see is that by working with a distribution instead of a deterministic estimate as input to the CNN, the data distribution is implicitly smoothed. It has been established that such smoothing reduces overfitting and improves generalization, e.g. by adding noise to inputs (An, 1996; Bishop,

1 https://github.com/cambridge-mlg/convcnp

1995, p.347) or more generally doing data augmentation (Volpi et al., 2018; Jackson et al., 2019). Working with a distribution instead of a deterministic estimate, we need to perform Monte-Carlo integration to get a prediction from our model. During training, however, we only use a single sample, as is commonly done e.g. in variational autoencoders when training with mini-batch stochastic gradient descent. To facilitate comparison, the kernel we use in our GP is the same as in ConvCNP, i.e. a Gaussian kernel with a learnable length scale.

7.2 EVALUATION

We design our experiments with the purpose of evaluating how well members of the Neural Process family, including the one we propose, are suited for the task of learning distributions over functions, specifically for time series data. Like the works we compare ourselves to, we evaluate both predictive performance (*How good is our prediction between context points?*) and reconstruction performance (*How good is our prediction at the context points?*). Predictive performance, which does not have a temporal meaning here, is measured via the predictive log-likelihood, and for models with a latent distribution (i.e. all except for ConvCNP) we average over 100 samples. To measure reconstruction performance, we resort to the root-mean-square error (RSME), because predictions directly at the context points are usually extremely narrow Gaussians, leading to unstable likelihoods.

As outlined in the introduction, one defining aspect of successfully learning a distribution over functions is a model's ability to generalize. This can mean several things, for example independence with respect to the input value range, called translation equivariance. This is a key feature of ConvCNP (as long as a stationary kernel is used for interpolation), and we retain this property in GP-ConvCNP. We evaluate two further attributes of generalization, both on real world data: one is the ability to extrapolate the context information, i.e. to produce good predictions well into the future by inferring an underlying pattern; the other is the ability to deal with a distribution shift at test time, in our case a shift from simulated to real world data.

On top of the above, we are also interested in how well the distribution of samples from a model matches the ideal distribution. In general, the latter is not accessible, but for some synthetic examples we describe below, specifically those from a Gaussian Process, we do have access, simply by using the generating GP as an oracle. We can then compare this reference—a Gaussian distribution—with the distribution of samples from our model. Note that one sample is a prediction at all target points at once, as seen for example in Figure 7.2. The majority of approaches that estimate differences between distributions fall into the categories of either f-*divergences* (Csiszár, 1963; Morimoto, 1963; Ali and Silvey, 1966) or *Integral Probability Measures* (Müller,

1997)—for an overview see for example Sriperumbudur et al. (2009). The former require evaluations of likelihoods for both distributions, while we only have individual samples from our model. We could place a density estimate on those samples, but that is often inaccurate in high dimensions (Vershynin, 2018). IPM only compare samples from the distributions and are thus suited for our scenario, so we opt for a parameter-free representative of this category, the Wasserstein distance W_2:

$$W_p(P, Q) = \min_{\pi} \left(\sum_{i=1}^{|P|} \|x_i - y_{\pi(i)}\|^p \right)^{1/p} \tag{7.10}$$

Here $P = \{x_i\}_i$ and $Q = \{y_i\}_i$ are collections of samples from the two distributions. In colloquial terms, the Wasserstein distance is the minimum overall distance between sample pairs, taken over all possible pairings between samples from the two distributions. For this reason the Wasserstein-1 distance is also called the *Earth Mover Distance*. p is the only hyperparameter we need to select, making this measure a very convenient choice. We set $p = 2$ so that the underlying distance metric becomes the Euclidean distance.

7.3 DATA

7.3.1 *Synthetic Data*

We initially test our method on diverse synthetic 1D functions. The first two have also been used in Gordon et al. (2020), and they allow us to evaluate the sample diversity, as outlined above:

1. Samples from a Gaussian Process with a Matern-5/2 kernel with lengthscale parameter $l = 0.5$. The kernel is given by

$$k(x, x') = \left(1 + \frac{\sqrt{5}|x - x'|}{l} + \frac{5|x - x'|^2}{3l^2}\right) \cdot \exp\left(-\frac{5|x - x'|}{l}\right) \tag{7.11}$$

2. Samples from a Gaussian Process with a weakly periodic kernel that is given by

$$k(x, x') = \exp\left(-\frac{|x - x'|^2}{8}\right) \cdot \exp\left((\cos(8\pi x) - \cos(8\pi x'))^2\right) \cdot \exp\left((\sin(8\pi x) - \sin(8\pi x'))^2\right) \tag{7.12}$$

3. Fourier series that are given by

$$f(x) = a_0 + \sum_{k=1}^{K} a_k \cos(kx - \phi_k) \tag{7.13}$$

where K is a random integer from $[10, 20)$ and a_k (including a_0) as well as ϕ_k are random real numbers drawn from $[-1, 1]$.

4. Step functions, where we draw S stepping points along the x-axis, with S a random integer from $[3, 10)$. The interval between two stepping points is assigned a constant value that is drawn from $[-3, 3]$. We ensure that each interval is at least 0.1 units wide and that the step difference is also at least 0.1 units in magnitude.

For all synthetic function draws we define the x-axis to cover the interval $[-3, 3]$. We draw N context points uniformly from this interval, with N a random integer from the range $[3, 100)$. We then draw M target points in the same manner, with M a random integer from $[N, 100)$. These choices follow the guidelines presented in Le et al. (2018b). During training, we add the context points to the target set, as done in Garnelo et al. (2018a) and Garnelo et al. (2018b), so that the methods also learn to reconstruct the context. Examples of the different functions can be seen in in Figure 7.2.

7.3.2 *Temperature Time Series*

The first real world dataset we look at are weather recordings for several different US, Canadian and Israeli cities. In particular we focus on temperature measurements in hourly intervals that have been collected over the course of 5 years[2]. Temperatures in each city are normalized by their respective means and standard deviations. We randomly sample sequences of ~1 month as instances and evaluate two tasks, taking US and Canadian cities as the training set and Israeli cities as the test set:

1. Interpolation, where we draw context points and target points randomly from the entire sequence (i.e. the same as in the synthetic examples).

2. Extrapolation, where context points are drawn from the first half of the sequence and performance is evaluated on the second half (as shown in Figure 7.3). We can reasonably be sure that temperature changes between day and night occur in the future with the same frequency, so extrapolating this pattern is a good test of a model's ability to generalize.

2 https://www.kaggle.com/selfishgene/historical-hourly-weather-data

Occasionally there are NaN values reported in the dataset, we either crop those when at the begging/end of a sequence or fill them via linear interpolation. For both training and testing we draw random sequences of length 720 (i.e. 30 days) from the corresponding set, and then draw N context points and M target points from the sequence, with N from the interval $[20, 100)$ and M from $[N, 100)$. We define the time range for a given sequence to be $[0, 3]$, so that one time unit is equivalent to 10 days. We evaluate each seed for a model with 100 random samples and report the mean and standard deviation over 5 seeds for each model.

7.3.3 *Population Dynamics*

The second real world dataset are measurements of a predator-prey population of lynx and hare by the Hudson Bay Company (Hewitt, 1921). Such population dynamics are often approximated by Lotka-Volterra equations (Leigh, 1968), so we train models on simulated population dynamics and test on both the simulated and real world data. Gordon et al. (2020) used this dataset as well, but only to qualitatively show that ConvCNP can be applied to it. The analysis will allow us to quantify how robust the models are to a shift in distribution at test time, as the simulation parameters are almost certainly not an ideal fit for the real world data.

The real data were recorded at the end of the 19th and the start of the 20th century by the Hudson's Bay Company. To the best of our knowledge, the data represent recorded trades of pelts from the two animals and not direct measurements of the populations. Earlier works that work with the same data point to Hewitt (1921) as the earliest source, but there is no unique source for the data in a tabular format. We used https://github.com/stan-dev/example-models/blob/master/knitr/lotka-volterra/hudson-bay-lynx-hare as our source. For evaluation, we normalize the data so that the mean population matches the mean of populations in the simulated data and the time interval matches the mean duration of a simulated population.

The simulated data were constructed as follows: let X be the number of predators at a given time and Y the number of prey. We draw initial numbers X from $[50, 100)$ and Y from $[100, 150)$. We then draw time increments from an exponential distribution and after each time increment one of the following events occurs:

1. A single predator is born with probability proportional to the rate $\theta_0 \cdot X \cdot Y$
2. A single predator dies with probability proportional to the rate $\theta_1 \cdot X$
3. A single prey is born with probability proportional to the rate $\theta_2 \cdot Y$

4. A single prey dies with probability proportional to the rate $\theta_3 \cdot X \cdot Y$

The rate of the exponential distribution we draw time increments from is the sum of the above rates. Each population is simulated for 10000 events, and we reject populations that have died out, populations that exceed a total number of 500 individuals at any given point, as well as those where the accumulated time is larger than 100 units. To get value ranges that are better suitable for training, we rescale the time axis by a factor 0.1 and the population axis by a factor 0.01. For each population we draw θ_0 from $[0.005, 0.01]$, θ_1 from $[0.5, 0.8]$, θ_2 from $[0.5, 0.8]$ and θ_3 from $[0.005, 0.01]$. These parameters result in roughly $2/3$ of the simulated populations matching our criteria. We also tried the parameters reported in Gordon et al. (2020), but found that we had to reject more than 90% of populations, which meant an unreasonably long training time, as the simulation process for the populations is difficult to parallelize and thus rather slow. The N context points and M target points are again drawn randomly from a population, with N from $[20, 100)$ and M from $[\max(70, N), 150)$

7.4 RESULTS

7.4.1 *Synthetic Data*

Table 7.1 shows results for the various synthetic function types. In this experiment the models are trained and tested on random samples generated in the same way, so these results measure *in-distribution* performance. We find that GP-ConvCNP is the overall best performing method, significantly so in terms of predictive performance for 3 out of 4 function types and performing on par with ConvCNP on the other. For reference, we also show the performance for a Gaussian Process with an EQ kernel (what our model uses as an initial estimate). When the initial estimate is good, our model can leverage that information, matching the performance of an oracle GP with a Matern-5/2 kernel. At the same time, the EQ estimate doesn't have to be good for our model to perform well.

Reconstruction performance is on par with ConvCNP in 3 out of 4 instances and significantly better in one. For examples originating from a Gaussian Process, we can evaluate the sample diversity with respect to the oracle GP, finding that GP-ConvCNP significantly outperforms the other methods in this regard. It is important to note, however, that this measure does not fully isolate the sample diversity. A low reconstruction error, for example, will also improve the W_2, which is likely the reason that ANP still performs better than NP, even though the former hardly displays any variation in its samples, as seen in Figure 7.2. The figure also shows how NP and ANP struggle to fit high frequency signals, while ConvCNP and GP-ConvCNP are

Table 7.1: Results for synthetically created data. Test data was generated with the same parameters as the training data, so we're looking at *in-distribution* performance. ↑/↓ indicate that higher/lower is better. Errors represent 1 standard deviation over 5 runs with different seeds (except for the GP entries, where we show the standard error of the mean), where each run was evaluated with 102 400 (30 720 for W_2) samples. Bold indicates that the method(s) are significantly better than all non-bold methods, significance being assumed when the difference is larger than the root sum of squares of the standard deviations. Overall, GP-ConvCNP outperforms the competing approaches, especially in terms of predictive log-likelihood and sample diversity (compared to an oracle) where applicable. In terms of reconstruction error, our method outperforms prior art on three datasets, but is on par with ConvCNP on two of those. Interestingly, the EQ-GP, which is what our model uses as an initial estimate, performs rather poorly in all but the first example. In the first example, where the EQ-GP is already a decent estimate, our approach leverages that information and matches the oracle GP in predictive performance! The reconstruction error and W_2 of the oracle are zero, so we don't show them here.

		Matern-5/2 GP	Weakly Per. GP
Predictive LL↑	GP (EQ)	*1.031 ± 0.075*	*−8.034 ± 2.260*
	GP (Oracle)	*1.933 ± 0.095*	*1.876 ± 0.026*
	NP	−0.496 ± 0.027	−1.161 ± 0.007
	ANP	0.723 ± 0.046	−1.047 ± 0.008
	ConvCNP	1.710 ± 0.038	−0.153 ± 0.033
	GP-ConvCNP	**1.930 ± 0.031**	**−0.090 ± 0.021**
Recon. Error↓	GP (EQ)	*0.001 ± 0.001*	*0.028 ± 0.001*
	NP	0.027 ± 0.001	0.500 ± 0.003
	ANP	**0.008 ± 0.002**	0.491 ± 0.004
	ConvCNP	0.025 ± 0.020	**0.109 ± 0.077**
	GP-ConvCNP	0.013 ± 0.002	**0.061 ± 0.007**
W_2 ↓	GP (EQ)	*4.294 ± 0.007*	*4.521 ± 0.003*
	NP	1.836 ± 0.021	2.745 ± 0.004
	ANP	1.369 ± 0.048	2.708 ± 0.002
	ConvCNP		
	GP-ConvCNP	**0.987 ± 0.086**	**1.800 ± 0.045**
		Fourier Series	Step Functions
Predictive LL↑	GP (EQ)	*−0.241 ± 0.752*	*-2×10^{17}*
	NP	−1.743 ± 0.020	−3.287 ± 0.491
	ANP	−0.976 ± 0.028	−65.141 ± 60.979
	ConvCNP	0.372 ± 0.065	**−0.522 ± 0.163**
	GP-ConvCNP	**1.632 ± 0.079**	**−0.532 ± 0.044**
Recon. Error↓	GP (EQ)	*0.004 ± 0.001*	*0.097 ± 0.001*
	NP	0.845 ± 0.074	0.292 ± 0.010
	ANP	0.181 ± 0.018	0.284 ± 0.013
	ConvCNP	**0.042 ± 0.027**	**0.121 ± 0.017**
	GP-ConvCNP	**0.040 ± 0.023**	**0.116 ± 0.017**

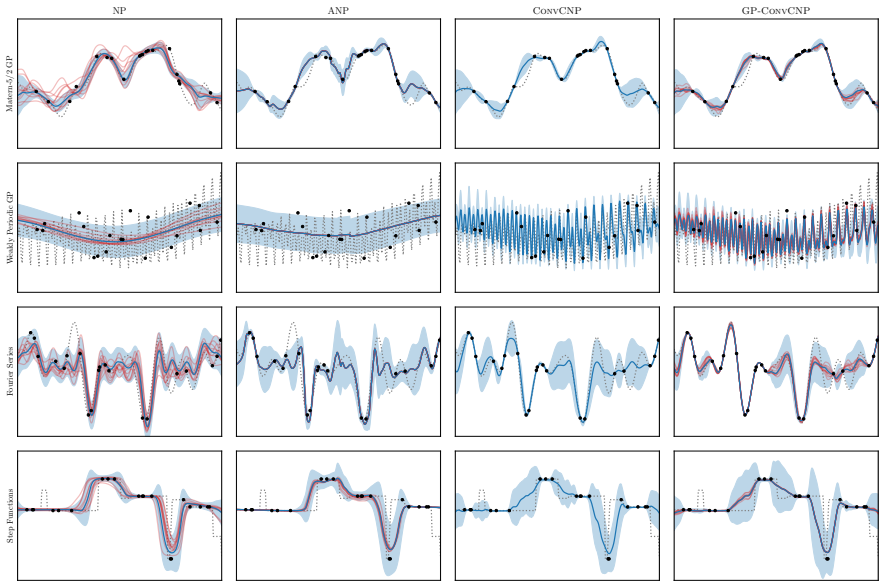

Figure 7.2: Examples for the different synthetic functions and methods evaluated in this work (mean prediction in blue, samples in red). Overall, the predictions from ConvCNP and GP-ConvCNP look very similar and significantly better than those from NP and ANP (with the exception of step functions, perhaps). NP and ANP are unable to fit high frequency signals and especially NP fits the context points (black dots) rather poorly. NP offers high sample diversity, while there is very little for ANP. GP-ConvCNP combines the high quality predictions of ConvCNP with the ability to sample. The variability of those samples depends of course on the data and the density of context points.

able to. The sample diversity in GP-ConvCNP is larger than in ANP, but samples are only significantly different from the mean prediction when further away from the context points in areas of high predictive uncertainty (shaded areas correspond to 1σ). In contrast, samples from the NP are more diverse throughout, at the expense of accurately matching the context points.

7.4.2 *Weather Time Series*

Examples from the temperature time series dataset can be seen in Figure 7.3. The key characteristic of the signal is the temperature change between day and night, making it a high frequency signal not unlike the weakly periodic GP samples in the synthetic dataset. NP and ANP were not able to fit these signals, so we don't show them here. The left side of Figure 7.3 shows an example of the regular interpolation task, the right side an example of the extrapolation task, which we deem an important aspect of generalization. The example was selected to show how our model improves over competing approaches. While ConvCNP and GP-ConvCNP are both able to interpolate the data well,

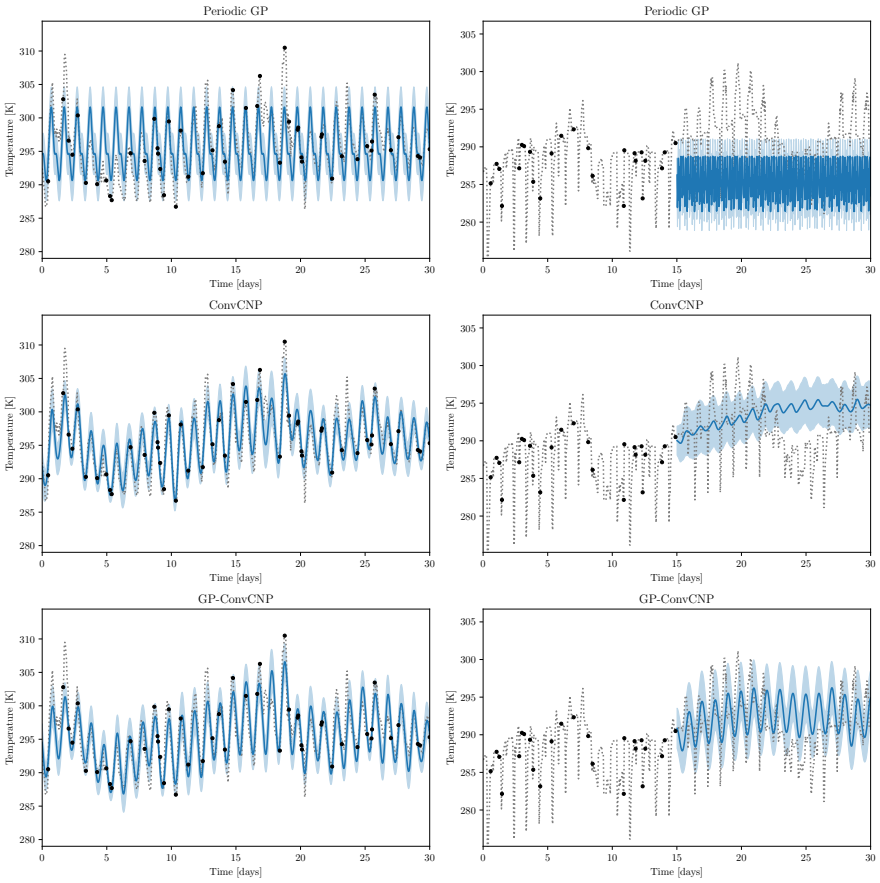

Figure 7.3: Examples from the temperature time series test set. For the interpolation task (left panel for each method) we provide context points from the full sequence, for the extrapolation task (right panel for each method) we provide context points in the first half of the sequence and evaluate the second. Both ConvCNP and GP-ConvCNP capture the periodicity of day/night changes in temperature well and are able to extrapolate it. However, GP-ConvCNP better matches the amplitude of the true signal, which likely explains its superior performance in Table 7.2. A periodic GP (ExpSineSquared kernel) can interpolate the signal with the correct frequency, but often has trouble extrapolating it. Note that the example was selected to show how a GP and ConvCNP fall short. In many cases, the predictions from ConvCNP and GP-ConvCNP are very similar. NP and ANP were unable to fit the data, similar to the weakly periodic GP data in Figure 7.2 and the results are not shown.

Table 7.2: Results on the temperature time series dataset. ↑/↓ indicate that higher/lower is better and errors represent 1 standard deviation over 5 runs with different seeds (except for the periodic GP, where the seed has negligible influence, so we show the standard error of the mean instead). For the temperature interpolation task (left), context points are randomly sampled from the test interval, for the temperature extrapolation task (right) we provide context points in the first half of the interval and measure performance on the second half (as seen in Figure 7.3). While ConvCNP and GP-ConvCNP perform roughly on par for the interpolation task, with slightly better prediction for GP-ConvCNP and slightly better reconstruction for ConvCNP, the difference between the two increases for the extrapolation task, where GP-ConvCNP now significantly outperforms all other methods. Somewhat surprisingly, the periodic GP seems to fail completely at the extrapolation task. We assume this is because of the small number of context points that often leads to poor estimation of the correct frequency, as seen in Figure 7.3.

		interpolation	extrapolation
Predictive LL↑	GP (per.)	*−2.075 ± 0.237*	*−46.611 ± 2.557*
	NP	−0.855 ± 0.003	−1.267 ± 0.011
	ANP	−0.733 ± 0.008	−1.938 ± 0.381
	ConvCNP	**−0.522 ± 0.008**	−1.261 ± 0.062
	GP-ConvCNP	**−0.515 ± 0.019**	**−1.190 ± 0.016**
Recon. Error↓	GP (per.)	*0.274 ± 0.001*	
	NP	0.238 ± 0.002	
	ANP	0.198 ± 0.007	
	ConvCNP	**0.106 ± 0.002**	
	GP-ConvCNP	**0.123 ± 0.018**	

Table 7.3: Results on the population dynamics data. ↑/↓ indicate that higher/lower is better and errors represent 1 standard deviation over 5 runs with different seeds. Models are trained on simulated data, so the real world data (also shown in Figure 7.4) is likely out-of-distribution, as evidenced by the stark drop in performance. This drop is by far the smallest for GP-ConvCNP, performing significantly better than all competing methods in terms of predictive log-likelihood.

		simulated	real
Predictive LL↑	NP	0.527 ± 0.051	-33.070 ± 7.636
	ANP	1.027 ± 0.033	-29.714 ± 9.210
	ConvCNP	$\mathbf{1.374 \pm 0.017}$	-23.540 ± 12.441
	GP-ConvCNP	1.337 ± 0.029	$\mathbf{-5.382 \pm 2.625}$
Recon. Error↓	NP	0.018 ± 0.001	1.053 ± 0.015
	ANP	0.008 ± 0.004	0.772 ± 0.020
	ConvCNP	$\mathbf{0.002 \pm 0.001}$	$\mathbf{0.374 \pm 0.019}$
	GP-ConvCNP	0.004 ± 0.001	$\mathbf{0.411 \pm 0.026}$

ConvCNP sometimes underestimates the amplitude of the signal in the extrapolation task. In many cases, however, both it and our model produce very similar extrapolations. This is reflected in Table 7.2, where the difference between GP-ConvCNP and ConvCNP is not spectacularly large, but still significant. We also show the performance of a periodic Gaussian Process with an ExpSineSquared kernel, which is able to estimate the correct frequency on the interpolation task, but often fails on the extrapolation task, likely because of the lower number of context points. In either case, it is unable to model the finer variations in the signal.

7.4.3 *Population Dynamics*

To measure how robust the different members of the Neural Process family are to a distribution shift at test time, we train models on population dynamics simulated as Lotka-Volterra processes, and evaluate performance both on simulated (*in-distribution*) and real world (*out-of-distribution*) data. The real world dataset, along with a simulated example, can be seen in Figure 7.4. While both ConvCNP and GP-ConvCNP fit the simulated data well, they struggle with the test interval on the real data. This is reflected in Table 7.3 as well, where we find that ConvCNP performs better than GP-ConvCNP (even significantly so, albeit not with a huge difference) on the simulated data.

