Causal Inference and Interpretable Machine Learning for Personalised Medicine

Inauguraldissertation

zur

Erlangung der Würde eines Doktors der Philosophie

vorgelegt der

Philosophisch-Naturwissenschaftlichen Fakultät

der Universität Basel

von

Sonali Parbhoo

aus Johannesburg, Südafrika und USA

2019

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät
auf Antrag von

Prof. Dr. Volker Roth, Dissertationsleiter

Prof. Dr. Niko Beerenwinkel, Korreferent

Basel, den 25. Juni 2019

Prof. Dr. Martin Spiess, Dekan

For my family

Abstract

In this thesis, we discuss the importance of causal knowledge in healthcare for tailoring treatments to a patient's needs. We propose three different causal models for reasoning about the effects of medical interventions on patients with HIV and sepsis, based on observational data. Both application areas are challenging as a result of patient heterogeneity and the existence of confounding that influences patient outcomes.

Our first contribution is a treatment policy mixture model that combines nonparametric, kernel-based learning with model-based reinforcement learning to reason about a series of treatments and their effects. These methods each have their own strengths: non-parametric methods can accurately predict treatment effects where there are overlapping patient instances or where data is abundant; model-based reinforcement learning generalises better in outlier situations by learning a belief state representation of confounding. The overall policy mixture model learns a partition of the space of heterogeneous patients such that we can personalise treatments accordingly.

Our second contribution incorporates knowledge from kernel-based reasoning directly into a reinforcement learning model by learning a combined belief state representation. In doing so, we can use the model to simulate counterfactual scenarios to reason about what would happen to a patient if we intervened in a particular way and how would their specific outcomes change. As a result, we may tailor therapies according to patient-specific scenarios.

Our third contribution is a reformulation of the information bottleneck problem for learning an interpretable, low-dimensional representation of confounding for medical decision-making. The approach uses the relevance of information to perform a sufficient reduction of confounding. Based on this reduction, we learn equivalence classes among groups of patients, such that we may transfer knowledge to patients with incomplete covariate information at test time. By conditioning on the sufficient statistic we can accurately infer treatment effects on both a population and subgroup level.

Our final contribution is the development of a novel regularisation strategy that can be applied to deep machine learning models to enforce clinical interpretability. We specifically train deep time-series models such that their predictions have high accuracy while being closely modelled by small decision trees that can be audited easily by medical experts. Broadly, our tree-based explanations can be used to provide additional context in scenarios where reasoning about treatment effects may otherwise be difficult. Importantly, each of the models we present is an attempt to bring about more understanding in medical applications to inform better decision-making overall.

Acknowledgments

I am both deeply grateful and consider myself extremely fortunate to have worked with my supervisor, Prof. Dr. Volker Roth, without whom none of the work in this thesis would have been possible. It is difficult to articulate in words how much the opportunity to work with you means to me. Thank you for providing me with what I consider to be one of the best experiences of my life. You never failed to guide and support me when I needed it, and you always asked the right questions, even if it meant I had to re-think my ideas at the time. These experiences have taught me how to become a better researcher overall. You also gave me the freedom to explore my own ideas, even if it meant collaborating with many other researchers around the world. Most of all, when I think about the kind of scientist I would like to be some day, I know I want to be just like you .... Thank you.

I am very thankful to Prof. Dr. Niko Beerenwinkel for reviewing my thesis as a co-referee and for providing valuable feedback during committee meetings and the Systems-X HIV-X project. I am also very grateful to Prof. Dr. Thomas Vetter for many thought-provoking discussions about research and philosophy during lab meetings and breaks. These discussions have been the source of a lot of inspiration during the final stages of my PhD.

A significant portion of the work in this thesis was performed under the guidance of Prof. Dr. Finale Doshi-Velez. Thank you for introducing me to a lot of interesting work conducted in your research group, and providing me the opportunity to understand these problems in a way that would not have been possible otherwise. I am very fortunate to have gained some experience working with you, and I am very excited about our future research together! I would also like to thank Prof. Dr. George Konidaris for introducing me to Prof. Dr. Finale Doshi-Velez, knowing that our research interests would align well.

I am especially grateful to my Masters' research advisor, Prof. Dr. Clint Van Alten, for encouraging me to pursue a PhD in the first place, and for providing support during one of the most difficult periods of my life. You recommended applying to Prof. Volker Roth after reading some of his work, and I have not regretted my decision ever since.

Outside the computer science department, I am glad to have been part of numerous collaborations with other researchers and medical professionals. I would especially like to thank Dr. Jasmina Bogojeska for many helpful discussions on non-parametric kernel methods, and whose work has inspired some of the ideas in this thesis. I am also grateful to Prof. Dr. Huldrych Günthard, Prof. Dr. Karin Metzner, Prof. Dr. Roger Kouyos and all the members of the Systems-X HIV-X project for giving me valuable insights in your areas of expertise. I would also like to thank my research collaborators overseas: Mike Wu, Michael Hughes, Andrew Ross and Omer Gottesman, for engaging in many

interesting research discussions with me, and for making collaborating a very easy and pleasant experience. I have learned so much from all of you and I look forward to learning more!

During my PhD, I was given the unique opportunity to conduct an overseas research visit and received career training for ten months through the antelope programme at the university. I am thankful to all the members and mentors, as well as all the inspiring young research scientists I had the opportunity to interact with and learn from. Most of all, I think it is especially progressive of a university to have such a programme in place and I am very fortunate to have been part of this.

I would especially like to thank all my colleagues and friends over the years at the university from both the Gravis and the BMDA group for enriching my PhD experience on both a professional and personal level. All of you have taught me so much over these years and have made my time in Basel incredibly special. A sincere thank you goes to my friends Mario Wieser, Maxim Samarin and Fabricio Arend Torres for taking time to read and comment on parts of this thesis. A special mention also goes to Mario Wieser, Adam Kortylewski and Aleksander Wieczorek and Dinu Kaufmann for introducing me to your cultures, food and languages, and for sharing many moments of laughter and fun! I would also like to thank Dana Rahbani, Mahnaaz Pariyaan, Silvia Ligabue and Manvi Bhatia for always being available for a hike, gym class or outing together when I needed to de-stress. Outside the university environment, special thanks go to Gianni, Silvia and Daniel for teaching me about your incredible culture and always encouraging my language learning pursuits. You have treated me like family in Switzerland and I'm so fortunate that our paths crossed. One day, somehow we will see one another again. Overall, this experience would have been exceptionally difficult without the support of all my friends from far and wide. I would especially like to thank Raymond, Bongani, Amrit, Francesco, Louis and Grace for sharing in all my experiences and happinesses in spite of the distance.

Last but not least, I am extremely grateful to each and every one in my family for their numerous visits, ongoing support, love and endless patience on this journey. Thank you for encouraging me to pursue my dreams even if these took me so far away from all of you, and for believing in me at times when I lacked belief in myself. I would especially like to thank my cousin, Siddharth, for always filling my refrigerator and ensuring I was fed when I had no time for this on my own. To my mother and father, Nisha and Anant, for never failing to call every day or proofread my research papers even if the subject was completely foreign to you. Finally, to my sister, Priya, for all the little things .... There's nothing bigger, is there?

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General Motivation

Machine learning has enabled us to answer many questions and vastly revolutionised several disciplines such as computer vision. The majority of these tasks have been tackled using training sets of data and prediction functions to identify a set of associations in the data. While associations may be useful for making accurate predictions, in fields such as medicine however, the driving questions are most often not associational, but rather *causal* in nature (Pearl et al., 2009). For instance, what is the effect of a drug on a particular individual or population? What would happen if a patient exhibited different symptoms to those observed? Does a particular genetic mutation cause resistance against a certain drug? Each of these questions is rooted in acquiring a deeper understanding of the reasons for a particular event, and may thus be referred to as causal inference or discovery questions. For such questions, understanding the distinction between association and causation is crucial. To illustrate the difference here, consider the following example from Guo et al. (2018). When temperatures are high, an ice-cream shop owner may observe both high sales and high electricity bills. While there may be a strong association between the electricity bill and sales, it is unlikely that the electricity bill is the cause of high sales. This could be demonstrated by observing what happens if the lights are left on for a prolonged period of time. Evidently, there would be no impact on sales. Rather, the association between sales and the electricity bill may be explained by a common cause variable or *confounder*, namely the temperature: If temperatures are high, there is a higher demand for ice-cream and more electricity is required to cool the increased supply. In this example, conducting an experiment to distinguish between cause and association is straightforward. In practice however, performing such experiments to reason about potential causes is not always possible. Causal discovery questions are generally challenging and cannot be answered solely on the basis of data, since they require some knowledge of the underlying data generating mechanism (Pearl et al., 2009).

To distinguish between tools for associational modelling and causal modelling, Pearl (2018) introduces a 3-level hierarchy based on the kind of information required to answer questions at each level. The three levels are: i) Association, ii) Intervention and iii) Counterfactuals. Table 1.1 illustrates this hierarchy, as well as examples of questions at each level. For instance, the first level identifies statistical relations *solely on the basis of data*, and can thus be used to determine associations. These are quantities such as correlations, odds ratios, risk ratios and statistical dependencies (Pearl et al., 2009). Since questions at the first level are entirely associational, they can typically be addressed with existing machine learning techniques such as regression. The next level, Intervention, focuses on understanding the *effects of causes*. Specifically, it requires not only observing existing data, but actively doing something to reason about the effects of interventions. The final layer, namely counterfactuals, may be used to reason about the *causes of effects*. Questions at this level involve retrospection and typically cannot be answered using associational or interventional information alone; for example, given a control experiment where patients are assigned particular treatments, we usually cannot re-execute the experiment to see what would have happened had the patient been treated otherwise. Rather, these questions require explicitly modelling the underlying data generation procedure and using such a model to make inferences. Overall, the hierarchy may viewed as directional since questions from a particular level can be addressed if information from that level or subsequent levels is available. That is, a model for counterfactuals can be used to address questions concerning the effects of interventions and identify associations, but the opposite does not necessarily hold. A schematic representation of this relationship is shown in Figure 1.1. Here, accounting for interventions by reasoning about their effects and learning causal models subsumes the task of identifying associations and making predictions.

Level	**Typical Activity**	**Typical Questions**	**Examples**
1. Association $P(y\|x)$	Seeing *e.g. Regression*	What is? How does X change my belief in Y?	What does a symptom tell us about the disease?
2. Intervention $P(y\|do(x), z)$	Doing, Intervening *e.g. Reinforcement Learning*	What if? What if I do X?	What if I take aspirin, will my headache be cured?
3. Counterfactuals $P_x(y\|x', y')$	Imagining, Retrospection *e.g. Structural Causal Models*	Was it X that caused Y? What if I acted differently?	Was it aspirin that stopped my headache?

Table 1.1: The 3-level hierarchy of tools for modelling causality (Pearl, 2018). $P(y|x)$ is the probability of outcome $Y = y$ given an observation $X = x$. $P(y|do(x), z)$ refers to the probability of $Y = y$ given that we explicitly intervene and set X to x and subsequently observe $Z = z$. Here, $P_x(y|x', y')$ refers to the probability of $Y = y$ had X been x given that we observe X as x' and Y as y'.

One of many principled ways to formulate queries concerning causation is using *Structural Causal Models (SCMs)*(Pearl, 1995, 2009). In its general form, an SCM denoted $(\mathcal{U}, \mathcal{V}, F)$ consists of two sets of variables $\mathcal{U}$ and $\mathcal{V}$, and a set of functions f_i that define or simulate how values are assigned to $V_i \in \mathcal{V}$. For example, $v_i = f_i(u, v)$ describes the process of how v_i is assigned a value based on the values u, v and the function f_i. The variables $\mathcal{U}$ are known as exogenous noise variables whose values are determined by external influences beyond the model, while the variables $\mathcal{V}$ are endogenous variables whose values are defined by other variables in the model. SCMs are frequently illustrated as causal graphs that capture the causal relationships over the variables. In the past, SCMs have found applications across domains such as economics (e.g. Imbens (2004)), social sciences (e.g Duncan (1975); Goldberger (1972); L. Morgan & Winship (2007)) and education (e.g. Dehejia & Wahba (1999); Hill (2011a); LaLonde

(1986)).

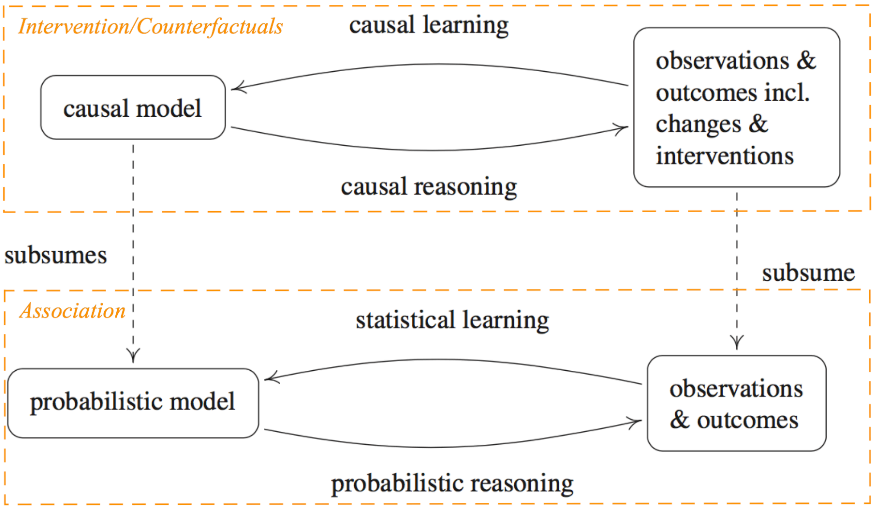


Figure 1.1: Illustration of the relationship between causal and associational modelling (adapted from Peters et al. (2017)). Causal models can be used to make predictions about the effects of interventions or (in some cases) counterfactual claims about a system. This requires modelling dependence relations as well as performing interventions, and hence subsumes probabilistic reasoning. Standard probabilistic models can account for associations but not for interventions. The orange boxes indicate how these models relate to Pearl's hierarchy.

1.2 Causal Inference for Personalised Medicine

In this thesis, we focus on the role of causal knowledge in healthcare, particularly for personalising treatments to a patient's needs. The fundamental question we address is: *What is the effect of a therapy on a particular patient?* We study this question in the context of two healthcare applications, namely treating patients with sepsis and Human Immunodeficiency Virus (HIV). We specifically introduce different causal models for this purpose. These models may be viewed as tools from either the Intervention or Counterfactual level in Pearl's hierarchy. In the following sections, we briefly introduce both healthcare applications and identify the major challenges in these domains. We subsequently embed these in the context of causal inference and machine learning, and highlight the key contributions of this thesis.

1.2.1 Human Immunodeficiency Virus

HIV[1] is a retrovirus that currently affects more than 36 million people worldwide and causes Acquired Immune Deficiency Syndrome (AIDS) (UNAIDS, 2015). If untreated, HIV attacks and destroys the immune system by causing progressive loss of white blood

[1] In this thesis we restrict our focus to HIV-1.

cells, such as CD4$^+$ T-lymphocytes. The gradual destruction of the immune system leaves a patient vulnerable to opportunistic infections that frequently result in death.

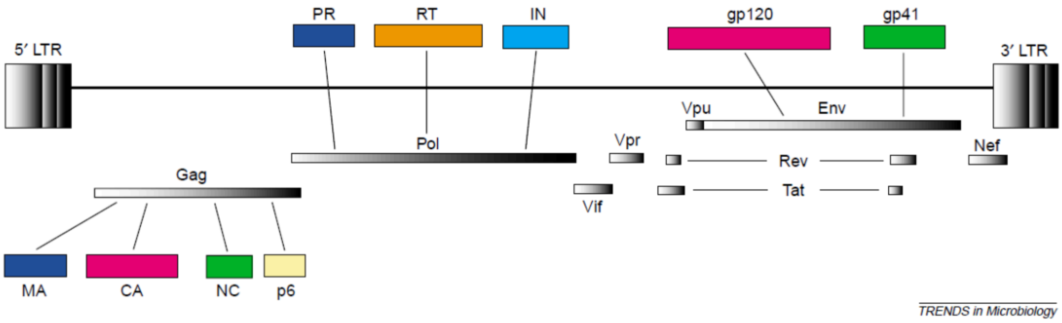


Figure 1.2: The viral genome of HIV. The Pol region serves as an active site for PIs, RTIs and Integrase Inhibitors (Freed, 2004).

To date, the only practical treatment for HIV is through life-long administration of combinations of antiretrovirals, known as Highly Active Antiretroviral Therapy (HAART), that target various phases of the viral life cycle. There are currently more than 20 drugs in use for HAART (Günthard et al., 2016). These may be classified as: Entry/Fusion Inhibitors, Reverse Transcriptase Inhibitors (RTIs), Integrase Inhibitors and Protease Inhibitors (PIs). Entry Inhibitors try to stop viral entry into an immune cell (De Clercq, 2009). RTIs bind to the virus's reverse transcriptase enzyme, thereby preventing the virus from converting its genomic material into DNA. Integrase Inhibitors prevent the viral DNA from integrating with the host's genome by inhibiting the function of the integrase enzyme in the virus (Lusic & Siliciano, 2017). Protease inhibitors target the formation and assembly of viral proteins that are crucial for assembling new viral particles (Hammer et al., 2006). Figure 1.2 illustrates the target sites of these drugs on the HIV genome. Overall, advances in drug therapies since the introduction of HAART in 1996, have meant that many individuals are able to suppress viral loads below detection limits ($<$ 40 copies/ml) and sustain immune functionality for prolonged periods of time.

1.2.2 Challenges with Treating HIV

High Mutagenicity. Despite the introduction of new antiretrovirals, the high evolutionary dynamics of the virus enable it to escape drug pressure by acquiring resistance mutations (Mansky & Temin, 1995). Resistance to a particular drug may also result in resistance to other drugs from the same family; this is known as *cross-resistance* (Thompson et al., 2010). This, together with the large number of available therapy combinations makes manually searching for an effective therapy particularly challenging.

Patient Heterogeneity. HIV may be classified into four groups, M, N, O and P (Robertson et al., 2000). Of these, group M accounts for the majority of the global pandemic and consists of ten different subtypes, A-K. However, many new recombinant strains can be formed from further recombination between subtypes in a host. Overall, HIV's high rate of mutation means that a patient may harbour many genetically heterogeneous populations that continue evolving. As a result, causal inference is important to tailor treatments to a patient's individual responses.

Influence of Confounding Factors. Existing sources of HIV data are high-dimensional and frequently biased by several confounding factors such as different treatment backgrounds, varying levels of therapy experience, demographics, and the occurrence of co-infections e.g. tuberculosis. These factors necessitate causal inference to reason about patient outcomes and administer appropriate therapies.

1.2.3 Sepsis

Sepsis is a complex multi-system disease resulting from the body's inflammatory response to infection (Brand et al., 2017). Treating patients suffering from sepsis is particularly challenging since they tend to exhibit a myriad of symptoms, depending on the nature of infection. In spite of this heterogeneity, sepsis is typically characterised by conditions such as a rapid rise or fall in body temperature, an elevated heart rate (known as tachycardia), vasodilation, hypotension and an elevated white blood cell count (Polat et al., 2017; Singer et al., 2016). The most severe form of sepsis, known as septic shock, occurs when patients experience severe hypotension that potentially results in organ dysfunction or failure. Septic shock is one of the leading contributors to mortality in the ICU with global estimates of between 20 and 30 million cases annually, and mortality rates typically exceeding 50% (Napolitano, 2018; Polat et al., 2017). As a result, sepsis is frequently viewed as a medical emergency that requires immediate intervention.

Treatments for sepsis vary depending on the underlying nature of infection, and in practice there is often little consensus about how patients should be treated (Peng et al., 2019). However, antibiotics, intravenous fluid resuscitators, corticosteroids and mechanical ventilation are typically necessary for combating hypotension and tachycardia (Hajj et al., 2018). When fluid resuscitation alone fails, vasopressors are frequently administered to restore adequate blood pressure and correct for excessive vasodilation (Brand et al., 2017). Depending on the severity of infection and loss of blood pressure, a number of vasopressors may be administered concurrently. These include dopamine, epinephrine, phenylephrine and vasopressin however, norepinephrine is frequently used as a first-line therapy (Brand et al., 2017). Overall, administering vasopressors for treating sepsis is particularly challenging, since vasopressors are associated with a number of dangerous outcomes such as fluid overload, kidney failure, high blood pressure and irregular heartbeat, all of which can have severe implications on a patient's mortality.

1.2.4 Challenges with Treating Sepsis

Patient Heterogeneity. The primary challenge of treating patients with sepsis is the lack of consensus as to what symptoms characterise the disease. Overall, sepsis can occur as a result of bacteria, viruses or parasites, severe trauma, pneumonia or other infections, and patients with each of these conditions may exhibit a wide variety of clinical and physiopathological symptoms (Polat et al., 2017). For these reasons, there is no universal diagnosis of the disease. As a result, the definition of sepsis has progressively evolved over the past thirty years to include new signs and symptoms (Napolitano, 2018).

Currently, an ICU patient may be classified as septic if they experience the following conditions: i) an increase in their sequential organ failure assessment score (known as SOFA) of more than 2 points; ii) a systolic blood pressure of less than 100 mmHg; iii) a respiratory rate of more than 22; iv) an altered mental state of less than 15 defined by the Glasgow Coma Scale (GCS) (Singer et al., 2016). Still, these criteria are difficult to use in practice and may not necessarily promote an understanding of the underlying disease process. In such scenarios, causal reasoning is important to understand how patients should be treated.

Influence of Confounding Factors. While sepsis is associated with a high mortality rate, it is also common for sepsis survivors to be re-admitted to hospital shortly after they are discharged. Not only do these patients have a higher chance of infection, but also have higher levels of inflammation – factors that may ultimately compromise quality of life and affect long-term life expectancy (Shankar-Hari et al., 2016). That is, a patient's outcomes following sepsis typically reflect a complex interplay between several factors such as their demographics, treatments in ICU, post-ICU care and the nature of initial infection. In such cases, identifying these factors and correcting for their influences may provide a better understanding of the disease and inform more effective interventions. Like with HIV, observational data of patients suffering from sepsis may be useful to promote such an understanding of the disease however, these too are biased by confounding. As a result, causal inference is crucial to be able to reason about patient outcomes and infer appropriate treatment strategies.

1.3 Contributions and Outline of the Thesis

Our overall aim is to understand the effect of a therapeutic intervention on a patient with HIV or sepsis. Consequently, our work is rooted in three closely related themes namely, *causal inference*, *explainability* and *decision-making.* In this thesis, we examine the relationship between these themes. In particular, we view both explainability and decision-making in light of causal inference as shown in Figure 1.3. While the use of machine learning in everyday systems is becoming increasingly common, the ability of a system to *explain* its reasoning is crucial in high-stake domains such as healthcare. That is, if we can interpret or explain why a system makes its predictions, we can verify whether or not this reasoning is correct (Doshi-Velez & Kim, 2017). Unfortunately however, there is little agreement on what model explainability is and how it should be evaluated. Similarly, in domains where decision-making is challenging, we usually require a deeper understanding of the effects of executing a particular decision before determining a suitable course of action. Causal reasoning may be helpful for tackling both of these issues simultaneously.