Applied to the real world dataset, all methods experience a large drop in performance, indicating that this is indeed a significant distribution shift. GP-ConvCNP is by far the best performing method here,

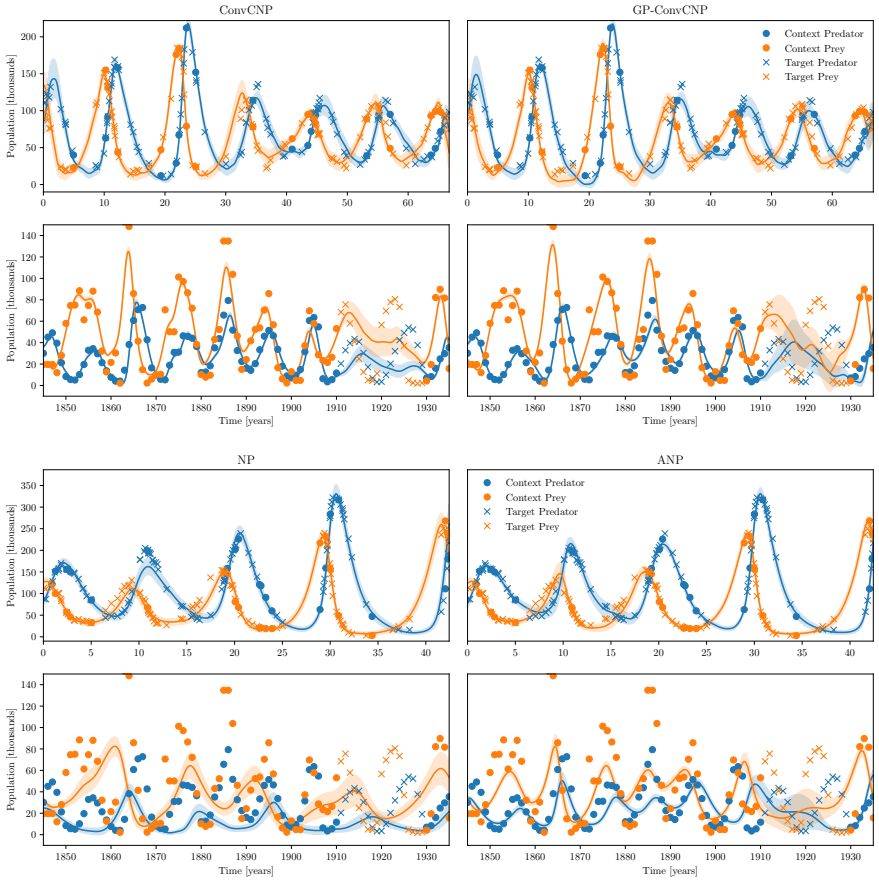

Figure 7.4: Example of ConvCNP and GP-ConvCNP (top half) as well as NP and ANP (bottom half) applied to the simulated Lotka-Volterra population dynamics (top panel for each method) and to the real Hudson Bay Company lynx-hare dataset (bottom panel for each method). Both ConvCNP and GP-ConvCNP perform well on the simulated (i.e. in-distribution) data and seem to struggle fitting the test interval on the real world data. Note however how the predicted uncertainty is larger for GP-ConvCNP. We display the best out of 5 models in each case, and for ConvCNP the performance is much more volatile, as evidenced by the results in Table 7.3. NP and ANP perform very poorly on the real world data.

Table 7.4: This table presents the same as the rightmost column of Table 7.3, but using a different evaluation mode. In Table 7.3, the evaluation was performed as seen in Figure 7.4, meaning one contiguous interval on the data was selected as the target region and the rest of the data is provided as context, following Gordon et al. (2020). Here we instead sample the context and target points randomly from the entire interval, like we do in the other experiments as well. For each seed, we average over 100 random draws and report the standard deviation over 5 seeds as errors. While ConvCNP maintains leading performance in terms of reconstruction error, GP-ConvCNP significantly outperforms the other methods in predictive performance, similar to what we found in Table 7.3. However, all methods perform worse compared to the evaluation method used in Table 7.3.

	Predictive LL↑	Recon. Error↓
NP	-36.735 ± 4.137	0.952 ± 0.024
ANP	-38.717 ± 3.572	0.718 ± 0.018
ConvCNP	-28.762 ± 1.958	$\mathbf{0.272 \pm 0.008}$
GP-ConvCNP	$\mathbf{-19.252 \pm 1.846}$	0.343 ± 0.020

which is likely because of a better estimate of the preditive uncertainty. Note how the uncertainty predicted by ConvCNP is smaller than that of GP-ConvCNP in Figure 7.4 (the figure shows 1σ). The predictions we show here are from the best performing seed in each case, other ConvCNP models predicted an even narrower distribution. Why did we select this particular interval for testing? It's the same interval Gordon et al. (2020) show in the ConvCNP paper. We also evaluated with context points drawn randomly from the entire interval (i.e. the same way we evaluate on the simulated data), and GP-ConvCNP still performs significantly better than the competing approaches (see Table 7.4).

7.5 RELATED WORK

Neural Processes have inspired a number of works outside of the ones we discuss. Louizos et al. (2019) propose to not merge observations into a global latent space, but instead learn conditional relationships between them. This is especially suitable for semantically meaningful clustering and classification. Singh et al. (2019) and Willi et al. (2019) address the problem of overlapping and changing dynamics in the generating process of the data, a special case we do not include here. With a simple Gaussian kernel, we wouldn't expect our model to perform well in that scenario, but one could of course introduce inductive bias in the form of e.g. non-stationary kernels, when translation equivariance is no longer desired. NPs have also been scaled to extremely complex output spaces like in *Generative Query Networks* (Eslami et al.,

2018; Kumar et al., 2018; Rosenbaum et al., 2018), where a single observation is a full image. GQN directly relates to the problem of (3D) scene understanding (Jimenez Rezende et al., 2016; Sitzmann et al., 2019; Engelcke et al., 2020).

Gordon et al. (2020) build their work (ConvCNP) upon recent contributions in the area of *learning on sets*, i.e. neural networks with set-valued inputs (Zaheer et al., 2017; Wagstaff et al., 2019), which has mostly been explored in the context of point clouds (Qi et al., 2017b; Qi et al., 2017a; Wu et al., 2019). Especially the work of Wu et al. (2019) is closely related to Gordon et al. (2020), also employing a CNN on a kernel density estimate, but their application is not concerned with time series. *Bayesian Neural Networks* (Neal, 1996; Graves, 2011; Hernández-Lobato and Adams, 2015) also address the problem of learning distributions over functions, but often implicitly, in the sense that the distributions over the weights are used to estimate uncertainty (Blundell et al., 2015; Pawlowski et al., 2017; Gal and Ghahramani, 2016b). We are interested in this too, but in our scenario we want to be able to condition on observations at test time.

The main limitation of *Gaussian Processes* is their computational complexity and many works are dedicated to improving this aspect, often via approximations based on inducing points (Snelson and Ghahramani, 2006; Titsias, 2009; Gardner et al., 2018; Wilson and Nickisch, 2015) but also other approaches (Deisenroth and Ng, 2015; Rahimi and Recht, 2007; Le et al., 2013; Cheng and Boots, 2017; Hensman et al., 2013; Hensman et al., 2015; Salimbeni et al., 2018), even for exact GPs (Wang et al., 2019b). Rather than competing with these approaches, our model will be able to leverage developments in this area. Some of the above try to find more efficient kernel representations and are thus closely related to the idea of *kernel learning*, i.e. the idea to combine the expressiveness of (deep) learning approaches with the flexibility of kernel methods, for example Yang et al. (2015), Wilson et al. (2016b), Wilson et al. (2016a), Tossou et al. (2019), and Calandra et al. (2016). The key difference to our work is that these approaches attempt to learn kernels as an input to a kernel method, while we learn to make the output of a kernel method more expressive.

7.6 DISCUSSION

We have presented a new model in the Neural Process family that extends ConvCNP by incorporating a Gaussian Process into it. We show on both synthetic and real time series that this improves performance overall, but most markedly when generalization is required: our model, GP-ConvCNP, can better extrapolate to regions far from the provided context points and is more robust when moving to real world data after training on simulated data. We further retain translation equivariance, a key feature of ConvCNP, as long as a stationary

kernel is used for the GP. The introduction of the latter also allows us to draw multiple samples from the model, where the distribution of samples from our model better matches the samples from an oracle than those from a regular Neural Process or an Attentive Neural Process do. Our model uses the prediction from a GP with an EQ-kernel as an initial estimate. Interestingly, this estimate needn't be very good, but when it is, our model can fully leverage it and even match the performance of an oracle, as seen in Table 7.1.

Of course, with the benefits of GPs we also inherit their limitations. GPs are typically slow, naively requiring $O(N^3)$ operations in the number of context observations. However, making GPs faster is a very active research area, as outlined in Section 7.5. We expect that our model is well suited to also work with approximate methods, as we modify the prediction from the GP with a powerful neural network. As we focused on time series or generally one-dimensional input spaces in this work, speed was not an issue. In general, scaling GP-ConvCNP to much larger problems presents an exciting opportunity for future work. While NPs have been scaled to impressive size, e.g. Eslami et al. (2018), Kumar et al. (2018), and Rosenbaum et al. (2018), where observations are entire images, it remains to be shown that the same works for ConvCNP and our model. For our model specifically it seems reasonable to leverage work on deep kernels (Wilson et al., 2016b) or to learn mappings before the GP prediction like in Calandra et al. (2016) in order to learn more meaningful GP posteriors that capture information about the training distribution.

We also tried to apply ConvCNP and GP-ConvCNP to the tumor volume prediction task, similar to NP and ANP in Section 5.3, but found both performed rather poorly. We suspect this is because of the extremely small number of context points. Gordon et al. (2020) also comment in their work that ConvCNP is best suited for scenarios where training data is abundant, for example *sim2real* tasks. While we also test on real world data—the temperature time series—, one of our selection criteria was that a large amount of training data be available. We leave the exploration of how much data these two approaches require, and how to make them more robust in a sparse setting, for future work. Because we could not successfully apply ConvCNP and our extension of it to our glioma growth dataset, we return to regular Neural Processes and Attentive Neural Processes for our next chapter.

8

ATTENTIVE SEGMENTATION PROCESSES

We saw in Chapter 4 that a probabilistic segmentation architecture can in principle be used to model glioma growth, learning the growth dynamics entirely from a data distribution. However, this approach is limited to input sequences of fixed length and equidistant spacing in time. In Chapter 5 and Chapter 6 we then saw that Neural Processes are a powerful tool to learn distributions over functions from data, such that predictions can be conditioned on observations available at test time and at arbitrary continuous-valued times. In this chapter, we will extend Neural Processes to segmentation tasks, both for general purposes and for modeling glioma growth. We show that a U-Net-like segmentation architecture (Ronneberger et al., 2015), i.e. an encoder-decoder structure with skip connections, can be made to perform as a Neural Process by introducing *spatial attention*. Compared to a regular Neural Process, meaning an encoder-decoder structure *without* skip connections, this allows the approach to model both global changes and finer variations. The work in this chapter is yet to be published.

8.1 INTRODUCTION & RELATED WORK

Neural Processes, which we introduced in detail in Section 6.2, can learn function spaces on a continuous domain. We now seek to do the same for *segmentation functions*, so it is only natural to try to apply Neural Processes in this context. As we will see, this is not quite as straightforward as it might sound. Neural Processes need to encode information from context observations into a *representation space*, from which new observations can be generated. This typically requires them to compress information significantly, leading to the loss of fine detail like higher frequency content, as we could show in Chapter 6. For segmentations this results in smooth predictions with little spatial variation, as seen for example in Figure 8.2. Compared to the original Neural Processes (Garnelo et al., 2018b; Garnelo et al., 2018a), Attentive Neural Processes (ANP) (Kim et al., 2019) alleviate the need to compress all context information into a single point in representation space, instead allowing a dynamic combination of multiple points for a given prediction via an attention mechanism. This greatly improves performance, but is still insufficient in some cases, as we show in our experiments below. Segmentation architectures typically use an encoder-decoder pattern with skip connections (see Section 2.3 for an introduction), which allows networks to retain very fine detail. We show that skip connections can be interpreted as separate represen-

tation spaces, but to make full use of the skip connections, attention based on time values alone is insufficient. Instead, we introduce *spatial attention* between individual spatio-temporal locations and term the result *Attentive Segmentation Processes*. These models show remarkably strong performance on a variety of tasks.

We have already discussed prior art related to Neural Processes in Section 6.3. There, we also mentioned the fact that Neural Processes have been succesfully applied to image-valued observation spaces in the form of Generative Query Networks (GQN) (Eslami et al., 2018; Kumar et al., 2018). These are Neural Processes (or Attentive Neural Processes in the case of follow up work by Rosenbaum et al. (2018)) with encoders similar to the ones we use here, but extremely large autoregressive decoders (Gregor et al., 2015). The authors show that they can predict unseen views of 3D scenes from context observations given at e.g. different angles, producing images with fine detail. This will seem at odds with our findings that Neural Processes struggle to produce segmentations with such high resolution detail. The difference is simply the size (in terms of number of parameters) of the decoder. In our work, we only evaluate models of comparable size. We have in fact published an open source implementation of GQNs[1] and found that to reproduce the simplest experiment from the reference publication, a single model needed to be trained for two weeks on a TITAN Xp GPU. For the toy experiments in this chapter, such compute requirements are beyond the scope of feasibility. Furthermore, the larger number of parameters in GQN requires a vastly larger amount of training data, precluding them from being applied to the glioma growth data in this chapter. So far, GQNs have never been demonstrated on any real world data. These compute and data requirements have also been recognized in later work in the context of 3D scene understanding, where researchers have replaced the learnable decoder with differentiable renderers (Sitzmann et al., 2019; Dupont et al., 2020; Mildenhall et al., 2020).

Prior art in the context of learning distributions of segmentation functions was already discussed in Section 4.1. As we saw there, no existing work is able to perform prediction conditioned on context observations on a continuous time axis, hence our desire to create such an approach in this chapter. We achieve this using attention, specifically scaled dot-product attention (Vaswani et al., 2017), which is by far the most commonly used attention mechanism. Blocks of repeated application of such a mechanism are often called *transformers*, and have been applied to image data in some exploratory work. Parmar et al. (2019) show that attention can be used to replace convolution layers in common image processing architectures to achieve the same performance at a lower parameter count, based on their earlier proof-of-concept work in Parmar et al. (2018). In Chen et al.

1 https://www.github.com/jenspetersen/gqn-pytorch

(2020a), the authors interpret images as pixel sequences and apply a powerful sequence transformer model, showing off impressive and diverse image completions. Carion et al. (2020) show that transformers can be used for object detection. To the best of our knowlege, there is no existing work that uses attention/transformers to learn distributions of segmentation functions. There is work that proposes an *Attention U-Net* for segmentation (Oktay et al., 2018), but the authors use a different definition of attention that is more commonly known as *gating*, which is only broadly related to our work.

8.2 METHODS

Our goal is to enable Neural Processes (Garnelo et al., 2018a; Garnelo et al., 2018b) to perform segmentation interpolation. A detailed introduction of the Neural Process framework is given in Section 6.2, so we only briefly repeat it here. Models of the family form a representation of some function space $\mathcal{F} = \{f_i\}, f_i : X \to Y$ by observing examples presented as context observations $C = \{(x_c, y_c)\}_{c=1}^{N} =: (\boldsymbol{x_c}, \boldsymbol{y_c})$ and target observations $T = \{(x_t, y_t)\}_{t=1}^{M} =: (\boldsymbol{x_t}, \boldsymbol{y_t})$. In Chapter 6 and Chapter 7, we restricted ourselves to low-dimensional scenarios, meaning $Y = \mathbb{R}$ or $Y = \mathbb{R}^D$ with a small dimension D. We now wish to predict segmentations, meaning $Y = [0; L]^{H \times W}$, where $H \times W$ are the spatial dimensions, L the number of labels (excluding background) and $[0; L]$ an integer interval. We further consider two scenarios: in the first, the input space is scalar like before, meaning $X = \mathbb{R}$. This means predicted segmentations have to be interpolated entirely from context observations given at test time. In the second scenario, we assume we also have input images available at the desired target locations, such that $X = \mathbb{R} \otimes \mathbb{R}^{\nu \times H \times W}$, ν being the number of channels in the input images. To understand the approach we propose in this chapter, it is best to begin with the implementation of Neural Processes for image-valued observations.

8.2.1 *Neural Process Implementation*

In Chapter 6 we implemented the encoder $E : C \to Z$ as an MLP that encodes each (x, y) pair into the representation space. Here we need to choose an architecture that is more suitable for image inputs, and a commonly used one consists of multiple (Conv, Activation, Conv, Activation, Pool)-blocks, optionally with additional instance or batch normalization layers, which we found to be unnecessary for our experiments. We use leaky ReLU activations and average-pooling with kernel size 2, meaning the spatial resolution is halved after each block. The convolutional layers work with a kernel size of 3, and the number of feature maps is doubled from the first to the second Conv-layer, while the first has the same number of feature maps as the second one

from the previous block, starting with 12 maps at the input. We stack 4 of these blocks, and the last one replaces the average-pooling of size 2 with global average pooling to remove all spatial resolution. A final convolution layer maps to the desired representation size, which is 128 in our case. This architecture is illustrated in Figure 8.1 on the left hand side.

The decoder $G : Z, X \to Y$ mirrors the encoder, with average-pooling replaced by linear interpolation to *increase* the spatial resolution, initially to 8×8 and then by a factor of 2 to match resolution of the encoder. The number of feature maps is halved instead of doubled between consecutive Conv-layers. Again, this pattern is illustrated in Figure 8.1. Note that this implementation *does not* use the skip connections depicted in the figure. If we just have target locations $\mathbf{x_t}$[2], the decoder does in fact work as $G : Z, X \to Y$, however, when we also have input images at the target locations, we encode these into another representation X' using an encoder that mirrors the context encoder, only adjusting the input channels to match the number of channels in the input image. The decoder then operates on the domain $G : Z, X, X' \to Y$. In CNP, the representations $\mathbf{r_{g_c}}$ (g for global scale as opposed to spatial as seen in Figure 8.1) of the individual context points are summed to form a global representation r_g (we take *global* to mean both non-spatial as well as over all context points, the meaning will be clear from context). In ANP, an attention mechanism is learned that aggregates the information:

$$r_g(\mathbf{x_t}) = \mathrm{att}(\mathbf{x_t}, \mathbf{x_c}, \mathbf{r_{G_c}}) \quad \text{or} \tag{8.1}$$

$$r_g(\mathbf{x_t}) = \mathrm{att}\big((\mathbf{x_t}, \mathbf{x'_t}), (\mathbf{x_c}, \mathbf{x'_c}), \mathbf{r_{g_c}}\big) \tag{8.2}$$

where the second line represents the case when input images are available. In practice, $\mathbf{x_t}$ and $\mathbf{x'_t}$ are just concatenated. The attention mechanism we use is multi-head scaled dot-product attention (Vaswani et al., 2017) with 8 heads and an embedding size of 128. A single head is defined by:

$$\mathrm{att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V} \tag{8.3}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ (short for query, key, value) are now matrices with each row representing one context/target item after applying learnable linear projections to $\mathbf{x_t}, \mathbf{x_c}, \mathbf{r_{g_c}}$. d is the dimension, i.e. the number of columns, of Q and K. For multi-head attention, several of these heads are applied, their outputs concatenated, and the result projected to a space of the desired dimensionality.

2 Like before, bold indicates a *set* of values

8.2.2 *Neural Processes with Skip Connections*

We mentioned above that segmentation architectures use skip connections to allow themselves to work on multiple scales of spatial resolution, so that their predictions incorporate both global information and fine localized detail. How can we enable Neural Processes to do the same? One way to do this would be to interpret the outputs of the encoder along those skips as additional representation spaces with a spatial resolution. This idea is depicted in Figure 8.1. The representations then become tuples of sets $(\mathbf{r}_{-K_c}, \ldots, \mathbf{r}_{-1_c}, \mathbf{r}_{g_c})$, where we choose backwards indexing so that the lowest skip connection above the global representation always has the same index regardless of the network's depth K. Just like in CNPs, we can just average the individual representations at each scale before feeding them to the decoder along with the target inputs $\mathbf{t}_t$. We call the resulting models *Segmentation Processes* (SP). Note that by doing so, there is no information exchange between the individual spatial locations in a given representation. Each pixel vector learns its own representation akin to $\mathbf{r}_g$, and it will turn out that summation works quite poorly in this setting. Above we referred to elements of the input space as $x \in X$; because representations now have a spatial resolution, we switch to more descriptive symbols, using $t \in X$ (=time) for the input space and letting x, y describe spatial locations.

We can of course also define ANPs in the same way. Instead of averaging the representations, we can learn an attention mechanism for each skip connection to combine context representations. In our implementation, we only use a single head of attention as described in Equation (8.3) for each skip connection, and perform no additional projection on the representation values, to save some computational expenses. We will find that attention along the skip connections works much better than summation, so whenever we mention skip connections in our experiments, they will use attention, unless the model is explicitly referred to as SP.

8.2.3 *Attentive Segmentation Processes*

Having established how we can employ a segmentation model with skip connections in the Neural Process framework, could there be potential factors that limit performance in this scenario? As we alluded to above, there is no spatially varying information flow in CNPs or ANPs with skip connections, regardless of whether averaging or attention is used to merge the individual context representations. In other words, the aggregate representation r_{-i} at some spatial location (x_t, y_t) only depends on the context representations *at that same location*. From an intuitive perspective, we would expect that the representation at (x_t, y_t) should also be influenced by different locations $(\mathbf{x_c}, \mathbf{y_t})$ in the

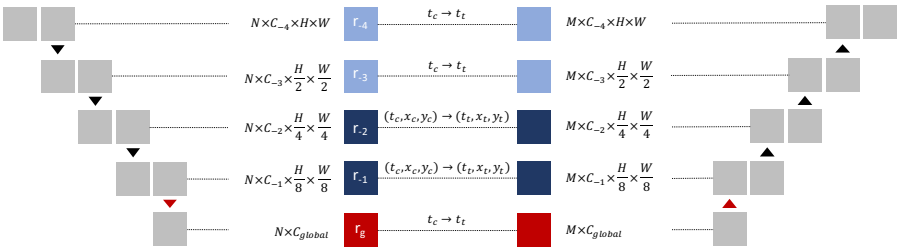

Figure 8.1: Illustration of the Attentive Segmentation Process architecture. An encoder-decoder setup with Conv-Activation pairs (grey blocks, decoder uses linear interpolation for upsampling) is augmented with skip connections, leading to a U-Net-like architecture. Along the higher skip connections, as well as the global representation, we aggregate the context for a given target point using only time information. At the coarser skip connections, we introduce spatial attention, meaning the representation for a target point (t_t, x_t, y_t) is constructed from all locations (t_c, x_c, y_c) available in the context representations. Indicated next to the representations are the dimensions of their tensors.

context, and in a way that depends on the individual values of (x_t, y_t) and $(\mathbf{x_c}, \mathbf{y_t})$. More formally:

$$\text{regular attention:} \quad r_{-i}(t_t, x_t, y_t) = f\big(\mathbf{r}_{-\mathbf{i_c}}(\mathbf{t_c}, x_t, y_t)\big) \tag{8.4}$$

$$\text{spatial attention:} \quad r_{-i}(t_t, x_t, y_t) = f\big(\mathbf{r}_{-\mathbf{i_c}}(\mathbf{t_c}, \mathbf{x_c}, \mathbf{y_c})\big) \tag{8.5}$$

where f is some mechanism that facilitates the aggregation of the individual $\mathbf{r}_{-\mathbf{i_c}}$. Because of the success of attention in ANPs compared to CNPs, we choose to implement f as an attention mechanism as well, and we call the resulting models *Attentive Segmentation Processes* (ASP). Note that spatial attention requires no additional model parameters compared to regular attention that only uses the input locations $\mathbf{t_t}, \mathbf{t_c}$. However, the dot-product in Equation (8.3) will result in much larger matrices that require correspondingly more GPU memory to store. To counteract this, we only implement spatial attention in the lowest N skips connections, and optionally include regular attentive skip connections at higher levels. The abbreviation ASP_N will refer to an Attentive Segmentation Process with spatial attention in the deepest N skip connections. ASP_0 thus becomes a regular ANP. Figure 8.1 depicts an illustration of ASP_2 with skip connections.