This thesis is divided into roughly three parts. In the first part of the thesis, we examine the problem of HIV therapy selection. Here, we show how tools from the Intervention layer of Pearl's hierarchy such as reinforcement learning (RL), can be combined with existing methods for associational modelling to reason about the effects of therapies. While in the past, the problem of HIV therapy selection has been studied extensively using regression methods, coupling this associational knowledge with RL enables us to personalise HIV therapies on the basis of a patient's history, while simultaneously accounting for the effects of confounders. In the second part of this

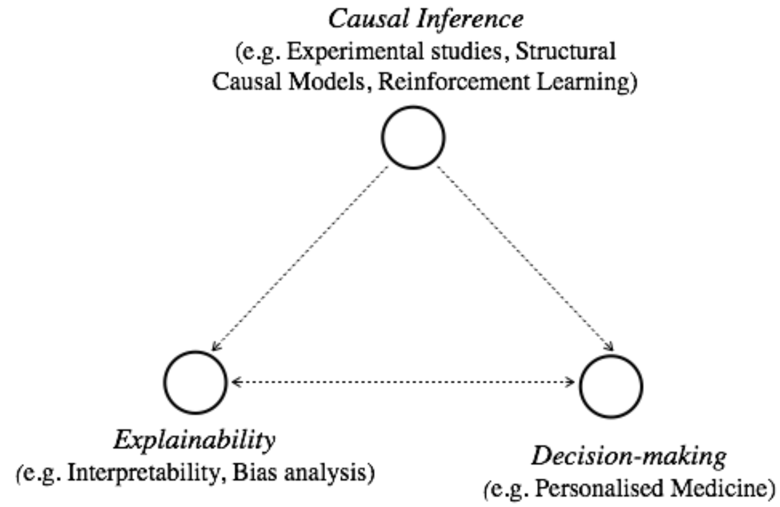

Causal Inference
(e.g. Experimental studies, Structural Causal Models, Reinforcement Learning)
Explainability
(e.g. Interpretability, Bias analysis)
Decision-making
(e.g. Personalised Medicine)

Figure 1.3: Illustration of the relationship between causal inference, decision-making and explainability. Explanations can aid decision-making if they are easy to understand and simple enough to be parsed. Similarly, reinforcement learning may be used to provide explainable recommendations via policies that are easy to step through, thus aiding decision-making. Causal inference is important for both explainability and decision-making since knowledge about interventions and their effects enables us to obtain meaningful explanations and informs better decisions.

thesis, we additionally study the problem of treating patients with sepsis. Here, we consider an alternative view to counterfactuals known as *decision-theoretic causality* which, like tools from the Intervention layer, enables us to directly examine the effects of an intervention. Specifically, we learn a low-dimensional compact representation of confounding that allows us to accurately estimate the effects of a therapy where only partial information is available. Finally, in the last part of this thesis, we introduce a new tree-based regularisation strategy for medical decision-making such that humans may step through and understand the predictions a system makes. Using both HIV and sepsis as application domains, we show that such explainability is important as it may shed light on the effects of interventions and allow us to tailor therapies to a patient's responses accordingly.

Having presented a brief overview of the thesis, we provide a detailed roadmap of how this thesis is structured. Chapter 2 serves as a general introduction to causal inference and provides an overview of two different perspectives of the field, namely the *potential outcomes framework* and *decision-theoretic causality.* We specifically show how both these views are related to Pearl's 3-level hierarchy and the theory of SCMs, as well as how they relate to each other. In the second part of Chapter 2, we introduce the RL framework which serves as the basis for two of our contributions in this thesis. In particular, we show that RL may also be expressed in terms of an SCM under certain conditions, thereby establishing a link between the Interventional and Counterfactual layers of the hierarchy. We subsequently describe how to perform inference and evaluate policies with such a model.

In Chapter 3, we present our first contribution where we combine RL with tools for associational modelling in a mixture-of-experts model to learn treatment policies for patients with HIV. In particular, the mixture-of-experts selectively alternates between a non-parametric expert and a parametric RL expert to personalise therapies according to a patient's particular needs. This approach is extended in Chapter 4 where we present our second major contribution. Here, instead of combining therapy policies as we do in Chapter 3, we directly incorporate the knowledge from associational modelling into an RL model to learn a more powerful causal model that we can use to generate counterfactuals and perform 'what-if' reasoning. This model allows us to simultaneously address patient heterogeneity and account for the effects of confounding. We then show how such a model can be used to infer state-of-the-art treatment strategies for both HIV and sepsis.

While the work in Chapter 3 and 4 combines different representations of causal knowledge with machine learning to learn better treatment policies, in Chapter 5 we shift our focus to learn better representations of confounders themselves. Specifically, we introduce our third major contribution, a variant of the Information Bottleneck method (Tishby et al., 2000) to learn a low-dimensional latent compression of confounding, and show how such a compression enables us to estimate both the average and specific causal effects of an intervention, even where covariate information is incomplete at test time. We subsequently apply this method to several benchmark problems as well as the tasks of treating sepsis and HIV.

Chapter 6 describes our final contribution where we introduce tree-based regularisation to optimise models for human simulatability. Importantly, this technique enforces that the predictions made by a model are explainable in terms of small decision-trees which can be traced through and audited. We demonstrate the importance of such explainability for decision-making and understanding treatment effects on both HIV and sepsis tasks. The thesis is concluded in Chapter 7 with a summary and a discussion on limitations of our work, as well as future research directions.

1.4 List of Publications

The following papers have resulted from some of the work presented in this thesis.

- *Cause-Effect Deep Information Bottleneck For Systematically Missing Covariates*
  Sonali Parbhoo, Mario Wieser, Aleksander Wieczorek, Volker Roth
  Under review, 2019.

- *Regional Tree Regularization for Interpretability in Black Box Models*
  Mike Wu, Sonali Parbhoo, Michael C. Hughes, Ryan Kindle, Leo Celi, Maurizio Zazzi, Volker Roth, Finale Doshi-Velez
  Under review, 2019.

- *Intelligent Policy Mixing for Improved HIV-1 Therapy Selection*
  Sonali Parbhoo, Jasmina Bogojeska, Mario Wieser, Fabricio Arend Torres, Maurizio Zazzi, Susana Posada Cespedes, Niko Beerenwinkel, Enos Bernasconi, Manuel Battegay, Alexander Calmy, Matthias Cavassini, Pietro Vernazza, Andri Rauch, Karin J. Metzner, Roger Kouyos, Huldrych Günthard, Finale Doshi-Velez, Volker Roth
  Under review, 2019.

- *Generative Subspace Learning Under Irrelevance Constraints in Continuous Domains*
  Mario Wieser, Sonali Parbhoo, Aleksander Wieczorek, Volker Roth
  Under review, 2019.

- *Determinants of HIV-1 Reservoir Size and Long-Term Dynamics*
  Nadine Bachmann, Chantal von Siebenthal, Valentina Vongrad, Teja Turk, Kathrin Neumann, Niko Beerenwinkel, Jasmina Bogojeska, Jacques Fellay, Volker Roth, Yik Lim Kok, Christian Thorball, Alessandro Borghesi, Sonali Parbhoo, Mario Wieser, Jurg Boni, Matthieu Perreau, Thomas Klimkait, Sabine Yerly, Manuel Battegay, Andri Rauch, Matthias Hoffmann, Enos Bernasconi, Matthias Cavassini, Roger Kouyos, Karin Metzner, Huldrych Günthard.
  To appear in Nature Communications, 2019.

- *Greedy Structure Learning of Hierarchical Compositional Models*
  Adam Kortylewski, Aleksander Wieczorek, Mario Wieser, Clemens Blumer, Andreas Morel-Forster, Sonali Parbhoo, Volker Roth and Thomas Vetter.
  To appear at CVPR, 2019.

- *Improving Counterfactual Reasoning with Kernelised Dynamic Mixing Models*
  Sonali Parbhoo, Omer Gottesman, Andrew Slavin Ross, Matthieu Komorowski, Aldo Faisal, Isabella Bon, Volker Roth, Finale Doshi-Velez
  PLoS One, Volume 13, Number 11, 2018.

- *Beyond Sparsity: Tree Regularization of Deep Models for Interpretability*
  Mike Wu, Michael C. Hughes, Sonali Parbhoo, Maurizio Zazzi, Volker Roth, Finale Doshi-Velez
  In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2018.

- *Combining Kernel and Model Based Learning for HIV Therapy Selection*
  Sonali Parbhoo, Jasmina Bogojeska, Maurizio Zazzi, Volker Roth, Finale Doshi-Velez
  AMIA Summits on Translational Science Proceedings, 2017.

- *Bayesian Markov Blanket Estimation*
  Dinu Kaufmann, Sonali Parbhoo, Aleksander Wieczorek, Sebastian Keller, David Adametz, Volker Roth
  In Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, 2016.

Chapter 2

Related Work

2.1 Introduction

In the first chapter, we discussed the notion of causality and its relevance for estimating treatment effects in the context of both HIV and sepsis. In particular, we emphasised the difference between *seeing* and *doing* using Pearl's hierarchy: Seeing involves passively observing a system in its natural condition; conversely, doing concerns the resulting behaviour of a system when it is disturbed upon *active intervention*. However, despite the widespread usage of statistics to address queries about association and seeing, there is little consensus as to how queries of causation and doing should be formalised. As a result, over the years several different formalisms for causal inference have been proposed. One such formalism is SCMs that we introduced in Chapter 1. In this chapter we focus on two different formalisms of causality, namely statistical decision theory and the potential outcomes framework. We specifically discuss the main theoretical concepts required for understanding the subsequent work presented in this thesis. We also draw links between causal inference and other machine learning methods for decision-making such as reinforcement learning, as well as supervised learning that appear in Chapters 3, 4 and 6.

2.2 Confounding and Simpson's Paradox

In healthcare, Randomised Control Trials (RCTs) are frequently viewed as the 'gold standard' for addressing causal questions and performing inference. The basic idea is to divide a set of patients into treatment and control groups. Patients from the treatment group are treated with a drug of interest, while patients from the control group are not allocated an intervention (termed a placebo control). The effects of the drug may then be assessed by comparing the outcomes of patients receiving treatment with the outcomes of those that do not. However, controlled experiments suffer from several drawbacks: they are expensive to conduct, may be ethically infeasible, are frequently susceptible to patient dropouts, and must typically be performed over several years. Alternatively, we must rely on observational data to draw inferences. However, in observational studies, the conditions across treatment groups may differ from those conditions we are interested in, and the characteristics of the individual subjects may not directly be comparable (Dawid, 2007a). As a result, it may be difficult to assess whether a particular effect is a direct result of an intervention or a consequence of other uncontrolled causes.

Consider the example of treating kidney stones (Charig et al., 1986). Table 2.1 shows the success of surgery (treatment A) vs. percutaneous nephrolithotomy (treatment B). While at first, treatment B seems appears more successful, it tends to be prescribed to patients with smaller, less severe, kidney stones. Is treatment B more effective, or is its success rate a consequence of being applied selectively to milder conditions? If we further divide the data on the basis of stone size, treatment A now achieves a better success rate for both patients with large and small kidney stones. This is known as Simpson's paradox (Simpson, 1951).

Treatment	**Overall**	**Patients with small stones**	**Patients with large stones**
A	78% (273/350)	**93%** (81/87)	**73%** (192/ 263)
B	**83%** (289/350)	87% (234/270)	69% (55/80)

Table 2.1: Illustration of Simpson's paradox. Treatment B appears better overall despite performing worse on patients with both large and small kidney stones (Charig et al., 1986).

Causal inference entails drawing conclusions based on responses to such interventions (Dawid, 2010). At its core, causal reasoning is centred around the concept of *confounding*. The example in Table 2.1 demonstrates the occurrence of confounding, where an observed outcome in the data may be a true effect, but may also be a result of several other factors that vary with the actual cause. These factors may sometimes be referred to as *confounding variables* whose consequences distort the true effect of an intervention. Together, the overall infeasibility of performing RCTs and the prevalence of confounding in observational data requires formalising the principles of causal inference.

2.3 Formal Frameworks For Causality

Several formal frameworks for causality have been proposed over the last few decades in an attempt to improve the rigour in addressing causal questions. These include graphical representations or causal diagrams, functional causal models, counterfactual models or models based on potential outcomes, and models based on statistical decision theory. We briefly describe each of these here before presenting more details in subsequent sections.

Graphical representations and causal diagrams are directed graphs that are used to visualise the causal relationships between variables in a model. Various different graphical representations have been proposed in e.g. Dawid (2002); Hernán & Robins (2006b); Pearl (2009). Functional causal models represent outcomes or effects of an intervention as a function of direct causes and some noise (Dawid, 2007a). In general, functional models assume that causal relationships can be represented entirely using deterministic functional equations; probabilities are introduced specifically under the assumption that certain variables are unobserved (Pearl, 2009). That is, the outcome is entirely deterministic given complete knowledge of the direct causes. SCMs are based on the idea of functional relationships, but coupled with graphical representations. Statistical models, unlike functional models, assume that there is always a degree of uncertainty that must be modelled, irrespective of whether a variable is observed or not. In particular, the decision-theoretic view of causal inference is based on probabilities and statistical modelling where the effect of a treatment is seen as the difference between two different probability distributions (Dawid, 2007a). Finally, counterfactual models measure causal effects as the difference between outcomes for an individual had they both taken and not taken the treatment (Sekhon, 2008). By definition, both of these outcomes cannot be observed simultaneously on a unit level, which necessitates joint probabilistic modelling of the underlying data distribution in order to make inferences[1].

Each of these frameworks has its merits, however we focus particularly on the decision-theoretic view of causal inference, and the counterfactual or potential outcomes perspective here. These formalisms differ in that they address different kinds of questions: the former can be used to address questions about effects of causes and is the primary focus of this thesis. In particular, when posing an effects-of-causes query, we ask a hypothetical question such as: *What would happen to my condition if I take a drug?* We can also consider alternative scenarios such as: *What would happen to my condition if I do not take the drug?* Conversely, the potential outcomes view (Rubin, 1974) can be used to address questions about causes of effects. These are questions such as: *What would have happened to my condition had I not taken a drug?* or *Was it because I took a drug that I observed certain outcomes?* The scenario is slightly different since the action has already been taken and an outcome has been observed (Dawid, 2007a). The query is counterfactual since it contradicts the fact that a certain action was taken in reality. Both of these frameworks may also be combined with graphical representations of causal models in order to make inferences. The subsequent sections describe these formalisms in more detail.

2.4 Graphs and Conditional Independence

We assume the reader is familiar with basic probability theory as described for instance by Klenke (2013). We also assume some familiarity of basic concepts in probabilistic graphical models, and graph theory such as d-separation (as in Koller & Friedman (2009)), but re-iterate some of these definitions where necessary. In particular, the properties of independence and conditional independence are crucial to understanding causality. We formalise these concepts here.

Definition 2.4.1 (Conditional Independence)**.** *A random variable X is independent of another random variable Y under the distribution P if $P(X \in \mathcal{X}|Y) = P(X \in \mathcal{X})$ for any set $\mathcal{X}$ of X. Two random variables X and Y are conditionally independent given a third random variable Z if and only if $P(X \in \mathcal{X}|Y, Z) = P(X|Z)$. We denote this as $X \perp\!\!\!\perp Y|Z$.*

Note that independence occurs as a special case of conditional independence where the conditioning variable Z is trivial or constant. The key properties of conditional independence can be found in Dawid (1979a,b). These properties are usually represented in terms of graphs. In particular, we consider the notion of a Directed Acyclic Graph (DAG) which allows us explicitly to visualise the specific dependence relations among a set of variables.

[1] SCMs may also be used as counterfactual models as presented in Chapter 1. Here, uncertainty is introduced specifically under the assumption that one outcome is unobserved. This relates back to Figure 1.1 in Chapter 1.

Definition 2.4.2 (Directed Acyclic Graph)**.** *Let $G = (\mathcal{V}, \mathcal{E})$ be a graph of vertices $\mathcal{V} := \{1, \ldots, p\}$ corresponding to random variables $X = (X_1, \ldots, X_p)$ with joint distribution P. G is called a DAG if there is no pair (X_j, X_k), for which directed paths from X_j to X_k or X_k to X_j exist.*

Conditional independence between two random variables in a DAG may be indicated by the absence of an edge between their corresponding vertices in a DAG. Note that the DAG representation satisfying a set of conditional independence properties is not necessarily unique.

Definition 2.4.3 (Markov Equivalence)**.** *Two DAGs are termed Markov equivalent if they represent identical collections of conditional independence relations.*

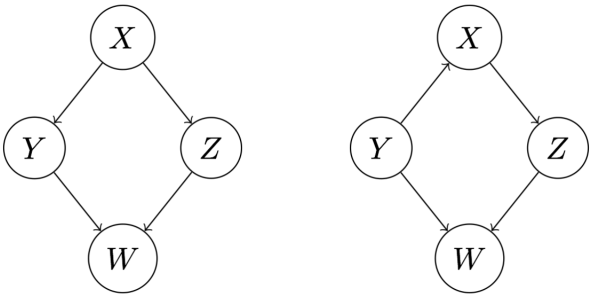


Figure 2.1: Illustration of Markov equivalence. Both DAGs represent the same conditional independences: $W \perp\!\!\!\perp X|Y, Z$ and $Z \perp\!\!\!\perp Y|X$.

Examples of two Markov equivalent DAGs are illustrated in Figure 2.1. Importantly, conditional independence is a *symmetric* relationship since $X \perp\!\!\!\perp Y|Z \implies Y \perp\!\!\!\perp X|Z$. However, a DAG representation by nature consists of *directed* edges between variables, thus implying a non-symmetric relationship between the nodes. This is a mere artefact of the graphical representation and should not be interpreted in terms of causes and effects (Dawid, 2010). To avoid ambiguity we refer to such DAGs as probabilistic DAGs.

2.4.1 Interventions and Causal Graphs

Thus far, we have informally described the concept of an intervention in terms of administering treatments. In order to distinguish between probabilistic DAGs and graphs representing causes and effects, we formalise this notion here.

Definition 2.4.4 (Intervention)**.** *An intervention refers to a forced change in a system that explicitly assigns a random variable X to a particular value x, written as $do(x)$*[2]*. The effects of an intervention on a random variable Y can be defined by the interventional distribution $P(Y|do(x))$. Note that in general, $P(Y|do(x)) \neq P(Y|X = x)$.*

[2] An in-depth discussion of do-calculus is beyond the scope of this thesis. We refer the reader to Pearl (1995, 2012a) for a detailed treatment.

At this point, we distinguish between the observational regime in which no interventions are performed, denoted by the indicator $F_\emptyset$, and the interventional regime, where we intervene on a random variable X and $do(x)$. We denote the latter as F_X hereafter.

In order to distinguish between probabilistic DAGs and DAGs that are capable of representing interventions and their effects, Definition 2.4.2 is extended in Dawid (2010) to include vertices that represent non-random regime variables.

Definition 2.4.5 (Influence Diagram). *An influence diagram is a probabilistic DAG that is extended to include non-random regime variables (indicated as squares) in addition to the nodes representing domain variables (indicated as circles).*

Influence diagrams that include interventional regime nodes F_X are termed *augmented DAGs* (Dawid, 2010). A particular instance of augmented DAGs is a *Pearlian DAG* (Pearl, 1995).

Definition 2.4.6 (Pearlian DAG). *A Pearlian DAG $\mathcal{G}$ is an augmented DAG, where for every random variable X there exists a corresponding intervention node F_X and an arrow pointing from F_X to X. An arrow from random variable X to Y in such a graph has a causal interpretation i.e X causes Y. Pearlian DAGs may thus also be referred to as causal graphs.*

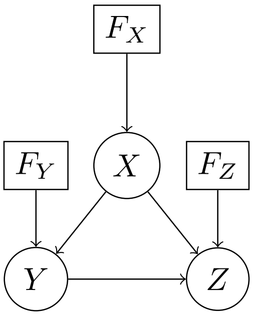


Figure 2.2: Illustration of a Pearlian DAG. Every random variable has a corresponding interventional node.

An example of Pearlian DAG is shown in Figure 2.2. It should be noted that while Pearlian DAGs have become a popular framework for causal reasoning, both Pearlian DAGs and augmented DAGs are special forms of influence diagrams that are frequently used in the decision-theoretic perspective of causality[3].

2.4.2 Causal Identification and The Backdoor Criterion

Evidently, estimating the causal effects on the basis of observational data requires correcting for confounding bias. This process is known as *causal identification.* Causal identification is formalised in Guo et al. (2018) in terms of the interventional distribution as,

[3] Note that augmented DAGs are Pearlian DAGs are not the only means of representing causal relations. Other representations include functional graphs. A detailed discussion of these representations is beyond the scope of this thesis. We refer the reader to Dawid (2007a) for an in-depth treatment of these.

Definition 2.4.7 (Causal Identification)**.** *A causal effect is identifiable if and only if the interventional distribution can be expressed as a function of probability distributions.*

The implication of this definition is that in order to identify the true causal effects of an intervention, we must account for irrelevant effects. In order to correct for confounding, one may estimate causal effects on subgroups where instances are homogenous with respect to confounding. Analogously, Pearl (2009) formulates this in terms of the *back-door criterion.*

Let $\mathcal{G}$ be a Pearlian DAG and $\mathcal{V} \subset \mathcal{G}$ be a set of observed variables from a non-experimental data set. Assume we would like to estimate the effect of interventions $do(X = x)$ on a set of outcome variables Y, where $X, Y \subset \mathcal{V}$. The back-door criterion provides a graphical test that can be applied to a Pearlian DAG to assess whether a subset $Z \subseteq \mathcal{V}$ of variables suffices in identifying the true causal effect. A back-door path in a Pearlian DAG is a directed path or set of edges from X_i to X_j with an arrow leading into X_i (Pearl, 2009). The back-door criterion may then be formalised as follows (Pearl, 2009).

Definition 2.4.8 (The Backdoor Criterion)**.** *A set of variables Z in $\mathcal{G}$ satisfies the back-door criterion relative to an ordered pair of variables (X_i, X_j) if and only if:*

- *No node in Z is a descendant of X_i, and*
- *Z blocks all back-door paths between X_i and X_j.*

Equivalently, if X and Y are two disjoint subsets of vertices $\mathcal{V}$ in $\mathcal{G}$, then Z satisfies the back-door criterion relative to (X, Y) if it satisfies the criterion relative to any pair (X_i, X_j) where $X_i \in X$ and $X_j \in Y$.

The first condition of the backdoor criterion is equivalent to having no back-door paths from Z to X_i. This generally occurs in randomised studies. The second condition may hold in observational studies as well. Overall, the backdoor criterion is fairly powerful as it can be used to identify whether there is confounding in a causal graph, and what variables must be conditioned on to correct for such confounding. That is, if Z satisfies the back-door criterion, it can be used to adjust for confounding and computing the causal effect of X on Y. This result is summarised in Theorem 2.4.1.

Theorem 2.4.1 (Backdoor Adjustment Theorem)**.** *If a set of variables Z satisfies the back-door criterion relative to (X, Y) then the causal effect of X on Y is identifiable and is given by $\sum_z P(y|x, z)P(z)$.*

2.5 Statistical Decision Theory for Causal Inference

This section discusses the decision-theoretic framework of causal inference. This perspective is rooted in basic concepts from probability theory and statistics. We refer to Dawid (2012) for the material throughout this section.

Consider the following example of a simple decision problem. Suppose I have a headache and I want to know whether taking an aspirin will help. This could be viewed as a statistical decision problem that consists of a non-stochastic decision variable T corresponding to the decision to treat with aspirin ($T = t$) or not to treat ($T = c$), and a stochastic outcome Y representing my outcomes in terms of e.g. the length of time

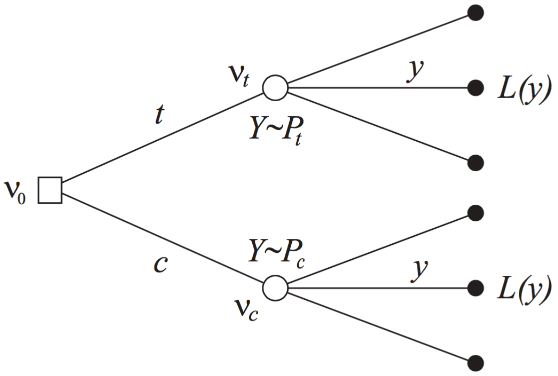


Figure 2.3: Tree representation of a decision problem (Dawid, 2012).

that the headache lasts for after I make decision T. For both treatment choices, I am uncertain about Y. This uncertainty in outcomes may be modelled using probability distributions. The decision-theoretic view of causal inference considers two separate distributions of outcomes given the treatment or control, P_t and P_c, and explicitly computes a loss $L(y)$ I suffer based on the true outcomes $Y = y$ for each action choice. The choice to treat with t is made using Bayesian decision theory if,

$$\mathbb{E}_{Y \sim P_t}[L(Y)] \leq \mathbb{E}_{Y \sim P_c}[L(Y)], \tag{2.5.1}$$

where $\mathbb{E}$ represents the expectation over a particular distribution. That is, I should take an aspirin if $\mathbb{E}_{Y \sim P_t}[L(Y)] - \mathbb{E}_{Y \sim P_c}[L(Y)] < 0$. The decision problem may be illustrated in terms of a decision tree shown in Figure 2.3. The arms in the decision tree correspond to the treatment decisions and respective outcomes.

This example of choosing whether or not to treat a headache with aspirin is rooted in causal inference since we would like to know: *What is the effect of the causal action of taking treatment T on outcomes Y?* Our goal in this setting is then to estimate the *Average Causal Effect (ACE)* of T on Y. If we assume $F_T = t$ or $F_T = c$ define the interventional regimes, and $F_T = \emptyset$ the observational regime, the ACE is given by,

$$ACE := \mathbb{E}[Y|F_T = t] - \mathbb{E}[Y|F_T = c]. \tag{2.5.2}$$

Importantly, because of its dependence on the interventional distributions, the ACE is considered a causal quantity rather than an associational one. Graphically, this may be illustrated as the difference between the means of two distributions as in Figure 2.4. Equation 2.5.2 shows that it is possible to assess the ACE in the interventional regime. However, in reality we are given purely observational data from which we would like to infer the causal effects. Hence we would like to be able to extend the notion of the ACE to the observational regime $F_T = \emptyset$. The subsequent sections discuss when this is possible, and how the ACE can be computed in these cases.

2.5.1 Estimating Treatment Effects with No Confounding

In some cases, it is possible to ignore the treatments assigned to patients. This occurs in certain situations such as randomised control studies. Here, we can use the observational distribution to estimate causal effects without having to correct for the effects

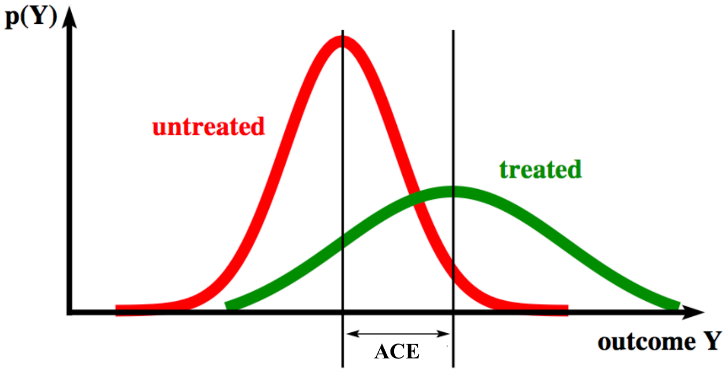


Figure 2.4: Illustration of the ACE. The ACE is the difference in average outcomes over interventional distributions $F_T = t$ (treated) and $F_T = c$ (untreated).

of confounding or biases. This property can be referred to as *no confounding* (Dawid, 2012), *ignorable treatment assignment* (Rosenbaum & Rubin, 1983), or *no unmeasured confounding*. In this case, the ACE may be computed by simply replacing the interventional distributions in Equation 2.5.2 with their observational counterparts. That is, the ACE is given by,

$$ACE := \mathbb{E}[Y|T = t, F_T = \emptyset] - \mathbb{E}[Y|T = c, F_T = \emptyset]. \tag{2.5.3}$$

No confounding may be illustrated in terms of the causal graph shown in Figure 2.5. Evidently, in this case we can ignore the interventional distribution F_T as the back-door criterion in Definition 2.4.8 is satisfied and $Y \perp\!\!\!\perp F_T|T$.

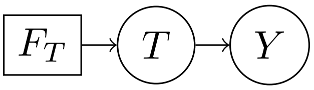


Figure 2.5: Causal graph of ignorable treatment assignment.

2.5.2 Estimating Treatment Effects with Confounding

In observational studies, we will almost always have confounding. In such scenarios, the ignorable treatment assignment assumption may not be directly applicable. However when we cannot assume ‘no confounding’, we might be able to tell an alternative story in terms of an additional set of variables U that, once conditioned on, ensure that we have no residual confounding. For example, consider the scenario where we have data from an observational study on patients that are treated by a particular doctor allocating treatments based on his own observations U of the general health of the patient. If we consider U to be the complete set of observations on a patient the doctor takes into account, it may be plausible to assume that there is no residual confounding. That is, $Y \perp\!\!\!\perp F_T|U, T$. In this situation, U is measured before the treatment decision is made and is termed an *unconfounder*. Because U is a pre-treatment variable, it has the same distribution in both interventional regimes $F_T = c$ and $F_T = t$. Importantly, if we can observe U, $P(Y|U, F_T = i) = P(Y|U, T = i, F_T = \emptyset)$ for $i = t$ or c. In this case, U is known as a *sufficient covariate*. This scenario is illustrated in Figure 2.6.

Two specific instances of unconfounding occur when either arm α or arm β in Figure 2.6 is removed. In the former case, the sufficient covariate has no impact on the choice of treatment i.e $T \perp\!\!\!\perp U|F_T$. This occurs when variables U that may have affected the decision did not in fact do so. This scenario is equivalent to a randomised study and we can treat it as such where $T \perp\!\!\!\perp U|F_T = \emptyset$. The latter case arises when $Y \perp\!\!\!\perp U|T$ and the outcome does not in fact depend on U. This scenario is equivalent to no confounding.

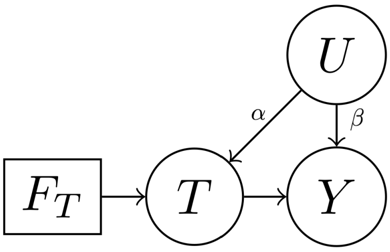


Figure 2.6: Causal graph of a sufficient covariate U.

In general however, if U is a sufficient covariate, we can use the back-door criterion (Pearl, 2009) to replace the interventional distribution with its observational counterpart and compute the ACE. In this case, the ACE may be computed in terms of the *Specific Causal Effect* (SCE). We formalise this concept as follows:

Definition 2.5.1 (Specific Causal Effect). *The Specific Causal Effect (SCE) of T on Y (relative to U) is the random variable*

$$\begin{aligned} SCE(U) &:= \mathbb{E}[Y|U, F_T = t] - \mathbb{E}[Y|U, F_T = c] && (2.5.4) \\ &:= \mathbb{E}[Y|U, T = t, F_T = \emptyset] - \mathbb{E}[Y|U, T = c, F_T = \emptyset]. && (2.5.5) \end{aligned}$$

The SCE is a random variable whose value is the average causal effect in the group of individuals with covariate $U = u$ and is thus sometimes referred to as the *Conditional Average Causal Effect*[4]. Because the SCE in Equation 2.5.4 is defined in terms of the interventional regime, it is considered a causal quantity. However, given sufficient covariate U, Equation 2.5.5 in Definition 2.5.1 shows that the SCE can also be estimated from observational data alone. Analogously, given the SCE, the ACE can be computed as,

$$ACE := \mathbb{E}[SCE(U)|F_T = \emptyset] \qquad (2.5.6)$$

Overall, since the decision-theoretic approach to causal inferences tackles the effects-of-causes question, it may be seen as part of the Intervention layer of Pearl's 3-level hierarchy. In the next section, we present an alternative approach to causal inference namely the potential outcomes framework. Unlike the decision-theoretic approach, the potential outcomes approach addresses the alternative causes-of-effects question and can be viewed as part of the Counterfactual layer of Pearl's hierarchy.

[4] If the group only consists of one individual, the SCE is analogous to the Individualised Causal Effect in the Potential Outcomes framework.

2.6 The Potential Outcomes Framework

The decision-theoretic approach to causal inference assumes we have a single response variable Y but different probability distributions for different regimes. An alternative formulation that is frequently used for modelling counterfactuals is the potential outcomes framework (Rubin, 1978; Splawa-Neyman, 1923, 1990). In contrast to the decision-theoretic approach, the response variable Y has several copies, each corresponding to the treatment variable T, but having a single joint distribution P. Assume we have two choices of taking a treatment t, and not taking a treatment (control) c. Let Y_t denote the outcomes under t and Y_c denote outcomes under the control c. The counterfactual approach assumes that there is a pre-existing joint distribution of potential responses $P(Y_t, Y_c)$. This joint distribution is hidden since t and c cannot be applied simultaneously. Applying an action t thus only reveals Y_t, but not Y_c. In this setting, computing the effect of an intervention involves computing the difference between when an intervention is made and when no treatment is applied (Morgan & Winship, 2015; Pearl, 2009). We would subsequently choose to treat with t if,

$$\mathbb{E}[L(Y_t)] \leq \mathbb{E}[L(Y_c)] \tag{2.6.1}$$

for loss L over Y_t and Y_c respectively. In this formulation, if there is no confounding, we can estimate the treatment effect if there is probabilistic independence between $Y = (Y_t, Y_c)$ and T. This corresponds to the decision-theoretic setting where $Y \perp\!\!\!\perp F_T | T$.

A key distinction between the potential outcomes framework and the decision-theoretic approach to causal inference is the concept of the *Individualised Causal Effect* (ICE) Dawid (2012). Let $\mathcal{I}$ denote the individuals or instances for which we wish to define the causal effect. The ICE may be formalised as follows.

Definition 2.6.1 (Individualised Causal Effect)**.** *Assuming binary treatments, the Individualised Causal Effect (ICE) for $i \in \mathcal{I}$ is defined as*

$$ICE(\mathcal{I}) := Y_t^i - Y_c^i, \tag{2.6.2}$$

for potential outcomes Y_t^i and Y_c^i.

Note that the ICE has no immediate counterpart in the decision-theoretic setting (Dawid, 2012). Given the definition of the ICE, it is possible to extend this to the ACE over the treated individuals and those that serve as controls. Here, the ACE may be calculated as,

$$ACE := \mathbb{E}[ICE(\mathcal{I})] = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} Y_t^i - Y_c^i, \tag{2.6.3}$$

where $|\mathcal{I}|$ denotes the size of the population (Guo et al., 2018). This is similar to Equation 2.5.6 in the decision-theoretic setting. However, unlike the decision theoretic setting, the key difficulty with the potential outcomes framework is existence of the so-called 'fundamental problem of causality': we can never simultaneously observe Y_t^i and Y_c^i for an individual i. As a result, there are parts of the bivariate distribution $P(Y)$ that cannot be learned on the basis of data alone. This was discussed in Chapter 1. Thus in such a setting, we require additional knowledge of the underlying data generating process. Hence in the potential outcomes setting, additional assumptions often have to be made in order to infer the effects of an intervention. We briefly describe two such assumptions (apart from the notion of unconfoundedness or no confounding) in what follows (Guo et al., 2018).

The Stable Unit (Instance) Treatment Value Assumption (SUTVA) This assumption requires that there are a) well-defined levels of treatment, and b) there is no interference. a) shows that for two different instances i and $i' \in \mathcal{I}$, their treatment variables are identical if they receive the same treatment. b) indicates that the potential outcomes for an individual i are independent of the potential outcomes of another individual i'.

Consistency This assumption means that the value of the potential outcomes does not vary according to how the treatment is observed or how an intervention is assigned.

Note that the decision-theoretic setting is largely free of such assumptions, and one may choose to either assume or not assume such conditions according to a particular scenario (Dawid, 2012).

2.7 The Backdoor Criterion and Supervised Learning

2.7.1 Regression Adjustment

The backdoor criterion from Section 2.4.2 enables us to determine how to learn causal effects by adjusting or conditioning on a set of variables that block all backdoor paths. In the case where all confounders are measured, one way to perform such an adjustment is via regression. If we consider the potential outcomes approach from the previous section, this entails inferring counterfactual outcomes using supervised learning. Specifically we can use a set of features or covariates X and treatment T to fit $P(Y|X,T)$. When all backdoor paths are blocked, no confounding bias remains. Counterfactual outcomes may be computed by considering treatments that differ to those taken. For instance, if we consider the simple case of treatment t and control c, if a patient i receives t, the outcomes are given by $P(Y_t^i|X, F_T = t)$, while the counterfactual outcomes are given by $\mathbb{E}[P(Y_c^i|X, F_T = c)]$. Alternatively, one may consider fitting two separate functions for estimating both potential outcomes. The ACE may subsequently be computed as,

$$ACE := \mathbb{E}[\hat{Y_t}^i - \hat{Y_c}^i|X, F_T = \emptyset], \tag{2.7.1}$$

where $\hat{Y_t^i}$ and $\hat{Y_c^i}$ are the regression estimates for Y_t and Y_c respectively. The key assumption of the potential outcomes framework is that outcomes $Y = (Y_t, Y_c)$ have a pre-existing joint distribution. However, fitting two separate functions for estimating both potential outcomes and computing the ACE using these is analogous to Equation 2.5.3 in the decision-theoretic setting, where $T = t$ or $T = c$.

2.7.2 Propensity Analysis

The regression approach for identifying causal effects suffers from several problems. In particular, if we have very different values of covariates for treatment and control groups, there may be little overlap between the two groups which makes it difficult to compare them. In such a case, the results will be highly sensitive to the particular regression model and any other apriori assumptions. *Propensity score methods* are an alternative to adjusting for observed confounding using regression (Horvitz & Thompson, 1952). They are based on the idea of statistical matching. The primary goal of matching is to reduce bias in observational studies by mimicking randomisation such that we can determine for every patient receiving treatment, a similar non-treated patient that is comparable across all observations. Here, a set of observational data is divided into strata and each stratum is subsequently viewed as a randomised study (Guo et al., 2018). In doing so, the ACE is identifiable and can be computed over each stratum. The key difficulty arises when strata are imbalanced i.e. contain data for only treated or untreated individuals, since the ACE cannot be computed in this case.

Several *weighting* methods have been proposed to overcome the problem of imbalanced strata. These are primarily rooted in the idea of a *propensity score* (Horvitz & Thompson, 1952). Assuming we have binary treatment T, outcomes Y and covariates X, the propensity score may be defined as the conditional probability of receiving treatment on the basis of the covariates, $P(T|X)$ (Rosenbaum & Rubin, 1983). Propensity scores may be computed by training a classifier as in standard supervised learning to predict the likelihood of receiving treatment on the basis of a covariate set. A popular example of this is using the logistic function as described by Rosenbaum & Rubin (1983). Given such a propensity score, different methods of propensity score weighting have been proposed to address data set imbalance. These include *propensity score matching (PSM)* and *inverse probability treatment weighting (IPTW)* (Guo et al., 2018).

PSM pairs each treated instance with a group of comparable, non-treated instances with similar propensity score estimates (Abadie & Imbens, 2011). Upon matching, the ACE can be estimated as the average over the difference between the outcomes observed for the treated and control groups. Several different approaches to matching propensity scores exist, such as nearest-neighbour or kernel matching, stratification matching and Mahalanobis matching (Becker & Ichino, 2002). The overall idea of these is to express the outcome for a treated individual as a weighted combination of the outcomes in a similar group.

Unlike matching, IPTW (Hirano et al., 2003; Horvitz & Thompson, 1952) tries to use all data but down-weight or up-weight instances such that a randomised control trial may be synthesised (Austin, 2011). This is accomplished by weighting instances according to the inverse of the probability of treatment received or the inverse of the propensity score. The weight w_i is frequently given by,

$$w_i = \frac{T^i}{P(T^i|X^i)} + \frac{1 - T^i}{1 - P(T^i|X^i)}, \tag{2.7.2}$$

where T^i and X^i denote the treatments assigned, and covariates corresponding to instance i (Guo et al., 2018). Given w_i for both treatment and control groups, the ACE may subsequently be estimated as the weighted averages over outcomes in these groups.

Recall that in Section 2.5, we introduced the decision-theoretic perspective of causality and showed that identifying the sufficient covariate enables us to transfer knowledge from an interventional regime to an observational setting to estimate treatment effects. Importantly, the sufficient covariate is not necessarily unique (Dawid, 2012). As a result, once we identify such a sufficient covariate, it may be possible to further reduce some information in the covariate. Propensity analysis might be used for this purpose. In Chapter 5, we present an alternative perspective using the Information Bottleneck Principle (Tishby et al., 2000) that allows us not only to perform a sufficient reduction of the covariate to infer treatment effects, but also allows us to further reduce the covariate if required while maintaining sufficiency. Unlike propensity analysis, the Information Bottleneck takes into account the relevance of information in order to perform such a reduction.

2.8 Reinforcement Learning and Causal Inference

In this section, we shift our focus to RL and decision-making and describe the importance of causal knowledge for RL systems. RL is a machine learning paradigm in which a decision-maker or *agent* interacts with an *environment* to learn a task (Sutton & Barto, 1998). Typically, this interaction consists of three components: a *state* summarising the current situation of the environment, an *action* (equivalent to an intervention) that allows an agent to intervene in the environment, and a *reward* outcome produced by the environment that provides the agent with feedback on its choice of intervention at a particular time. The overall aim of RL is to determine a series of interventions or *policy* such that the future reward outcomes may be maximised (Sutton & Barto, 1998). As in the previous sections, this involves reasoning about the effects of interventions. While the theory of RL is largely based on engineering techniques used in process control problems and planning, it relies on certain causal assumptions and can be formulated in terms of SCMs (Bellman, 1958), thereby allowing us to connect both the Interventional and Counterfactual levels of Pearl's hierarchy in Chapter 1. Next, we show how RL may be formalised and draw these connections.

The basic setup of an RL problem consists of an agent interacting with an environment to learn a task. At a particular time[5] $\mathtt{t}$, the agent finds itself in state $s_{\mathtt{t}}$ from a set of possible states $\mathcal{S}$ describing the condition of the environment; the agent takes an action or performs an intervention $a_{\mathtt{t}} \in \mathcal{A}$ defined by a policy $\pi(a_{\mathtt{t}}|s_{\mathtt{t}})$, observes a reward outcome $r_{\mathtt{t}} \in \mathbb{R}$ with expectation $\mathcal{R}(s_{\mathtt{t}}, a_{\mathtt{t}})$, and moves to a subsequent state $s_{\mathtt{t+1}}$ based on a transition function $\mathcal{T}(s_{\mathtt{t+1}}|s_{\mathtt{t}}, a_{\mathtt{t}})$. This process is repeated such that the agent continues to collect rewards. Together, the states, actions, rewards and environment dynamics constitute a *Markov Decision Process (MDP)*. That is, an MDP is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R})$.

The agent's goal in RL is to maximise the cumulative reward over the future (or the *return*),

$$R = \sum_{\mathtt{t}=0}^{\infty} \gamma^{\mathtt{t}} r_{\mathtt{t}}, \tag{2.8.1}$$

where $\gamma \in [0, 1]$ is a discount factor that favours immediate rewards.

Evidently, the problem of tailoring treatments to a patient's needs (e.g. for HIV or sepsis) may be formulated as an RL problem wherein, a patient finds themselves in a particular state of health, based on which a doctor can prescribe certain therapies and observe patient responses. Given such a series of interactions between the patient and the doctor, the overall aim of the doctor is to deduce a suitable therapy policy for the patient such that their outcomes (e.g. chances of survival) may be optimised in the future. This goal may also be formulated in terms of a *value function* that describes how good it is to be in a particular state. The value function, V for a particular policy π, is given by,

$$V(s) = \mathbb{E}\left[R|s_0 = s\right]. \tag{2.8.2}$$

[5] To avoid confusion, we use $\mathtt{t}$ to denote time steps, while t denotes treatment assignments as in the decision-theoretic and potential outcomes frameworks.

Alternatively, it is often more useful to estimate the value of a state-action pair, $Q(s,a)$, that determines how good it is to be in a particular state and take a certain action. This is given by,

$$Q(s,a) = \mathbb{E}\left[R|s_0 = s, a_0 = a\right]. \tag{2.8.3}$$

The optimal action a^* from a state s is the one that maximises the value function $Q(s,a)$,

$$a^* = \arg\max_a Q(s,a). \tag{2.8.4}$$

Importantly, the environment satisfies the Markov Property, such that both transitions from one state to another and rewards are independent of the agent's history based on the current state and action. This enables us to re-express Equation 2.8.3 recursively as a *Bellman Equation*,

$$Q(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{T}(s'|s,a) \sum_{a'} \pi(a'|s') Q(s',a'), \tag{2.8.5}$$

where s' and a' denote future states and actions respectively. Here, the first term describes the immediate return while the second corresponds to the discounted expected future reward under a policy π. Equation 2.8.5 may be optimised thus,

$$Q^*(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{T}(s'|s,a) \max_{a'} Q^*(s',a'), \tag{2.8.6}$$

where we replace $\pi^*(a'|s')$ with $\arg\max_a' Q(s',a')$. This equation forms the foundation for learning optimal intervention policies and therapy planning.

2.8.1 Formulating Reinforcement Learning as a Causal Model

RL methods may sometimes be divided into *model-based* and *model-free* learning (Sutton & Barto, 1998). Model-based methods explicitly construct a model of the agent's interaction with the environment to estimate the value function. Model-free methods try to estimate the value function directly on the basis of experience, However, since both model-based and model free methods rely on the concept of a state (which may or may not be hidden), causal knowledge plays an important role in both methods of learning (Gershman, 2017). Specifically, RL assumes the following causal relations hold: the state and action *cause* the reward outcome; the state and action *cause* the subsequent state; in partially observable cases, the hidden state *causes* an observation outcome. Consequently, it can be shown that MDPs may be formulated as SCMs. In this case, the following structural equations hold,

$$\begin{aligned} a_{\mathtt{t}} &= \pi(s_{\mathtt{t}-1}) + \epsilon_{\mathtt{t}}, \\ r_{\mathtt{t}} &= \mathcal{R}(s_{\mathtt{t}}, a_{\mathtt{t}}) + \epsilon_{\mathtt{t}'}, \\ s_{\mathtt{t}} &= \mathcal{T}(s_{\mathtt{t}-1}, a_{\mathtt{t}}) + \epsilon_{\mathtt{t}''}. \end{aligned} \tag{2.8.7}$$

Note that in this form, rewards and transitions are viewed as functions, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, rather than probabilities (or probability distributions) as in the classical formulation of RL, where knowledge about noise terms $\epsilon_{\mathtt{t}}, \epsilon_{\mathtt{t}'}, \epsilon_{\mathtt{t}''}$, allows us to specify the MDP completely. In general, Equations 2.8.7 may be replicated for all time steps.[6] Given this formulation, it is possible to restate the goal of RL as learning a policy π^* such that a sequence of interventions $F_T = \{a_1 = F_{T1}, a_2 = F_{T2}, \ldots\}$ maximises $\mathbb{E}[R|F_T]$. Based on this, there are two primary strategies to causal inference using reinforcement learning, namely *online learning* and *off-policy learning.* Both online learning and off-policy learning play an important role in the models we develop in Chapters 3 and 4. We briefly provide an overview of both types of learning.

2.8.2 Online Learning

Online learning occurs where an agent actively performs an experiment or intervention themselves (Bareinboim, 2018). In this case, the agent receives as input a series of experiments and observes outcomes. For instance, a doctor may actively intervene with F_T and observe patient outcomes now in terms of a R. Given the set of *experiments* and corresponding outcomes for subjects i, $\{F_{Ti}, R_i\}$, the agent learns the probability of outcomes $P(R|F_T)$. Based on this, it is straightforward to estimate $\mathbb{E}[R|F_T]$ through a series of randomised experiments as in a RCT as in Figure 2.6 where α is removed, or via MDPs or partially observable MDPs (POMDPs) (Bareinboim, 2018).

In the next two chapters, we explicitly make use of online search algorithms such as forward search that enable policy execution and action planning in a reinforcement learning setting. Importantly, at each stage of these algorithms, we may actively intervene by performing a particular action and simulate experience following such interventions, analogous to performing interventions via experimentation. In doing so, we may also generate counterfactuals, since we can adjust our interventions to explore alternative scenarios via simulation. This procedure is analogous to using SCMs for counterfactual reasoning.