In many scenarios where attention is used, there is no intrinsic concept of queries ($\mathbf{t_t}, \mathbf{x_t}, \mathbf{y_t}$ in our case) and keys ($\mathbf{t_t}, \mathbf{x_t}, \mathbf{y_t}$ in our case). For language modeling, transformers are extremely successful—see e.g. the recently published GPT-3 (Brown et al., 2020) which has a rather extensive overview of related work. These models use a concept called *self-attention*, which simply means that the inputs to the projections that produce Q, K and V in Equation (8.3) are identical. We do something similar in our case: while the inputs for

the V-projections are just the context representations, we not only use $\mathbf{t_c}, \mathbf{x_c}, \mathbf{y_c}$ as the key inputs, but instead concatenate them with the context representations as well. As input for the query projections, we either use just the targets $\mathbf{t_t}, \mathbf{x_t}, \mathbf{y_t}$ or, in case we have input images available at the target locations, a concatenation of them with the representations from the target encoder, which are of the same shape as the context representations.

8.2.4 *Variational ASP and Optimization*

So far, everything we described above relates to deterministic models, which we can train by directly optimizing the likelihood of the ground truth data under a predicted distribution:

$$\max_{\theta} \log p_\theta(\mathbf{y_t}|\mathbf{x_t}, \mathbf{x_c}, \mathbf{y_c}) \tag{8.6}$$

In Chapter 4, where we also modeled segmentations, we implemented p_θ as a categorical distribution, which results in a cross entropy loss as the minimzation target. Unfortunately, in the experiments in this chapter, we found training with only cross entropy to be very unstable. Instead, we use the sum of cross entropy and Dice loss (Milletari et al., 2016), a common choice for segmentation problems. The Dice loss, sometimes also called soft Dice, is a differentiable formulation of the Dice overlap and does not implicitly assume that the target distribution factorizes across individual pixels. It typically results in more spatial consistency in predicted segmentations. The full loss for deterministic models thus becomes:

$$\text{Loss} = \frac{1}{N_{pixels}} \sum_{l=0}^{L} \sum_{x,y} -S_l \log p_\theta(l) \quad + \tag{8.7}$$

$$\sum_{l=0}^{L} \frac{-\sum_{x,y} 2 S_l p_\theta(l)}{\sum_{x,y} \left(2 S_l p_\theta(l) + S_l(1 - p_\theta(l)) + (1 - S_l) p_\theta(l)\right)} \tag{8.8}$$

where l sums over all labels (including background) and x, y over the individual output pixels. $p_\theta(l)$ is the predicted softmax-probability for label l and S is a one-hot encoding of the segmentation. Both have a dependence on x, y that we drop for better readability. We use the Adam optimizer (Kingma and Ba, 2015) with an initial learning rate of 1×10^{-4} that is decayed with a factor of $\gamma = 0.98$ after 1000 iterations. We train for a total of 200 000 of batches with batch size 32 (128 for the glioma growth experiments).

We will find in our experiments that none of the models we compare can actually predict glioma growth in any meaningful way. This supports the hypothesis of our work in Chapter 4, where we argue that

one should focus on modeling multiple growth trajectories instead of trying to predict a single one. Our proposed ASP models can naturally be extended akin to variational Neural Processes (Garnelo et al., 2018b) or the variational formulation of ANPs (Kim et al., 2019), which allows them to produce multiple, globally consistent growth trajectories. For variational ASPs, we let the encoder predict mean and variance of a Gaussian distribution at the global scale, which we average to form a global distribution. The representations with a spatial resolution along the skip connections remain deterministic. The optimization target then becomes, as seen in Section 6.2.2:

$$\max_{\theta} \mathop{\mathbb{E}}_{z\sim q_\theta(Z|\mathbf{x_t},\mathbf{y_t})} \log p_\theta(\mathbf{y_t}|\mathbf{x_t}, z) - \lambda \cdot D_{KL}(q_\theta(z|\mathbf{x_t},\mathbf{y_t})||q_\theta(z|\mathbf{x_c},\mathbf{y_c})) \tag{8.9}$$

where the first term is the (negative) loss of Equation (8.8) for which we take the expectation using a single sample during training. The second term is the KL divergence between the prior and the posterior. We introduced a factor λ, because the original formulation of this so-called *evidence lower bound* (ELBO) assumes a factorized target distribution (like when we only use cross entropy). The Dice loss breaks this assumption, so the KL divergence will implicitly have a larger weight which we need to balance. We found that $\lambda = 1 \times 10^{-4}$ results in latent losses in the same order of magnitude as what we observed in Chapter 4, so we fix it to this value.

The prior is the distribution predicted by the encoder at the global scale when presented with the context observations. The posterior is the same distribution, conditioned on both context and target observations. Note that prior and posterior only refer to the representation at the global scale, i.e. the bottom of the U-Net. During training, we will first encode the context representations, which includes the prior. We then encode the context and target points together to obtain the posterior, discarding all representations with a spatial resolution. We then sample from the posterior and combine the sample with the context representations at the non-global scales as input to the decoder.

8.3 EXPERIMENTS & EVALUATION

8.3.1 *Toy Examples*

There are two capabilities we consider essential in the context of learning distributions of functions that perform segmentation. One is the ability to interpolate between context observations, inferring global changes like the position or size of objects, but at the same time retaining fine detail. We refer to this as *segmentation interpolation*. As

we will see in the following section, models that encode information into a global representation and generate new observations from there (meaning a typical encoder-decoder structure without skip connections) generally struggle to produce fine detail, and will result in very smooth segmentations unless a very large and computationally expensive decoder is used like in Eslami et al. (2018) and Kumar et al. (2018). We test this by constructing a toy task where a shape moves on a random linear trajectory through a fixed frame of size 64^2, changing size and orientation along the way. The shape we use is a star shape, which can also change its "pointiness", i.e. the difference between the radius of outer points vs. the radius if inner points. An example of this can be seen in Figure 8.2. The end points of the trajectory are assigned values $t_{start} = 0$ and $t_{end} = 1$. We sample start and end values for the size, orientation and pointiness randomly and interpolate them linearly along the trajectory. We then draw C context points and T target points along it, with $C \in [3; 9]$ and $T \in [10; 24]$. We measure the segmentation performance using the Dice overlap.

The second capability we seek in models is the combination of information provided at test time via context segmentations and information available in input images available for target locations. For lack of a better name, we refer to this as *generalization* below. To test this, we construct a toy task as follows: we start with an empty, i.e. zero-valued, image volume of size 64^3. We then place a random ellipsoid, i.e. an ellipsoid with random position and random major semi-axis values in the volume, filling it with a grey value sampled from a standard normal. This process is repeated 10 times, while later ellipsoids will usually overwrite parts of the already placed ones, to obtain a decently random image volume with some structure. Finally, we add Gaussian noise, the variance of which is sampled from $[0; 1]$. From the input volume we obtain a corresponding segmentation by splitting the grey value range into 10 bins of equal size. To remove any fixed correspondence between grey value ranges and classes, the class assignment to the bins is also random. We then provide models with C context slices and let them segment T target slices, with C and T equal to the above values. An example for this task can be seen in Figure 8.3. We measure performance in terms of Dice, but because we find that some models tend to produce many small disconnected regions, we also measure the absolute difference between the correct number of connected components and the number of connected components in the predicted segmentation, which we call the "#Objects Error".

8.3.2 *Interactive Segmentation*

The second toy example can be interpreted as a form of interactive segmentation, where a user annotates a number of slices to produce a segmentation for the entire image volume. The toy task is relatively

easy to solve, so in order to test the same concept on a more challenging task we apply our model to interactive brain tumor segmentation, using the dataset we have already employed in Chapter 4 and Chapter 5. The input data has 4 channels: T1, T1c, T2, FLAIR and the available segmentations contain labels for edema, enhancing tumor and the combination of necrosis and non-enhancing tumor. We split the dataset into 5 random subsets (on a patient level to ensure there is no data leakage) and generate predictions by training on 4 subsets and predicting the last. This task can be solved quite well by directly segmenting the input images, so we shuffle the classes for each example like before, and we qualitatively observe if the our approach is still able to extrapolate class information from context segmentations to target slices. Because our model is relatively expensive in terms of GPU memory requirements, we first downsample the input volumes to a resolution of 128^3 (from 192^3) and then extract patches of size 64^3 centered around the tumor. Individual slices are sampled along the axial/transversal orientation, and C and T are the same as in the toy examples. For the context we ensure it always contains the slice through the tumor center as well as its limiting slices.

8.3.3 *Tumor Growth Modeling*

We motivate this thesis using glioma growth as an example where we would like to learn a distribution of functions by observing examples. In Chapter 4 we applied a Probabilistic U-Net, which is a conditional VAE combined with a U-Net segmentation architecture, to this task, but the approach is limited to a fixed number of observations that are required to have a fixed spacing in time. Our approach can perform segmentation interpolation on a continuous time axis, so naturally we wish to find out if it can also be applied to glioma growth modeling. We use the same data as above, meaning we work with volumes of size 64^3 centered around the tumor, but because working in 3D+t is too computationally demanding, we treat each axial slice in a volume as a separate entity, reducing the problem to 2D+t. We again split the full datasets into 5 subsets on a patient level and create predictions for each subset by training on the other four. We sample random sequences with a length between 3 and 6 and provide all but the last time point as context. Models are trained to predict the future time point and also reconstruct the context. Because we found in Chapter 4 that there was no difference between using image volumes or segmentations as input, we choose to work directly in segmentation space and discard the MRI scans entirely. We also add a random shift between -100 days and 100 days to the time values to artificially increase the amount of available data.

As a first experiment, we essentially reproduce the volume regression experiments from Chapter 5 and see how well different models

can predict a true future segmentation from a varying number of input observations. Following this, we qualitatively inspect different models and their ability to model complex spatial growth patterns. We will find that there is negligible predictive value in all models, so we also test the variational implementation of our model, inspecting sample diversity qualitatively and then measuring performance in terms of overall likelihood (via the ELBO, see Equation (8.9)) as well as its individual components, including the KL divergence (or *surprise*) like in Chapter 4.

8.4 RESULTS

8.4.1 *Toy Examples*

To evaluate how well a model can interpolate segmentations on a continuous axis, we construct a toy task by moving a star shape along a random linear trajectory within a fixed frame. The star can rotate along the axis, change its size and also change its pointiness. The goal is then to interpolate frames from a small number of context observations. An example of this can be seen in Figure 8.2. We find that a regular CNP struggles with this task. While it generally predicts the correct location, rotation and size, it only produces smoothed out versions of the desired star shape. A Segmentation Process (SP), which is the same as a CNP but with skip connections that also perform a summation of the context, can reconstruct the given context well, but predictions at other points generally only capture the position and size correctly, losing the characteristic star shape. ANPs generally handle the task well, but sometimes struggle to fully reproduce the star shape (see second-to-last row, $t = 0.83$). Our model, ASP_1, solves this particular example almost perfectly and overall outperforms the other approaches, as seen in Table 8.1, where it achieves the highest average Dice score.

The second capability we seek in continuous interpolation of segmentations is the dynamic adaptation to information presented at test time. To this end we construct a toy example by generating random image volumes for which we then generate a segmentation by thresholding the image. However, we shuffle the classes for each example, so that models can't learn any correspondence of grey values to classes. We provide a small number of annotated slices from the volume and the models need to segment the full volume by inferring a mapping from image intensities to labels for the given example only from the context slices. An example for this task is shown in Figure 8.3. We don't show the ANP, because here the model collapsed to predicting only a single class for the full volume. CNP is hardly able to solve the task, with segmentations exhibiting strong discontinuities and misclassifications. SP performs better, classifying grey values mostly

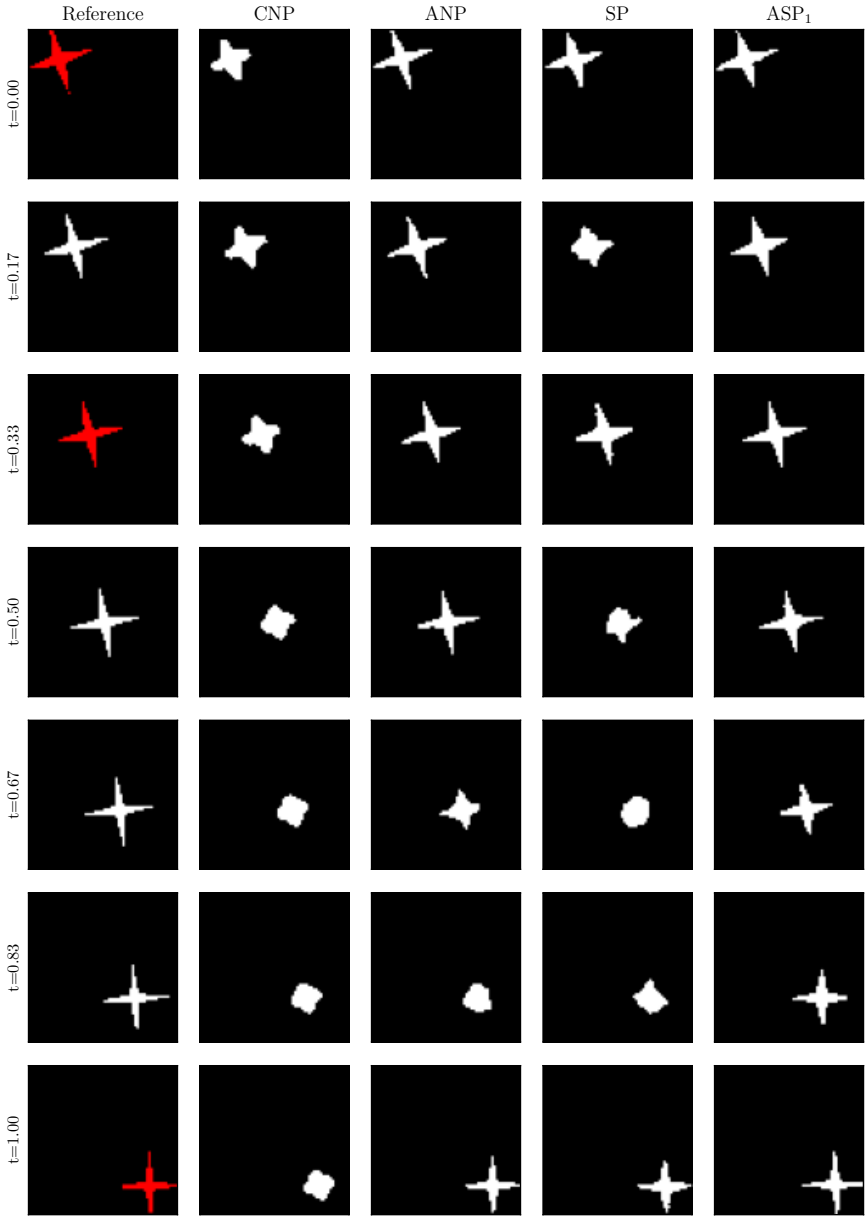

Figure 8.2: Example for the segmentation interpolation toy task, context shown in red. The models are trained on segmentations of a star shape, which moves along a random linear trajectory trough the image. Along the trajectory the a star rotates by a random amount, changes size and changes pointiness. Only our ASP$_1$ model is able to fully interpolate all of these characteristics.

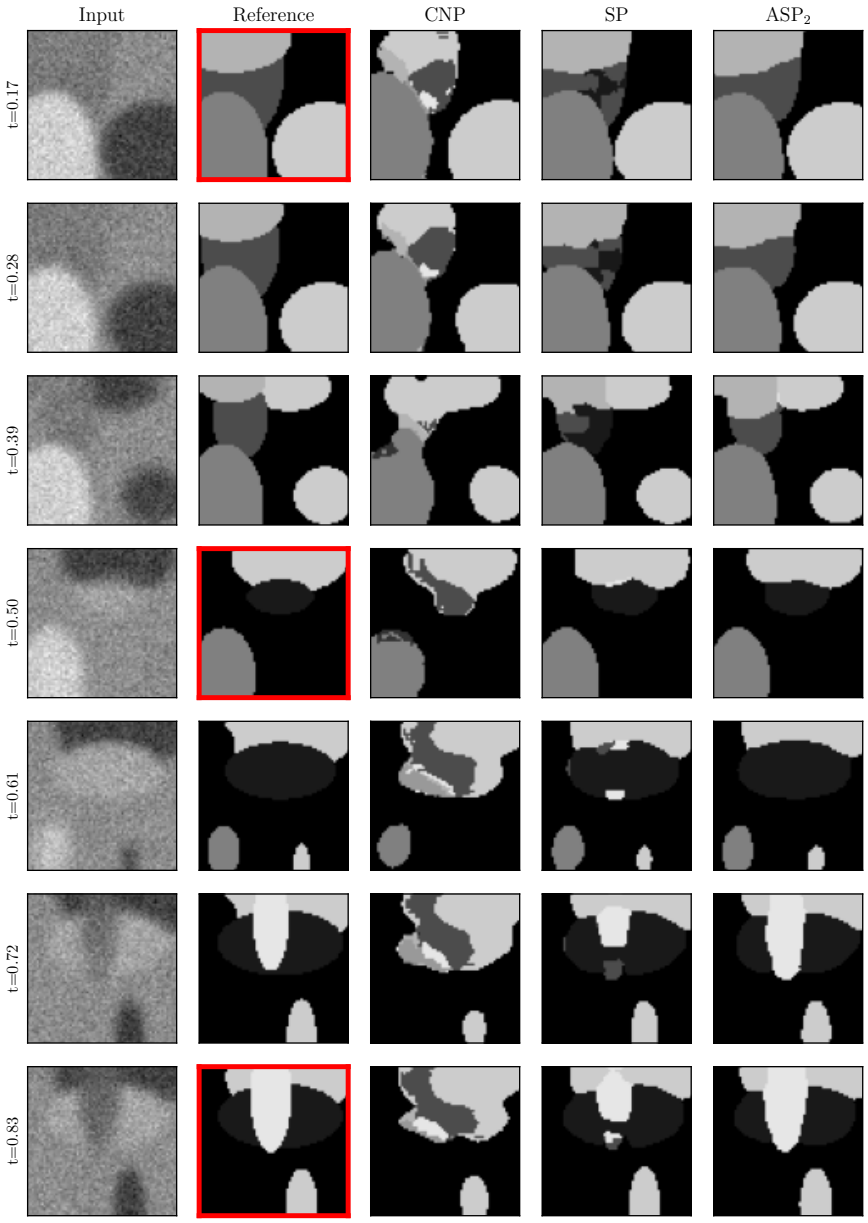

Figure 8.3: Example for the generalization toy task. We generate image volumes by placing ellipsoids randomly and adding noise with random magnitude at the end. Segmentations are generated by thresholding the input, but classes are shuffled randomly to remove any correlation between image intensities and classes. The models need to infer this information at test time from the provided context segmentations (marked in red). CNP performs poorly on the task, while SP generally manages to assign the correct labels, but regions often appear disconnected erroneously. ASP_2 is able to solve the task almost completely.

Table 8.1: Results for the segmentation interpolation toy task, ↑ indicates that higher is better. All results are averaged over 32 000 test examples and the errors represent the standard error of the mean. ASP_1 outperforms the baselines, with ANP coming closest in performance, similar to what what we see in Figure 8.2.

	CNP	ANP	SP	ASP_1
Dice [1e-2] ↑	75.0 ± 0.1	87.7 ± 0.1	84.5 ± 0.1	$\mathbf{89.5 \pm 0.1}$

Table 8.2: Results for the generalization toy task, ↑ / ↓ indicate that higher/lower is better. The Dice score measures the overlap of the true and the predicted segmentation, averaged over all classes in a given example. To measure the connectedness of the predicted segmentations, we also evaluate the absolute difference between the numbers of connected components in the ground truth and the prediction, which we call "#Objects Error". All results are averaged over 32 000 test examples and the errors represent the standard error of the mean.

	CNP	ANP	SP	ASP_2
Dice [1e-2] ↑	95.4 ± 0.1	86.9 ± 0.1	98.4 ± 0.1	$\mathbf{98.7 \pm 0.1}$
#Objects Error ↓	130.3 ± 0.5	7.4 ± 0.1	22.4 ± 0.2	$\mathbf{4.4 \pm 0.1}$

correctly, segmentations are still often discontinuous. Our model ASP_2 (with downsampling to 8^2 resolution before the spatial attention to save time) solves the task almost perfectly, with only minor misclassifications (e.g. in the third row). This is confirmed by the results in Table 8.2, where ASP_2 shows both the best segmentation performance and the lowest error in the number of connected components.

8.4.2 *Interactive Segmentation*

Our proposed model solves the toy tasks remarkably well, so as a next step we seek to show that it is also able to work on real-world data. To this end, we apply it to interactive segmentation of glioma. Like in the second toy example, we provide models with a few annotated slices of 3D MRI scans of brain tumor patients, from which they need to construct a segmentation of the entire 3D volume. Because this task can be solved quite well by just segmenting the input image directly, without utilizing the context information, we again shuffle the classes randomly for each example. This way the model can't assign certain imaging features to any particular class. Figure 8.4 shows that this does indeed work. The fourth column shows the prediction of our model with the correct context slices provided, and unsurprisingly it segments the input images quite well, only misclassifying some blood vessels as enhancing tumor. However, if we decide that the enhancing

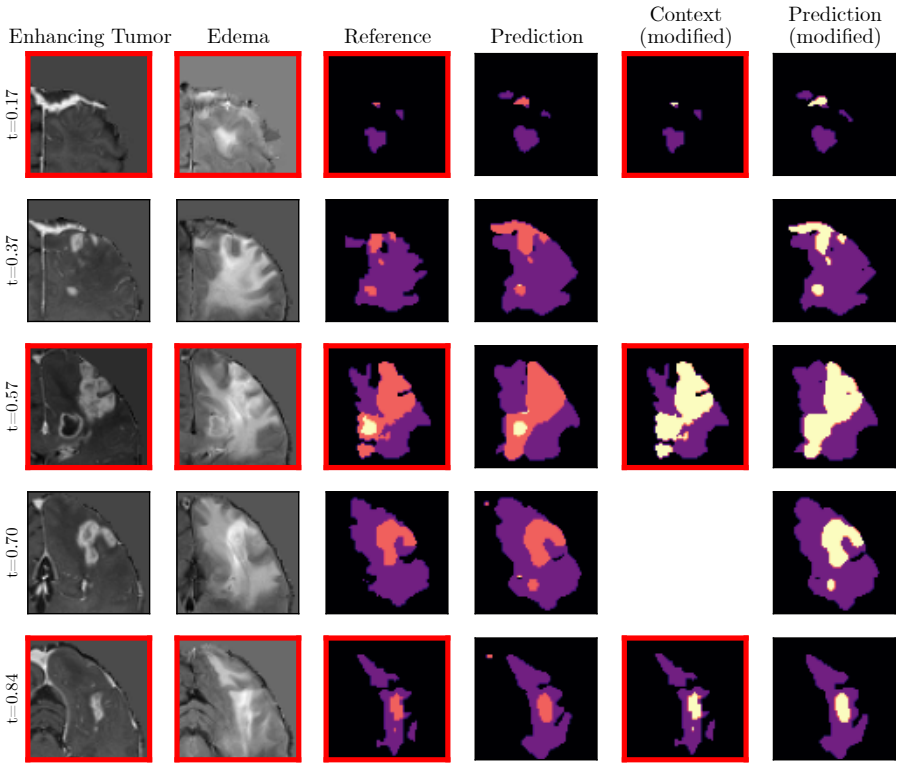

Figure 8.4: Example for the interactive segmentation task, with context marked in red. Displayed classes are edema (purple), enhancing tumor (orange) and necrosis/non-enhancing tumor (light yellow). We provide segmentation slices from a volume and the model (ASP_2 with skip connections) must interpolate the provided information to other image slices. Because there is relatively little ambiguity, the models will usually directly segment the input images and not utilize the context segmentations. For this reason, we shuffle the classes in each example like we did in the second toy example. When we reassign enhancing tumor to necrosis in the context slices, our model interpolates this change to other slices as well. For the input, we only show the T1c and FLAIR channels, which best visualize the enhancing tumor and the edema. We also see that our model does not perfectly reconstruct the context segmentations, and that it mistakenly segments blood vessels as enhancing tumor.