2.8.3 Off-Policy Learning and Evaluation

Frequently, an agent uses the retrospective observational data based on another agent's actions to either learn an optimal policy or evaluate a particular policy of interest. This is termed *off-policy learning/evaluation.* Here we are given a set of input *samples* rather than experiments, and corresponding outcomes, $\{F_{Ti}, R_i\}$; the agent again learns the probability of outcomes $P(R|F_T)$. However, because the estimate $\mathbb{E}[R|F_T]$ is based off the data from other agents operating under unknown policies, this requires additional assumptions that the same variables were randomised and the situations or contexts were similar (Bareinboim, 2018). Off-policy learning is important across several domains such as healthcare, where active experimentation is not possible to collect samples (Schulam & Saria, 2017). The key challenge is to use the sequence of states, actions and rewards of an agent operating under some unknown behavioural policy, π_b, to estimate the reward under a target policy of interest, π_e (Schulam & Saria, 2017). Evidently, this requires adjusting for the difference in observational and interventional regimes as discussed earlier in this chapter. To do so, off-policy algorithms rely on variants of reweighting or matching strategies similar to IPTW to estimate the expected reward. Conceptually, all these estimators try to identify a subset of the data where π_b coincides with π_e and assign weights to samples such that they appear as if they were drawn

[6] In this form, MDPs may also be viewed as a sequence of contextual bandits, where the context is an endogenous variable that depends on previous states and actions (Bottou et al., 2013).

from the policy of interest π_e. In this thesis, we make use of importance sampling (IS), weighted importance sampling (WIS) and doubly robust (DR) estimators. The classic IS estimator (Kahn & Marshall, 1953; Koller & Friedman, 2009; Rubinstein, 1981) over the value function V of policy π_e is given by,

$$\hat{V}^{\pi_e}_{IS} = \frac{1}{N}\sum_{n=1}^{N} w^{h_n} R^{h_n}, \tag{2.8.8}$$

where h_n is the history of a patient n, R^{h_n} is the total reward accumulated over the patient's history, and w^{h_n} is an importance ratio that reflects how relevant the n^{th} sample is for estimating $V^{\pi_e}_{IS}$. Here, histories that are unlikely are given a smaller weight when evaluating a policy. The importance ratios w^{h_n} (Precup, 2000) may be computed according to,

$$w^{h_n} = \prod_{\mathtt{t}=0}^{T_{h_n}} \frac{\pi_e(a^{h_n}_{\mathtt{t}}|s^{h_n}_{\mathtt{t}})}{\pi_b(a^{h_n}_{\mathtt{t}}|s^{h_n}_{\mathtt{t}})}, \tag{2.8.9}$$

where T_{h_n} denotes the number of time steps in history n, and $\pi_e(a^{h_n}_{\mathtt{t}}|s^{h_n}_{\mathtt{t}})$ and $\pi_b(a^{h_n}_{\mathtt{t}}|s^{h_n}_{\mathtt{t}})$ are the probabilities of taking action $a^{h_n}_{\mathtt{t}}$ from a state $s^{h_n}_{\mathtt{t}}$ at time $\mathtt{t}$ under the evaluation and behavioural policies respectively. Evidently, when a clinician's policy differs significantly from the evaluation policy of interest, the corresponding importance weight will be small. Since the IS estimator is unbiased but prone to high variance, a variant known as weighted-IS is often used for off-policy evaluation. This estimate can be computed as a weighted average of the samples,

$$\hat{V}^{\pi_e}_{WIS} = \frac{\frac{1}{N}\sum_{n=1}^{N} w^{h_n} R^{h_n}}{\frac{1}{N}\sum_{n=1}^{N} w^{h_n}}. \tag{2.8.10}$$

The doubly robust off policy evaluation scheme (DR) (Jiang & Li, 2015; Thomas & Brunskill, 2016) further attempts to reduce the variance (with potentially additional bias) by adding control variates. Specifically, it combines IS estimates from Equation 2.8.9 with an approximate (regression) model of the action-value $\hat{Q}^{\pi_e}$ and value $\hat{V}^{\pi_e}$ based on a held-out set. The estimated value of π_e can then be computed using,

$$\hat{V}^{\pi_e}_{DR} = \hat{V}^{\pi_e} + \sum_{n=1}^{N} w^{h_n}(R^{h_n} - \hat{Q}^{\pi_e}(s^{h_n}_{\mathtt{t}}, a^{h_n}_{\mathtt{t}})) \tag{2.8.11}$$

This evaluation scheme works well if either the approximate model or the IS weights are reasonably accurate. Importantly, all the approaches above require some parameterisation of the patient's history h_n to compute the weights in Equation 2.8.9. We will discuss the specific details of how to apply these strategies in our evaluation in Chapters 3 and 4.

Additionally, one may directly compare the differences in expected reward under the behavioural policy and the evaluation policy. This difference in expected reward is in some senses, conceptually similar to the ACE of applying the proposed evaluation policy vs. the observed behavioural policy. Specifically,

$$\Delta := \mathbb{E}_{\pi_e}[R^{h_n}] - \mathbb{E}_{\pi_b}[R^{h_n}]. \tag{2.8.12}$$

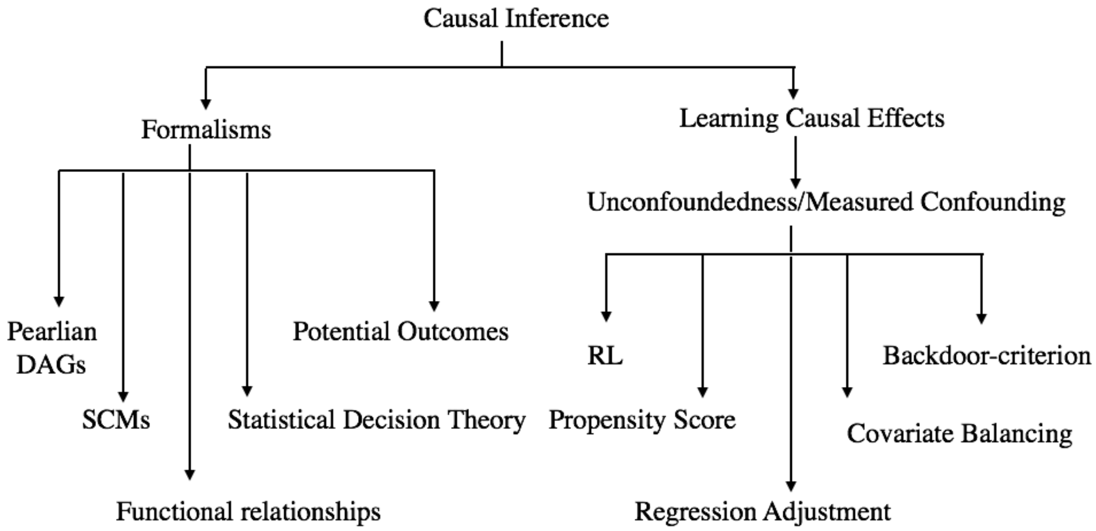


Figure 2.7: Summary of causal inference frameworks and methods for estimating causal effects under measured confounding.

2.9 Conclusion

This chapter presented an overview of the key concepts that form the basis of our contributions of this thesis. A schematic representation of these is shown in Figure 2.7. Specifically, in this chapter we presented an overview of various perspectives to causal inference and highlighted the assumptions and distinctions between each of these frameworks. We also showed how MDPs can be formulated as SCMs using Equations 2.8.7. The implication of this are that RL models may also be used to perform counterfactual reasoning. We will revisit this idea in the next two chapters. In what follows, we present each of the key contributions of this thesis and discuss the significance of each of these in light of medical decision-making and personalised medicine.

Chapter 3

Policy Mixture Models for Therapy Selection

3.1 Introduction and Background

In this chapter, we present a new model for reasoning about the effects of a series of interventions on a patient's outcomes. Our approach combines model-based RL and kernel-based reasoning in a mixture-of-experts model that partitions the space of patients on the basis of their individual characteristics, to infer suitable treatments. Recall that in Chapter 2, we discussed how kernel methods could be used in matching to adjust for the effects of confounding. The general idea is to compute treatment effects for an individual based on similar, comparable instances or nearest neighbours, where similarity is defined by a kernel function. In contrast, techniques such as model-based RL that allow us to build causal models, rely on the idea of a state space representation to capture confounding, and condition on this information to reason about treatment choices. In doing so, these methods allow us to actively simulate experience much like performing active interventions via experimentation, such that we can reliably infer patient outcomes. By combining the policies obtained using both approaches in a mixture-of-experts model, we demonstrate how we can exploit the knowledge from both methods to learn more effective treatment policies. The work in this chapter is based on a combination of Parbhoo et al. (2017) and Parbhoo et al. (2019). While the approach is general enough to be applied to a number of medical contexts, we specifically focus on HIV.

Overall, significant advances in therapeutics since 1996, have transformed HIV infection from a life-threatening illness to a treatable, chronic condition and have vastly improved a patient's chances of survival (Deeks et al., 2013). Despite the availability of comprehensive treatment recommendations, choosing appropriate interventions in practice remains a challenging task since much of the success in managing HIV with ART depends on the mutagenicity of the viral variants infecting an individual. Typically, a rapid turnover and an error-prone replication cycle mean that HIV is frequently capable of developing drug-resistant variants in response to drug pressure, which presents a major obstacle in establishing effective therapies for a patient, as many treatment options may be rendered ineffective (Bogojeska et al., 2012; Günthard et al., 2016). This makes manually searching for a feasible therapy particularly challenging, especially for patients with long treatment histories.

Since the advent of ART, several studies have proposed using computational techniques to address the challenges associated with HIV therapy selection (e.g Bickel et al. (2008), Altmann et al. (2007), Rosen-Zvi et al. (2008)). Among these, Altmann et al. (2007) utilise the evolutionary information of a viral population to construct a tree and predict a patient's immediate response on the basis of this tree. Similarly, Revell et al. (2010) predicts the probability of a short-term reduction of the viral load using random forests, while Prosperi et al. (2010) integrates HIV genotypic information (from the viral subtype and mutations with respect to a reference wild type) to predict phenotypic susceptibility to single drugs. Several methods also focus on examining the dependencies among mutations as a means of understanding treatment efficacy: in particular, Singh (2017) use linear combinations of mutations to build a resistance profile on the basis of which treatments may be selected more effectively, while Beerenwinkel et al. (2007) use graphical models called Conjunctive Bayesian Networks to formulate the accumulation of genetic mutations. Perhaps most closely related to our work is the method of Bogojeska et al. (2012), where the authors present a kernel-based regression approach for predicting whether a particular therapy choice will be successful using a patient's treatment history and variant information. Specifically, treatment success is characterised by the viral load dropping below 40 copies/mL after at least 21 days under the therapy. The premise here is that patients with similar treatment histories are likely to respond to treatment in a similar way. While some of these approaches aim to understand the relations among viral variants, the majority of these methods use this information to classify whether a particular choice of therapy results in a short-term reduction in the viral load overall. As a result, they fail to account for the sequential, causal nature of the therapy selection process — that a current choice of therapy may cause drug-resistant viral variants that are more difficult to control later. Model-based RL methods make this sequential process explicit: by building a causal model, a treatment policy is learned via a series of repeated exchanges between an agent (clinician) and an environment (patient); the agent learns how to take decisions that not only optimise a patient's immediate virological response, but also their long-term future outcomes. In the past, general RL methods have been applied for optimising treatments in several other medical contexts and simulation scenarios e.g Ernst et al. (2006); Parbhoo (2014); Pineau et al. (2009); Raghu et al. (2017); Zhao et al. (2009), but reasoning about futures from limited data has curbed the utility of these in practice.

Here, we present a unified approach to HIV therapy selection that simultaneously accounts for patient heterogeneity and confounding by combining kernel-based reasoning from Bogojeska et al. (2012) with model-based RL. Specifically, we demonstrate that both of these methods are complementary: kernel methods excel where there is significant overlap among patients, as they can model the idiosyncrasies in viral response specific to those patients. However, their prediction quality drops for patients that are not part of a tight cluster. In contrast, model-based RL first builds a causal model of a patient's expected response to reason about how well a series of interventions will perform. Generally, these models tend to find simpler patterns of response – a better alternative for patients outside clusters. By combining the policies learned using both approaches in a mixture-of-experts setting, we demonstrate that we can infer more effective treatment strategies overall.

In what follows, we present the details of the proposed methodology and show that our method outperforms both kernel methods and model-based RL alone for the task of HIV therapy selection. Throughout this chapter, we assume that all confounders are measured.

3.2 Model and Inference

In Chapter 2, we introduced MDPs as a mathematical framework for decision-making in RL. In this section, we consider an extended formulation of MDPs for partially observable settings known as Partially Observable MDPs (POMDPs) (Kaelbling et al., 1998) for the task of therapy selection. The primary difference here is that states are hidden. Like MDPs, POMDPs may also be seen as SCMs and can thus be used to perform counterfactual reasoning. Assume that we are given a collection $\mathcal{D} = \{h_{nT_n}\}_{n=1}^{N}$ of N patient histories of length T_n, where each history is composed of a sequence of interventions a, observations o and rewards corresponding to a patient's outcomes to treatments r, such that $h_{nT_n} = \{a_{n1}, o_{n1}, r_{n1}, \dots, a_{nT_n}, o_{nT_n}, r_{nT_n}\}$. The overall problem of HIV therapy selection may be viewed as identifying a treatment policy π_e that takes as input some parameterisation of the patient's history h_{nT_n} and outputs actions that will maximise that patient's expected long-term return $\mathbb{E}[\sum_{\mathtt{t}} \gamma^{\mathtt{t}} r_{\mathtt{t}}]$. The approach we propose here identifies such a policy in two different ways using kernel-based learning and model-based RL. We subsequently integrate the policies obtained from both methods in a mixture-of-experts network to learn an optimal policy that is tailored to each patient's particular scenario. A schematic overview of the approach is shown in Figure 3.1.

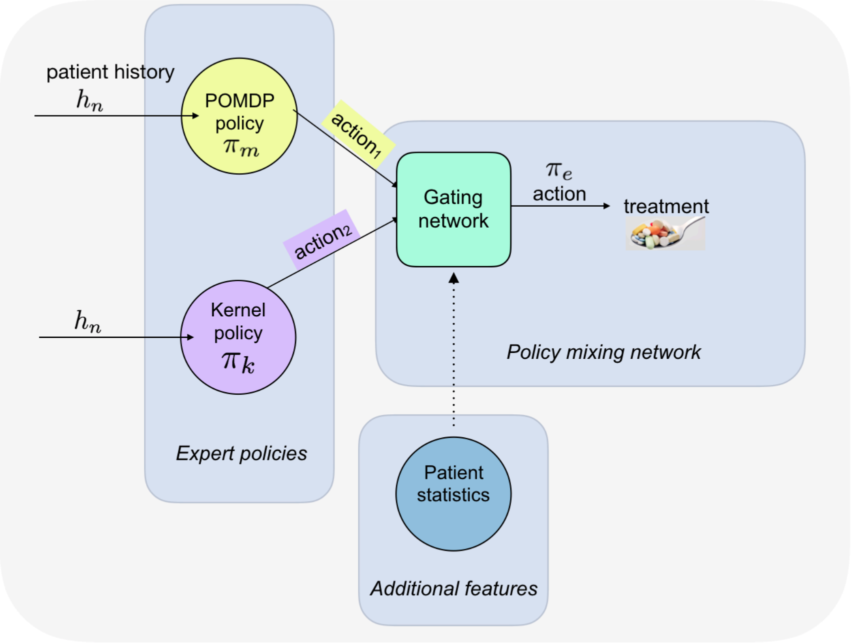


Figure 3.1: The mixture-of-experts framework for HIV therapy selection. We combine policies at test time using a mixture-of-experts gating network that weights POMDP and kernel policies for each patient on the basis of their history.

3.2.1 Learning a POMDP model

Formally a discrete state POMDP consists of the n-tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \Omega, \mathcal{R}, \gamma\}$. $\mathcal{S}, \mathcal{A}$ and $\mathcal{O}$ are sets of hidden states, actions and observations respectively, while $\mathcal{T}$ defines the distribution of next states s' from state s when taking an action a, and Ω is a distribution over observations o that occur from state s when taking action a. $\mathcal{R}$ specifies the immediate reward r in a state s when taking action a. Like MDPs, POMDPs can be seen as SCMs. Specifically, the hidden state may be viewed as a latent cause variable that, together with interventions, jointly causes rewards, outcomes and transitions (see Gershman (2017) for a detailed discussion about this). In the SCM formulation of POMDPs, one may view $\mathcal{T}, \Omega, \mathcal{R}$ as functions rather than probabilities (or probability distributions) as in the classical RL formulation, where $\mathcal{T}$ and $\mathcal{R}$ are defined as described in Section 2.8.1, while $\Omega : \mathcal{S} \times \mathcal{A} \to \mathcal{O}$. In this case, the following structural equations hold for each time step $\mathtt{t}$:

$$\begin{aligned} a_{\mathtt{t}} &= \pi(s_{\mathtt{t-1}}) + \epsilon_{\mathtt{t}}, \\ r_{\mathtt{t}} &= \mathcal{R}(s_{\mathtt{t}}, a_{\mathtt{t}}) + \epsilon_{\mathtt{t}'}, \\ s_{\mathtt{t}} &= \mathcal{T}(s_{\mathtt{t-1}}, a_{\mathtt{t}}) + \epsilon_{\mathtt{t}''}, \\ o_{\mathtt{t}} &= \Omega(s_{\mathtt{t}}, a_{\mathtt{t}}) + \epsilon_{\mathtt{t}'''}. \end{aligned} \tag{3.2.1}$$

In general, making decisions in a partially observable setting requires the entire history. Fortunately, there exists a succinct sufficient statistic for the history: the belief $b \equiv P(s|h)$, the distribution over states given the history. Given the belief $b_{\mathtt{t-1}}$, an action $a_{\mathtt{t}}$, and a new observation $o_{\mathtt{t}}$, the subsequent belief $b_{\mathtt{t}}$ can be computed via Bayes' rule:

$$b_{\mathtt{t}}(s) = \Omega(o_{\mathtt{t}}|s, a_{\mathtt{t}}) \sum_{s' \in \mathcal{S}} \frac{\mathcal{T}(s|s', a_{\mathtt{t}}) b_{\mathtt{t-1}}(s')}{P(o_{\mathtt{t}}|b_{\mathtt{t-1}}, a_{\mathtt{t}})}, \tag{3.2.2}$$

where $p(o_{\mathtt{t}}|b_{\mathtt{t-1}}, a_{\mathtt{t}}) = \sum_{s' \in \mathcal{S}} \Omega(o_{\mathtt{t}}|s', a_{\mathtt{t}}) \sum_{s \in \mathcal{S}} \mathcal{T}(s'|s, a_{\mathtt{t}}) b_{\mathtt{t-1}}(s)$. Assuming we are given a reward function $\mathcal{R}$ in this setting, the problem of solving a POMDP model may thus be reformulated as determining an optimal POMDP policy π_m via online planning (e.g forward search) using Equation 3.2.2, once the parameters $\mathcal{T}, \Omega$ are learned. Model-based RL methods typically interleave between these two phases. To learn the POMDP model, we take a Bayesian approach by using available histories to estimate the parameters $\mathcal{T}$ and Ω. The basic procedure may be summarised as follows.

1. Sample a set of states using Forward Filtering Backward Sampling (Carter & Kohn, 1994)

2. Sample transition parameters $\mathcal{T}$ from a Multinomial Dirichlet.

3. For a continuous observation setting, sample the emission parameters Ω from a Normal Inverse Wishart where for each state i, we have

$$\mu_i, \Sigma_i \sim \mathcal{N}(y|\mu_i, \Sigma_i)\mathcal{N}(y|0, \Sigma_i)IW(\Sigma_i)$$

4. Alternatively for a discrete observation setting, sample the emission parameters Ω from a Multinomial Dirichlet.

The procedure above gives us a set of parameters $\mathcal{T}, \Omega$, which together with our reward function $\mathcal{R}$ may be used to perform planning and update our beliefs about our states. Based on this, we may subsequently infer a POMDP policy. We describe this procedure next.

3.2.2 Forward Planning with a POMDP

Given a POMDP model m, we make use of stochastic forward search to learn a suitable π_m. The general idea of online POMDP planners (Ross et al., 2011, 2008) is to construct a forward looking tree rooted at the current belief b_t in order to compute a value function. At each stage, the tree branches on each action a the agent may take and the observation o the agent may observe. At each action node, the agent computes its expected immediate reward $\mathcal{R}(a) = \mathbb{E}_{s|m}[\mathcal{R}(\cdot|s,a)]$. The value of taking action a in belief state $b(s)$ is

$$Q(a,b) = \mathcal{R}(a,b) + \gamma \sum_{o} \Omega(o|b,a) \max_{a'} Q(a', b^{ao}), \tag{3.2.3}$$

where b^{ao} is the agent's belief after taking action a and observing o from belief state b, and $\mathcal{R}(a,b) = \sum_s b(s)\mathcal{R}(s,a)$, and the action-value $Q(a', b^{ao})$ is recursively calculated down the tree to some depth D. Because our observation space is large in this case, we approximate the sum above using samples from $\Omega(o|b,a)$. Based on the Q-values of the leaves, we can compute a POMDP policy π_m for each belief state b. We combine this information with the kernel policy from the next section to train our mixture-of-experts network.

3.2.3 Learning a Kernel Policy

Suppose we are given a set of pairs of patient histories and long-term return $\{h_{n\mathtt{t}}, R_n\}$. For each history $h_{n\mathtt{t}}$, we can predict its long-term return $\hat{R}$ via a non-parametric regression where our predictions are expressed by averaging over nearby histories $h'_{n\mathtt{t}}$ as follows,

$$\hat{R}' = \sum_{h'_{n\mathtt{t}}} k(h_{n\mathtt{t}}, h'_{n\mathtt{t}}) R_n, \forall h'_{n\mathtt{t}} \in \mathcal{H}. \tag{3.2.4}$$

Here, $k(h_{n\mathtt{t}}, h'_{n\mathtt{t}}) \geq 0$ is a weighting kernel function in Reproducing Kernel Hilbert Space (RKHS) satisfying $\sum_{h'_{n\mathtt{t}}} k(h_{n\mathtt{t}}, h'_{n\mathtt{t}}) = 1$, $\forall h_{n\mathtt{t}} \in \mathcal{H}$, and $\mathcal{H}$ represents the set of patient histories. Intuitively, this implies that one can assess the long-term value of taking an action a by examining the training data of histories where a has been applied and weighting over their long-term values; the kernel policy π_k is subsequently given by the distribution of actions a taken over those instances that maximise the predicted long-term return. Overall, π_k and π_m from the previous section are aggregated together an additional set of patient statistics in order to train a mixture-of-experts model. We describe this procedure next.

3.2.4 Combining Kernel and POMDP Policies in a Mixture of Experts Model

The mixture-of-experts (Jordan & Jacobs, 1994) is an ensemble method where multiple learners or experts partition the input space into regions. A gating network is then used to automatically assign these regions to an expert for prediction. In doing so, two experts with complementary properties may specialise in different regions of the input space, allowing us to take advantage of both. In general, a mixture-of-experts model may be viewed as an associational mixture model where each distribution in a mixture is instead replaced with a conditional distribution associated with a particular region of the input space. The nature of the mixture-of-experts model makes it particularly suited to the task of HIV therapy selection, where patients exhibit a large amount of heterogeneity and it is typically difficult for a single model to provide reasonable predictions across all patients. The kernel expert is able to draw inference from similar patient groups, and performs well when patients are part of a cluster. The POMDP expert uses experience to build a simplified model of a patient's responses over a set of discrete patient states. The mixture-of-experts combines kernel and POMDP policies to learn a context-specific policy.

We build a mixture-of-experts network that uses a set of patient statistics about their history, to map the kernel and POMDP policies from the previous sections to an expert policy π_e. This procedure is illustrated in Figure 3.1. Specifically, the network uses a vector x, consisting of the length of a patient's treatment history, lower quantiles of the distance between a patient and their neighbours, a patient's gender, age, resistance mutations and viral load, to learn the weights assigned to the POMDP and kernel policies, π_m and π_k, respectively. The mixture-of-experts assigns probabilities to each expert by learning the weights for each of the features in x. These probabilities are given by,

$$p_k = \frac{\exp(ux + u_0)}{\exp(ux + u_0) + 1} \tag{3.2.5}$$

$$p_m = 1 - p_k \tag{3.2.6}$$

Given the expert probabilities p_k and p_m respectively, the mixture of experts policy may then be defined as,

$$\pi_e = p_k \pi_k + p_m \pi_m. \tag{3.2.7}$$

Specifically, the mixture-of-experts network is trained using the Adam optimiser (Kingma & Ba, 2014) which depends on various hyperparameters such as the learning rate, batch size, maximum epochs, and early stopping criterion. We preset the learning rate to 0.01, the maximum number of epochs to 70, the early stopping criterion to 20 and the batch size to 75 for our experiments which we discuss next.

3.3 Experiments

In this section, we show how the mixture-of-experts model can be applied for learning a suitable treatment policy in the context of HIV. We subsequently apply off-policy evaluation using the techniques discussed in Section 2.8.3 to provide a quantitative assessment of the performance of the learned policies. In what follows, we first detail the cohorts that we use for training and testing the proposed approach, as well as how the data from these cohorts was pre-processed for experimentation. We subsequently discuss how to learn POMDP and kernel-based policies in this context by introducing a new reward criterion specifically for HIV, and specifying the kernel function we make use of. Finally, we perform an analysis of the scenarios where the mixture-of-experts has a higher preference for the POMDP model in comparison to the kernel, and discuss how this performance changes with modifications to the evaluation strategy and reward criterion.

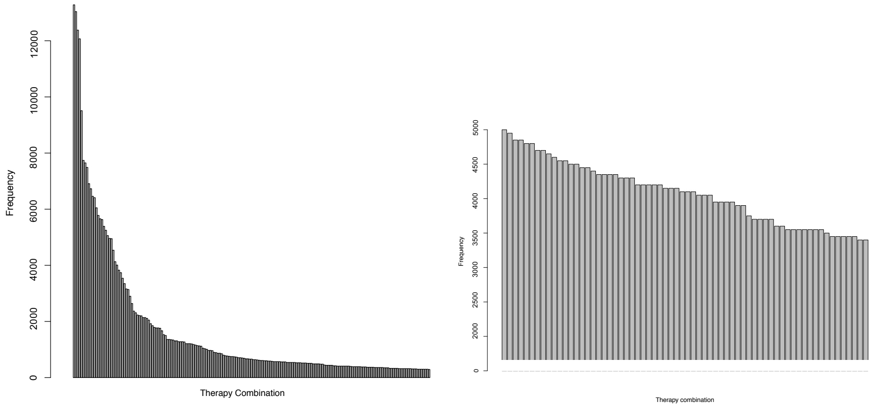
3.3.1 Data Cohorts

We assemble a set for building the mixture-of-experts predictor and its kernel and model-based components using the EuResist Integrated Database (EIDB) (Zazzi et al., 2012). Our data set consists of data from 1980 to 2016 of 32 960 HIV-infected patients, independent of their disease stage or degree of immunosuppression. Detailed information on patient demographics, mode of HIV acquisition, risk information, clinical events and treatment are collected. Our inclusion criteria for this study are patients whose complete viral genotype data is available at baseline, together with CD4^{+} and HIV viral load measurements collected and recorded at 6-monthly intervals. We performed roughly a 80%-10%-10% train-test-validation split of this data set to evaluate the performance of the mixture-of-experts approach presented in this chapter.

In addition to the held-out subset of data from the EIDB used for testing, we assembled an additional independent test set of 9 565 patients from the Swiss HIV Cohort Study (SHCS). The SHCS is a prospective study established in 1988, with ongoing enrolment of HIV-positive patients of ages 16 years and older from seven outpatient clinics and their affiliated hospitals or private practices in Switzerland. Detailed information about patient demographics, mode of HIV-infection, risk information, clinical events, treatments, as well as baseline viral genotype data, blood count and HIV viral load measurements are available at 6-monthly intervals. We consider only those patients from 1988 to 2016. Using two independent sets of patients as test sets enables us to draw comparisons between the learned treatment policies and assess how well the method generalises to unseen populations. As features, in addition to the viral genotype, treatment and response data, we include information concerning a patient's age, ethnicity, gender, risk group, co-infections and any prior treatments. Variables with excessive missingness from either data set were excluded from both sets for consistency.

3.3.2 Restricting the Space of Treatments

There are over 20 antiretrovirals available for treating HIV, resulting in a large space of therapy combinations to explore when optimising treatments. In particular, certain drugs that were previously used in the past have been phased out or reformulated as part of single-tablet regimes with other drugs. An obvious way to safe-guard against obsolete treatments would be to consider only those combinations of drugs consistent with current drug standards and encode these guidelines as hard constraints in our model. Unfortunately, this requires removing a significant portion of the data across both cohorts which impacts on learning. Instead, we restrict the space of treatments here based on calendar time. Specifically, we consider the 75 most frequently occurring drug combinations in a period of 10 calendar years and learn separate POMDP and kernel policies over each decade. Since both data cohorts cover patients from roughly 1980 to 2016, this results in four decades. In doing so, we can predict treatments for a patient with history h on the basis of current drug standards, as well as earlier standards. Figure 3.2 (a) shows the distribution of drug combinations across the training data when the set of treatments is not pruned, while Figure 3.2 (b) shows the distribution of the 75 most frequently occurring combinations of drugs between 1999 and 2000 in the EuResist set. Evidently, the latter appears significantly more balanced. In general, the pruning procedure produces similar distributions of therapies across the other decades too.



(a) (b)

Figure 3.2: (a) Distribution of drug combinations across training data. The distribution of treatments is considerably imbalanced. (b) Pruned distribution of most frequently occurring drug combinations from 1990 and 2000 across training data. Pruning the treatment space according to the periods in which drugs have been introduced and actively used produces a more balanced distribution of therapies.

3.3.3 A Long-Term HIV Success Criterion

Existing approaches to HIV therapy selection focus on a short term reduction in the viral load below detection limits (e.g Bogojeska et al. (2012)). Here, we propose using a new reward criterion following Ernst et al. (2006); Parbhoo (2014), that accounts for both a reduction in viral load as well as mutations and as well a patient's short term immune response in terms of CD4$^+$ cells. Specifically,

$$r_{\mathtt{t}} = \begin{cases} -0.7 \log V_{\mathtt{t}} + 0.6 \log C_{\mathtt{t}} - 0.2|M_{\mathtt{t}}|, & \text{if } V_{\mathtt{t}} \text{ is above detection limits} \\ 5 + 0.6 \log C_{\mathtt{t}} - 0.2|M_{\mathtt{t}}|, & \text{if } V_{\mathtt{t}} \text{ is below detection limits} \\ -10 & \text{if the patient died,} \end{cases}$$

where $V_{\mathtt{t}}$ is the viral load (in copies/mL), $C_{\mathtt{t}}$ is the CD4$^+$ count (in cells/mL), and $|M_{\mathtt{t}}|$ is the number of mutations at time $\mathtt{t}$ respectively. The reward function penalises instances where a patient's viral load increases and rewards instances where a patient's CD4$^+$ count increases (more weight is placed on the viral load, as it is an earlier indicator of whether a therapy is working). We also penalise on the basis of the number of mutations a patient has at a particular time, as these may ultimately contribute to resistance and therapy failure. There is also a bonus for if the viral load is below detectable limits as this is something we would like to sustain over time. Finally, a large penalty is provided if a patient died during the course of treatment. Summing the immediate rewards over a patient's history produces a long-term accumulated return. In our experiments, we demonstrate that optimising over long-term patient outcomes in terms of the accumulated return, produces different treatment policies to optimising for short-term gains. A discussion about the sensitivity of the mixture-of-experts model with respect to this function is provided in Appendix A.

3.3.4 Specifying a Suitable Kernel for Therapy Selection

A patient's prior treatment history is frequently considered a key factor in predicting the efficacy of subsequent HIV therapies (Prosperi et al., 2010; Revell et al., 2010). In this light, Bogojeska et al. (2012) train a history alignment model that measures the similarity between two patient histories or sequences and predicts treatment outcomes on the basis of this similarity. Here, a sequence refers to a sequence of therapies or drug combinations taken by a patient over time. Specifically, two therapies are deemed similar if they are comprised of similar drugs, are administered in a similar order, and result in similar genomic fingerprints in the viral population. The history alignment model first constructs a *resistance mutations kernel* to quantify the pairwise similarities between different therapy combinations. The kernel assumes that similarity between the different drug groups is additive, and thus can be assumed to act independently as each drug class has different modes of actions and therapeutic targets. Bogojeska et al. (2012) subsequently use the resistance mutations kernel to calculate a therapy sequence alignment kernel $k(h, h')$ by adapting the Needleman-Wunsch score frequently used for assessing the quality of an alignment of a protein or nucleic acid sequences (Needleman & Wunsch, 1970). The therapy histories h and h' of two patients are aligned and compared to assess their similarity. This produces a *history alignment kernel.* In doing so, the approach accounts not only for similarity across the genetic fingerprints of potential latent virus populations, but also similarity across therapy histories. Further details of this kernel function are provided in the Appendix A.

Bogojeska et al. (2012) use $k(h, h')$ to train a regression model for predicting therapy outcomes in terms of short-term virological success or failure. We call this the *Short-Term History Alignment (ST Kernel)* model. As described in Section 3.2.3, we can perform a similar procedure using the long-term return in terms of Equation 3.3.3. Based on this, we may deduce a kernel policy in terms of the distribution of actions a taken over those instances that maximise the predicted long-term return. We call this a *Long Term History Alignment (LT Kernel)* model. We subsequently combine the kernel policy obtained using the long-term success criterion with a POMDP policy via a gating function to learn mixture-of-experts policy.

3.3.5 Results and Discussion

In the following section, we make use of all three of the off-policy evaluation methods described in Section 2.8.3 in order to assess the performance of each of the policies in terms of patient outcomes. Importantly, we make the significant assumption that the belief state from the POMDP is a sufficient statistic for a patient's history and use these where necessary to perform off-policy evaluation. We compare the performance of the mixture-of-experts policy against a kernel expert that uses the long-term reward criterion introduced earlier (LT kernel), as well as a kernel policy that only optimises immediate outcomes (ST kernel), a POMDP policy and a random policy in which a completely random therapy choice is made. In each case, we evaluate the policies produced across two different test sets. The first test set consists of 3 000 held-out patients from the EIDB. The second test set contains 9 565 patients from the SHCS. Estimates of the discounted expected return over a period of 5 years (or forward search depth of $D = 10$) for each policy are provided for the EIDB in Table 3.1. A higher value indicates a better performing treatment policy. A similar set of results for the SHCS is shown in Table 3.2.

	DR	**IS**	**WIS**
Random	–2.31 ± 1.42	–3.48 ± 1.36	–2.80 ± 1.27
ST Kernel	2.17 ±1.4	2.18 ±1.20	2.16 ±1.71
LT Kernel	9.47 ± 1.70	5.72 ± 1.81	6.97 ± 1.29
POMDP	6.04 ± 2.18	4.15 ± 2.28	6.67 ± 1.74
Mixture-of-experts	**11.83 ± 1.26**	**12.50 ± 1.19**	**11.07 ± 1.21**

Table 3.1: Off-policy evaluation using importance sampling, weighted importance sampling and doubly robust methods for different therapy selection models across EIDB test set ($\gamma = 0.98$). Overall, the mixture-of-experts produces the largest immune response while reducing the viral load.

	DR	**IS**	**WIS**
Random	–6.33 ± 3.47	–5.57 ± 2.17	–6.18 ± 3.24
ST Kernel	1.64 ±1.86	2.03 ±1.81	2.17 ±1.74
LT Kernel	9.67 ± 1.49	7.38 ± 1.72	7.64 ± 1.92
POMDP	5.46 ± 2.05	6.72 ± 2.88	7.76 ± 2.10
Mixture-of-experts	**10.73 ± 1.02**	**13.59 ± 1.57**	**11.83 ± 1.31**

Table 3.2: Off-policy evaluation using importance sampling, weighted importance sampling and doubly robust methods for different therapy selection models across SHCS test set ($\gamma = 0.98$). The mixture-of-experts produces the largest immune response while reducing the viral load on a different cohort of patients.

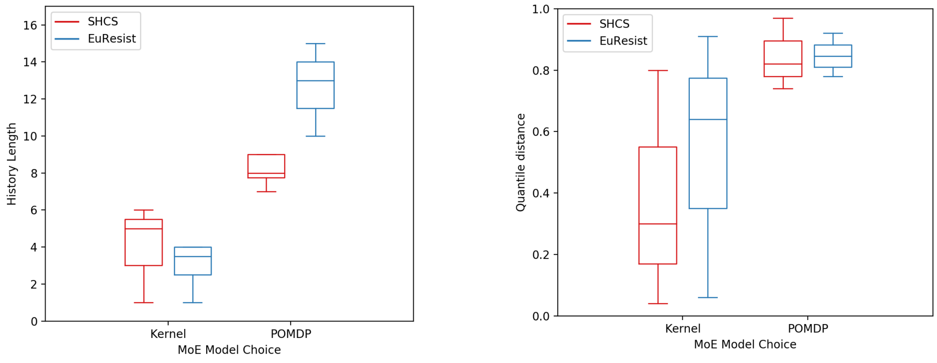
The results from Tables 3.1 and 3.2 show that optimising long-term outcomes produces different policies to optimising immediate outcomes (choosing treatments based on the short-term kernel policy results in lower long-term rewards than choosing treatments based on any of the methods that consider the long-term rewards). This suggests that treatments which may initially be feasible, may result in poor patient outcomes later on—a result consistent with the occurrence of resistance amongst HIV-infected individuals. Specifically, resistance against a particular drug may lead to cross-resistance against another, leading to long-term dependencies in therapy response.

We also compared the performance of these approaches when using the unpruned set of treatments. These results are provided in the Appendix A in Tables A.3 and A.4. In both cases, the mixture-of-experts approach still outperforms its kernel and model-based counterparts, but the performance gains are not as pronounced, particularly for the SHCS data set.

Interpreting the mixture-of-experts. Regardless of the choice of test data set, *the mixture-of-experts policy significantly outperforms the kernel and model-based policies alone* ($p < 0.05$). In both cases, the POMDP policy performs worse than the long-term kernel policy. This suggests that the models tend to make prediction mistakes in different regions of the input space. Combining the two approaches via a mixture-of-experts overcomes these issues. A post-hoc examination of the mixture-of-experts policy shows that the model has a preference for the POMDP policy approximately 28% of the time, while relying heavily on the kernel policy for the remaining 72%. These results are very similar across both cohorts.

Feature	W_k
History length	0.3721
Quantile distance	0.4619
CD4$^+$ count	-0.0579
Viral Load	0.1846
Age	0.0026
Male	-0.0718
PR90M	0.0671
RT215YF	-0.0830

Table 3.3: Feature weights for mixture-of-experts gating function. The history length and quantile distance have the largest weights.



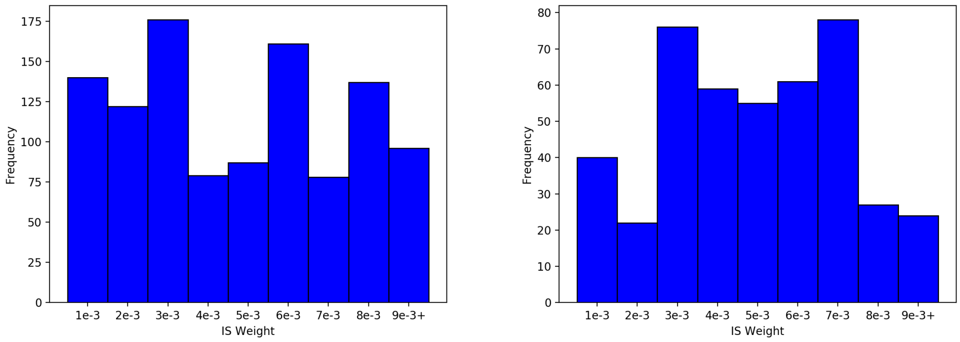
(a) (b)

Figure 3.3: Interpreting the features of the mixture-of-experts network that have the highest weights. The history length and quantile distances between patients have the highest weight. The mixture-of-experts prefers the kernel policy for patients with short histories that are closer and more similar to other patients as shown in (a). The mixture-of-experts prefers the POMDP policy when patients have longer histories that are distinct from other patients as in (b).

In order to interpret and understand the behaviour of the mixture-of-experts network for learning π_e, we examine the gating function's weight parameters W_k for those features with the highest weights. The features with the largest weights are shown in Table 3.3. In particular, the quantile distance and history length have higher weights than the other input features. We further analyse this result by examining when the mixture-of-experts has a strong preference towards the kernel policy in comparison to the POMDP policy. These results are shown in Figure 3.3. The mixture-of-experts favours the kernel policy when patients have similar histories across both data sets. The mixture-of-experts favours the POMDP policy for outlier patients with longer treatment histories across both cohorts. These differences are likely a result of the way in which each method uses a patient's history to represent confounding for inferring treatment effects: the

POMDP incorporates this knowledge implicitly through its beliefs and actions, each influenced by past observations, treatments and mutations. The kernel policy, on the other hand, suffices where history data is similar to what is seen or where there are significant overlaps, but does not extrapolate well for outlier cases. This result is largely consistent with our discussion about kernel matching in Section 2.7.2. In general, while the belief state representation of confounding in the POMDP may be approximate, this representation is preferable to mapping a patient to another dissimilar patient, and following their treatment policy as a result.

Assessing the quality of evaluation. The WIS and DR estimates of the value of a policy are highly dependent on having a significant number of patient histories in the evaluation set with non-zero importance ratios w^{h_n}. We verify that this is indeed the case for the mixture-of-experts policy where 83% and 76% of the weights have non-zero values with respect to the EIDB and SHCS data sets. This indicates that at test time, the vast majority of the data may be used to evaluate the policies learned – an important factor for building trust in our results. While most of these weights are still small (in the range of $[10^{-3}, 10^{-2}]$), there are several samples with weights that are in the order of 10^{-1} that are likely to have a much larger effect on the overall evaluation of the policies learned. These results are shown in Figures 3.4a and 3.4b respectively.



(a) (b)

Figure 3.4: Distributions of frequencies of non-zero IS weights for (a) EIDB and (b) SHCS data sets respectively. Overall, our treatments are fairly consistent with those in the data sets since the distributions are relatively balanced.

Clinical assessment of the learned policies. We also assess the learned policies from the kernel, POMDP and mixture-of-experts approaches for clinical validity. Here, the learned policies of ART regimens were compared against existing WHO and IAS-USA clinical guidelines (in terms of the types of drugs administered in different scenarios, as well as the frequency of drug switching (Carpenter et al., 2000, 1996, 1998; Günthard et al., 2014, 2016; Hammer et al., 2008, 2006; Thompson et al., 2010, 2012; Yeni et al., 2002, 2004) as well as the recommendations of clinicians. Specifically, we applied the guidelines available for a particular year and decade to the respective learned policies and checked their consistency. Overall, the learned polices are consistent with clinical guidelines 87% of the time. We further classify whether the learned ART regimen follows the a) recommended regimen, b) alternative regimen, or c) a drug combination in violation of the recommendations of clinicians. These results for both the EIDB and SHCS are shown in Table 3.4. We examine the baseline characteristics of those patients with policies in violation of recommendations. Specifically, several patients with lower CD4$^+$ counts (below 220 cells/mL) at the start of ART were more likely to receive therapy combinations that violated recommendations. These are less frequently occurring patients with AIDS-defining symptoms early on in their treatment history that typically require more nuanced treatment strategies. In these cases, the mixture-of-experts suggests policies based on the kernel, with fewer drugs (typical for entry-level patients) since there is a limited history available. Policies for patients initiating ART between the decade 1980 and 1990 were also more likely to violate recommendations. This is however, a result of having less training data available for this period.

Period	**Recommended (%)**	**Alternative (%)**	**Violation (%)**
1980-1990	56	17	27
1990-2000	73	11	16
2000-2010	80	19	1
2010 –	84	11	5
Average violation (%)	–	–	12.25

Table 3.4: Percentages of mixture-of-expert policies in violation with clinical recommendations.

In general, we observe that first-line therapies from the mixture-of-experts consist of mostly NNRTIs and NRTIs. This is consistent with clinical guidelines that typically recommend combining 1 NNRTI with 2 or more NRTIs for first-line therapy (Günthard et al., 2014). For second-line therapies, the mixture-of-experts frequently prescribes PI-boosted NRTI drug combinations which is also consistent with clinical guidelines.

3.4 Conclusion

In this chapter, we developed a new method for reasoning about the effects of a sequence of interventions to learn a suitable treatment policy. Our contribution highlights that parametric and non-parametric approaches have complementary strengths for reasoning about the effects of interventions: the non-parametric kernel approach can accurately estimate treatment effects where there is considerable overlap between patient instances or where data are abundant; in contrast, learning a parametric causal model using RL can generalise better about treatment effects in situations that are not frequently observed. We attribute this difference directly to the way in which model-based RL and the kernel approach use a patient's history and represent confounding: the POMDP incorporates this knowledge implicitly in its beliefs, each influenced by past observations, treatments and mutations, which when conditioned on can be used to learn treatment effects, while the kernel approach tries to match patients according to a similarity measure for the same purpose. Our mixture-of-experts approach combines parametric and nonparametric approaches to personalise treatment policies according to a patient's particular characteristics and hence outperforms both approaches individually.

Most importantly, while we have combined non-parametric and parametric models for causal inference and learning treatment policies, the overall mixture-of-experts approach could also be more broadly applicable as a procedure for off-policy evaluation in general. Here, the kernel weights may be viewed as a variant of PSM while the parametric model (in this case a POMDP) could serve as an estimate of the value function much like in doubly robust off-policy evaluation in Equation 2.8.11. This is a particularly interesting direction of research since it allows us to reformulate the problem of off-policy evaluation in terms of planning. A direct application of this would be to use the mixture-of-experts approach as an off-policy evaluation strategy rather than a treatment planning strategy to rank existing clinical policies in different scenarios.

Chapter 4

Dynamic Mixture Models for Counterfactual Reasoning

4.1 Introduction

In the previous chapter, we combined the policies obtained from using parametric causal models and non-parametric kernel-based reasoning in a mixture-of-experts setting to reason about the effects of a series of interventions on a patient and personalise treatment policies for individuals with HIV. In this chapter, we instead try to combine both of these approaches to build a new causal model for simulation. Simulation-based approaches to disease progression are generally desirable as they allow us to make counterfactual predictions about the effects of an untried series of treatment choices. However, in healthcare and medicine, building accurate simulators (e.g using causal models) for disease progression is challenging, hence limiting the practical utility of these approaches for real world treatment planning. Here, we show that our approach learns state-of-the-art treatment policies and can make accurate forward predictions about the effects of treatments on unseen patients. The majority of the work presented in this chapter is based on Parbhoo et al. (2018a). As before, we assume all confounders may be measured. Since much of the material in this chapter builds on work in the previous chapter, there may be some overlap but the overall model we present in this chapter is different and uses a different algorithm hence warranting a separate discussion.

Despite progress in machine learning methods for clinical decision support (e.g. Che et al. (2015); Choi et al. (2016); Miotto et al. (2016)), machine learning algorithms usually operate as uninterpretable black-boxes which clinicians are often hesitant to trust and adopt as tools. Given this context, simulation-based approaches to managing disease progression are appealing because they allow us to make counterfactual predictions about the possible future outcomes associated with different treatment options. Especially in high-stakes decisions, simulatability can help guide and audit recommendations. For example, a clinician who sees that the current set of HIV treatments will lead to future drug resistance, may choose a different set of therapies. Alternatively, an intensivist may see a physiologically implausible blood-pressure trajectory accompanying a treatment recommendation and correctly decide to ignore the recommendation. In this way, simulations provide a complementary context than a set of guidelines or recommendations.

At its core, building a simulator requires building a model. In disease progression modelling, we commonly posit that a patient has some underlying (and unobserved) disease state s that evolves according to the choice of treatments or actions a they take, governed by some transition function $\mathcal{T}(s'|s,a)$. We assume that we cannot observe the true state of the patient, and can only measure partial observations o, governed by some probability function $\Omega(o|s,a)$. For example, in an oncology setting, the true disease state s might be patient's cancer stage, while the observations o might be measured biomarkers and symptoms such as fatigue or weight loss. Given the model, we may subsequently use it to forward simulate potential histories and identify the most optimal treatments.

Unfortunately, disease progression is complex, and building models accurate enough for making decisions is challenging. Thus in many treatment recommendation settings, kernel-based regressors are much more common (e.g. Bogojeska et al. (2012), Rabinowitz et al. (2005), Seibert et al. (2007)). These approaches work by identifying similar patients and recommending the (usually one-step ahead) action that worked best for those similar patients. Kernel-based regressors have also been built into models: Fukumizu et al. (2013); Nishiyama et al. (2012) and Boots et al. (2013) all build dynamical system models that predict the patient's next physiological state based on the next-states of the patient's nearest neighbours. Using this kind of non-parametric predictor, rather than being confined to some parametric model, greatly improves model accuracy, especially if the underlying dynamics are complex and the data are dense.

However, kernel-based approaches to building models still have an important failure mode: because they work by matching patients with similar conditions, they perform poorly for patients with uncommon conditions. This limitation is an important concern for healthcare applications of kernel methods, as there often exists a large tail of distinct cases.