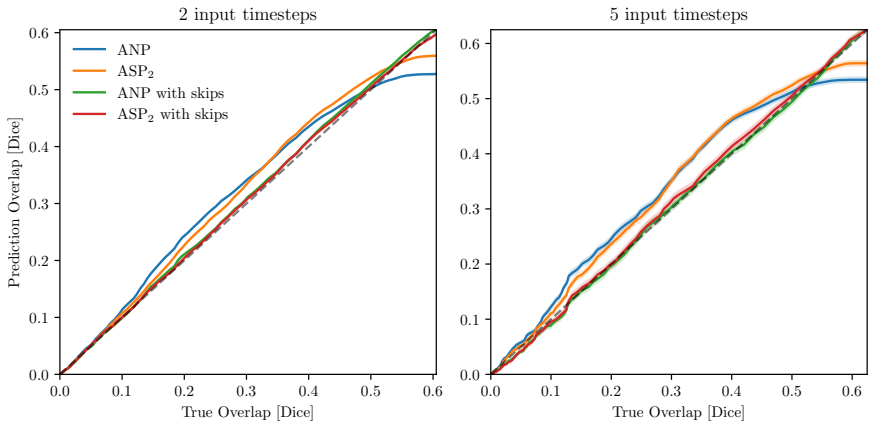

Figure 8.5: Predictive Dice for ANP and ASP_2, for two input time points (left) and 5 input time points (right). We threshold the true overlap between the last context point and the target point, and display the median predictive dice vs. the median true overlap. The shaded area (almost negligibly small) represents the standard error of the mean in each case. The dashed line indicates the predictive performance (measured as Dice overlap) if we just predict the last context point for the target point. We find that averaging over the all test cases (rightmost point), no model outperforms predicting no change. Models with skip connections will on average perform as well as predicting no change. Models without skips connections will perform worse than predicting no change when we include all test cases, with ASP_2 performing better than ANP, but they exhibit better predictive performance on cases with larger change. These results are for the whole tumor region, i.e. the union of all classes. The same graphs for the individual classes are shown in Figure A.8.

tumor should rather be necrosis or non-enhancing tumor, as indicated in the modified context, our model also incorporates this change in the target slices. Admittedly, this particular change is hardly justified given the underlying T1c scan shown in the first column, but it's easy to imagine how this can be used in more ambiguous cases when the model misclassifies a certain region.

8.4.3 *Tumor Growth Modeling*

Following the guiding example of this thesis, we apply Attentive Segmentation Processes to tumor growth modeling. While our first attempt at this in Chapter 4 only worked with discrete time steps and a fixed number of inputs, Attentive Segmentation Processes work on a continuous time domain and can incorporate arbitrarily many context observations. We also saw in Chapter 5 that ANPs have at least some predictive capability in terms of estimating tumor growth, i.e. they have a lower predictive error than predicting no change. We repeat this

experiment, but now measure the Dice score of predictions and again compare it with the Dice score obtained by not predicting any change. The result can be seen in Figure 8.5, where we show the average prediction overlap vs. the average baseline overlap, thresholded at different values. We compare our ASP_2 with ANP as a baseline, both with and without skip connections. We find that no model performs better than predicting no change when averaged over all test cases (the rightmost point of the graphs), but when we only look at a subset of the test cases that exhibits larger change, there is a predictive value to both ANP and ASP_2, with ASP_2 performing better when all cases are considered. Models with skip connections, however, have absolutely no predictive value and perform exactly as well as no change prediction at all thresholds.

The main goal in this growth modeling context is that the model predictions are realistic growth trajectories. We find that this indeed the case, with one example given in Figure 8.6. ANP and ASP_2 both produce very smoothed out segmentation, with ASP_2 offering a little more spatial resolution. Interestingly, while ANP predicts very little change for the future time points, ASP_2 actually anticipates some growth. Both models without skip connections can't accurately reconstruct the context observations and essentially produce temporally smoothed out versions of the context. Compare this to Figure 5.6, where ANP was applied to volume regression. There, too, the model produces an interpolation that doesn't go directly through the context points, so it is unsurprising to see similar behaviour here. Introducing skip connections, however, does allow the models to reproduce the context segmentations accurately, and also to predict more spatially complex predictions in the future. Like the ANP without skip connections, the model with skip connections predicts very little change over time in the future. The ASP_2 with skip connections predicts growth like the model without skips, but it anticipates a very interesting scenario where enhancing tumor forms around the necrotic core. This is realistic behaviour that can be observed in the training data as well, and quite impressively our model is able to capture this complex growth pattern. With the findings from above, it is unsurprising that none of the models actually predict the observed growth trajectory correctly.

Without predictive value, deterministic models are quite useless, even if they predict realistic growth patterns like in Figure 8.6. We thus cast our model in a variational framework, which allows us to sample from it. This is done by replacing the lowest level of the U-Net with a Gaussian distribution, just like in Neural Processes. Figure 8.7 shows exemplary samples from a variational ASP_2. We find that the model produces diverse growth trajectories while still being able to reconstruct the context segmentations quite accurately, with some averaging effect especially between the second and third context point.

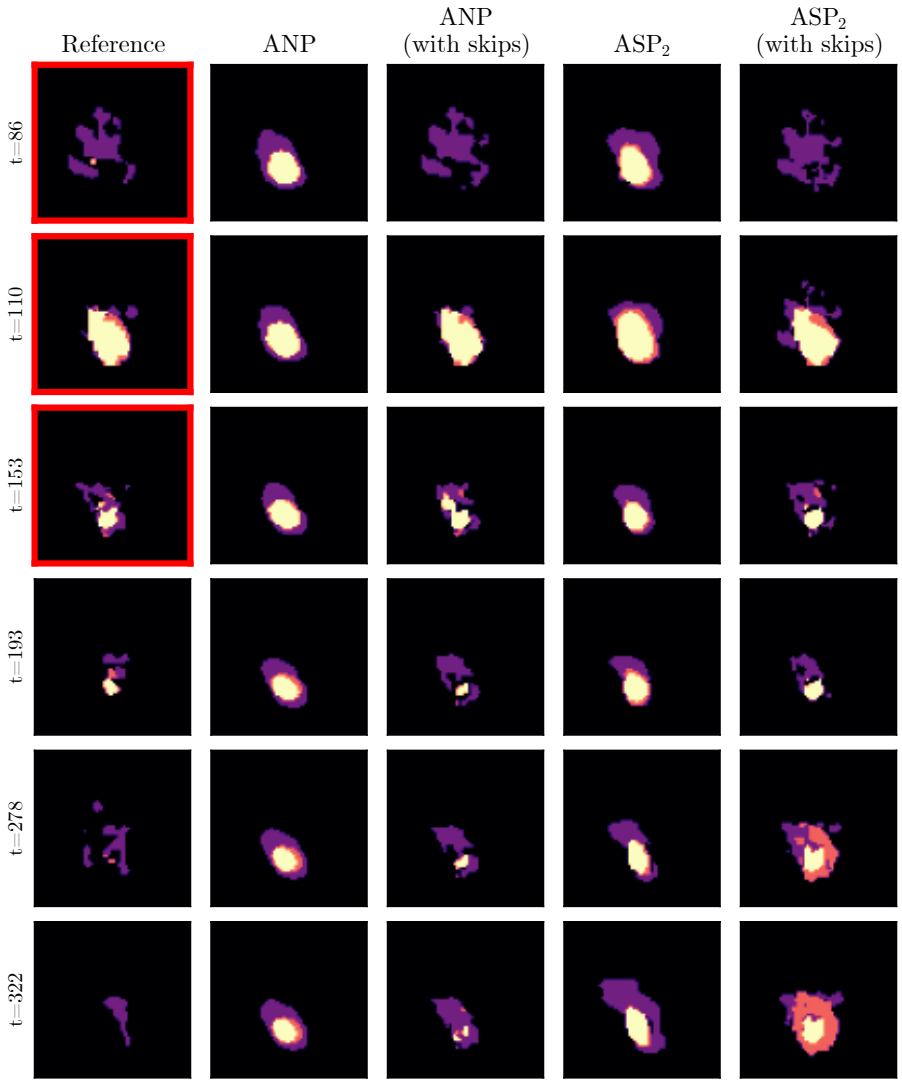

Figure 8.6: Example growth predictions for different models, with context marked in red. Times are given in days. Displayed classes are edema (purple), enhancing tumor (orange) and necrosis/non-enhancing tumor (light yellow). ANP (and any other model that compresses inputs into a representation without spatial resolution) is only able to produce very smooth predictions and can't reconstruct the provided context well. ASP$_2$ improves on this, but still produces relatively smoothed out segmentations. Introducing skip connections allows models to reconstruct the context segmentations accurately, but in the case of ANP, predictions in the future often look very similar. Our model, ASP$_2$ reproduces the context segmentations well and at the same time produces a realistic future growth pattern, where enhancing tumor forms around the necrotic core. However, none of the models actually predict the correct growth trajectory.

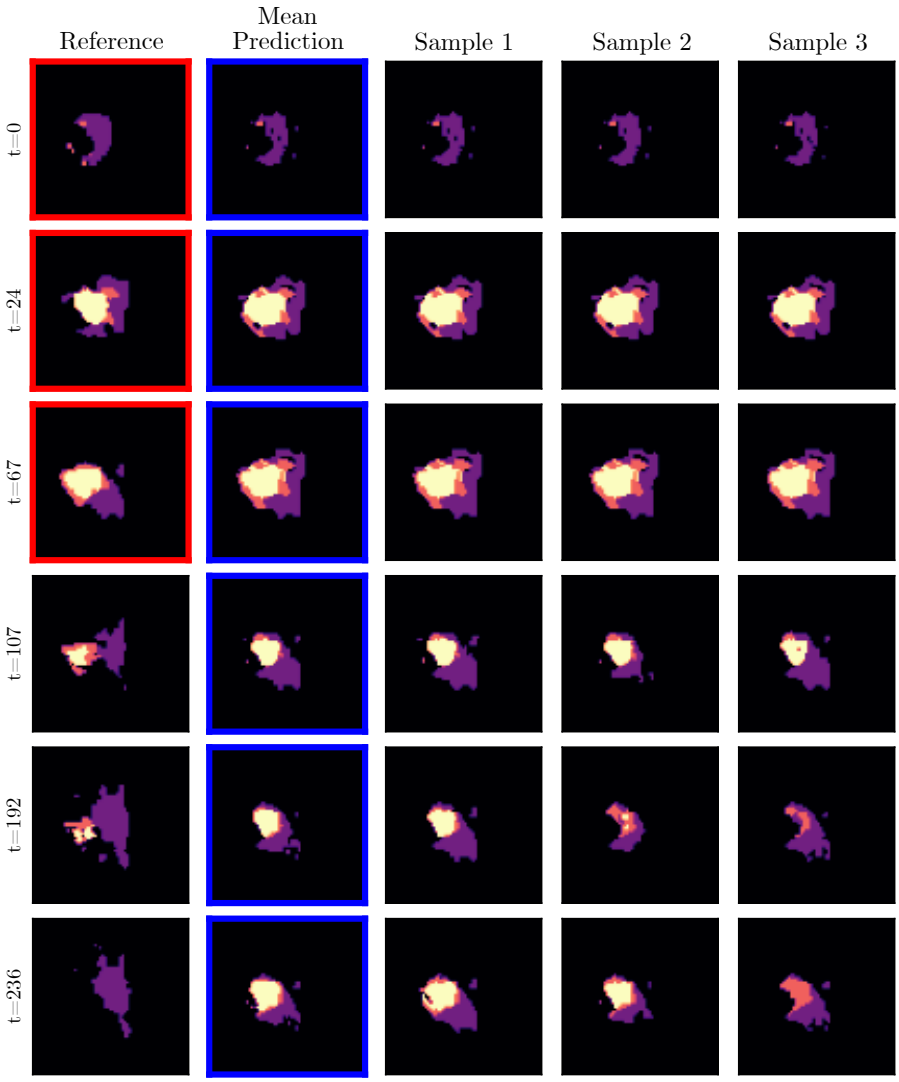

Figure 8.7: Sample diversity for growth predictions from ASP_2 with skip connections, with context marked in red and the mean prediction marked in blue. Displayed classes are edema (purple), enhancing tumor (orange) and necrosis/non-enhancing tumor (light yellow). The rightmost columns show samples from the predicted distribution. While there is virtually no variation at the context points, as is desirable, sample diversity increases moving further away from the provided context. While no sample matches the true growth trajectory, all samples look like realistic patterns. Importantly, each column represents one globally consistent growth sample.

Table 8.3: Results for tumor growth modeling with Attentive Segmentation Processes, ↑ / ↓ indicate that higher/lower is better. Log-likelihood is the negative cross entropy and measures prediction/reconstruction quality. The differentiable (soft) formulation of measures the same, and the sum of the two represents what our models are trained to maximize. The KL divergence is also sometimes called the *surprise*, as it measures how far the predicted prior is from the posterior, and we give the value *per channel*. For reference, in Chapter 4 we obtained a surprise of 3.688 ± 0.059 per channel, but because the models here were trained differently, the values are not directly comparable. The aggregate ELBO is the negative loss (Equation (8.8)), evaluated on the test set. Errors represent the standard error of the mean. These results were obtained by providing the models with two context observations in each case.

	NP	NP with skips	ASP_2	ASP_2 with skips
Log-Likelihood ↑	-0.198 ± 0.002	-0.197 ± 0.002	-0.214 ± 0.003	$\mathbf{-0.118 \pm 0.009}$
Dice (Soft) ↑	0.913 ± 0.001	0.914 ± 0.001	0.917 ± 0.001	$\mathbf{0.955 \pm 0.004}$
KL (Surprise) ↓	2.766 ± 0.033	2.740 ± 0.031	$\mathbf{0.557 \pm 0.008}$	1.191 ± 0.072
Aggregate (ELBO) ↑	0.680 ± 0.002	0.682 ± 0.002	0.696 ± 0.003	$\mathbf{0.822 \pm 0.009}$

At the first time point in the future, there is still very little variation in the samples, but as we move further away from the context, samples become more diverse. Note also that by construction, samples always represent global realizations of a single trajectory, so that each column in Figure 8.7 represents a consistent set of predictions over time.

To evaluate the quality of the learned distribution, we measure the three components that also make up our loss in Table 8.3: both the log-likelihood (i.e. the negative cross entropy) and the Dice score, or more precisely its continuous implementation, measure the quality of the predicted segmentation at a target point, while the KL divergence measure the difference between prior and posterior. If the predicted prior is far from the posterior, it means that the model assigns a lower likelihood to the particular observation, which is why the KL divergence is also sometimes called the *surprise*. We find that in terms of segmentation quality, ASP_2 with skip connections performs best, while in terms of surprise, ASP_2 without skips exhibits the lowest KL. When aggregating the individual values following Equation (8.9), i.e. essentially the total negative loss (deterministic and KL divergence) evaluated on the test set, ASP_2 with skip connections outperforms the other models by a large margin. Note that these values are only comparable for models trained in exactly the same framework, so the values for the surprise can't be compared with those obtained in Chapter 4.

8.5 DISCUSSION

In this chapter we present an approach that leverages the capabilities of Neural Processes in a segmentation context. NPs are able to learn function distributions in a way that allows them to condition their predictions to observations given at test time. However, we find that in their default implementation, which encodes observations into a joint representation space and sums their representation, these models are hardly suitable for segmentation purposes, where modeling of fine details is required. To remedy this, we first cast a U-Net architecture, i.e. an encoder-decoder structure with skip connections, as a Neural Process by interpreting the skip connections as separate representations, but find that summation in these skip representations results in artifacts in the predictions and overall poor performance. Combining the representations with an attention mechanism instead, like in Attentive Neural Processes, leads to an improvement, but the resulting model still fails to model more complex spatial patterns. We propose the introduction of *spatial attention* in the coarser scales of the U-Net, meaning attention not only between input locations t_i and t_j, but between tuples (t_i, x_i, y_i) and (t_j, x_j, y_j). We call such models *Attentive Segmentation Processes* (ASP) and show that they outperform the aforementioned approaches by a large margin. ASPs can be applied in scenarios where input images are available at the target locations, e.g. for interactive segmentation, as well as in scenarios where segmentations have to be interpolated fully from available context segmentations.

We apply ASPs to the tumor growth modeling task already introduced in Chapter 4, and find that they are able to produce realistic, complex and spatially varying growth patterns. At the same time, much in line with the findings in Chapter 5, it was impossible to accurately predict changes in tumor appearance and size based on earlier observations alone. As a result, we also present a variational version of ASPs that can sample multiple growth trajectories. These trajectories are globally consistent and can be very diverse, including growth of different tissue types. Again, our variational model outperforms the baselines—variational Neural Processes with and without skip connections—by a large margin.

While our approach is both versatile and powerful, it has one main disadvantage in its compute requirements, specifically GPU memory consumption. The number of additional parameters our spatial attention introduces is small, but it entails extremely large matrix multiplications. Assume we work with images of size N^D and we'd like to predict T target points from C context points. GPU memory usage then scales with $O(C \cdot T \cdot N^{2D})$, which prevents us from conducting our tumor growth experiments on 3D data and from introducing spatial attention at the higher level skip connections in the U-Net.

This is a well-known problem of attention mechanisms in general and a very active area of research. Wang et al. (2020b) propose a low-rank approximation that reduces the problem to linear instead of quadratic complexity, but this requires a fixed sequence length. Kitaev et al. (2020) introduce binning of similar queries/keys, which reduces the complexity to $O(L \log L)$ (L being an abstract sequence length containing both N and C/T). One can also use sparse attention (Child et al., 2019), which just ignores some context points and is more appropriate for a large and dense context. Working with images, the most natural way to reduce the computational cost would be to limit spatial attention to only a neighborhood of a given point (x_i, y_i), which works well in principle (Parmar et al., 2018). Unfortunately, there is currently no way to implement this efficiently in commonly used deep learning frameworks that would actually result in memory savings.

9 DISCUSSION

9.1 SUMMARY

We have presented a number of contributions that share a common desire: unlike most deep learning research, we wish to learn representations of function spaces—as opposed to single functions—so that we can leverage these representations and perform predictions conditioned on additional observations available at a later time. Moreover, we are not satisfied with representations that operate only on discrete domains, seeking instead representations of continuous functions. Finally, the desired approach should be able to handle both deterministic and probabilistic scenarios.

As a guiding example where all these factors come into play we looked at glioma growth. Having monitored the disease in a patient for some time, an estimate of its future development—both spatially and with respect to the total tumor burden—would be invaluable for treatment planning. As we outline in Section 2.2, there is no consensus on the mechanism that dictates glioma growth, so it seems only natural to instead try to learn it from examples. At the same time, it can be assumed that the process is at least in part stochastic, so that purely deterministic growth estimates would be of little help.

It is no secret that deep learning requires comparatively large amounts of annotated data for training. Generating those annotations is especially challenging in the medical domain, where expert knowledge is required. To enable ourselves to conduct deep learning research, our initial contribution in Chapter 3 was to investigate how interactive segmentation of glioblastoma tissue on MRI data can be done most efficiently. Our main finding was that expert users should be encouraged to iteratively correct the classifier instead of trying to minimize its uncertainty, which somewhat contradicts much of the literature in the field of so-called *active learning*, where researchers seek to find new measures that better describe classifier uncertainty. There could certainly be better instruments for this than what the employed—the commonly used probability entropy—but we suspect the finding will generalize to other uncertainty metrics. We anticipate that more efficient interactive segmentation techniques are to be expected from the deep learning domain and give an overview of the field in Section 3.6. As of now, these techniques still require annotated data to be trained, while the Random Forest we use is trained from scratch for each new image volume, but recently there has been a lot of progress in unsupervised representation learning (Oord et al., 2018;

Hénaff et al., 2019; He et al., 2020; Chen et al., 2020b). We imagine that these advancements could be leveraged to enable deep interactive segmentation without the need for training labels.

We made a first attempt at learned glioma growth in Chapter 4. Instead of trying to predict growth, which we argue is hardly possible anyway, we focused on learned *modeling*: can the network, a Probabilistic U-Net (Kohl et al., 2018), represent a distribution of different growth trajectories, given some context observations? We found that it can, but the limitations of this approach are plentiful. In fact, none of our requirements from above are fulfilled. While the Probabilistic U-Net predicts a distribution of future tumor appearances, it doesn't actually represent a distribution of *functions*. It only works with a fixed number of inputs and requires a fixed time difference between them. The main goal was to show as a proof of concept that purely learned growth modeling is possible, something that hasn't been done before, and the results encouraged us to explore the problem further.

Chapter 5 was used to further illustrate our research objective and to give the reader a better understanding of the data we're working with. We applied polynomial regression to predict future tumor volume measurements and found that linear regression is the best choice, narrowly beating a trivial baseline that does not use any longitudinal information. However, when an uncertainty estimate is desired—which it certainly is for this purpose—and the linear regression is performed in a Bayesian framework, we couldn't beat a trivial baseline in terms of average predictive likelihood. We then introduced Neural Processes (Garnelo et al., 2018a; Garnelo et al., 2018b) as a tool that, in principle, meets all desiderata outlined above. These models are indeed able to learn representations of function spaces on a continuous domain, and can also be formulated in a variational framework to handle probabilistic scenarios. Applying Neural Processes to tumor volume prediction, we found that they performed much better than the other approaches we tried, but at the same time their average errors were still too large to be able to say that they can *predict* tumor growth. For us, this illustrated two things: a) that there is great value in learning function spaces compared to manually specifying them, which is one of the main factors motivating our work; and b) that there is no point in trying to deterministically predict tumor growth, at least from the data available to us, meaning we should instead focus on the modeling aspect, like we did in Chapter 4. Still, the convincing performance of Neural Processes persuaded us to make them the basis of our later work.

Knowing that Neural Processes are able to learn representations of function spaces, we were curious what these representations look like. After all, these are finite-dimensional representations of an infinite-dimensional space. Using synthetic one-dimensional examples, we found that the individual dimensions of the learned representations

correspond to different regions of the input space, i.e. the domain of the function space. In variational Neural Processes this usually results in a partitioning, sometimes very sharp, so that a context input at a certain location t would "activate" a representation dimension. A more interesting pattern was visible in the representations of deterministic Neural Processes, where the correspondence between representation dimensions and input regions exhibited an oscillating behaviour. More importantly, the frequency of these oscillations varied among representation channels, which led us to conclude that Neural Processes perform a frequency decomposition of the function space. We further tested this by deriving a theoretical upper bound on the maximum signal frequency that can be represented in a Neural Process with a given representation size and empirically validating that the bound does indeed hold. Finally, as another confirmation that Neural Processes learn frequencies, we showed that they suppress frequencies in a test signal they have not seen in the training data. This essentially means that they are trainable band-pass and band-stop filters.

While these are some surprising insights into the inner workings of Neural Processes, there is still a lot we were not able to explain entirely. It is not impossible for variational Neural Processes to learn frequency decomposition, and it would be interesting to know how we can encourage this behaviour in them to a greater extent. It would also be helpful to derive some theoretical conditions that allow a function space to be represented, or rather decide how good the representation will be. Because our work only focuses on continuous *time* functions, we didn't investigate higher-dimensional domains to see if frequency decomposition still occurs. Another open question is how our findings translate to scenarios that disallow a classical definition of frequency, for example when the observations are images. Finally, we are curious if the construction of Neural Processes perhaps forces them to learn a basis of the function space. We will further elaborate on this below in Section 9.3.

A more recent contribution to the Neural Process family, ConvCNP (Gordon et al., 2020), reported impressive performance compared to prior art, but these models are fully deterministic. In an attempt to recover the ability to predict multiple samples, we used Chapter 7 to combine ConvCNP with Gaussian Processes. This does indeed allow for sampling from the model, but we also found that it improves generalization. The Gaussian Process allows our model to better extrapolate far from the provided context points and also makes it more robust to distribution shifts at test time. Unfortunately, both ConvCNP and our model did not work on tumor volume regression, we suspect that they need a minimum number of context points that is larger than what the glioma growth problem offers. It is also not immediately obvious how these models can be applied to very high-dimensional (e.g. images) observation spaces. Compared to

ConvCNP the introduction of the Gaussian Process comes at some computational cost that could be restrictive for higher-dimensional input spaces or extremely large numbers of context points. As we point out in Section 7.6, approximate methods could alleviate this problem.

In Chapter 8 we demonstrate segmentation in the Neural Process framework. In principle, the skip connections in a segmentation architecture can be interpreted as additional representation spaces that retain a spatial resolution, but simple summation of context representations like in conventional Neural Processes results in poor performance. Using attention over time, i.e. combining an Attentive Neural Process (Kim et al., 2019) with a segmentation architecture, improves performance to an extent. We show that spatial attention, which incorporates both time and spatial location in the representations, results in vastly improved performance. The resulting model, which can be used both with and without input images available at target points, is able to interpolate complex spatial shapes and can dynamically propagate information from the context segmentation, e.g. by assigning certain structures in an input image to a class specified in the context.