To address this challenge, we propose *kernelised dynamical mixing* (KDM), a hybrid approach that combines parametric (standard model-based) and non-parametric (kernel-based) predictors into one dynamical model of disease progression. Conceptually, when trying to predict how a specific patient's disease will evolve given a specific intervention, we build a gating network that will select whether it is more accurate to use a kernel-based prediction, which can model more complex functions but extrapolates poorly, or a model-based prediction, which is simpler but therefore extrapolates more smoothly. We demonstrate that our approach allows us to make both better forward predictions of disease progression and better treatment recommendations than either alone. Specifically,

- We introduce a hybrid strategy called kernelised dynamic mixing (KDM) that permits dynamically combining parametric (model-based) and non-parametric (kernel-based) counterfactual predictions of events within a forward planning setting.

- On two real clinical tasks, managing HIV and managing sepsis, our KDM-based approach produces more accurate predictions of future disease states compared to either parametric or non-parametric models alone.

- On those tasks, we show our KDM-based approach not only makes better treatment recommendations than either parametric or non-parametric models alone, but also makes better treatment recommendations than other non-model-based approaches (Bogojeska et al., 2012; Rabinowitz et al., 2005; Seibert et al., 2007).

4.2 Related Work

Kernel-based methods have a long history in reinforcement learning. Ormoneit & Sen (2002) assess the value of a particular state by averaging over histories passing near it. Other works, notably (Fukumizu et al., 2013; Grünewälder et al., 2011; Nishiyama et al., 2012; Song et al., 2016), use kernels to explicitly build models. For example, Fukumizu et al. (2013); Nishiyama et al. (2012) take a non-parametric view of learning policies by representing distributions over states, actions, and observations as embeddings in Hilbert spaces, and defining policies and value functions over these embeddings. Song et al. (2016) establish a principled connection between Bayesian inference and posterior distribution embeddings via the kernel Bayes' rule. Specifically, the authors express kernel Bayesian inference as a vector-valued regression problem and impose additional regularisation terms to control the resulting posterior embeddings, thus incorporating side information or domain knowledge into a problem. However, all of these approaches make predictions only from the data; while the choice of feature space may provide some regularisation effect, these approaches cannot be expected to generalise far from the observed histories.

Also related to our work, are methods that combine knowledge from different sources. Talvitie (2014, 2017); Weber et al. (2017) use rollouts with variants of experience replay to prevent sample degradation; they augment the training data used to learn a model with samples from a hallucinated context, and replay this experience to correct the model when it produces errors. Marco et al. (2017) trade off knowledge from simulations and physical experiments by explicitly representing the costs of different sources of information in a Gaussian process model, and use an entropy-based search to minimise quality of information costs while optimising performance. Chebotar et al. (2017) integrate model-based policy optimisation with model-free updates to improve a policy. While similar in spirit, this method is not designed to produce accurate future trajectories; it only aims to identify the optimal policy.

Other approaches try to capture model uncertainty more effectively. For example, Deisenroth & Rasmussen (2011); Gal et al. (2016) use probabilistic transition models such as Gaussian processes to incorporate uncertainty in the transition distribution into planning. These approaches are best suited for continuous, low-dimensional action spaces—not the norm in healthcare applications—and neither combines models with data in forward planning as we propose here.

Finally, other works combine models and data at the *policy level*, rather than for forward simulation. Specifically, in the previous chapter we introduced a mixture-of-experts model that combined policies from a simple kernel regression with policies derived from a causal model learned on the same data. The major downfall of this approach is that it cannot be used to *simulate* what might happen if the combined policy is followed. In this chapter, we instead propose an approach for combining kernel and model-based approaches on a *model* level.

4.3 Preliminaries and notation

As in the previous chapter, we assume that all confounding is measured and that $\mathcal{D} = \{h_{nT_n}\}_{n=1}^{N}$ is a collection of N patient histories of length T_n where each history is comprised of a sequence of treatments (actions) a, observations o, and outcomes (rewards) r, $h_{nT_n} = \{a_{n1}, o_{n1}, r_{n1}, \ldots, a_{nT_n}, o_{nT_n}, r_{nT_n}\}$. In general, the treatment that optimises a patient's immediate outcomes do not necessarily guarantee a patient's health in the long term. Our goal is to, for any patient history h, identify a policy $a = \pi(h)$ or sequence of treatments that optimises a patient's expected long-term outcomes $R := \mathbb{E}[\sum_{\mathtt{t}=0}^{T} \gamma^{\mathtt{t}} r_{\mathtt{t}}]$, where γ is a discount factor that trades between the importance of current and future rewards.

In Chapter 3, we discussed two ways of deriving such a policy: in the first, we learn a parametric dynamical system model of disease progression such as a POMDP, and use this in with online planning to infer a treatment policy; the second approach uses non-parametric regression to directly learn a policy without explicitly learning a model first. A third alternative is to model dynamical systems non-parametrically for instance, in a kernel-based setting. Notable works that take this approach include Nishiyama et al. (2012), Song et al. (2016) and Fukumizu et al. (2013). These approaches construct models specifically by representing distributions $\mathcal{T}(s'|s,a)$, $\Omega(o|s,a)$ and beliefs b as embeddings in Reproducing Kernel Hilbert Space (RKHS), and performing belief updates in accordance to Kernel Bayes' rule (Fukumizu et al., 2011). Approaches based on Kernel Bayes' rule can however be difficult to use in practice, as they require explicit knowledge about the hidden state in order to learn the embeddings of the distributions from training samples.

As an alternative to the aforementioned approaches, kernel-based learning may be used to directly sample subsequent observations $o_{\mathtt{t}+1}$. In this case, $o_{\mathtt{t}+1}$ may be drawn by considering the observations of the nearest neighbours and weighting these according to kernel function $k(h_{\mathtt{t}}, \cdot)$. In doing so, it is possible to deduce a kernel-based probability estimate of $\Omega(o|h) \propto \sum_{h'_{\mathtt{t}}} k(h_{\mathtt{t}}, h'_{\mathtt{t}})\delta(o = o_{\mathtt{t}+1}|h'_{\mathtt{t}})$ from which $o_{\mathtt{t}+1}$ may be sampled. Since the forward search in Equation. 3.2.3 only requires simulations of the next observation, these observations may be incorporated directly into model-based planning. We build on this idea in this paper.

4.4 Model and Inference

Both the parametric POMDP-based modelling approach and the non-parametric kernel-based modelling approach have their advantages: the simpler discrete POMDP tends to extrapolate better, whereas the kernel-based approach tends to be more accurate in regions of dense data. In this section, we present a modelling approach that dynamically mixes between these two approaches to build a simulator that is more accurate than either alone; given this simulator, we can then identify treatments using the online planning. Importantly, because predictions are combined in an *model-based setting*, all the advantages associated with model-based approaches apply here. Through forward simulation, we can assess a treatment policy holistically in terms of the particular observations that may result from a particular choice of drug, and perform counterfactual reasoning about the subsequent series of events that may follow, or alternative hypothetical scenarios that may arise as a result of changes in the context. An overview of our model-based approach is shown in Figure. 4.1.

Main Algorithm Both the discrete POMDP and the kernel-based model can be used to sample future observations given a history. Our approach combines these predictions to make this simulation more accurate. Specifically, we consider models such that

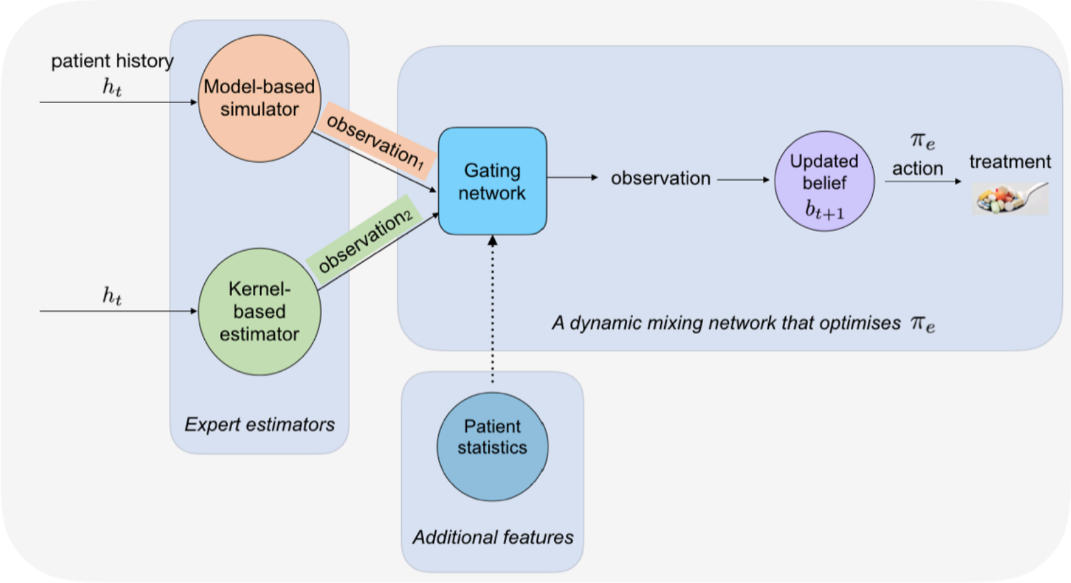


Figure 4.1: In our model-mixing approach, we create a simulator that chooses between parametric (discrete POMDP) and non-parametric (kernel) approaches for performing the forward simulation and use this simulator for planning. Specifically we now incorporate knowledge from the kernel directly into estimation of belief states based on which we can infer suitable treatments. The combined belief state representation here may be viewed as a means of representing confounding for causal inference.

the probability of an observation given a history $\Omega(o|h)$ is a linear combination of the probabilities under the POMDP model $\Omega_m(o|h)$ and the kernel-based approach $\Omega_k(o|h)$:

$$\Omega(o|h) = \theta(h)\Omega_m(o|h) + (1-\theta(h))\Omega_k(o|h) \tag{4.4.1}$$

where $\theta(h) \in [0,1]$ is some mixing parameter that trades between the two estimates. (We do not consider learning transition and observation models directly because, as noted in Nishiyama et al. (2012), these would require access to the hidden state s.) We note that the mixing in Equation 4.4.1 is complementary to kernelised reinforcement learning approaches such as kernelised POMDPs and PSRs (Nishiyama et al., 2012; Song et al., 2016). Both of these approaches regularise the kernel-based predictions through a bottleneck of the belief over states or core test predictions. In contrast, we include the parametric POMDP model over future observations, Ω_m, as an equal player in the prediction task, as if it were another special kind of patient history with kernel weight $\theta(h)$.

Once we have the function $\Omega(o|h)$, we can extend a history h given an action a by sampling from $\Omega(o|h)$. We can continue this forward simulation process for as long as we want; at each stage, we shall have a new history h' to compare to the batch of our histories in the kernel-based model and a new belief b' to be the sufficient statistic for our POMDP-based model. The final step to use this new simulator to optimise for new policies is to define the reward function on the basis of history h'. In our work, we use the POMDP alone to determine the immediate reward, although in principle the kernel could also be used. Our approach to using the POMDP to determine rewards is analogous to the approach from in (Nishiyama et al., 2012). Given the rewards we can apply forward search to find an optimal policy via Section 3.2.2 (see description in Algorithm 1).

Algorithm 1 Kernelised Dynamic Mixing Planner

```
Require:
    $\Theta(\cdot, W)$: MLP prediction function, with parameters $W$
    $B = \{b_{\mathtt{t}}\}_{n=1}^{N}$: belief states for each patient at time $t$
    $H = \{h_{\mathtt{t}}\}_{n=1}^{N}$: histories of each patient at time $\mathtt{t}$
    $k(\cdot, \cdot)$, $\Omega_k$: kernel parameters
    $\Omega_m, \mathcal{T}, \mathcal{R}$: POMDP parameters
function KDM($\theta$)
    while search depth has not been reached do
        Branch on an action $a_{\mathtt{t}}$
        Predict $\theta = \Theta(\cdot, W)$ based on $\mathcal{T}$, $k(\cdot, \cdot)$, and history length
        Set $\Omega = \theta(h_{\mathtt{t}})\Omega_m + (1 - \theta(h_{\mathtt{t}}))\Omega_k$
        Sample new observation $o_{\mathtt{t}}$ from $\Omega$
        Use $o_{\mathtt{t}}$, $h_{\mathtt{t}}$ and $a_{\mathtt{t}}$ to predict $R$
        Update belief $b_{\mathtt{t}}$ according to $o_{\mathtt{t}}$ and $a_{\mathtt{t}}$ using Equation(3.2.2)
        Add $o_{\mathtt{t}}$, $a_{\mathtt{t}}$ and $r_{\mathtt{t}}$ to existing history $h_{\mathtt{t}}$
    Backpropagate values up through the search tree to get $a^*_{\mathtt{t}}$
    return Updated $b_t$ and optimal action $a^*_{\mathtt{t}}$
```

4.4.1 Learning the mixing proportion $\theta(h)$

The key question, of course, is how to define the mixing function $\theta(h)$ to make our probability of observation estimate $\Omega(o|h)$ in Equation. 4.4.1 as accurately as possible for new histories. To do so, we note that while at test time the next observation $o_{\mathtt{t+1}}$ is not observed, our training set will contain many histories that can be cut into some past history and some next observation. That is, we have access to $o_{\mathtt{t+1}}$. Thus we can consider

$$\max_{\theta} \frac{1}{N} \sum_{n}^{N} \frac{1}{T_n} \sum_{\mathtt{t}}^{T_n} \log(\theta_{\mathtt{nt+1}} \Omega_m(o_{\mathtt{t+1}}|h_{\mathtt{nt}}) + (1 - \theta_{\mathtt{nt+1}})\Omega_k(o_{\mathtt{t+1}}|h_{\mathtt{nt}})) \tag{4.4.2}$$

In the formulation above where our goal is to predict the true next observation correctly, we note that either the POMDP or the kernel must necessarily be more accurate; thus, the optimal choice of $\theta_{\mathtt{nt}}$ at any time will be to select that more accurate model. During training, rather than fit to a binary target, we consider the softmax version

$$\theta(h_{\mathtt{nt}}) := \frac{\exp(\Omega_m(o_{\mathtt{t+1}}|h_{\mathtt{nt}}))}{\exp(\Omega_m(o_{\mathtt{t+1}}|h_{\mathtt{nt}})) + \exp(\Omega_k(o_{\mathtt{t+1}}|h_{\mathtt{nt}}))}. \tag{4.4.3}$$

The softmax target is akin to having a classifier predict which method makes most sense to use at each point in time. Specifically, it provides a probabilistic interpretation of which method is more likely to produce the observed future values, and hence determines which method should be given a higher weight for that time step.

Finally, we note that while one could train the weighting term θ to simply be a function of the history h, that is, some $\theta(h)$, the *relationship* between the history of interest h and the other histories in the training set is very important—as we mentioned before, we expect the kernel-based approach to be more accurate in regions where the data are dense and the POMDP to be more accurate otherwise. Thus, we include additional inputs to the predictor θ: patient statistics in terms of the history length of the current history h, along with the 5-quantiles of the function $k(h, \cdot)$ with respect to the training set. We call this collection of statistics ς, so our predictor is now $\theta(\varsigma)$.

Given the batch of histories, we can now create a collection $\{\varsigma_{\mathtt{nt}}, \theta_{\mathtt{nt}}\}$, where $\varsigma_{\mathtt{nt}}$ are the properties of the history and its relationship to the data and $\theta_{\mathtt{nt}}$ is the softmax target (equation 4.4.3). We train a multilayer perceptron (MLP) Θ as a mixing network to predict $\theta_{\mathtt{nt}}$ given parameters ς. Let vector W denote the parameters of the MLP. Then we write the training objective as

$$\min_{W} \sum_{\mathtt{n,t}} (\theta_{\mathtt{nt}} - \Theta(\varsigma_{\mathtt{nt}}, W))^2 + \lambda ||W||_2^2, \tag{4.4.4}$$

This loss is differentiable, and thus we can optimise it with gradient descent.

4.5 Experiment Setup: Evaluation Measures and Baselines

Our experiments focus on two related goals: (1) to characterise the performance of KDM in comparison in existing baselines, and (2) to assess the quality (in terms of forward predictions) and interpretability of approach in comparison to existing methods. Below we describe our metrics as well as our baselines.

4.5.1 Evaluation: Forward Simulation Quality

The KDM procedure described in the previous section provides a principled means of dynamically integrating kernel-based predictions into model-based RL to not only learn suitable treatment policies, but also provide counterfactual predictions. It is relatively straightforward to evaluate the quality of the predictions on retrospective data—at any time point, we have our distribution over possible next-observations, and we can compute the log-loss with respect to that distribution given what observation actually occurred. Additionally, we provide illustrations of the deviation between our counterfactual predictions and the ground truth in terms of the observations produced.

4.5.2 Evaluation: Policy Quality

While evaluating the quality of the forward simulation (above) was relatively straightforward, evaluating policy quality is much more difficult. We apply a collection of importance-sampling based estimators from Section 2.8.3 to evaluate our policies. (We report several, because each have different bias-variance trade-offs.) Conceptually, all of these methods try to determine a subset of the data over which the behavioural policy, π_b, coincides with the evaluation policy π_e.

4.5.3 Baselines

For each of our experiments, we compare the performance of a policy obtained from KDM to several baselines. Our first baseline is a policy based on a non-parametric (kernel-based) model computed by estimating the long-term reward from the samples falling in an ϵ radius of a particular patient at a certain time point. The kernel policy successively applies the action from the nearby samples associated with the largest expected long-term reward. Note that despite the similarities KDM shares with the Hilbert Space Embedding of the POMDP (kPOMDP) (Nishiyama et al., 2012), we cannot directly compare them since the kPOMDP requires knowledge of the true state representation during training—a severe limitation of the approach that makes it largely infeasible in practice. Here, the non-parametric model is used to approximate the kPOMDP. We also compare the KDM policy against a policy computed using a POMDP model alone. The third baseline is a mixture-of-experts model from Chapter 3, where we combine both parametric and non-parametric policy estimates using a gating network and choose actions accordingly. Across all tasks, we make the simplifying assumption that the belief state is a sufficient statistic for the history, and thus the policy is a function of the belief $\pi(b)$.

4.5.4 Training Parameters

To optimize the loss in Equation 4.4.4 we use L2 regularisation with strength $\lambda > 0$ and perform cross-validation against the true values of θ. We use $J = 500$ labeled pairs for training the mixing network on a toy example and $J = 4000$ for real world datasets. Optimisation of the mixing network's objective is done via gradient descent. We use Autograd (Maclaurin et al., 2015) to compute gradients of the loss in Equation 4.4.4 with respect to ξ, then use Adam (Kingma & Ba, 2014) to compute descent directions with step sizes set to 0.01 for the toy experiment and 0.001 for the medical applications. Across all three tasks a discount factor of $\gamma = 0.9$ is used, which puts weight on not only immediate rewards, but also long-term future rewards. In doing so, we can optimise not only a patient's immediate, but also their long-term health outcomes. (We do not use a very large γ as the domain does not require a particularly deep look-ahead to solve.) Further details of the training parameters are discussed in the next section.

4.6 Results

Below we show results on three domains. The first is a synthetic domain that highlights the how mixing parametric and non-parametric approaches when building a model can be beneficial. Next, we present two medical applications for administering treatments for patients with HIV and sepsis. In both cases, we present a quantitative evaluation of the policy and the forward simulation (note that for the forward simulation, we can only compare the model-based approaches; the mixture-of-experts cannot produce counterfactual predictions). Our KDM approach produces better policies and is able to simulate counterfactual scenarios more accurately than the baselines.

4.6.1 Demonstration on a Synthetic Domain

Consider a system that evolves deterministically through 4 states: S_1, S_2 or S_3, and finally absorbs in S_4. Each agent has a variant that belongs to one of two types: A and B. Agents with variants of type A deterministically go through state S_2, and agents with variants of type B deterministically go through S_3. At each stage, there are three actions available: 0, 1 or 2. At each time step, the agent observes its variant (which is one of the two types), as well as its reward, which is given by:

$$S_1 \begin{cases} r(a_0) = -10 \\ r(a_1) = 5 \\ r(a_2) = 5 \end{cases} S_2 \begin{cases} r(a_0) = 0 \\ r(a_1) = 5 \\ r(a_2) = -10 \end{cases} S_3 \begin{cases} r(a_0) = 0 \\ r(a_1) = -10 \\ r(a_2) = 5 \end{cases} S_4 \left\{ r = 0. \right.$$

The optimal policy for all agents is to initially take either action 1 or 2. Next, agents with variants of type A transition to S_2 where the optimal action is action 1; agents with variants of type B transition to S_3 where the optimal action is action 2. Action 0 is safe in states S_2 or S_3. By construction, a four-state POMDP cannot learn the optimal policy for this model since the dynamics depend on the hidden type of the agent's variant. Without the variant information, from the POMDP's perspective, it is equally likely to transition from S_1 or S_2 starting from S_0; not knowing where it will end up, it will initially suggest the safe policy of selection action 0 at the second time-step. For the kernelised planning approach, we use a kernel that matches based on the length of the agent's history, action choices, and an observation dependent on the hidden variant. Such a choice will lead to optimal policies for agents with common variants. However, agents with rare variants will match to some arbitrary other agent, and we can expect the performance of the kernelised planner for those agents to be poor. In such cases, falling back on the POMDP will produce the optimal policy. An illustration of the toy example is shown in Figure 4.2. The numbers in brackets indicate the action taken from a particular state, followed by the associated reward. We compared the performance of KDM against the baselines described earlier in this section, using a forward search depth of 4. Our mixing network for KDM consists of 15 input units and a hidden layer of 25 units. We trained the models using a data set of $N = 250$ sequences, each with $T_n = 4$ time steps. A separate test set of the same size was used for evaluating performance. Table 4.1 compares the performance of KDM against the aforementioned baselines. The

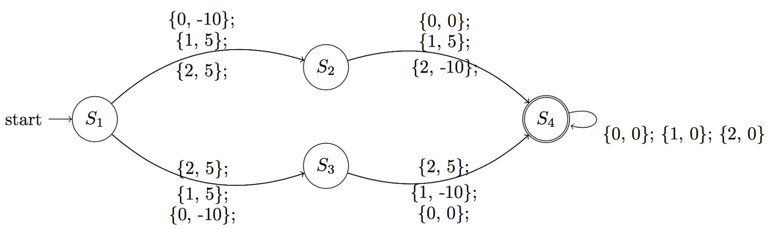


Figure 4.2: Illustration of dynamics for the toy example. The optimal sequence of actions for a type A variant is to initially take action 1 or 2, followed by action 1. For type B variants, the optimal sequence of actions is to first take actions 1 or 2, followed by action 2.

toy example illustrates that dynamically mixing kernel and model-based methods during simulation outperforms using either approach on its own. The quantitative differences between KDM and MoE policies suggest that combining parametric and non-parametric predictions on a model level results in different policies than combining these approaches on a policy level. Specifically, on a test set of 250 sequences, KDM learns the optimal policy 92% of the time, while in comparison the MoE approach learns the optimal policy 87% of the time.

	DR	**WIS**	**IS**
Random	–5.84 ± 2.61	–7.79 ± 3.71	–8.46 ± 3.24
Kernel	4.39 ± 1.74	4.86 ± 2.85	4.14 ± 2.72
POMDP	3.09 ± 1.16	3.62 ± 1.71	3.84 ± 2.42
Mixture-of-Experts	5.62 ± 1.02	5.81 ± 2.37	5.42 ± 2.74
KDM	**6.08 ± 1.14**	**6.19 ± 1.03**	**6.32 ±1.46**

Table 4.1: Performance comparison of KDM vs. baselines across 250 test sequences for the toy example. A higher value corresponds to a higher accumulated reward, and indicates a better performing policy.

4.6.2 HIV Therapy Selection

Cohort: Data for these patients were obtained from the EuResist database (Zazzi et al., 2012). We extracted the genotype and treatment response data of $N = 32\,960$ patients together with their CD4$^+$ and viral load measurements, gender, age, risk group and prior recorded treatments. The measurements are collected at approximately 6 month intervals corresponding to hospital visits. Variables with excessive missingness were removed, and any remaining missing values were imputed. We restrict the space of therapy combinations to the 312 most frequently occurring combinations in the cohort. These drug combinations span 20 drugs in total. Table 4.2 provides a summary of the cohort statistics used.

Number of Patients	32960
Average Sequence Length	14
Feature Dimensionality	134
Number of Actions	312

Table 4.2: Summary of HIV cohort statistics.

Reward function: We use the same reward function as before from Equation 3.3.3.

Experimental setup: We performed a random 80%-10%-10% train-test-validation split of our cohort of patients and compared the performance of KDM against the baselines. This split resulted in a held-out test set consisting of 3000 patients with the same distribution as patients in the training set. The training set was the largest split as we needed to learn the large number of parameters governing the kernel, POMDP, and dynamic mixing network. The random policy selects a therapy randomly for each forward time step across all patients. For the kernel policy we used a modified version of the history alignment kernel based on Bogojeska et al. (2012). This kernel compares therapy histories of patients on the basis of the drugs used, the order in which they are administered and the subsequent resistance mutations they produce. We modify this kernel to also match on the basis of other mutations from the Pol region of the viral genome in Figure 1.2, in addition to the resistance mutations. Importantly, two therapy histories in this case are considered similar if they contain similar drugs that are administered in a similar order and produce similar mutations. For the POMDP policy, we learn a POMDP model with 30 states as specified in Chapter 3. For planning, we perform a forward search for therapy choices that optimise patient outcomes over a 30-month horizon (corresponding to 5 forward time steps, which was chosen for tractable planning). Our mixing network for KDM consists of 100 input units and 2 hidden layers of 50 units each, where the number of parameters is selected by performing cross-validation on an independent hold-out set. Since the problem is non-linear by nature, our mixing network requires enough parameters to adequately approximate a smooth mapping between inputs and the mixing proportion. At the same time, over-parameterisation results in overfitting. To prevent the latter, we regularise the network with an L2 regularisation of strength $\lambda = 15$.

Results: Table 4.3 summarises the performance of KDM compared to the aforementioned baselines. The KDM policy produces the highest accumulated immune response while reducing the viral load, outperforming the other baselines over a 30-month long-term horizon. The choice of time horizon is made on the basis of how frequently an HIV patient visits the hospital for treatment, medical guidelines and drugs available. Patient visits usually occur on a bi-annual basis, while medical guidelines and available drugs for treating HIV may change over longer periods of time. In general however, KDM may also be applied to extended time horizons.

	DR	**WIS**	**IS**
Random	–7.31 ± 3.72	–11.48 ± 4.36	–10.64 ± 4.81
Kernel	9.35 ± 2.61	6.42 ± 3.93	6.73 ± 3.62
POMDP	3.37 ± 2.15	3.86 ± 2.38	3.74 ± 2.46
Mixture-of-Experts	11.52 ± 1.31	12.25 ± 2.01	11.36 ± 2.97
KDM	**12.47 ± 1.38**	**14.25 ± 1.27**	**14.48 ± 1.41**

Table 4.3: Performance comparison of KDM vs. baselines for HIV therapy selection across 3000 held-out patients using a POMDP model with 30 states. KDM produces the largest immune response while reducing the viral load, thus outperforming its competitors.

From observing the quantitative differences between the performance of KDM and the mixture-of-experts policy, we can conclude that both the policies are different. Importantly, the model-based nature of KDM has several key benefits (particularly in a high-risk setting such as therapy selection). We highlight these differences with a motivating example: Consider an HIV-infected patient whose underlying health status is unknown, but with a baseline viral load of 589 copies/mL . If a patient is treated with a first-line therapy of EFV/3TC/TDF, we obtain a set of observations and rewards from which subsequent treatments may be selected. Based on the treatment of EFV + 3TC + TDF and the patient's particular observations, KDM predicts that the viral load will drop below detection limits for a period of 6 months (which may or may not change the patient's overall health status). At 12 months, KDM predicts that the virus reappears in the patient's bloodstream, but falls below detection limits again shortly after this period. The mixture-of-experts policy suggests a treatment change at 12 months from first line therapy to a more aggressive second-line therapy of AZT + 3TC + TDF + LPV/r.

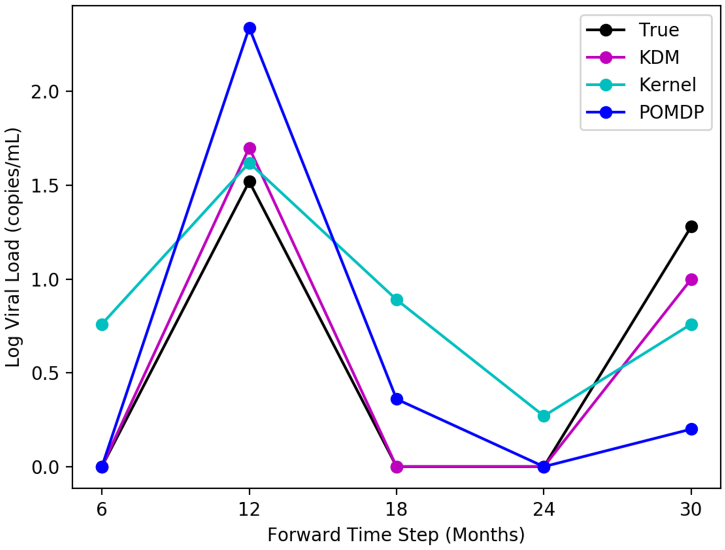


Figure 4.3: Simulating the viral load in an HIV patient when the viral load is below detection limits (indicated by 0). KDM can detect the occurrence of blips at 12 and 30 months, unlike a MoE. No treatment change should be administered here.

Because however, KDM actively simulates a patient's future trajectory, it is able to predict the occurrence of a blip in the viral load at 12 months. As a result, the KDM policy continues using the same first-line therapy over this period, without suggesting a change in treatments. The implications of this are important: through actively forward simulating a patient's long-term future, we can analyse the impact of making treatment decisions in terms of the particular outcomes that they may produce. The example here, highlights the fact that KDM is able to forward simulate such occurrences as blips in the viral load and use this information to deduce whether or not a therapy switch is necessary. In this case, switching treatments to a more aggressive treatment is unnecessary and potentially reduces a patient's future therapy options. Importantly, the KDM policy may be easily interpreted through explicitly examining and auditing our forward simulations. This interpretability is key to building trust in machine learning methods in high-risk settings. Figure 4.3 illustrates forward simulating the viral load for the test patient described here. The ground truth, and respective kernel and POMDP-based predictions are shown. Since the mixture-of-experts approach combines kernel and model-based learning on a *policy* level, it is impossible to obtain a set of forward predictions of a patient's viral load (hence we cannot illustrate a trajectory for it here). The corresponding predictive log-likelihood is shown in Figure 4.4. Here, KDM's forward predictions are closer to the ground truth and ultimately result in learning a more effective treatment policy overall. While obviously a single-patient anecdote, we found many such situations in which the KDM predicted deviations in trajectories.

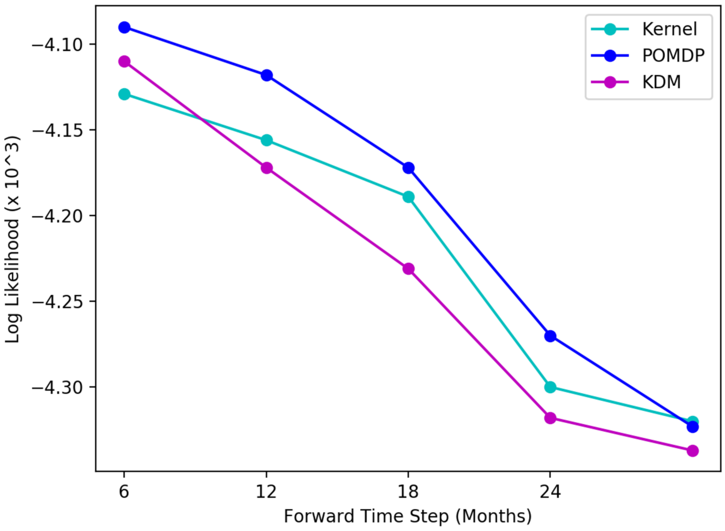


Figure 4.4: Comparison of predictive log-likelihood across baselines for HIV for a typical test patient. KDM's predictions are more accurate across the forward time steps.

We obtain similar results on the rest of the patients in the test set. Figure 4.5 illustrates the deviations in counterfactual predictions of the viral load over a 30-month horizon. KDM is able to model and predict counterfactuals more accurately than the other baselines. This performance is sustained across all time steps.

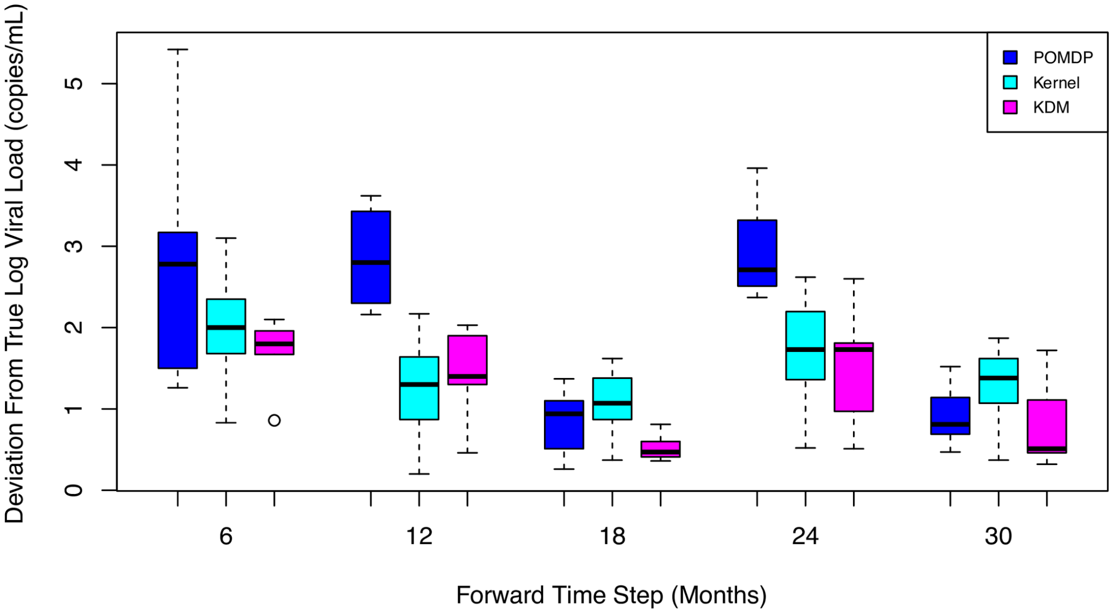


Figure 4.5: Box plot of viral load predictions across 3000 test patients under baselines over a 30-month horizon. KDM's predictions are closer to the ground truth than POMDP or kernel predictions.

Number of Patients	18200
Average Sequence Length	13
Feature Dimensionality	47
Number of Actions	100

Table 4.4: Summary of sepsis cohort statistics

4.6.3 Sepsis Management

Cohort: Data for these patients were obtained from the publicly available Multiparameter Intelligent Monitoring in Intensive Care (MIMC-III v1.4) database (Johnson et al., 2016b), containing hospital admissions for approximately 38600 adults (at least 15 years old). We extracted a cohort of patients fulfilling Sepsis-3 criteria (Singer et. al, 2016). A summary of the populations can be found in Table 4.4. We extracted the appropriate physiological parameters such as demographics, lab values, vital signs and intake-output events. The data were aggregated into 4 hour windows, where the mean or sum was recorded (as appropriate) when several data points were present in a window. Variables with excessive missingness were excluded, and other missing values were imputed. This produced a feature vector of size 47×1 per patient for each time step. The values of each feature were passed through a sigmoid function to reduce the effect of outliers and subsequently normalised to zero mean and unit variance.

The action space of medical interventions was defined to cover the space of intravenous (IV) fluid, and maximum vasopressor (VP) dosage, as well as whether or not to sedate and ventilate a patient in a given four hour window. We discretised the action space into per-drug quartiles based on all non-zero dosages of the two drugs, and converted each drug at every time step into integer values representing the respective quartile bin. We included a special case of no drug given as bin 0. This created an action representation of interventions as tuples of (total IV in, maximum VP in, sedation, mechanical ventilation) at each time step.

Reward function: Our overall goal in this task is to reduce patient mortality. Mortality, however, is a sparse outcome: whether a patient survived can only be known at the end of the stay. At the recommendation of our clinical colleagues, we use the log odds of in-hospital mortality as described in Raghu et al. (2017); Ross et al. (2017b) as an intermediate cost function for treating sepsis at each time step (we note, more broadly, that there exists relatively little clinical literature on optimisation criteria for sepsis). This reward function is trained on a held-out subset of the sepsis data cohort. Summing the log odds of in-hospital mortality over a patient's future allows us to explicitly quantify a patient's odds of mortality over this period. Since our goal is to reduce mortality, a lower accumulated cost corresponds to a better performing treatment policy in this case. (We also emphasise that our dynamic mixing procedure is general in that it can be applied to any cost or reward function, and retrained as domain experts refine their cost functions.)

Experimental setup: Once again we performed a random 80%-10%-10% train-test-validation split of our cohort of patients and compared the performance of KDM against the baselines on a held-out set of 3000 patients. For the kernel policy, we use a kernel that matches based on the length of the agent's history, action choices, and observations. For the POMDP policy, we learn a POMDP model with 75 states with Gaussian emissions, corresponding to the observation space of lab values, vital signs and intake-output events described above.

For the planning, we perform a forward search for therapy choices that optimise patient outcomes over a 20-hour horizon, again corresponding to 5 forward time steps that was both the limit of tractable planning and reasonable given that stays in the ICU are relatively short. Our mixing network for KDM consists of 40 input units and 2 hidden layers of 25 units each. The number of network parameters is again selected by performing cross-validation on an independent hold-out set.

Results: Table 4.5 summarises the performance of KDM compared to the aforementioned baselines for sepsis management. The KDM policy significantly reduces the risk of mortality for held-out patients over a 20-hour horizon, once again outperforming the other baselines.

	DR	**WIS**	**IS**
Random	4.31 ± 1.72	3.52 ± 1.76	4.26 ± 1.82
Kernel	-0.88 ± 0.41	-1.47 ± 0.33	-1.63 ± 0.48
POMDP	1.73 ± 1.69	1.73 ± 1.25	1.86 ± 1.29
Mixture-of-Experts	-1.42 ± 0.71	-1.85 ± 0.57	-1.46 ± 0.79
KDM	**-1.87 ± 0.39**	**-2.25 ± 0.77**	**-2.86 ± 0.80**

Table 4.5: Performance comparison of KDM vs. baselines for treating sepsis across 3000 held-out patients using a POMDP model with 75 states. The KDM policy significantly reduces the odds of mortality (indicated by a lower value here), and outperforms existing baselines.

In the context of sepsis too, the quantitative differences between the performance of KDM and the mixture-of-experts policy indicates that the policies are different. As with HIV, we provide an illustrative example. Consider a patient whose blood pressure, heart rate and respiratory rate are all within normal limits. $\mathrm{SpO_2}$ is used to quantify the saturation of oxygen in the blood. If a patient is initially not ventilated, sedated, or prescribed any vasopressors, we obtain a set of observations and rewards from which subsequent treatments may be selected. Based on the lack of sedation or need to mechanically ventilate initially, KDM predicts the blood oxygen saturation is within normal limits ranging between $90\% - 100\%$. Over the course of 30 hours, this prediction varies marginally when there are minor changes in blood pressure, heart rate and respiratory rate. Throughout this period, no vasopressors are required or prescribed. This is clinically reasonable since vasopressors are typically used to raise the blood pressure hypotensive patients, and are thus not required in this situation. Figure 4.6 illustrates forward simulating $\mathrm{SpO_2}$ for the patient described here. The corresponding predictive log-likelihood is shown in Figure 4.7. As before, the ground truth and respective kernel and POMDP-based predictions are also shown. KDM's forward predictions are visibly more accurate with respect to the ground truth and contribute to learning a better treatment policy.

Again, we obtain similar results on the rest of the patients in the test set. Figure. 4.8

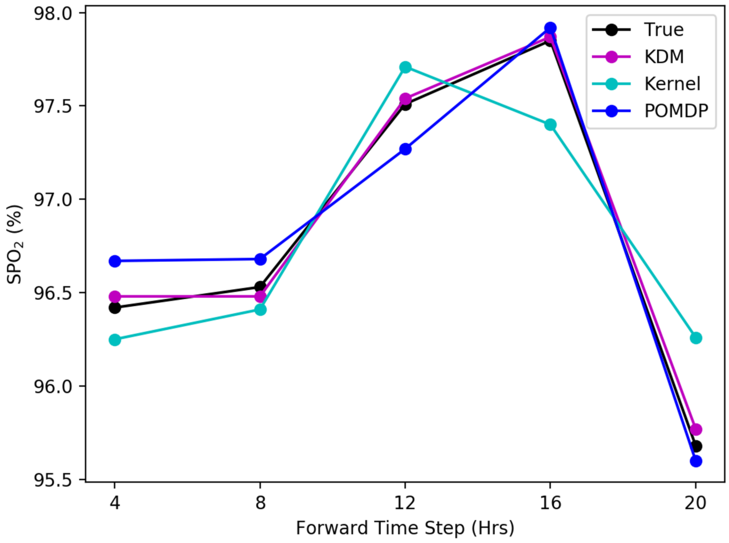


Figure 4.6: Simulating the SpO$_2$ of a sepsis test patient under baselines over a 20-hour horizon. Counterfactual predictions of SpO$_2$ levels are more accurate using KDM than existing baselines.

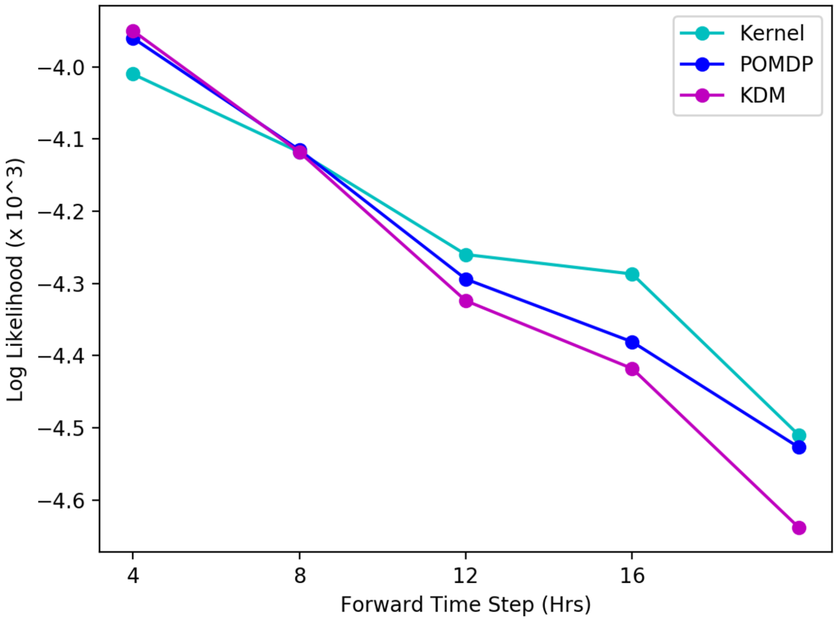


Figure 4.7: Comparison of predictive log-likelihood across baselines for sepsis for a typical test patient. KDM's predictions are more accurate across the forward time steps.

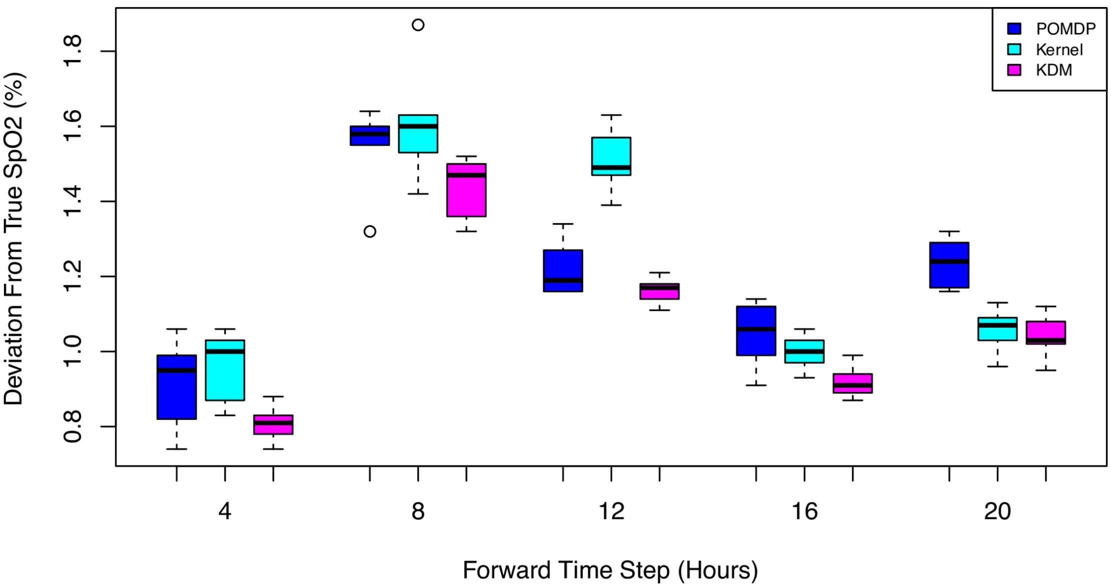


Figure 4.8: Box plot of SpO_2 predictions across 3000 test patients under baselines over a 20-hour horizon. KDM's predictions are closer to the ground truth than POMDP or kernel predictions.

illustrates the deviations in counterfactual predictions of SpO_2 over a 20-hour horizon. KDM is able to model and predict counterfactuals more accurately than the other baselines. This performance is sustained across all time steps.

4.7 Discussion

KDM produces accurate forward predictions. The KDM policy results in more accurate counterfactual predictions over observation across both the HIV and sepsis tasks. Figure. 4.5 and 4.8 show the differences at each forward time step between counterfactual predictions using the kernel, POMDP and KDM, and the ground truth across HIV and sepsis patients respectively. Note that these differences cannot be calculated for the MoE policy as this approach does not permit simulating counterfactuals. We observe that across all time steps, the KDM policy tends to predict counterfactuals that are generally closer to the ground truth than those predictions made using the kernel or POMDP methods.

While the kernel and POMDP policies vary considerably over time in their closeness to the true observation, the KDM policy is able to make accurate predictions by combining these predictions and weighting them appropriately. We can also examine the predictive log-likelihoods of all three approaches for both tasks across each of the forward time steps. An example of these is shown in Figure 4.4 where we see considerable differences between these values across the methods in the HIV task. For each method, the predictive log-likelihood tends to increase with each forward time step. This is likely a result of more data being available at each successive simulation step in which the histories are grown. Nonetheless, KDM significantly outperforms both the POMDP and kernel approaches at most forward steps. These results are summarised in Tables 4.6 and 4.7 for both HIV and sepsis tasks, where we perform a Friedman's statistical significance test with post-hoc analysis to measure the differences in predictive performance of KDM against the POMDP and the kernel respectively across all test patients. A p-value < 0.05 here indicates a significant result.

t	1	2	3	4	5
POMDP	**0.046**	**0.041**	**0.047**	**0.042**	0.073
Kernel	0.057	0.086	**0.047**	0.058	**0.042**

Table 4.6: Friedman's test measuring predictive performance differences of KDM against POMDP and kernel methods across `t` in HIV. Bold p-values correspond to steps where counterfactual predictions from KDM are significantly more accurate than the respective methods. Comparisons with policy-based approaches like the mixture-of-experts cannot be drawn here as these methods cannot be used for counterfactual predictions.

t	1	2	3	4	5
POMDP	**0.041**	**0.038**	**0.049**	0.083	**0.046**
Kernel	**0.038**	**0.036**	**0.041**	0.091	0.083

Table 4.7: Friedman's test measuring predictive performance differences of KDM against POMDP and kernel methods across `t` in sepsis. Bold p-values correspond to steps where counterfactual predictions from KDM are significantly more accurate than the respective methods. Comparisons with policy-based approaches like the mixture-of-experts cannot be drawn here as these methods cannot be used for counterfactual predictions.