Translating our initial attempt at glioma growth modeling from Chapter 4 to a continuous time domain, we applied the above *Attentive Segmentation Processes* to the problem. We began by testing the predictive capabilities of our model, as well as regular Neural Processes and Attentive Neural Processes, and found that none of them offered any predictive value when evaluated on the full dataset. Even though we found in Chapter 5 that Neural Processes can learn to predict future tumor volumes to some extent, this evidently doesn't translate to the image space, where we use an overlap measure (Dice) to evaluate predicted segmentations. Our earlier hypothesis that we should focus more on *modeling* glioma growth instead of *predicting* it was thus reinforced, and we consequently proposed a variational version of our model that can predict a distribution of possible growth trajectories for a given set of context observations. We showed qualitatively that our model can represent complex spatial variations of the tumor and that the individual growth trajectories are consistent over time. Finally, we showed that our model performs significantly better than all competing approaches in terms of test loss.

Even though we have successfully demonstrated learned glioma growth modeling on a continuous time domain, the approach can clearly not be considered much more than a proof of concept. Judging from the results in Chapter 5 we suspect that it should be possible to at least have some predictive value in the models. Including treatment information would certainly be of help, but we did not have access to it in a structured format. Regardless of predictive capabilities, there could be a benefit in using the model to identify regions of high and

low probability of tumor infiltration e.g. in the context of radiation therapy planning. A comparison of our model with biological growth models, which we summarize in Section 2.2, would also be interesting, but it is not clear how exactly such a comparison could be done. After all, the growth mechanism in these approaches is always modeled as deterministic, so we could only compare predictive capabilities. Overall, the most difficult aspect of glioma growth modeling remains: it is entirely unclear what the upper bound on performance is and how good our models can ever hope to be.

9.2 CLINICAL TRANSLATION

The contributions we make in this work, in particular those related to glioma growth, clearly have a long way to go before they could be of practical use and be deployed to the "real world" in any way. Nevertheless, the field of medical image analysis has seen impressive progress in recent years, thanks to advances in deep learning-based image processing. Numerous studies have been published attesting AI models the ability to perform certain clinical tasks as well as or better than humans, for example skin lesion classification (Esteva et al., 2017), diagnosis of retinal disease (Gulshan et al., 2016; Fauw et al., 2018), breast cancer screening (McKinney et al., 2020) or segmentation of organs of risk for radiation therapy (Nikolov et al., 2018). Alternatively, deep learning could improve the standard of care, either by incorporating more data into a diagnosis (Jäger et al., 2017) or by performing tasks that would be too labor-intensive to do manually, like tumor volumetry (Kickingereder et al., 2019).

In light of these promising findings, it might come as somewhat of a surprise that the number of AI-based products that have made into the healthcare market is still very limited. One of the limiting factors is certainly time; to gain regulatory approval proposed diagnostic systems need to undergo rigorous testing. And for good reason, as there is often a certain *translational gap* that presents itself when we move from a controlled research environment to the uncertainties and individualities of clinical routine. As an example, one study we mention above developed a model to detect diabetic retinopathy in retinal fundus images (Gulshan et al., 2016). The model was subsequently deployed in a prospective study (which might still be ongoing) across multiple sites in Thailand. In an initial report (Beede et al., 2020) the authors state that in clinical routine the model's performance was inferior compared to the initial results. They attribute this to changed environmental factors such as lighting conditions, but also to differences in routine workflows. Overall, these changes can be described as *distribution shifts* between training and test data distributions (Kelly et al., 2019; Oakden-Rayner et al., 2020) that models are not robust to.

To overcome this problem and to truly be able to deliver clinical impact, it makes sense to test developed models in a clinical routine setting as early as possible; at least that was our desire for the very successful segmentation model we developed in Isensee et al. (2018). While not directly related to the topic of this thesis, it is still an effort worth mentioning: we developed a processing infrastructure that can be used to deploy deep learning models (or any form of automated processing) in clinical routine. The system is detailed in Petersen et al. (2018) and Kickingereder et al. (2019), what follows is a very practical description, a guideline of sorts for those who might be interested to reproduce it.

The basic idea is that whenever imaging data is acquired for which automated processing is desired, the data is not only sent to the PACS[1] but also to a dedicated processing server that is integrated into the clinical network infrastructure to ensure adherence to data protection regulations. The process of sending acquisitions to multiple receivers can be automated by adjusting scanner protocols. The processing server uses an open source PACS alternative called XNAT (Marcus et al., 2007). Using its "Container Service" plugin, XNAT can start Docker[2] containers, which are encapsulated processing environments that ensure independence with respect to the system on which they are executed. Researchers, in the vast majority of cases, work with processing scripts that take in a number of files and output other files, for our purposes those will be the inputs to and outputs from a deep neural network. We developed a wrapper that handles communication with the XNAT (and also with the PACS if desired) and can receive and send files from and to them, from within a Docker container. Using this wrapper, researchers can quickly convert their research scripts into Docker containers that can be deployed to run on our system, meaning directly parallel to clinical routine and on data that has not been processed or even quality-controlled after acquisition at the scanner. Patient consent is of course still required and ensured by our clinical partners.

The simplicity of this setup has a number of advantages: it only uses open source components, meaning there is no cost involved beyond the potential need to buy dedicated hardware for the processing server; the practical overhead for clinicians and researchers is small; there is no interference with clinical routine workflows; and data protection regulations are observed. As a first example, we applied the segmentation model developed in (Isensee et al., 2018) for the purpose of tumor volumetry. As we outline in Section 2.2, tumor size is one of the key features tracked in glioblastoma progression monitoring. Using the above system to process a large number of

1 Picture Archiving and Communication System, a system hospitals use to store imaging data.

2 www.docker.com

cases automatically, we could show that automated volumetry based on CNNs is superior to the standard of care that uses tumor diameters as a proxy (Wen et al., 2010), because manual volumetry is too time-consuming (Kickingereder et al., 2019). We also automatically generate a graph that displays the longitudinal change of the tumor volume and send it to the PACS as an additional source of information for routine radiological diagnosis. The system is of course only being employed in a research context, but it has now been in active use at the Heidelberg University Hospital for many months and has so far processed data from over 1000 patients, enabling further studies (Brugnara et al., 2020). We hope that this brief description of our work beyond this thesis can serve as an indication that the difficulties of clinical translation can be addressed even with simple tools and a small budget.

9.3 FUTURE DIRECTIONS

We will close with some speculation what future research in a similar direction as this thesis could look like, focusing on two aspects in particular where we see a lot of potential for impactful contributions.

9.3.1 *Do Neural Processes Learn Bases?*

Above we already mentioned that the construction of Neural Processes might force them to learn a basis of the function space they are trained on, a hypothesis we form from the observation that they decompose the function space into different frequencies—akin to a representation in a Fourier basis. Recall that a set of functions $\{f_i\}_{i=1}^N$ is called a basis if it holds that $\forall g \in \mathcal{G} : g = \sum_i \alpha_i f_i$, where $\mathcal{G}$ is some function space that is also a vector space[3]. Otherwise a basis doesn't exist, but it would of course be interesting to observe how Neural Processes behave in that case. Assume now we have a perfectly trained Neural Process with encoder E and decoder G. In that case we know that for some represented function f:

$$f(x) = G(x, \mathbf{r}) \quad , \quad r_i \propto \int E_i(x', f(x'))dx' \tag{9.1}$$

where i indexes representation dimensions and we use an integral instead of a sum to indicate that we pass all points through the Neural Process. It's easy to see that by choosing $E_i(x', f(x')) = f(x')e^{jix'}$ we could recover a Fourier transform. However, as we saw in Chapter 6, the Neural Process doesn't actually learn such a Fourier transform, probably because there is nothing that would encourage orthogonality

3 Some sources define function spaces as any set of functions with shared domain and co-domain, others explicitly require them to be a vector space.

in the representations. To further investigate our conjecture, it would make sense to look at function spaces that don't have a Fourier basis. Would the Neural Process be able to approximate other bases as well? Either way, some metric would be required that essentially tells us how well the learned representation covers the function space, because whatever representation we learn, it will at best *approximate* a basis. It might also make sense to learn representations for *finite-dimensional* vector spaces. That would potentially make it easier to evaluate orthogonality and linear separability. How would the Neural Process behave if the representation size is smaller than the dimensionality of the vector space and the data cannot be approximated with fewer dimensions (e.g. for points sampled randomly from the unit ball)? Such investigations could yield further insight into learned representations of function spaces and perhaps into function approximation in general, which would encompass many areas of machine learning.

9.3.2 *Learned Differential Equations*

The idea to combine differential equations with neural networks is not new. Consider the first order ordinary differential equation $\dot{y} = f(y)$. If we have observations for $y(t)$ but don't actually know $f(y)$, it might make sense to approximate it with a neural network. The difficulty then lies in simultaneously solving the ODE and the optimization problem for f. In practice, the problem is usually rephrased as $\dot{y} = y + f'(y)$, and simpler implementations of f' (e.g. a single projection with activation) date back several years (Beer and Gallagher, 1992; Beer, 1997). These approaches, called *continuous-time recurrent neural networks*, didn't use gradient descent but instead relied on evolutionary algorithms to find f'.

While the field of learned differential equations has continuously progressed—we won't discuss older works here—it can be argued that it recently experienced a surge in interest after the publication of *Neural ODEs* (Chen et al., 2018), where the authors connect the above formulation of ODEs to residual neural networks (He et al., 2016) and propose to use the adjoint method to efficiently perform gradient descent on f' while solving the ODE. Since then, their work has been extended in various ways, e.g. to stochastic differential equations (Li et al., 2020), for generative modeling (Yıldız et al., 2019; Grathwohl et al., 2019), or by moving the dynamics to a latent space (Rubanova et al., 2019). These models have mostly been demonstrated on low-dimensional observations, but some initial explorations also perform simple experiments on image time-series, e.g. by extrapolating rotating MNIST digits in (Yıldız et al., 2019). How they fare on more spatially complex data remains to be seen.

For glioma growth in particular it seems like a natural next step to phrase the problem as a learned diffusion equation, which to the

best of our knowledge has not been done yet. The reaction term (see Section 2.2 for an introduction) could be replaced by a small neural network, for example, and we imagine that the diffusion tensor could also be estimated from the available MRIs (if DTI was not performed anyway) in an end-to-end fashion. As we outline in Section 2.2, some works approximate the diffusion process with an eikonal equation, meaning an equation that describes a travelling wavefront (Konukoglu et al., 2007; Konukoglu et al., 2010a). Some recent works explore learning with eikonal equations (Lichtenstein et al., 2019; Smith et al., 2020; Waheed et al., 2020) and it would interesting to see if these contributions can be leveraged for glioma growth modeling.

A

APPENDIX

A.1 EFFICIENT EXPERT ANNOTATIONS IN INTERACTIVE SEGMENTATION

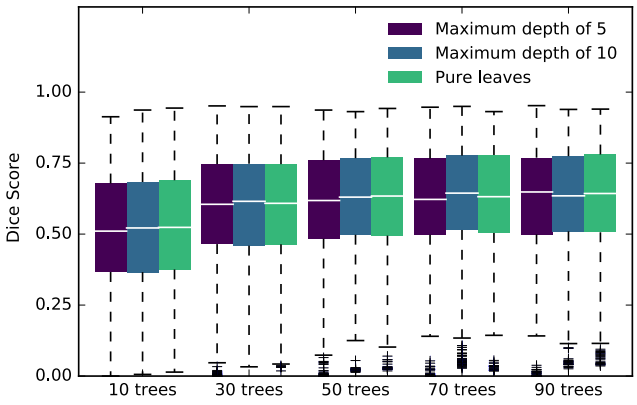

Figure A.1: Results of the parameter grid search. We repeated our experiments with different combinations of the number of trees and the maximum depth of the forest. The results here show the median Dice score with boxes extending from lower to upper quartile values for the best performing method MISCLASS-B over the interval from 10 to 30 interactions, taking into consideration data from necrotic, edema and enhancing regions. The depth of the forest has little influence on the results, while they seem to improve with the number of trees, especially from 10 to 30. However the differences are not statistically significant.

Table A.1: Pairwise comparison of all interactive annotation methods. Please see Table 3.1 for more information.

Methods	Edema			Non-enhancing Abnormalities		
	Statistic	p	ΔMedian	Statistic	p	ΔMedian
UNCERTAIN v MISCLASS	**13**	**<0.001**	**−0.038**	88	0.526	−0.031
UNCERTAIN v MISCLASS-B	13	**<0.001**	**−0.024**	**10**	**<0.001**	**−0.114**
UNCERTAIN v UNCERTAIN-MB	58	0.079	−0.002	**4**	**<0.001**	**−0.176**
UNCERTAIN v CERTAIN-MB	33	0.007	0.100	45	0.025	−0.062
MISCLASS v MISCLASS-B	62	0.108	0.015	**15**	**<0.001**	**−0.086**
MISCLASS v UNCERTAIN-MB	36	0.010	0.040	**11**	**<0.001**	**−0.145**
MISCLASS v CERTAIN-MB	**0**	**<0.001**	**0.138**	96	0.737	−0.031
MISCLASS-B v UNCERTAIN-MB	32	0.006	0.025	84	0.433	−0.061
MISCLASS-B v CERTAIN-MB	**0**	**<0.001**	**0.123**	**12**	**<0.001**	**0.052**
UNCERTAIN-MB v CERTAIN-MB	**1**	**<0.001**	**0.098**	**12**	**<0.001**	**0.114**

A.2 A MOTIVATING EXAMPLE: TUMOR VOLUME PREDICTION

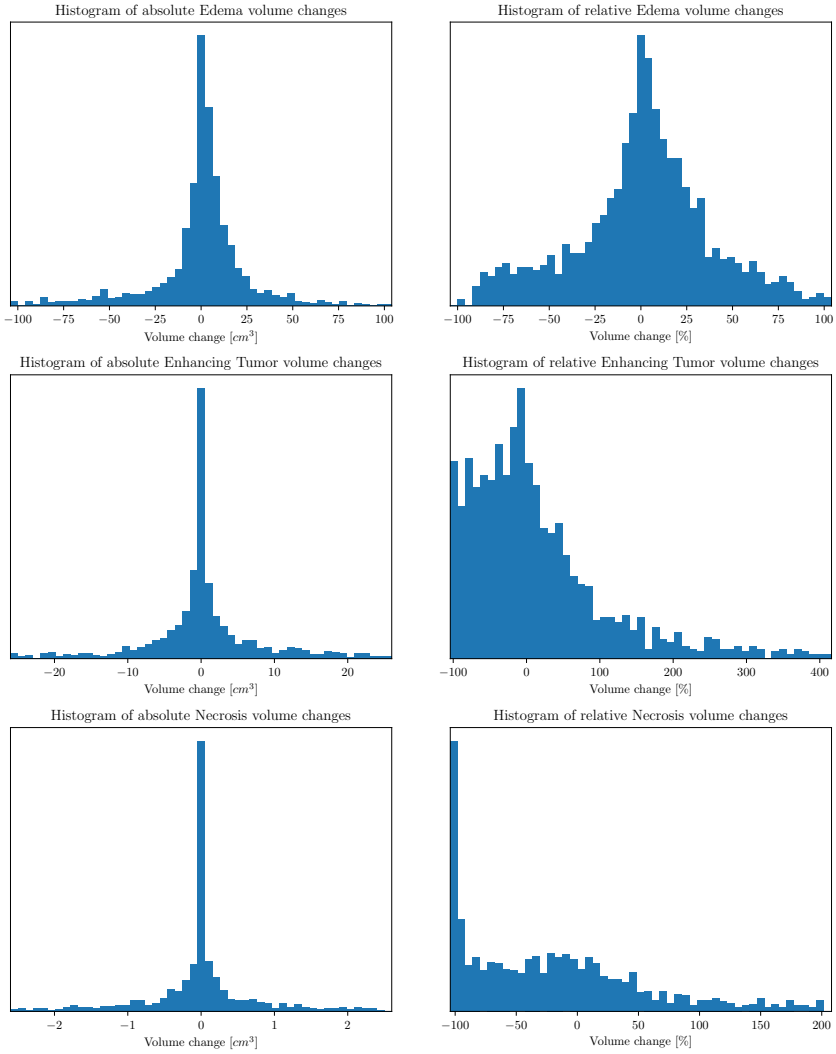

Figure A.2: Changes in tumor size in the dataset for tumor tumor tissue types. The outermost bins collect all points outside of the given range. For the enhancing tumor region, the distribution of absolute changes is shifted to the negative growth region, an indication of the efficacy of the administered treatment.

A.3 FREQUENCY DECOMPOSITION IN NEURAL PROCESSES

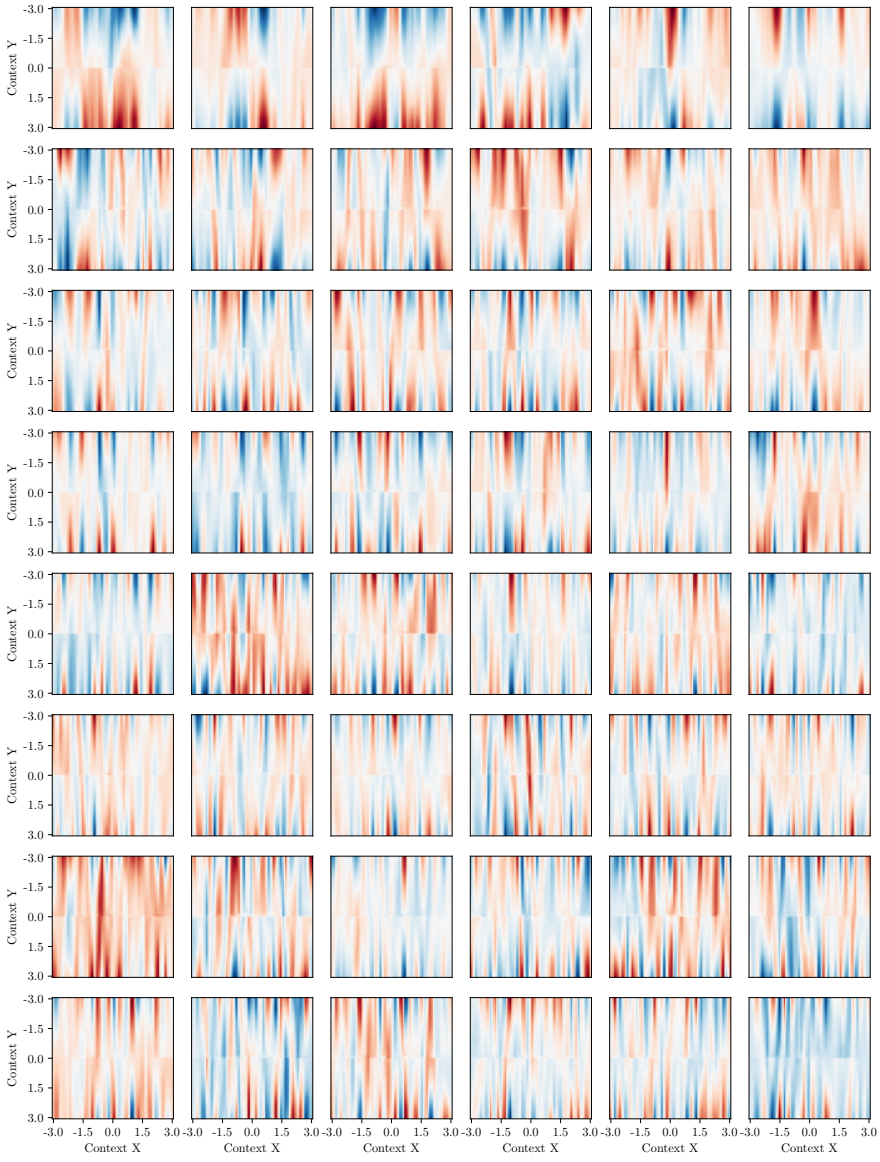

Figure A.3: Influence of the context on the learned representations in a CNP with $D_r = 128$ and data coming from Fourier series. X refers to the input space (i.e. time), Y to the output space. These are the first 48 representations ordered by their average Fourier components at $y = -3$ and $y = 3$ (left-to-right, top-to-bottom). Note that each panel is normalized separately, so color values are not comparable. This is the same as Figure 6.6 with different data. We find again that CNP performs a decomposition of the signal space into different frequencies, but compared to the GP data the frequencies are much higher.

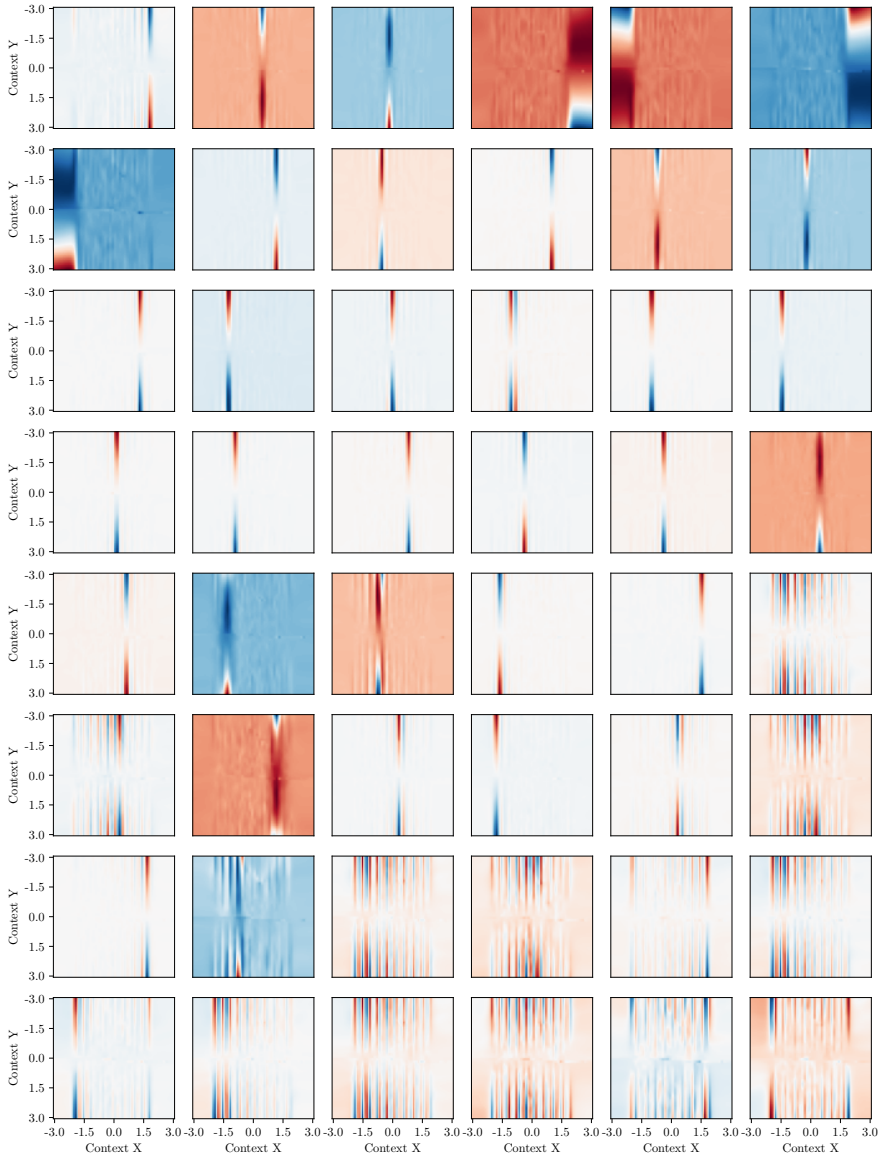

Figure A.4: Influence of the context on the learned representations in a NP with $D_r = 128$ and data coming from Fourier series. X refers to the input space (i.e. time), Y to the output space. These are the first 48 representations ordered by their average Fourier components at $y = -3$ and $y = 3$ (left-to-right, top-to-bottom). Note that each panel is normalized separately, so color values are not comparable. This is the same as Figure 6.8 with different data. Even though a representation of Fourier series data in frequency space would be beneficial, the NP learns a spatial partitioning of the signal space which is much sharper compared to the GP data.

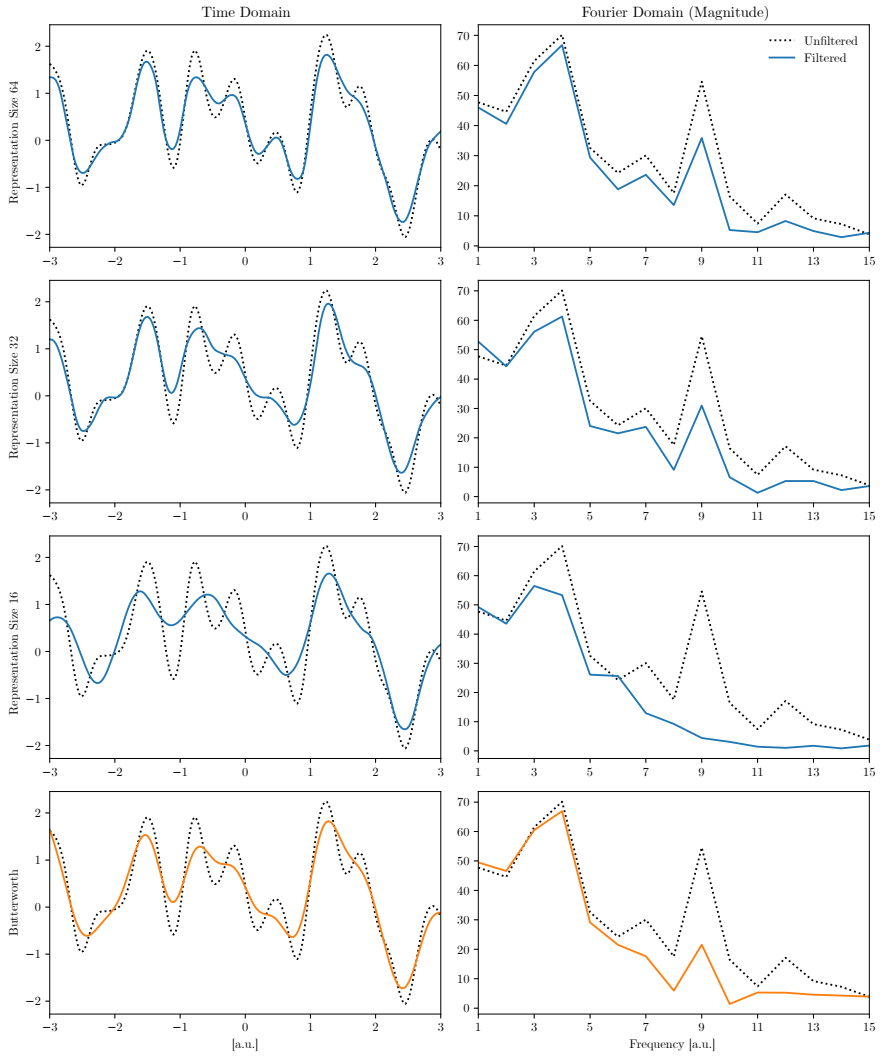

Figure A.5: Example reconstructions of GP data in NPs with varying representation sizes. A decrease in representation size leads to the omission of higher frequency components. In other words, the NP acts like a low-pass filter. For comparison, we also show a simple 3^{rd} order Butterworth filter (Butterworth, 1930) with a cutoff frequency manually selected for visual similarity to the $D_r = 16$ CNP model. Compared to the CNP in Figure 6.3, the NP exhibits stronger dampening, with the $D_r = 32$ NP being very similar to the $D_r = 16$ CNP.

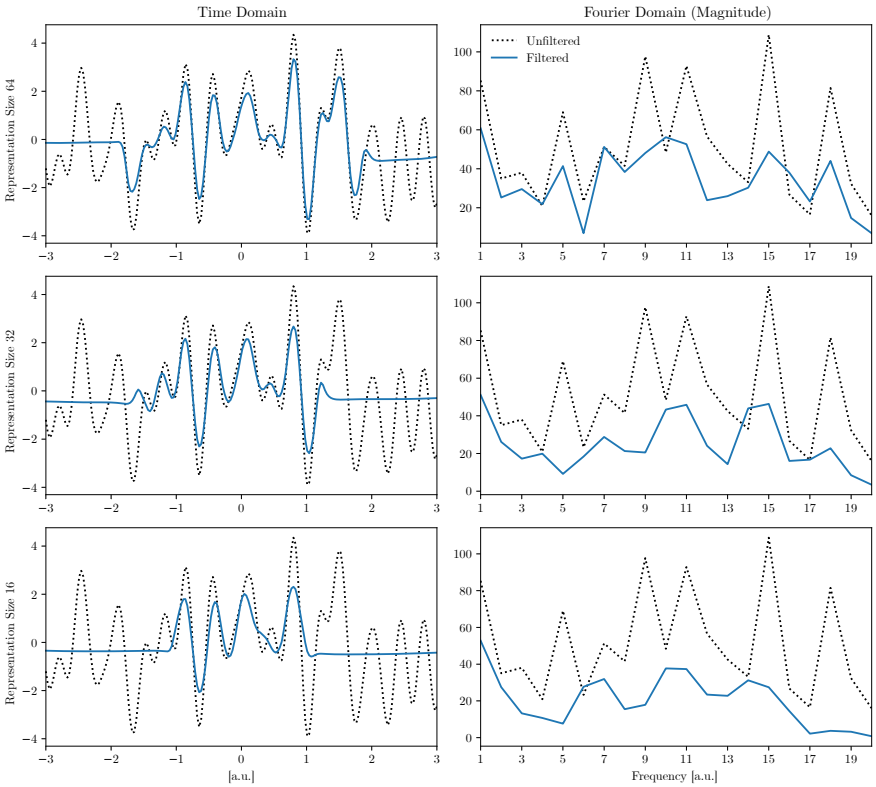

Figure A.6: Example reconstructions of Fourier data in NPs with varying representation sizes. A decrease in representation size leads to the model ignoring the outer regions of the input space as well as suppressing some higher frequency content. The corresponding figure for a CNP is Figure 6.5.

A.4 GP-CONVCNP

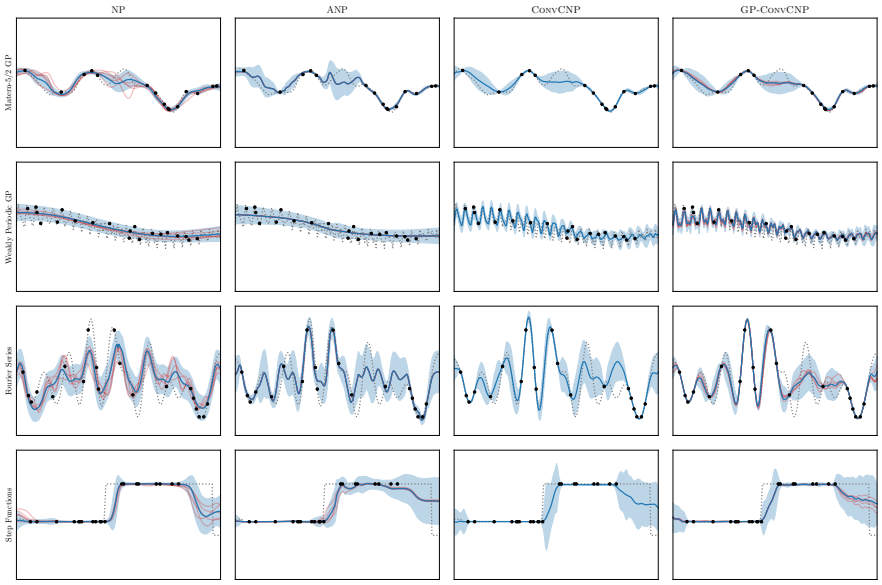

Figure A.7: These are additional examples for the synthetic data presented in Chapter 7. The figure is otherwise identical to Figure 7.2.

A.5 ATTENTIVE SEGMENTATION PROCESSES

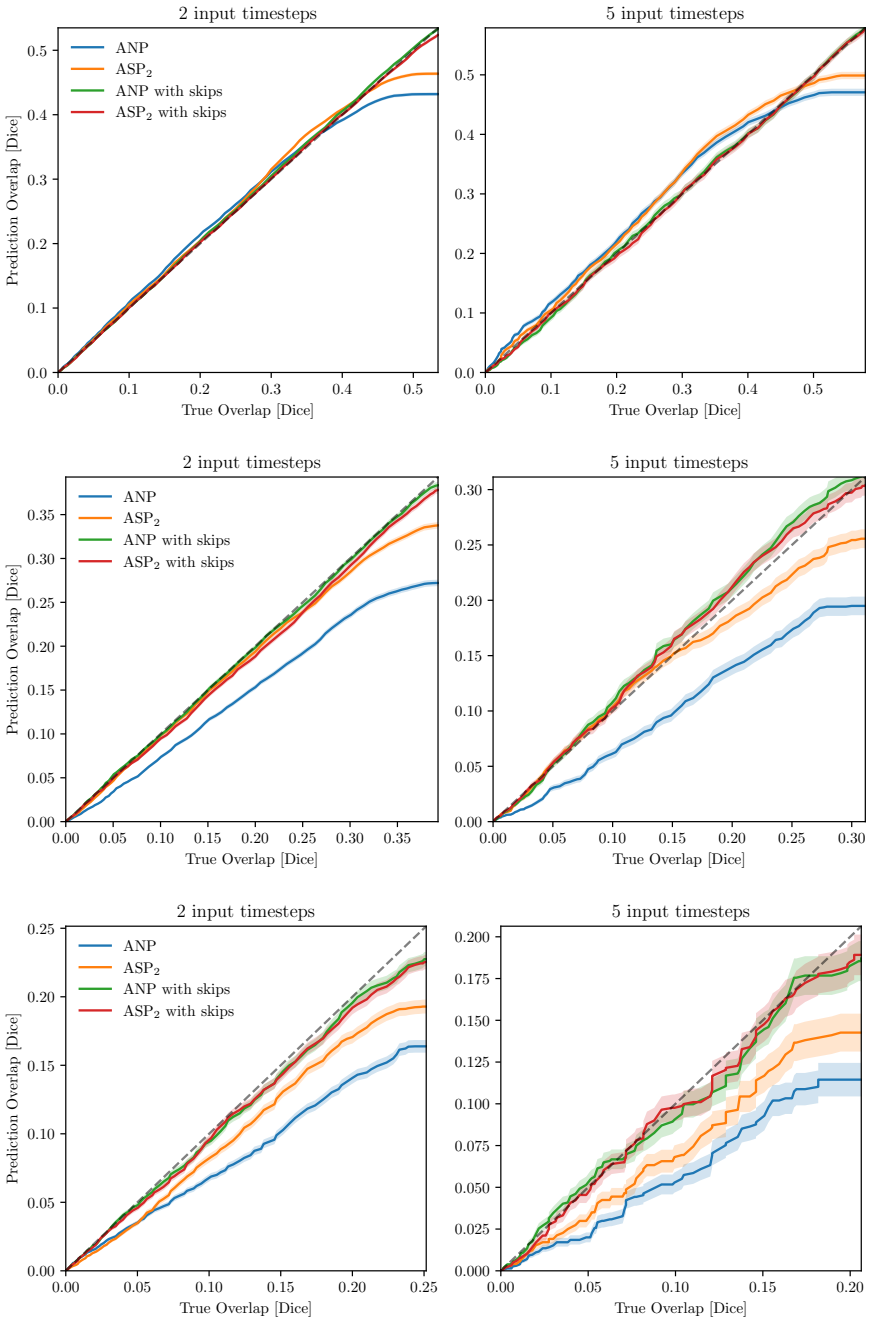

Figure A.8: Predictive Dice for ANP and ASP$_2$, for two input timepoints (left) and 5 input timepoints (right). This is the same as Figure 8.5, but for the individual tumor classes, with the top row showing edema, the middle row showing enhancing tumor, and the bottom row showing necrosis.

BIBLIOGRAPHY

Abragam, A. (1983). *Principles of Nuclear Magnetism*. International Series of Monographs on Physics. Oxford University Press. 614 pp.

Agustsson, Eirikur, Jasper R. Uijlings, and Vittorio Ferrari (2019). "Interactive Full Image Segmentation by Considering All Regions Jointly". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11614–11623.

Akbari, Hamed, Luke Macyszyn, Xiao Da, Michel Bilello, Ronald L. Wolf, Maria Martinez-Lage, George Biros, Michelle Alonso-Basanta, Donald M. O'Rourke, and Christos Davatzikos (2016). "Imaging Surrogates of Infiltration Obtained Via Multiparametric Imaging Pattern Analysis Predict Subsequent Location of Recurrence of Glioblastoma". In: *Neurosurgery* 78.4, pp. 572–580.

Ali, S. M. and S. D. Silvey (1966). "A General Class of Coefficients of Divergence of One Distribution from Another". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 28.1, pp. 131–142.

An, G. (1996). "The Effects of Adding Noise During Backpropagation Training on a Generalization Performance". In: *Neural Computation* 8.3, pp. 643–674.

Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla (2015). "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *arXiv:1511.00561 [cs]*.

Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla (2017). "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Bakas, Spyridon et al. (2019). "Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge". In: *arXiv:1811.02629 [cs, stat]*.

Bauer, Stefan, Roland Wiest, Lutz-P. Nolte, and Mauricio Reyes (2013). "A survey of MRI-based medical image analysis for brain tumor studies". In: *Physics in Medicine & Biology* 58.13, R97–R129.

Baumgartner, Christian F., Kerem C. Tezcan, Krishna Chaitanya, Andreas M. Hötker, Urs J. Muehlematter, Khoschy Schawkat, Anton S. Becker, Olivio Donati, and Ender Konukoglu (2019). "PHiSeg: Capturing Uncertainty in Medical Image Segmentation". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*.

Beede, Emma, Elizabeth Baylor, Fred Hersch, Anna Iurchenko, Lauren Wilcox, Paisan Ruamviboonsuk, and Laura M. Vardoulakis (2020). "A Human-Centered Evaluation of a Deep Learning System Deployed in Clinics for the Detection of Diabetic Retinopathy". In:

2020 CHI Conference on Human Factors in Computing Systems, pp. 1–12.

Beer, Randall D. (1997). "The dynamics of adaptive behavior: A research program". In: *Robotics and Autonomous Systems*. Practice and Future of Autonomous Agents 20.2, pp. 257–289.

Beer, Randall D. and John C. Gallagher (1992). "Evolving Dynamical Neural Networks for Adaptive Behavior". In: *Adaptive Behavior* 1.1, pp. 91–122.

Bernstein, Matt A., Kevin F. King, and Xiaohong Joe Zhou (2004). *Handbook of MRI Pulse Sequences*. Elsevier.

Bishop, Christopher M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Inc.

Bishop, Christopher M. (2006). *Pattern recognition and machine learning*. Information science and statistics. Springer. 738 pp.

Blei, David M., Alp Kucukelbir, and Jon D. McAuliffe (2017). "Variational Inference: A Review for Statisticians". In: *Journal of the American Statistical Association* 112.518, pp. 859–877.

Blundell, Charles, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra (2015). "Weight Uncertainty in Neural Networks". In: *International Conference on Machine Learning*, pp. 1613–1622.

Bondiau, Pierre-Yves, Olivier Clatz, Maxime Sermesant, Pierre-Yves Marcy, Herve Delingette, Marc Frenay, and Nicholas Ayache (2008). "Biocomputing: numerical simulation of glioblastoma growth using diffusion tensor imaging". In: *Physics in Medicine & Biology* 53.4, p. 879.

Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik (1992). "A training algorithm for optimal margin classifiers". In: *Annual Workshop on Computational Learning Theory*, pp. 144–152.

Branco, J. R., J. A. Ferreira, and Paula de Oliveira (2014). "Mathematical modeling of efficient protocols to control glioma growth". In: *Mathematical Biosciences* 255, pp. 83–90.

Breiman, Leo (2001). "Random forests". In: *Machine Learning* 45.1, pp. 5–32.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020). "Language Models are Few-Shot Learners". In: *arXiv:2005.14165 [cs]*.

Brugnara, Gianluca, Fabian Isensee, Ulf Neuberger, David Bonekamp, Jens Petersen, Ricarda Diem, Brigitte Wildemann, Sabine Heiland, Wolfgang Wick, Martin Bendszus, Klaus Maier-Hein, and Philipp

Kickingereder (2020). "Automated volumetric assessment with artificial neural networks might enable a more accurate assessment of disease burden in patients with multiple sclerosis". In: *European Radiology* 30.4, pp. 2356–2364.

Burgess, P. K., P. M. Kulesa, J. D. Murray, and E. C. Alvord (1997). "The interaction of growth rates and diffusion coefficients in a three-dimensional mathematical model of gliomas". In: *Journal of Neuropathology and Experimental Neurology* 56.6, pp. 704–713.

Burnet, Neil G, Simon J Thomas, Kate E Burton, and Sarah J Jefferies (2004). "Defining the tumour and target volumes for radiotherapy". In: *Cancer Imaging* 4.2, pp. 153–161.

Butterworth, Stephen (1930). "On the theory of filter amplifiers". In: *Wireless Engineer* 7.6, pp. 536–541.

Calandra, Roberto, Jan Peters, Carl Edward Rasmussen, and Marc Peter Deisenroth (2016). "Manifold Gaussian Processes for Regression". In: *International Joint Conference on Neural Networks*.

Carion, Nicolas, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko (2020). "End-to-End Object Detection with Transformers". In: *arXiv:2005.12872 [cs]*.

Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille (2015). "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs". In: *International Conference on Learning Representations*.

Chen, Mark, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever (2020a). *Generative Pretraining from Pixels*. Tech. rep.

Chen, Ricky T. Q., Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud (2018). "Neural Ordinary Differential Equations". In: *Advances in Neural Information Processing Systems*, pp. 6571–6583.

Chen, Ting, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton (2020b). "A Simple Framework for Contrastive Learning of Visual Representations". In: *arXiv:2002.05709 [cs, stat]*.

Cheng, Ching-An and Byron Boots (2017). "Variational Inference for Gaussian Process Models with Linear Complexity". In: *Advances in Neural Information Processing Systems*, pp. 5184–5194.

Chicoine, Michael R. and Daniel L. Silbergeld (1995). "Assessment of brain tumor cell motility in vivo and in vitro". In: *Journal of Neurosurgery* 82.4, pp. 615–622.

Child, Rewon, Scott Gray, Alec Radford, and Ilya Sutskever (2019). "Generating Long Sequences with Sparse Transformers". In: *URL https://openai.com/blog/sparse-transformers*.

Chyzhyk, Darya, Rosalía Dacosta-Aguayo, Maria Mataró, and Manuel Graña (2015). "An active learning approach for stroke lesion segmentation on multimodal MRI data". In: *Neurocomputing* 150, pp. 26–36.

Çiçek, Özgün, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger (2016). "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 424–432.

Clatz, Olivier, Maxime Sermesant, Pierre-Yves Bondiau, Hervé Delingette, Simon K. Warfield, Grégoire Malandain, and Nicholas Ayache (2005). "Realistic Simulation of the 3D Growth of Brain Tumors in MR Images Coupling Diffusion with Biomechanical Deformation". In: *IEEE Transactions on Medical Imaging* 24.10, pp. 1334–1346.

Cordts, Marius, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele (2016). "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine Learning* 20.3, pp. 273–297.

Crum, W.R., O. Camara, and D.L.G. Hill (2006). "Generalized Overlap Measures for Evaluation and Validation in Medical Image Analysis". In: *IEEE Transactions on Medical Imaging* 25.11, pp. 1451–1461.

Cruywagen, Gerhard C., Diana E. Woodward, Philippe Tracqui, Grace T. Bartoo, J. D. Murray, and Ellsworth C. Alvord (1995). "The modelling of diffusive tumours". In: *Journal of Biological Systems* 03.4, pp. 937–945.

Csiszár, I. (1963). "Eine informationstheoretische Ungleichung und ihre Anwendung auf den Beweis der Ergodizitat von Markoffschen Ketten". In: *Magyar. Tud. Akad. Mat. Kutato Int. Kozl.* 8, pp. 85–108.

Cybenko, G. (1989). "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2.4, pp. 303–314.

Deisenroth, Marc and Jun Wei Ng (2015). "Distributed Gaussian Processes". In: *International Conference on Machine Learning*, pp. 1481–1490.

Dice, Lee R. (1945). "Measures of the Amount of Ecologic Association Between Species". In: *Ecology* 26.3, pp. 297–302.

Dupont, Emilien, Miguel Angel Bautista, Alex Colburn, Aditya Sankar, Carlos Guestrin, Josh Susskind, and Qi Shan (2020). "Equivariant Neural Rendering". In: *arXiv:2006.07630 [cs, stat]*.

Ellingson, Benjamin M., Hyun J. Kim, Davis C. Woodworth, Whitney B. Pope, Jonathan N. Cloughesy, Robert J. Harris, Albert Lai, Phioanh L. Nghiemphu, and Timothy F. Cloughesy (2014). "Recurrent glioblastoma treated with bevacizumab: contrast-enhanced T1-weighted subtraction maps improve tumor delineation and aid prediction of survival in a multicenter clinical trial". In: *Radiology* 271.1, pp. 200–210.

Engelcke, Martin, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner (2020). "GENESIS: Generative Scene Inference and Sampling with Object-Centric Latent Representations". In: *International Conference on Learning Representations*.

Engwer, Christian, Thomas Hillen, Markus Knappitsch, and Christina Surulescu (2015). "Glioma follow white matter tracts: a multiscale DTI-based model". In: *Journal of Mathematical Biology* 71.3, pp. 551–582.

Eslami, S. M. Ali, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis (2018). "Neural scene representation and rendering". In: *Science* 360.6394, pp. 1204–1210.

Esmaeili, Morteza, Anne Line Stensjøen, Erik Magnus Berntsen, Ole Solheim, and Ingerid Reinertsen (2018). "The Direction of Tumour Growth in Glioblastoma Patients". In: *Scientific Reports* 8.1, p. 1199.

Esteva, Andre, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun (2017). "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542.7639, pp. 115–118.

European Commission (2018). *Artificial Intelligence for Europe*. Communication. URL: https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=51625 (visited on 09/25/2020).

Fauw, Jeffrey De, Joseph R. Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, George van den Driessche, Balaji Lakshminarayanan, Clemens Meyer, Faith Mackinder, Simon Bouton, Kareem Ayoub, Reena Chopra, Dominic King, Alan Karthikesalingam, Cían O. Hughes, Rosalind Raine, Julian Hughes, Dawn A. Sim, Catherine Egan, Adnan Tufail, Hugh Montgomery, Demis Hassabis, Geraint Rees, Trevor Back, Peng T. Khaw, Mustafa Suleyman, Julien Cornebise, Pearse A. Keane, and Olaf Ronneberger (2018). "Clinically applicable deep learning for diagnosis and referral in retinal disease". In: *Nature Medicine* 24.9, pp. 1342–1350.

Fedorov, Andriy, Reinhard Beichel, Jayashree Kalpathy-Cramer, Julien Finet, Jean-Christophe Fillion-Robin, Sonia Pujol, Christian Bauer, Dominique Jennings, Fiona Fennessy, Milan Sonka, John Buatti, Stephen Aylward, James V. Miller, Steve Pieper, and Ron Kikinis (2012). "3D Slicer as an image computing platform for the Quantitative Imaging Network". In: *Magnetic Resonance Imaging* 30.9, pp. 1323–1341.

Fernandez-Delgado, Manuel, Eva Cernadas, Senen Barro, and Dinani Amorim (2014). "Do we need hundreds of classifiers to solve real world classification problems?" In: *Journal of Machine Learning Research* 15.1, pp. 3133–3181.

Floridi, Luciano (2020). "AI and Its New Winter: from Myths to Realities". In: *Philosophy & Technology* 33.1, pp. 1–3.

Gal, Yarin and Zoubin Ghahramani (2016a). "Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference". In: *International Conference on Learning Representations – Workshop*.

Gal, Yarin and Zoubin Ghahramani (2016b). "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *International Conference on Machine Learning*, pp. 1050–1059.

Gallant and White (1988). "There exists a neural network that does not make avoidable mistakes". In: *IEEE International Conference on Neural Networks*. Vol. 1, pp. 657–664.

Gardner, Jacob, Geoff Pleiss, Ruihan Wu, Kilian Weinberger, and Andrew Wilson (2018). "Product Kernel Interpolation for Scalable Gaussian Processes". In: *International Conference on Artificial Intelligence and Statistics*, pp. 1407–1416.

Garnelo, Marta, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami (2018a). "Conditional Neural Processes". In: *International Conference on Machine Learning*, pp. 1704–1713.

Garnelo, Marta, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh (2018b). "Neural Processes". In: *International Conference on Machine Learning – Workshop on Theoretical Foundations and Applications of Deep Generative Models*.

Gaw, Nathan, Andrea Hawkins-Daarud, Leland S. Hu, Hyunsoo Yoon, Lujia Wang, Yanzhe Xu, Pamela R. Jackson, Kyle W. Singleton, Leslie C. Baxter, Jennifer Eschbacher, Ashlyn Gonzales, Ashley Nespodzany, Kris Smith, Peter Nakaji, J. Ross Mitchell, Teresa Wu, Kristin R. Swanson, and Jing Li (2019). "Integration of machine learning and mechanistic models accurately predicts variation in cell density of glioblastoma using multiparametric MRI". In: *Scientific Reports* 9.1, p. 10063.

Giese, A., R. Bjerkvig, M. E. Berens, and M. Westphal (2003). "Cost of migration: invasion of malignant gliomas and implications for treatment". In: *Journal of Clinical Oncology* 21.8, pp. 1624–1636.

Giese, A., L. Kluwe, B. Laube, H. Meissner, M. E. Berens, and M. Westphal (1996). "Migration of human glioma cells on myelin". In: *Neurosurgery* 38.4, pp. 755–764.

Goch, Caspar J., Jasmin Metzger, Martin Hettich, André Klein, Tobias Norajitra, Michael Götz, Jens Petersen, Klaus H. Maier-Hein, and Marco Nolden (2018). "Automated Containerized Medical Image Processing Based on MITK and Python". In: *Bildverarbeitung für die Medizin*, pp. 315–315.

Goetz, M., C. Weber, F. Binczyk, J. Polanska, R. Tarnawski, B. Bobek-Billewicz, U. Koethe, J. Kleesiek, B. Stieltjes, and K. H. Maier-Hein (2016). "DALSA: Domain Adaptation for Supervised Learning From Sparsely Annotated MR Images". In: *IEEE Transactions on Medical Imaging* 35.1, pp. 184–196.

Goodenberger, McKinsey L. and Robert B. Jenkins (2012). "Genetics of adult glioma". In: *Cancer Genetics* 205.12, pp. 613–621.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. MIT Press.

Gordon, Jonathan, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner (2020). "Convolutional Conditional Neural Processes". In: *International Conference on Learning Representations*.

Grathwohl, Will, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud (2019). "FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models". In: *International Conference on Learning Representations*.

Graves, Alex (2011). "Practical Variational Inference for Neural Networks". In: *Advances in Neural Information Processing Systems*, pp. 2348–2356.

Gregor, Karol, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra (2015). "DRAW: A Recurrent Neural Network For Image Generation". In: *International Conference on Machine Learning*. Vol. 37, pp. 1462–1471.

Gulshan, Varun, Lily Peng, Marc Coram, Martin C. Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, Ramasamy Kim, Rajiv Raman, Philip C. Nelson, Jessica L. Mega, and Dale R. Webster (2016). "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs". In: *JAMA* 316.22, pp. 2402–2410.

Haacke, E. Mark, Robert W. Brown, Michael R. Thompson, and Ramesh Venkatesan (1999). *Magnetic Resonance Imaging: Physical Principles and Sequence Design*. 1st ed. New York: John Wiley & Sons. 944 pp.

Hanson, Floyd B. and Charles Tier (1982). "A stochastic model of tumor growth". In: *Mathematical Biosciences* 61.1, pp. 73–100.

Harpold, Hana L. P., Ellsworth C. Alvord, and Kristin R. Swanson (2007). "The evolution of mathematical modeling of glioma proliferation and invasion". In: *Journal of Neuropathology and Experimental Neurology* 66.1, pp. 1–9.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2016). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. 2nd ed. Springer. 767 pp.

He, Kaiming, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick (2020). "Momentum Contrast for Unsupervised Visual Representation Learning". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9726–9735.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017). "Mask R-CNN". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep Residual Learning for Image Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Heimann, Tobias and Hans-Peter Meinzer (2009). "Statistical shape models for 3D medical image segmentation: A review". In: *Medical Image Analysis* 13.4, pp. 543–563.

Hénaff, Olivier J., Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord (2019). "Data-Efficient Image Recognition with Contrastive Predictive Coding". In: *arXiv:1905.09272 [cs]*.

Hensman, James, Nicolò Fusi, and Neil D. Lawrence (2013). "Gaussian processes for Big data". In: *Conference on Uncertainty in Artificial Intelligence*, pp. 282–290.

Hensman, James, Alexander Matthews, and Zoubin Ghahramani (2015). "Scalable Variational Gaussian Process Classification". In: *International Conference on Artificial Intelligence and Statistics*, pp. 351–360.

Hernández-Lobato, José Miguel and Ryan P. Adams (2015). "Probabilistic backpropagation for scalable learning of Bayesian neural networks". In: *International Conference on Machine Learning*. Vol. 37, pp. 1861–1869.

Hewitt, C. Gordon (1921). *The conservation of the wild life of Canada*. C. Scribner's Sons.

Hochreiter, Sepp, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber (2001). "Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies". In: *A Field Guide to Dynamical Recurrent Networks*. IEEE, pp. 237–243.

Hogea, Cosmina, Christos Davatzikos, and George Biros (2007). "Modeling glioma growth and mass effect in 3D MR images of the brain". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Vol. 10, pp. 642–650.

Hogea, Cosmina, Christos Davatzikos, and George Biros (2008). "An image-driven parameter estimation problem for a reaction–diffusion glioma growth model with mass effects". In: *Journal of Mathematical Biology* 56.6, pp. 793–825.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (1989). "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5, pp. 359–366.

Hron, Jiri, Alexander G. de G. Matthews, and Zoubin Ghahramani (2017). "Variational Gaussian Dropout is not Bayesian". In: *Advances in Neural Information Processing Systems – Bayesian Deep Learning Workshop*.

Hu, Yang, Andrea Soltoggio, Russell Lock, and Steve Carter (2019). "A fully convolutional two-stream fusion network for interactive image segmentation". In: *Neural Networks* 109, pp. 31–42.

Iglesias, Juan Eugenio and Mert R. Sabuncu (2015). "Multi-atlas segmentation of biomedical images: A survey". In: *Medical Image Analysis* 24.1, pp. 205–219.

Isensee, Fabian, Paul F. Jäger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein (2020). "Automated Design of Deep Learning Methods for Biomedical Image Segmentation". In: *Nature Methods (accepted)*.

Isensee, Fabian, Jens Petersen, Andre Klein, David Zimmerer, Paul F. Jaeger, Simon Kohl, Jakob Wasserthal, Gregor Koehler, Tobias Norajitra, Sebastian Wirkert, and Klaus H. Maier-Hein (2018). "nnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation". In: *arXiv:1809.10486 [cs]*. Winning entry of the Medical Segmentation Decathlon.

Jackson, Philip T, Amir Atapour-Abarghouei, Stephen Bonner, Toby P Breckon, and Boguslaw Obara (2019). "Style Augmentation: Data Augmentation via Style Randomization". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) – Workshop*, pp. 83–92.

Jäger, Paul F., Sebastian Bickelhaupt, Frederik Bernd Laun, Wolfgang Lederer, Daniel Heidi, Tristan Anselm Kuder, Daniel Paech, David Bonekamp, Alexander Radbruch, Stefan Delorme, Heinz-Peter Schlemmer, Franziska Steudle, and Klaus H. Maier-Hein (2017). "Revealing Hidden Potentials of the q-Space Signal in Breast Cancer". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 664–671.

Jang, Won-Dong and Chang-Su Kim (2019). "Interactive Image Segmentation via Backpropagating Refinement Scheme". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5292–5301.

Jbabdi, Saâd, Emmanuel Mandonnet, Hugues Duffau, Laurent Capelle, Kristin Rae Swanson, Mélanie Pélégrini-Issac, Rémy Guillevin, and Habib Benali (2005). "Simulation of anisotropic growth of low-grade gliomas using diffusion tensor imaging". In: *Magnetic Resonance in Medicine* 54.3, pp. 616–624.

Jégou, Simon, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio (2017). "The One Hundred Layers Tiramisu:

Fully Convolutional DenseNets for Semantic Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, pp. 1175–1183.

Jenkinson, Mark, Christian F. Beckmann, Timothy E. J. Behrens, Mark W. Woolrich, and Stephen M. Smith (2012). "FSL". In: *NeuroImage* 62.2, pp. 782–790.

Jimenez Rezende, D., S. Mohamed, and D. Wierstra (2014). "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *International Conference on Machine Learning*.

Jimenez Rezende, Danilo, S. M. Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess (2016). "Unsupervised Learning of 3D Structure from Images". In: *Advances in Neural Information Processing Systems*, pp. 4996–5004.

Johnson, Derek R. and Brian Patrick O'Neill (2012). "Glioblastoma survival in the United States before and during the temozolomide era". In: *Journal of Neuro-Oncology* 107.2, pp. 359–364.

Karimi, Davood and Septimiu E. Salcudean (2020). "Reducing the Hausdorff Distance in Medical Image Segmentation With Convolutional Neural Networks". In: *IEEE Transactions on Medical Imaging* 39.2, pp. 499–513.

Kelly, Christopher J., Alan Karthikesalingam, Mustafa Suleyman, Greg Corrado, and Dominic King (2019). "Key challenges for delivering clinical impact with artificial intelligence". In: *BMC Medicine* 17.1, p. 195.

Kelly, Patrick J. and Cathy Hunt (1994). "The Limited Value of Cytoreductive Surgery in Elderly Patients with Malignant Gliomas". In: *Neurosurgery* 34.1, pp. 62–67.

Kickingereder, Philipp, Fabian Isensee, Irada Tursunova, Jens Petersen, Ulf Neuberger, David Bonekamp, Gianluca Brugnara, Marianne Schell, Tobias Kessler, Martha Foltyn, Inga Harting, Felix Sahm, Marcel Prager, Martha Nowosielski, Antje Wick, Marco Nolden, Alexander Radbruch, Jürgen Debus, Heinz-Peter Schlemmer, Sabine Heiland, Michael Platten, Andreas von Deimling, Martin J van den Bent, Thierry Gorlia, Wolfgang Wick, Martin Bendszus, and Klaus H Maier-Hein (2019). "Automated quantitative tumour response assessment of MRI in neuro-oncology with artificial neural networks: a multicentre, retrospective study". In: *The Lancet Oncology* 20.5, pp. 728–740.

Kim, Hyunjik, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh (2019). "Attentive Neural Processes". In: *International Conference on Learning Representations*.

Kingma, D. P. and J. Ba (2015). "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations*.

Kingma, D. P and M. Welling (2014). "Auto-Encoding Variational Bayes". In: *International Conference on Learning Representations*.

Kingma, Durk P, Tim Salimans, and Max Welling (2015). "Variational Dropout and the Local Reparameterization Trick". In: *Advances in Neural Information Processing Systems*, pp. 2575–2583.

Kitaev, Nikita, Lukasz Kaiser, and Anselm Levskaya (2020). "Reformer: The Efficient Transformer". In: *International Conference on Learning Representations*.

Kleesiek, Jens, Armin Biller, Gregor Urban, U. Koethe, Martin Bendszus, and F. Hamprecht (2014). "Ilastik for multi-modal brain tumor segmentation". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI) – BraTS Workshop*, pp. 12–17.

Kleesiek, Jens, Jens Petersen, Markus Döring, Klaus Maier-Hein, Ullrich Köthe, Wolfgang Wick, Fred A. Hamprecht, Martin Bendszus, and Armin Biller (2016). "Virtual Raters for Reproducible and Objective Assessments in Radiology". In: *Scientific Reports* 6, p. 25007.

Kohl, Simon A. A., Bernardino Romera-Paredes, Klaus H. Maier-Hein, Danilo Jimenez Rezende, S. M. Ali Eslami, Pushmeet Kohli, Andrew Zisserman, and Olaf Ronneberger (2019). "A Hierarchical Probabilistic U-Net for Modeling Multi-Scale Ambiguities". In: *Medical Imaging meets NeurIPS*.

Kohl, Simon, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R. Ledsam, Klaus Maier-Hein, S. M. Ali Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger (2018). "A Probabilistic U-Net for Segmentation of Ambiguous Images". In: *Advances in Neural Information Processing Systems*, pp. 6965–6975.

Kontschieder, Peter, Pushmeet Kohli, Jamie Shotton, and Antonio Criminisi (2013). "GeoF: Geodesic forests for learning coupled predictors". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 65–72.

Konukoglu, E., O. Clatz, B. H. Menze, B. Stieltjes, M. Weber, E. Mandonnet, H. Delingette, and N. Ayache (2010a). "Image Guided Personalization of Reaction-Diffusion Type Tumor Growth Models Using Modified Anisotropic Eikonal Equations". In: *IEEE Transactions on Medical Imaging* 29.1, pp. 77–95.

Konukoglu, Ender, Olivier Clatz, Pierre-Yves Bondiau, Hervé Delingette, and Nicholas Ayache (2010b). "Extrapolating glioma invasion margin in brain magnetic resonance images: suggesting new irradiation margins". In: *Medical Image Analysis* 14.2, pp. 111–125.

Konukoglu, Ender, Maxime Sermesant, Olivier Clatz, Jean-Marc Peyrat, Hervé Delingette, and Nicholas Ayache (2007). "A Recursive Anisotropic Fast Marching Approach to Reaction Diffusion Equation: Application to Tumor Growth Modeling". In: *Information Processing in Medical Imaging*. Vol. 4584, pp. 686–699.

Konyushkova, Ksenia, Raphael Sznitman, and Pascal Fua (2015). "Introducing Geometry in Active Learning for Image Segmentation". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 2974–2982.

Kotelnikov, V. (1933). "On the Transmission Capacity of the "Ether" and Wire in Electrocommunications". In: *Proceedings of the first All-Union Conference on the technological reconstruction of the communications sector and the development of low-current engineering*.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems*, pp. 1097–1105.

Kumar, Ananya, S. M. Ali Eslami, Danilo J. Rezende, Marta Garnelo, Fabio Viola, Edward Lockhart, and Murray Shanahan (2018). "Consistent Jumpy Predictions for Videos and Scenes". In: *Advances in Neural Information Processing Systems – Bayesian Deep Learning Workshop*.

Kuroiwa, Toshihiko, M. Ueki, Q. Chen, H. Suemasu, I. Taniguchi, and R. Okeda (1994). "Biomechanical Characteristics of Brain Edema: the Difference Between Vasogenic-Type and Cytotoxic-Type Edema". In: *Brain Edema IX*. Acta Neurochirurgica, pp. 158–161.

Kyriacou, S.K., C. Davatzikos, S.J. Zinreich, and R.N. Bryan (1999). "Nonlinear elastic registration of brain images with tumor pathology using a biomechanical model [MRI]". In: *IEEE Transactions on Medical Imaging* 18.7, pp. 580–592.

Le, Hoang, Long Mai, Brian Price, Scott Cohen, Hailin Jin, and Feng Liu (2018a). "Interactive Boundary Prediction for Object Selection". In: *European Conference on Computer Vision (ECCV)*, pp. 20–36.

Lê, M., H. Delingette, J. Kalpathy-Cramer, E. R. Gerstner, T. Batchelor, J. Unkelbach, and N. Ayache (2017). "Personalized Radiotherapy Planning Based on a Computational Tumor Growth Model". In: *IEEE Transactions on Medical Imaging* 36.3, pp. 815–825.

Le, Quoc, Tamas Sarlos, and Alexander Smola (2013). "Fastfood - Computing Hilbert Space Expansions in loglinear time". In: *International Conference on Machine Learning*, pp. 244–252.

Le, Tuan Anh, Hyunjik Kim, Marta Garnelo, Dan Rosenbaum, Jonathan Schwarz, and Yee Whye Teh (2018b). "Empirical Evaluation of Neural Process Objectives". In: *Advances in Neural Information Processing Systems – Bayesian Deep Learning Workshop*.

Leigh, Egbert G (1968). *Ecological role of Volterra's equations*. Lectures on mathematics in the life sciences. Princeton University.

Lenczner, G., B. Le Saux, N. Luminari, A. Chan-Hon-Tong, and G. Le Besnerais (2020). "DISIR: DEEP IMAGE SEGMENTATION WITH INTERACTIVE REFINEMENT". In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* V-2-2020, pp. 877–884.

Li, Xuechen, Ting-Kam Leonard Wong, Ricky T. Q. Chen, and David Duvenaud (2020). "Scalable Gradients for Stochastic Differential Equations". In: *International Conference on Artificial Intelligence and Statistics*, pp. 3870–3882.

Li, Zhuwen, Qifeng Chen, and Vladlen Koltun (2018). "Interactive Image Segmentation With Latent Diversity". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 577–585.

Liang, Zhi-Pei and Paul C. Lauterbur (2000). *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. SPIE Optical Engineering Press. 416 pp.

Lichtenstein, Moshe, Gautam Pai, and Ron Kimmel (2019). "Deep Eikonal Solvers". In: *Scale Space and Variational Methods in Computer Vision*, pp. 38–50.

Liew, JunHao, Yunchao Wei, Wei Xiong, Sim-Heng Ong, and Jiashi Feng (2017). "Regional Interactive Image Segmentation Networks". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 2746–2754.

Lighthill, M. J. (1958). *An Introduction to Fourier Analysis and Generalised Functions*. Cambridge Monographs on Mechanics. Cambridge University Press. DOI: 10.1017/CBO9781139171427.

Lin, Di, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun (2016). "ScribbleSup: Scribble-Supervised Convolutional Networks for Semantic Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3159–3167.

Lipková, Jana, Panagiotis Angelikopoulos, Stephen Wu, Esther Alberts, Benedikt Wiestler, Christian Diehl, Christine Preibisch, Thomas Pyka, Stephanie E. Combs, Panagiotis Hadjidoukas, Koen Van Leemput, Petros Koumoutsakos, John Lowengrub, and Bjoern Menze (2019). "Personalized Radiotherapy Design for Glioblastoma: Integrating Mathematical Tumor Models, Multimodal Scans, and Bayesian Inference". In: *IEEE Transactions on Medical Imaging* 38.8, pp. 1875–1884.

Lipton, Zachary C. (2018). "The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3, pp. 31–57.

Lipton, Zachary C., John Berkowitz, and Charles Elkan (2015). "A Critical Review of Recurrent Neural Networks for Sequence Learning". In: *arXiv:1506.00019 [cs]*.

Lipton, Zachary C. and Jacob Steinhardt (2018). "Troubling Trends in Machine Learning Scholarship". In: *International Conference on Machine Learning Debates Workshop*.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440.

Louis, David N., Arie Perry, Guido Reifenberger, Andreas von Deimling, Dominique Figarella-Branger, Webster K. Cavenee, Hiroko Ohgaki, Otmar D. Wiestler, Paul Kleihues, and David W. Ellison (2016). "The 2016 World Health Organization Classification of

Tumors of the Central Nervous System: a summary". In: *Acta Neuropathologica* 131.6, pp. 803–820.

Louizos, Christos, Xiahan Shi, Klamer Schutte, and Max Welling (2019). "The Functional Neural Process". In: *Advances in Neural Information Processing Systems*, pp. 8743–8754.

Mahadevan, Sabarinath, Paul Voigtlaender, and Bastian Leibe (2018). "Iteratively Trained Interactive Segmentation". In: *British Machine Vision Conference*.

Maiora, Josu, Borja Ayerdi, and Manuel Graña (2014). "Random forest active learning for AAA thrombus segmentation in computed tomography angiography images". In: *Neurocomputing*, pp. 71–77.

Maninis, K.-K., S. Caelles, J. Pont-Tuset, and L. Van Gool (2018). "Deep Extreme Cut: From Extreme Points to Object Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 616–625.

Mann, Justin, Rohan Ramakrishna, Rajiv Magge, and A. Gabriella Wernicke (2018). "Advances in Radiotherapy for Glioblastoma". In: *Frontiers in Neurology* 8.

Marcus, Daniel S., Timothy R. Olsen, Mohana Ramaratnam, and Randy L. Buckner (2007). "The Extensible Neuroimaging Archive Toolkit: an informatics platform for managing, exploring, and sharing neuroimaging data". In: *Neuroinformatics* 5.1, pp. 11–34.

McKinney, Scott Mayer, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg C. Corrado, Ara Darzi, Mozziyar Etemadi, Florencia Garcia-Vicente, Fiona J. Gilbert, Mark Halling-Brown, Demis Hassabis, Sunny Jansen, Alan Karthikesalingam, Christopher J. Kelly, Dominic King, Joseph R. Ledsam, David Melnick, Hormuz Mostofi, Lily Peng, Joshua Jay Reicher, Bernardino Romera-Paredes, Richard Sidebottom, Mustafa Suleyman, Daniel Tse, Kenneth C. Young, Jeffrey De Fauw, and Shravya Shetty (2020). "International evaluation of an AI system for breast cancer screening". In: *Nature* 577.7788, pp. 89–94.

Menze, B. H., A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M. A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, C. Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H. C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput (2015). "The Multimodal Brain Tumor

Image Segmentation Benchmark (BRATS)". In: *IEEE Transactions on Medical Imaging* 34.10, pp. 1993–2024.

Menze, Bjoern H., Koen Van Leemput, Antti Honkela, Ender Konukoglu, Marc-André Weber, Nicholas Ayache, and Polina Golland (2011a). "A Generative Approach for Image-Based Modeling of Tumor Growth". In: *Information Processing in Medical Imaging* 22, pp. 735–747.

Menze, Bjoern H, Erin Stretton, Ender Konukoglu, and Nicholas Ayache (2011b). *Image-based modeling of tumor growth in patients with glioma*. Tech. rep.

Mildenhall, Ben, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng (2020). "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *arXiv:2003.08934 [cs]*.

Milletari, Fausto, Nassir Navab, and Seyed-Ahmad Ahmadi (2016). "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation". In: *International Conference on 3D Vision*, pp. 565–571.

Minaee, Shervin, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos (2020). "Image Segmentation Using Deep Learning: A Survey". In: *arXiv:2001.05566 [cs]*.

Mohamed, Ashraf and Christos Davatzikos (2005). "Finite Element Modeling of Brain Tumor Mass-Effect from 3D Medical Images". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Vol. 3749, pp. 400–408.

Monteiro, Miguel, Loïc Le Folgoc, Daniel Coelho de Castro, Nick Pawlowski, Bernardo Marques, Konstantinos Kamnitsas, Mark van der Wilk, and Ben Glocker (2020). "Stochastic Segmentation Networks: Modelling Spatially Correlated Aleatoric Uncertainty". In: *arXiv:2006.06015 [cs]*.

Morimoto, Tetsuzo (1963). "Markov Processes and the H-Theorem". In: *Journal of the Physical Society of Japan* 18.3, pp. 328–331.

Morris, Marianne, Russell Greiner, Jörg Sander, Albert Murtha, and Mark Schmidt (2006). "Learning a Classification-based Glioma Growth Model Using MRI Data". In: *Journal of Computers* 1.7.

Mosayebi, Parisa, Dana Cobzas, Albert Murtha, and Martin Jagersand (2012). "Tumor invasion margin on the Riemannian space of brain fibers". In: *Medical Image Analysis* 16.2, pp. 361–373.

Müller, Alfred (1997). "Integral Probability Metrics and Their Generating Classes of Functions". In: *Advances in Applied Probability* 29.2, pp. 429–443.

Nadaraya, E. A. (1964). "On Estimating Regression". In: *Theory of Probability & Its Applications* 9.1, pp. 141–142.

Nam, Joo Yeon and John F. de Groot (2017). "Treatment of Glioblastoma". In: *Journal of Oncology Practice* 13.10, pp. 629–638.

Neal, Radford M. (1996). *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics. Springer.

Nikolov, Stanislav, Sam Blackwell, Ruheena Mendes, Jeffrey De Fauw, Clemens Meyer, Cían Hughes, Harry Askham, Bernardino Romera-Paredes, Alan Karthikesalingam, Carlton Chu, Dawn Carnell, Cheng Boon, Derek D'Souza, Syed Ali Moinuddin, Kevin Sullivan, DeepMind Radiographer Consortium, Hugh Montgomery, Geraint Rees, Ricky Sharma, Mustafa Suleyman, Trevor Back, Joseph R. Ledsam, and Olaf Ronneberger (2018). "Deep learning to achieve clinically applicable segmentation of head and neck anatomy for radiotherapy". In: *arXiv:1809.04430 [physics, stat]*.

Nolden, Marco, Sascha Zelzer, Alexander Seitel, Diana Wald, Michael Müller, Alfred M. Franz, Daniel Maleike, Markus Fangerau, Matthias Baumhauer, Lena Maier-Hein, Klaus H. Maier-Hein, Hans -Peter Meinzer, and Ivo Wolf (2013). "The Medical Imaging Interaction Toolkit: challenges and advances". In: *International Journal of Computer Assisted Radiology and Surgery* 8.4, pp. 607–620.

Oakden-Rayner, Luke, Jared Dunnmon, Gustavo Carneiro, and Christopher Re (2020). "Hidden stratification causes clinically meaningful failures in machine learning for medical imaging". In: *ACM Conference on Health, Inference, and Learning*, pp. 151–159.

Ohgaki, Hiroko and Paul Kleihues (2005). "Population-based studies on incidence, survival rates, and genetic alterations in astrocytic and oligodendroglial gliomas". In: *Journal of Neuropathology and Experimental Neurology* 64.6, pp. 479–489.

Oktay, Ozan, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y. Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert (2018). "Attention U-Net: Learning Where to Look for the Pancreas". In: *International Conference on Medical Imaging with Deep Learning (MIDL*.

Oord, Aaron van den, Yazhe Li, and Oriol Vinyals (2018). "Representation Learning with Contrastive Predictive Coding". In: *arXiv:1807.03748 [cs, stat]*.

Painter, K.J. and T. Hillen (2013). "Mathematical modelling of glioma growth: The use of Diffusion Tensor Imaging (DTI) data to predict the anisotropic pathways of cancer invasion". In: *Journal of Theoretical Biology* 323, pp. 25–39.

Palenzuela, Yaser Martinez (2020). *Awesome GPT-3*. URL: https://github.com/elyase/awesome-gpt3 (visited on 09/25/2020).

Parmar, Niki, Prajit Ramachandran, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens (2019). "Stand-Alone Self-Attention in Vision Models". In: *Advances in Neural Information Processing Systems*, pp. 68–80.

Parmar, Niki, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran (2018). "Image Transformer". In: *International Conference on Machine Learning*.

Paulsson, Anna K., Kevin P. McMullen, Ann M. Peiffer, William H. Hinson, William T. Kearns, Annette J. Johnson, Glenn J. Lesser, Thomas L. Ellis, Stephen B. Tatter, Waldemar Debinski, Edward G. Shaw, and Michael D. Chan (2014). "Limited margins using modern radiotherapy techniques does not increase marginal failure rate of glioblastoma". In: *American Journal of Clinical Oncology* 37.2, pp. 177–181.

Pawlowski, Nick, Martin Rajchl, and Ben Glocker (2017). "Implicit Weight Uncertainty in Neural Networks". In: *Advances in Neural Information Processing Systems – Bayesian Deep Learning Workshop*.

Pereira, Sergio, Adriano Pinto, Victor Alves, and Carlos A. Silva (2016). "Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images". In: *IEEE Transactions on Medical Imaging* 35.5, pp. 1240–1251.

Petersen, Jens, Martin Bendszus, Jürgen Debus, Sabine Heiland, and Klaus H. Maier-Hein (2016). "A Software Application for Interactive Medical Image Segmentation with Active User Guidance". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI) – Interactive Medical Image Computing Workshop*.

Petersen, Jens, Martin Bendszus, Jürgen Debus, Sabine Heiland, and Klaus H. Maier-Hein (2017a). "Effective User Guidance in Online Interactive Semantic Segmentation". In: *SPIE Medical Imaging*.

Petersen, Jens, Martin Bendszus, Jürgen Debus, Sabine Heiland, and Klaus H. Maier-Hein (2017b). "Effective User Interaction in Online Interactive Semantic Segmentation of Glioblastoma Magnetic Resonance Imaging". In: *Journal of Medical Imaging* 4.3, p. 034001.

Petersen, Jens, Sabine Heiland, Martin Bendszus, Jürgen Debus, and Klaus H. Maier-Hein (2017c). "Quantification of Guidance Strategies in Online Interactive Semantic Segmentation of Glioblastoma MRI". In: *Bildverarbeitung für die Medizin*, pp. 231–236.

Petersen, Jens, Sabine Heiland, Martin Bendszus, Jürgen Debus, Marco Nolden, Caspar J. Goch, and Klaus H. Maier-Hein (2018). "Leveraging Open Source Software to Close Translational Gaps in Medical Image Computing (Abstract)". In: *Bildverarbeitung für die Medizin*, pp. 22–22.

Petersen, Jens, Paul F. Jäger, Fabian Isensee, Simon A. A. Kohl, Ulf Neuberger, Wolfgang Wick, Jürgen Debus, Sabine Heiland, Martin Bendszus, Philipp Kickingereder, and Klaus H. Maier-Hein (2019). "Deep Probabilistic Modeling of Glioma Growth". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 806–814.

Petersen, Jens, Paul Jäger, Gregor Köhler, David Zimmerer, Fabian Isensee, and Klaus H. Maier-Hein (2020a). "Frequency Decompo-

sition in Neural Processes". In: *International Conference on Learning Representations (under review)*.

Petersen, Jens, Gregor Köhler, David Zimmerer, Fabian Isensee, Paul Jäger, and Klaus H. Maier-Hein (2020b). "GP-ConvCNP: Improving Generalization in Convolutional Conditional Neural Processes". In: *AAAI Conference on Artificial Intelligence (under review)*.

Price, S. J., N. G. Burnet, T. Donovan, H. a. L. Green, A. Peña, N. M. Antoun, J. D. Pickard, T. A. Carpenter, and J. H. Gillard (2003). "Diffusion tensor imaging of brain tumours at 3T: a potential tool for assessing white matter tract invasion?" In: *Clinical Radiology* 58.6, pp. 455–462.

Qi, Charles R., Hao Su, Mo Kaichun, and Leonidas J. Guibas (2017a). "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85.

Qi, Charles Ruizhongtai, Li Yi, Hao Su, and Leonidas J Guibas (2017b). "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space". In: *Advances in Neural Information Processing Systems*, pp. 5099–5108.

Rahimi, Ali and Benjamin Recht (2007). "Random Features for Large-Scale Kernel Machines". In: *Advances in Neural Information Processing Systems*, pp. 1177–1184.

Rasmussen, Carl Edward and C. K. I. Williams (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Razavi, Ali, Aaron van den Oord, and Oriol Vinyals (2019). "Generating Diverse High-Fidelity Images with VQ-VAE-2". In: *Advances in Neural Information Processing Systems*, pp. 14866–14876.

Reiser, Maximilian F., Wolfhard Semmler, and Hedvig Hricak, eds. (2008). *Magnetic Resonance Tomography*. Springer.

Rekik, Islem, Stéphanie Allassonnière, Olivier Clatz, Ezequiel Geremia, Erin Stretton, Hervé Delingette, and Nicholas Ayache (2013). "Tumor growth parameters estimation and source localization from a unique time point: Application to low-grade gliomas". In: *Computer Vision and Image Understanding* 117.3, pp. 238–249.

Rezende, Danilo Jimenez and Fabio Viola (2018). "Taming VAEs". In: *arXiv:1810.00597 [cs, stat]*.

Rockafellar, R. Tyrrell and Roger J.-B. Wets (1998). *Variational Analysis*. Grundlehren der mathematischen Wissenschaften. Springer.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 234–241.

Rosenbaum, Dan, Frederic Besse, Fabio Viola, Danilo J. Rezende, and S. M. Ali Eslami (2018). "Learning models for visual 3D localization with implicit mapping". In: *Advances in Neural Information Processing Systems – Bayesian Deep Learning Workshop*.

Rubanova, Yulia, Ricky T. Q. Chen, and David K Duvenaud (2019). "Latent Ordinary Differential Equations for Irregularly-Sampled Time Series". In: *Advances in Neural Information Processing Systems*, pp. 5320–5330.

Sakinis, Tomas, Fausto Milletari, Holger Roth, Panagiotis Korfiatis, Petro Kostandy, Kenneth Philbrick, Zeynettin Akkus, Ziyue Xu, Daguang Xu, and Bradley J. Erickson (2019). "Interactive segmentation of medical images through fully convolutional neural networks". In: *arXiv:1903.08205 [cs]*.

Salimbeni, Hugh, Ching-An Cheng, Byron Boots, and Marc Deisenroth (2018). "Orthogonally Decoupled Variational Gaussian Processes". In: *Advances in Neural Information Processing Systems*, pp. 8711–8720.

Schroff, F., A. Criminisi, and A. Zisserman (2008). "Object Class Segmentation using Random Forests". In: *British Machine Vision Conference*, pp. 54.1–54.10.

Schwab, Klaus (2016). *The Fourth Industrial Revolution: what it means and how to respond*. URL: https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/ (visited on 09/25/2020).

Settles, Burr (2010). *Active learning literature survey*. Tech. rep. 1648. University of Wisconsin, Madison.

Shannon, C.E. (1949). "Communication in the Presence of Noise". In: *Proceedings of the IRE* 37.1, pp. 10–21.

Silbergeld, Daniel L. and Michael R. Chicoine (1997). "Isolation and characterization of human malignant glioma cells from histologically normal brain". In: *Journal of Neurosurgery* 86.3, pp. 525–531.

Silver, David, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis (2018). "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play". In: *Science* 362.6419, pp. 1140–1144.

Simonyan, Karen and Andrew Zisserman (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *arXiv:1409.1556 [cs]*.

Simpson, Amber L., Michela Antonelli, Spyridon Bakas, Michel Bilello, Keyvan Farahani, Bram van Ginneken, Annette Kopp-Schneider, Bennett A. Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M. Summers, Patrick Bilic, Patrick F. Christ, Richard K. G. Do, Marc Gollub, Jennifer Golia-Pernicka, Stephan H. Heckers, William R. Jarnagin, Maureen K. McHugo, Sandy Napel, Eugene Vorontsov, Lena Maier-Hein, and M. Jorge Cardoso (2019). "A large annotated medical image dataset for the development and evaluation of segmentation algorithms". In: *arXiv:1902.09063 [cs, eess]*.

Singh, Gautam, Jaesik Yoon, Youngsung Son, and Sungjin Ahn (2019). "Sequential Neural Processes". In: *Advances in Neural Information Processing Systems*, pp. 10254–10264.

Sitzmann, Vincent, Michael Zollhoefer, and Gordon Wetzstein (2019). "Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations". In: *Advances in Neural Information Processing Systems*, pp. 1119–1130.

Smith, Jonathan D., Kamyar Azizzadenesheli, and Zachary E. Ross (2020). "EikoNet: Solving the Eikonal equation with Deep Neural Networks". In: *arXiv:2004.00361 [physics, stat]*.

Snelson, Edward and Zoubin Ghahramani (2006). "Sparse Gaussian Processes using Pseudo-inputs". In: *Advances in Neural Information Processing Systems 18*, pp. 1257–1264.

Soerensen, T.J. (1948). "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons". In: vol. 5. Biologiske Skrifter 4. Det Kongelige Danske Videnskabernes Selskab, pp. 1–34.

Sofiiuk, Konstantin, Ilia Petrov, Olga Barinova, and Anton Konushin (2020). "F-BRS: Rethinking Backpropagating Refinement for Interactive Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8620–8629.

Sohn, Kihyuk, Honglak Lee, and Xinchen Yan (2015). "Learning Structured Output Representation using Deep Conditional Generative Models". In: *Advances in Neural Information Processing Systems*. Vol. 28, pp. 3483–3491.

Sommer, Christoph, Christoph Straehle, Ullrich Koethe, and Fred A. Hamprecht (2011). "ilastik: Interactive learning and segmentation toolkit". In: *IEEE International Symposium on Biomedical Imaging*, pp. 230–233.

Sriperumbudur, Bharath K., Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert R. G. Lanckriet (2009). "On integral probability metrics, phi-divergences and binary classification". In: *arXiv:0901.2698 [cs, math]*.

Stensjøen, Anne Line, Ole Solheim, Kjell Arne Kvistad, Asta K. Håberg, Øyvind Salvesen, and Erik Magnus Berntsen (2015). "Growth dynamics of untreated glioblastomas in vivo". In: *Neuro-Oncology* 17.10, pp. 1402–1411.

Stretton, E., E. Geremia, B. Menze, H. Delingette, and N. Ayache (2013). "Importance of patient DTI's to accurately model glioma growth using the reaction diffusion equation". In: *IEEE International Symposium on Biomedical Imaging*, pp. 1142–1145.

Sutton, Richard (2020). *The Bitter Lesson*. Tech. rep. URL: http://www.incompleteideas.net/IncIdeas/BitterLesson.html (visited on 09/25/2020).

Swan, Amanda, Thomas Hillen, John C. Bowman, and Albert D. Murtha (2018). "A Patient-Specific Anisotropic Diffusion Model for Brain Tumour Spread". In: *Bulletin of Mathematical Biology* 80.5, pp. 1259–1291.

Swanson, K R, E C Alvord, and J D Murray (2002). "Virtual brain tumours (gliomas) enhance the reality of medical imaging and highlight inadequacies of current therapy". In: *British Journal of Cancer* 86.1, pp. 14–18.

Swanson, K. R., E. C. Alvord, and J. D. Murray (2000). "A quantitative model for differential motility of gliomas in grey and white matter". In: *Cell Proliferation* 33.5, pp. 317–329.

Tao, Andrew, Karan Sapra, and Bryan Catanzaro (2020). "Hierarchical Multi-Scale Attention for Semantic Segmentation". In: *arXiv:2005.10821 [cs]*.

Thomas, Philipp, Guillaume Terradot, Vincent Danos, and Andrea Y. Weiße (2018). "Sources, propagation and consequences of stochasticity in cellular growth". In: *Nature Communications* 9.1, p. 4528.

Titsias, Michalis (2009). "Variational Learning of Inducing Variables in Sparse Gaussian Processes". In: *International Conference on Artificial Intelligence and Statistics*, pp. 567–574.

Top, Andrew, Ghassan Hamarneh, and Rafeef Abugharbieh (2010). "Spotlight: Automated confidence-based user guidance for increasing efficiency in interactive 3D image segmentation". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI) – Medical Computer Vision Workshop*, pp. 204–213.

Top, Andrew, Ghassan Hamarneh, and Rafeef Abugharbieh (2011). "Active learning for interactive 3D image segmentation". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 603–610.

Tossou, Prudencio, Basile Dura, Francois Laviolette, Mario Marchand, and Alexandre Lacoste (2019). "Adaptive Deep Kernel Learning". In: *arXiv:1905.12131 [cs, stat]*.

Tracqui, P., G. C. Cruywagen, D. E. Woodward, G. T. Bartoo, J. D. Murray, and E. C. Alvord (1995). "A mathematical model of glioma growth: the effect of chemotherapy on spatio-temporal growth". In: *Cell Proliferation* 28.1, pp. 17–31.

Triebel, Rudolph, Jan Stuehmer, Mohamed Souiai, and Daniel Cremers (2014). "Active online learning for interactive segmentation using sparse gaussian processes". In: *German Conference on Pattern Recognition*, pp. 641–652.

Unkelbach, Jan, Bjoern H. Menze, Ender Konukoglu, Florian Dittmann, Nicholas Ayache, and Helen A. Shih (2014a). "Radiotherapy planning for glioblastoma based on a tumor growth model: implications for spatial dose redistribution". In: *Physics in Medicine and Biology* 59.3, pp. 771–789.

Unkelbach, Jan, Bjoern H. Menze, Ender Konukoglu, Florian Dittmann, Matthieu Le, Nicholas Ayache, and Helen A. Shih (2014b). "Radiotherapy planning for glioblastoma based on a tumor growth model: improving target volume delineation". In: *Physics in Medicine and Biology* 59.3, pp. 747–770.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems*, pp. 5998–6008.

Vershynin, Roman (2018). *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Vezhnevets, Alexander, Joachim M. Buhmann, and Vittorio Ferrari (2012). "Active learning for semantic segmentation with expected change". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3162–3169.

Vijayanarasimhan, Sudheendra and Kristen Grauman (2009). "What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2262–2269.

Vlaardingerbroek, Marinus T. and Jacques A. Boer (2003). *Magnetic Resonance Imaging: Theory and Practice*. 3rd ed. Springer.

Volpi, Riccardo, Hongseok Namkoong, Ozan Sener, John Duchi, Vittorio Murino, and Silvio Savarese (2018). "Generalizing to Unseen Domains via Adversarial Data Augmentation". In: *Advances in Neural Information Processing Systems*, pp. 5339–5349.

Wagstaff, Edward, Fabian B. Fuchs, Martin Engelcke, Ingmar Posner, and Michael Osborne (2019). "On the Limitations of Representing Functions on Sets". In: *International Conference on Machine Learning*.

Waheed, U. B., E. Haghighat, T. Alkhalifah, C. Song, and Q. Hao (2020). "Eikonal Solution Using Physics-Informed Neural Networks". In: *EAGE Annual Conference & Exhibition Workshop Programme*.

Wang, Guotai, Wenqi Li, Maria A. Zuluaga, Rosalind Pratt, Premal A. Patel, Michael Aertsen, Tom Doel, Anna L. David, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren (2018). "Interactive Medical Image Segmentation Using Deep Learning With Image-Specific Fine Tuning". In: *IEEE Transactions on Medical Imaging* 37.7, pp. 1562–1573.

Wang, Guotai, Maria A. Zuluaga, Wenqi Li, Rosalind Pratt, Premal A. Patel, Michael Aertsen, Tom Doel, Anna L. David, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren (2019a). "DeepIGeoS: A Deep Interactive Geodesic Framework for Medical Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.7, pp. 1559–1572.

Wang, J., K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao (2020a). "Deep High-

Resolution Representation Learning for Visual Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP.

Wang, Ke, Geoff Pleiss, Jacob Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson (2019b). "Exact Gaussian Processes on a Million Data Points". In: *Advances in Neural Information Processing Systems*, pp. 14622–14632.

Wang, Sinong, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma (2020b). "Linformer: Self-Attention with Linear Complexity". In: *arXiv:2006.04768 [cs, stat]*.

Wasserman, Richard, Raj Acharya, Claudio Sibata, and K. H. Shin (1996). "A patient-specific in vivo tumor model". In: *Mathematical Biosciences* 136.2, pp. 111–140.

Watson, Geoffrey S. (1964). "Smooth Regression Analysis". In: *Sankhya: The Indian Journal of Statistics, Series A (1961-2002)* 26.4, pp. 359–372.

Wen, Patrick Y., David R. Macdonald, David A. Reardon, Timothy F. Cloughesy, A. Gregory Sorensen, Evanthia Galanis, John Degroot, Wolfgang Wick, Mark R. Gilbert, Andrew B. Lassman, Christina Tsien, Tom Mikkelsen, Eric T. Wong, Marc C. Chamberlain, Roger Stupp, Kathleen R. Lamborn, Michael A. Vogelbaum, Martin J. van den Bent, and Susan M. Chang (2010). "Updated response assessment criteria for high-grade gliomas: response assessment in neuro-oncology working group". In: *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology* 28.11, pp. 1963–1972.

Whittaker, E. T. (1915). "XVIII.—On the Functions which are represented by the Expansions of the Interpolation-Theory". In: *Proceedings of the Royal Society of Edinburgh* 35, pp. 181–194.

Wick, Wolfgang, Thierry Gorlia, Martin Bendszus, Martin Taphoorn, Felix Sahm, Inga Harting, Alba A. Brandes, Walter Taal, Julien Domont, Ahmed Idbaih, Mario Campone, Paul M. Clement, Roger Stupp, Michel Fabbro, Emilie Le Rhun, Francois Dubois, Michael Weller, Andreas von Deimling, Vassilis Golfinopoulos, Jacoline C. Bromberg, Michael Platten, Martin Klein, and Martin J. van den Bent (2017). "Lomustine and Bevacizumab in Progressive Glioblastoma". In: *New England Journal of Medicine* 377.20, pp. 1954–1963.

Wilcoxon, Frank (1945). "Individual Comparisons by Ranking Methods". In: *Biometrics Bulletin* 1.6, pp. 80–83.

Willi, Timon, Jonathan Masci, Jürgen Schmidhuber, and Christian Osendorfer (2019). "Recurrent Neural Processes". In: *arXiv:1906.05915 [cs, stat]*.

Wilson, Andrew G, Zhiting Hu, Russ R Salakhutdinov, and Eric P Xing (2016a). "Stochastic Variational Deep Kernel Learning". In: *Advances in Neural Information Processing Systems*, pp. 2586–2594.

Wilson, Andrew Gordon, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing (2016b). "Deep Kernel Learning". In: *International Conference on Artificial Intelligence and Statistics*.

Wilson, Andrew Gordon and Hannes Nickisch (2015). "Kernel Interpolation for Scalable Structured Gaussian Processes (KISS-GP)". In: *International Conference on Machine Learning*.

Wolf, Ivo, Marcus Vetter, Ingmar Wegner, Marco Nolden, Thomas Bottger, Mark Hastenteufel, Max Schobinger, Tobias Kunert, and Hans-Peter Meinzer (2004). "The medical imaging interaction toolkit (MITK): a toolkit facilitating the creation of interactive software by extending VTK and ITK". In: *SPIE Medical Imaging*.

Woodward, D. E., J. Cook, P. Tracqui, G. C. Cruywagen, J. D. Murray, and E. C. Alvord (1996). "A mathematical model of glioma growth: the effect of extent of surgical resection". In: *Cell Proliferation* 29.6, pp. 269–288.

Wu, Wenxuan, Zhongang Qi, and Li Fuxin (2019). "PointConv: Deep Convolutional Networks on 3D Point Clouds". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9621–9630.

Xu, Ning, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang (2016). "Deep Interactive Object Selection". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yang, Zichao, Andrew Wilson, Alex Smola, and Le Song (2015). "A la Carte – Learning Fast Kernels". In: *International Conference on Artificial Intelligence and Statistics*, pp. 1098–1106.

Yıldız, Çağatay, Markus Heinonen, and Harri Lähdesmäki (2019). "ODE$ˆ2$VAE: Deep generative second order ODEs with Bayesian neural networks". In: *arXiv:1905.10994 [cs, stat]*.

Yuan, Jianjun and Lipei Liu (2016). "Brain glioma growth model using reaction-diffusion equation with viscous stress tensor on brain MR images". In: *Magnetic Resonance Imaging* 34.2, pp. 114–119.

Yuan, Jianjun, Lipei Liu, and Qingmao Hu (2013). "Mathematical modeling of brain glioma growth using modified reaction–diffusion equation on brain MR images". In: *Computers in Biology and Medicine* 43.12, pp. 2007–2013.

Zaheer, Manzil, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola (2017). "Deep Sets". In: *Advances in Neural Information Processing Systems*, pp. 3391–3401.

Zhang, Ling, Le Lu, Ronald M. Summers, Electron Kebebew, and Jianhua Yao (2018). "Convolutional Invasion and Expansion Networks for Tumor Growth Prediction". In: *IEEE Transactions on Medical Imaging* 37.2, pp. 638–648.

Zhou, Zongwei, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang (2020). "UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation". In: *IEEE Transactions on Medical Imaging* 39.6, pp. 1856–1867.

Zhumekenov, Abylay, Malika Uteuliyeva, Olzhas Kabdolov, Rustem Takhanov, Zhenisbek Assylbekov, and Alejandro J. Castro (2019). "Fourier Neural Networks: A Comparative Study". In: *arXiv:1902.03011 [cs]*.

Zikic, Darko, Ben Glocker, Ender Konukoglu, Antonio Criminisi, C. Demiralp, Jamie Shotton, O. M. Thomas, T. Das, R. Jena, and S. J. Price (2012). "Decision forests for tissue-specific segmentation of high-grade gliomas in multi-channel MR". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 369–376.

Zimmerer, David, Fabian Isensee, Jens Petersen, Simon Kohl, and Klaus Maier-Hein (2019a). "Unsupervised Anomaly Localization using Variational Auto-Encoders". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 289–297.

Zimmerer, David, Simon A. A. Kohl, Jens Petersen, Fabian Isensee, and Klaus Maier-Hein (2019b). "Context-encoding Variational Autoencoder for Unsupervised Anomaly Detection". In: *International Conference on Medical Imaging with Deep Learning (MIDL – Extended Abstract Track*.

Zimmerer, David, Jens Petersen, Simon A. A. Kohl, and Klaus H Maier-Hein (2018). "A Case for the Score: Identifying Image Anomalies using Variational Autoencoder Gradients". In: *Medical Imaging meets NeurIPS*.