Mixing kernel and model-based RL on a model level produces different policies to mixing on a policy level. Just from the quantitative results, it is clear that the policies produced by our KDM and the MoE are different. We attribute these differences directly to the way in which KDM computes its policy: KDM mixes approaches on the model level, and incorporates these predictions into its belief states for learning an optimal policy. In this way it is able to account for variations across patients at different time points and use these variations to draw new examples of observations from which it can learn. For example in the HIV task, we observe that the KDM policy tends to contain less switches between drug combinations in comparison to the MoE policy. This occurs specifically in cases where patients experience temporary blips or spikes in their viral loads as shown in Figure 4.3 at 12 and 30 months in the future respectively. Because the KDM policy directly mixes kernel and model based approaches in simulating observations, it can identify these cases more effectively. In these situations, the typical KDM policy does not call for a change in treatments, whereas a MoE policy does. While spurious blips are not regular occurrences, in a clinical setting, it is still important to be able to detect them since it prevents a clinician from potentially exhausting a patient's future treatment options and exposing them to more potential side effects than necessary.

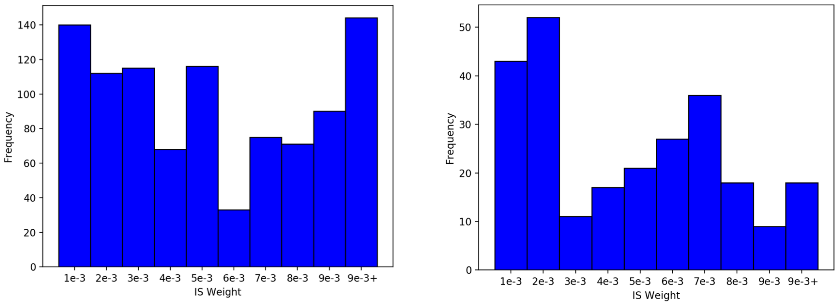
KDM leads to interpretable treatment decisions that are clinically face-valid. In both the toy and real experiments, we can demonstrate that the policies obtained using KDM make sense. For the toy task for variants of Type A, KDM correctly chooses a_1 at the second time step, while for variants of Type B, it chooses a_2 here. Since the POMDP is unable to make any informed choice here, the KDM policy typically assigns a higher weight to the nearest neighbour predictions at the second time step and uses these to determine the correct action choice at this step.

For the HIV task, we observe that test patients with higher baseline viral loads tend to sustain higher viral loads and lower CD4$^+$ counts in our forward simulations. This is consistent with medical literature that suggests patients with higher baseline viral loads tend to have faster disease progression (Langford et al., 2007; Socias et al., 2011). In these cases, the KDM policy typically consists of using a nucleoside reverse transcriptase inhibitor (NRTI) such as Zidovudine (AZT), in conjunction with a protease inhibitor (PI) such as Liponavir/ritonavir (LPV/r). Our clinical collaborators confirm that these choices are valid, since a single boosted PI and an NRTI are typically recommended for second-line ART when first-line therapy fails (as indicated by sustaining a viral load above detection limits) (Sungkanuparph et. al., 2007). We also checked our treatment policies against current ART guidelines (Günthard et al., 2016; Günthard et al., 2016). Overall, we found that our policies were consistent with the recommended first and second-line therapy guidelines 81% of the time. In contrast, the policies obtained from the MoE approach were consistent 76% of the time. KDM policies in violation of IAS-USA recommendations were slightly more likely for patients who started in ART in the early 90s, as standards for combination ART differed significantly at that time. MoE policies in violation of IAS-USA recommendations were more likely for patients experiencing single episodes of low-level viremia or blips, which typically have no clinical consequences, as well as cases where patients were infected by multiple HIV strains. In general, patients infected by multiple HIV strains tend to be more difficult to treat since chances of drug resistance are higher. This, in general, motivates the need for more nuanced treatment policies (e.g. via forward simulation) as suggested by KDM.

There exist less consistent guidelines for the management of fluid and vasopressor administration for patients with sepsis, but we find that the policies recommended by KDM still have many sensible properties, including being consistent with prior work by Raghu et al. (2017). KDM frequently (72% of the time) learn policies where no vasopressors are prescribed. This result is reasonable as vasopressors are used to raise arterial blood pressure in hypotensive patients, and the majority of the test patients do not fall into this category. The KDM policy suggests mechanically ventilating patients with SpO$_2$ predictions below 85%, when corresponding predictions of their respiratory rates exceed 29 breaths per second. Several other methods have also been suggested for detecting events such as desaturation and transient hypoxia, but there is frequently a high false alarm rate as described in Bodilovskyi & Popov (2013). In these instances, further clinical expertise is required before intervening. KDM gives us thresholds that we can discuss and debate.

Most importantly, across all three tasks, it is the ability to explicitly step through our forward predictions via KDM that enables us to interpret the policies easily. Overall, we hope that the generative approach of the KDM could help better assess a patient's overall prognosis and offer more informed therapy choices for intervention.

The policies obtained from KDM are stable over multiple runs. We tested the performance of KDM over multiple runs on the test data. While the sampled ob-



(a)

(b)

Figure 4.9: Distributions of frequencies of non-zero IS weights for (a) HIV and (b) sepsis respectively. Our treatments are fairly consistent with those in the data sets.

servations and trajectories obtained may differ during forward simulation, the therapy policies obtained across the real world data sets remained virtually identical. Specifically, we obtained fidelity scores of 95% for the HIV domain and 93% for the sepsis task. This stability is crucial to building trust in our policies. A related issue that is frequently encountered when using off-policy evaluation is that only a small fraction of the data contains the treatments suggested by the policies we learn. Figure 4.9(a) demonstrates that our treatments for HIV are fairly consistent with those in the data set, and at least 1/3 of the test values have non-zero weights. Similar results hold for the sepsis data set in 4.9(b). This spread is also essential for building trust in our results. That said, these off-policy estimators can be sensitive to the choice of reward and representation; a limitation of all approaches relying on off-policy evaluation is that the reward function is often some surrogate for what we actually wish to optimise, and that we have to assume that the POMDP belief is a sufficient statistic for the history.

4.8 Conclusion

In this chapter, we introduced kernelised dynamic mixing (KDM), as a novel approach for forward simulating counterfactuals. The approach combines the forward predictions from non-parametric kernel-based learning with predictions based on a causal model to produce a more accurate simulator. By simulating outcomes more accurately, we can subsequently learn more effective treatment policies. In particular, using KDM significantly improves upon the policy performance in two real medical tasks for HIV and managing sepsis, while also providing the ability to interpret and interrogate the policies via simulating counterfactual scenarios. These steps take us toward being able to provide better decision-support in situations where clinicians must plan over sequences of decisions.

Like the policy mixing approach we presented in Chapter 3, we have used KDM to construct a more accurate causal model specifically in order to infer suitable treatment policies. However, the overall idea of combining forward predictions to produce a more accurate simulator could be used as a back off strategy when modelling value functions in off-policy evaluation methods such as doubly robust off-policy evaluation. In this case, we would once again re-cast the problem of off-policy evaluation in terms of planning and use this as a basis to rank existing or hypothetical clinical policies in different scenarios.

Chapter 5

Cause-Effect Information Bottleneck For Systematically Missing Data

5.1 Introduction

Previously, we combined RL and nearest-neighbour approaches to individualise treatment policies to a patient's needs over time. Both of the methods presented in Chapters 3 and 4 allowed us to simultaneously address patient heterogeneity and account for the effects of measured confounding. In the RL case, this information was captured through the notion of a hidden state that we conditioned on to learn a suitable treatment policy, while the nearest-neighbour approach used a similarity score instead. However, it may still be difficult to decipher what information is confounding based on a hidden state representation, as these may be high dimensional in practice; analogously, it may not always be clear what similarity score to use. This information is especially important if we would like to transfer this knowledge to different sets of patients in the future. Hence in this chapter, we focus on explicitly learning better representations of confounding, such that we can account for these effects even in the absence of complete information at test time. Overall, this allows us to estimate treatment responses more accurately. The majority of the material we present is based on Parbhoo et al. (2018b).

As we have seen so far, predicting the causal effects of an intervention from observational data requires accounting for confounding. In healthcare, the problem is often complicated by the fact that we may have a complete, high-dimensional set of observational measurements for a group of patients, but only have an incomplete set of measurements for a potentially larger group of patients for whom we would like to infer treatment effects at test time. For instance, a doctor treating patients with HIV may readily have access to routine measurements such as blood count data for all their patients, but only have the genotype information for some patients as a result of medical costs or resource limitations.

In such a situation, conventional approaches to adjust for confounding such as propensity reweighting or covariate shift (Hernán & Robins, 2006a; Rosenbaum & Rubin, 1984) do not suffice, since they do not explicitly address the issue of missingness in data. A naive strategy to address this problem would be to remove all those features that are missing at test time and infer treatment effects on the basis of the reduced space of features. Alternatively, one may attempt imputing the incomplete dimensions for the same purpose. Both of these solutions however, fail in high-dimensional settings, particularly if the missingness is systematic as in this case, or if many dimensions are missing. Other approaches account for incomplete data during training, for instance by assuming hidden confounding. These methods typically try to build a joint model on the basis of noisy representatives of confounders (see for instance Greenland & Lash (2008); Kuroki & Pearl (2014); Louizos et al. (2017); Pearl (2012b)). However, in high-dimensional-settings, it is unclear what these representatives might be, and whether our data meets such assumptions. Regardless of these assumptions, none of these approaches addresses systematic missingness at test time.

A more natural approach would be to assume one could measure everything that is relevant for estimating treatment effects for a subset of the patients, and attempt to transfer this distribution of information to a potentially larger set of test patients. However, this is a challenging task given the high dimensionality of the data that we must condition on. In this case, one might try to first perform dimensionality reduction via Principal Component Analysis (PCA) to first construct a new representation of the data with fewer dimensions, and subsequently attempt to transfer this information to the set of test patients with missing covariates in order to infer treatment effects. In general however, the performance of PCA is highly dependent on the choice of distortion function that defines what attributes in the data are important to preserve (Rey, 2015), and certain distortion functions are not applicable to every type of data (Joe, 1989). Moreover, in the context of causal inference, it may be unclear as to what distortion function makes sense and how one would subsequently transfer this knowledge to a set of test patients whose covariates are partially missing. The key question is thus how can we perform such a distribution transfer to a set of patients with missing covariates in practice? Here, we propose tackling this question from the decision-theoretic perspective of causal inference. The overall idea is to use the Information Bottleneck (IB) criterion (Alemi et al., 2016; Tishby et al., 2000) to perform a sufficient reduction of the covariate (see Section 2.5.2) and recover a distribution of the confounding information. Unlike traditional dimensionality reduction techniques, the IB is expressed entirely in terms of information-theoretic quantities rather than distortion functions. As a result, the IB principle is particularly appealing in this context as it allows us to define a good representation of confounding, by trading off learning a reduced covariate with predicting treatment effects. Specifically, by conditioning on this reduced covariate, the IB enables us to build a discrete reference class over patients with complete data, to which we can map patients with incomplete data at test time, and subsequently estimate treatment effects on the basis of these groups.

In what follows, we describe the details of this approach and demonstrate our method outperforms existing methods across established causal inference benchmarks, as well as the tasks of treating sepsis and HIV.

5.2 Background and Prior Work

Before we describe the details of the IB method, we briefly introduce the concepts of *Kullback-Leibler (KL) divergence* or *relative entropy* and *mutual information* that are relevant for the rest of this chapter. We refer the reader to Cover & Thomas (2012); MacKay (2003) for a more detailed description of these.

Definition 5.2.1 (Kullback-Leibler Divergence)**.** *Let $X \in \mathcal{X}$ be a random variable, and $P(X)$ and $Q(X)$ denote two continuous probability distributions over X. The Kullback-Leibler divergence of $Q(X)$ from $P(X)$, denoted $D_{KL}(P(X)||Q(X))$ is a measure of how different the two probability distributions are. Formally, $D_{KL}(P(X)||Q(X))$ is defined as,*

$$D_{KL}(P(X)||Q(X)) := \int_{X \in \mathcal{X}} P(X) \log \frac{P(X)}{Q(X)} dX. \tag{5.2.1}$$

Note that $D_{KL}(P(X)||Q(X)) \geq 0$ and equals 0 if and only if $P = Q$.

Based on the definition of the KL-divergence, we can define mutual information as follows.

Definition 5.2.2 (Mutual Information)**.** *Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ denote two random variables, whose joint distribution is defined by $P(X, Y)$. The mutual information between X and Y, denoted $I(X; Y)$, is given by,*

$$I(X;Y) := D_{KL}(P(X,Y)||P(X)P(Y)) := \int_{\mathcal{X}} \int_{\mathcal{Y}} P(X,Y) \log \frac{P(X,Y)}{P(X)P(Y)} dX dY, \tag{5.2.2}$$

where $P(X)$ and $P(Y)$ are the marginal distributions of X and Y.

Evidently, since the mutual information is expressed as a KL-divergence, it can also be re-formulated in terms of entropies. Hence mutual information satisfies the following properties:

1. *Non-negativity: $I(X; Y) \geq 0$*

2. *Symmetry: $I(X; Y) = I(Y; X)$*

3. *Relation to conditional and joint entropy:*

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \\ &= H(X,Y) - H(X|Y) - H(Y|X), \end{aligned}$$

   where $H(X)$ and $H(Y)$ are the marginal entropies of X and Y, $H(X|Y)$ and $H(Y|X)$ are the conditional entropies, and $H(X, Y)$ is the joint entropy of X and Y respectively.

5.2.1 The Information Bottleneck

The IB method is a compression technique first introduced by Tishby et al. (2000), that considers the *relevance* of information to deduce a meaningful compression. The classical IB method describes an information-theoretic approach to constructing a compressed representation Z of a random variable X that is most informative about another random variable Y. Let I denote the mutual information between two random variables. Achieving such a compression requires solving the problem,

$$\max_{P(z|x)} - I(Z; X) + \lambda I(Z; Y), \tag{5.2.3}$$

under the assumption that $Z \perp\!\!\!\perp Y \mid X$, or that the variables satisfy the Markov relation $Z - X - Y$. This relation indicates that compression Z cannot contain more information about Y than the original data X. Specifically, $I(X;Z)$ measures how close the compression is to the original data: a high value corresponds to a lower compression. Analogously, $I(Z;Y)$ measures the information that the compression contains about Y; a high value indicates that more relevant information is preserved. The Lagrange parameter λ controls the degree of compression, by trading off compression of X with preserving information about Y. A smaller λ will favour compression. Adjusting λ allows for a task-dependent compression Z.

In its classical form, the IB is defined for discrete random variables such that Z is a discrete cluster structure over X. However in recent years, multiple IB relaxations and extensions, such as for Gaussian (Chechik et al., 2005) and meta-Gaussian variables (Rey & Roth, 2012), have been proposed. Among these extensions, the Gaussian bottleneck assumes that both X and Y are jointly Gaussian. In this case, $Z = AX + \xi$ where $\xi \sim \mathcal{N}(0, \Sigma_\xi)$, may be viewed as a noisy projection of X that is also jointly Gaussian with (X, Y) (Rey, 2015). Since the Gaussian IB admits an analytic form, it is widely applicable to many problems. Specifically, when Y is a one-dimensional random variable, the solution to the Gaussian IB is equivalent to least squares regression, while if Y is a noisy version of X, the Gaussian IB is analogous to PCA; for a general Y, the Gaussian IB may be viewed as an asymmetric version of Canonical Correlation Analysis (Rey, 2015).

More recently, Alemi et al. (2016) proposed a latent variable formulation of the IB problem, where the linear mean function in the Gaussian IB, $\mu = AX$, is replaced by a non-linear neural network. In this formulation of the IB, one assumes structural equations of the form,

$$\begin{aligned} z &= f(x) + \eta_z, \\ y &= g(z) + \eta_y. \end{aligned} \tag{5.2.4}$$

These equations give rise to a different conditional independence assumption, $X - Z - Y$. While both independence assumptions from the classical formulation of the IB and its latent variable counterpart cannot hold in the same graph, in the limiting case where the noise term $\eta_y \to 0$, $Z \perp\!\!\!\perp X \mid Y$. Like the Gaussian IB, the latent variable formulation of the IB may viewed as an asymmetric version of CCA. The model we develop in this chapter assumes the latent variable formulation of the IB. Unlike each of these formulations, we extend the latent variable form of the IB to learn a sufficiently reduced covariate such that we may infer causal effects.

5.2.2 Deep Latent Variable Models

Deep latent variable models have recently received remarkable attention and have been applied to a variety of problems. Among these, variational autoencoders (VAEs) employ the reparameterisation trick introduced in Kingma & Welling (2013); Rezende et al. (2014) to infer a variational approximation over the posterior distribution of the latent space $P(z|x)$. Important work in this direction includes Kingma et al. (2014) and Jang et al. (2017). Most closely related to the work we present here, is the application of VAEs in a healthcare setting by Louizos et al. (2017). Here, the authors introduce a Cause-Effect VAE (CEVAE) to estimate the causal effect of an intervention in the presence of noisy proxies. In high-dimensional settings, this approach requires many strict modelling assumptions.

Despite their differences, it has been shown that there are several close connections between the VAE framework and the previously described latent variable formulation of the IB principle (Alemi et al., 2016; Wieczorek et al., 2018). This is essentially a VAE where X is replaced by Y in the decoder. In contrast, our approach considers the IB principle to perform *causal inference* in scenarios where only *partial covariate data* is available at test time.

5.2.3 Models for Causal Inference with Missing Data

A sizeable amount of work has also been done on both causal inference with missing data, and transfer learning for estimation of causal effects. Previously, Cham & West (2016) presented an empirical example of the performance of propensity score estimation methods when adapted to incompletely observed covariates. More recently, Kallus et al. (2018) perform a low-rank matrix factorisation on a noisy set of covariate matrices to deduce a set of confounders based on which one can infer treatment effects. The approach is general enough to adapt to scenarios where covariates are missing at random and can be used as a preprocessing step for other bias correction techniques such as propensity reweighting. Unlike both of these, we make use of the IB criterion to learn treatment effects and adjust for confounding.

5.3 Model and Inference

In this section, we present an approach based on the IB principle for estimating the causal effects of an intervention with partial covariates at test time. We refer to this model as a *Cause-Effect Information Bottleneck (CEIB)*. In recent years, there has been a growing interest in the connections between the IB principle and deep neural networks (Alemi et al., 2016; Tishby & Zaslavsky, 2015; Wieczorek et al., 2018). Here, we use the non-linear expressiveness of neural networks with the IB criterion to learn a sufficiently reduced representation of confounding, based on which we can approximate the effects of an intervention more effectively. Specifically, we interpret our model from the decision-theoretic view (Dawid, 2007b) of causal inference.

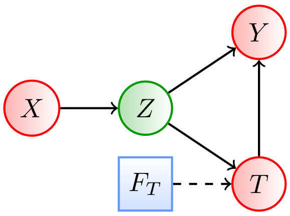


Figure 5.1: Influence diagram of the CEIB. Red and green circles correspond to observed and latent random variables respectively, while blue rectangles represent interventions. We identify a low-dimensional representation Z of covariates X to estimate the effects of an intervention on outcome Y where partial covariate information is available.

Problem Formulation Like other approaches in the decision-theoretic setting, our goal is to estimate the ACE of T on Y. We employ the following assumptions and notation. Let $X = (X_1, X_2)$ denote a set of patient covariates based on which we would like to estimate treatment effects. During training, we assume that all covariates $X \in \mathbb{R}^d$ can be observed as in a medical study, where dimension d is large. Outside the study at test time however, we assume covariates X_1 are not usually observed, e.g. due to the expensive data acquisition process. That is, we assume the same feature dimensions are missing for all patients at testing. Let $Y \in \mathbb{R}$ denote the outcomes following treatments T. For simplicity and ease of comparison with prior methods on existing benchmarks, we consider treatments T that are binary, but our method is applicable for any general T. We assume strong ignorability or that all confounders are measured for a set of patients. The causal model we assume is depicted in Figure 5.1. Importantly, estimating the ACE in this case, only requires computing the distribution $Y|Z, T$, provided Z is a sufficient covariate. In what follows, we use the IB to learn such a sufficient covariate, that allows us to approximate this distribution.

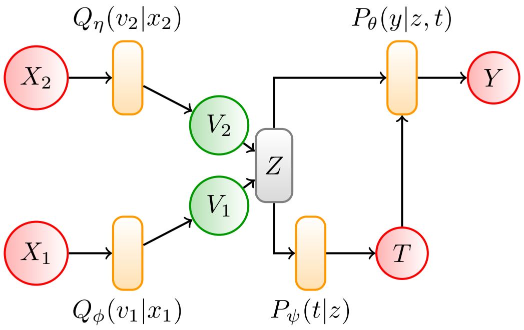


Figure 5.2: Graphical illustration of the CEIB. Orange rectangles represent deep networks parameterising the random variables

Performing a Sufficient Reduction of the Covariate We propose modelling this task with an extended formulation of the IB using the architecture shown in Figure 5.2. Here, our model consists of encoder networks Q_ϕ and Q_η, and a decoder network P_θ. The IB allows us to learn a low-dimensional compression of relevant information during training, such that we can infer treatment effects where covariate information is incomplete at test time.

We adapt the IB for learning the outcome of a therapy when incomplete covariate information is available for X_2 at test time. To do so, we consider the following extended parametric form of the IB,

$$\max_{\phi,\theta,\psi,\eta} -I_\phi(V_1; X_1) - I_\eta(V_2; X_2) + \lambda I_{\phi,\theta,\psi,\eta}(Z; (Y, T)), \tag{5.3.1}$$

where V_1 and V_2 are low-dimensional discrete representations of the covariate data, $Z = (V_1, V_2)$ is a concatenation of V_1 and V_2 and I represents the mutual information parameterised by networks ϕ, ψ, θ and η respectively. Each of the conditionals $Q_\phi(v_1|x)$,

$Q_\eta(v_2|x)$, $P_\theta(y|t,z)$, $P_\psi(t|z)$ thus has a parametric form. We describe each of the terms in 5.3.1 in turn. Expanding on the first term, we have

$$\begin{aligned} I_\phi(V_1;X_1) &= D_{KL}(Q_\phi(v_1|x_1)P(x_1)||P(v_1)P(x_1)) \\ &= \mathbb{E}_{P(x_1)}\left[D_{KL}(Q_\phi(v_1|x_1)||P(v_1))\right], \end{aligned}$$

where our encoder model $Q_\phi(v_1|x_1)$ is a variational approximation to $P(v_1)$ with parameters ϕ.

Similarly, we have,

$$\begin{aligned} I_\eta(V_2;X_2) &= D_{KL}(Q_\eta(v_2|x_2)P(x_2)||P(v_2)P(x_2)) \\ &= \mathbb{E}_{P(x_2)}\left[D_{KL}(Q_\eta(v_2|x_2)||P(v_2))\right], \end{aligned}$$

where $Q_\eta(v_2|x_2)$ is a variational approximation to $P(v_2)$ with parameters η.

Finally for our decoder model, we have

$$\begin{aligned} I_{\phi,\theta,\psi,\eta}(Z;(Y,T)) &\geq \mathbb{E}_{P(x,y,t)}\mathbb{E}_{P_{\phi,\eta}(z|x)}\big[\log P_\theta(y|t,z) \\ &\quad + \log P_\psi(t|z)\big] + H(y,t), \end{aligned} \tag{5.3.2}$$

where the lower bound follows from the fact that the mutual information of Z and (Y,T) can be expressed as a sum of the expected value of $\log P_\theta(y|t,z) + \log P_\psi(t|z)$, entropy $H(y,t)$ and two KL-divergences, which are by definition non-negative. This can be seen in the following derivation, where we drop the subscripts ϕ, ψ, η and θ for readability,

$$\begin{aligned} & \mathbb{E}_{P(y,t|x)}\left[\int P(z|x,y,t)\log P(y,t|z,x)\,dz\right] - \mathbb{E}_{P(y,t|x)}\left[\int P(z|x)\log P(y,t|z,x)\,dz\right] \\ = & \int\int [P(y,t,z|x) - P(y,t|x)P(z|x)]\log\frac{P(y,t,z|x)P(y,t|x)}{P(z|x)P(y,t|x)}\,dy\,dt\,dz \\ = & \int\int P(y,t,z|x)\log\frac{P(y,t,z|x)}{P(z|x)p(y,t|x)}\,dy,t\,dz + \int\int P(y,t|x)P(z|x)\log\frac{P(z|x)P(y,t|x)}{P(y,t,z|x)}\,dy\,dt\,dz \\ & + \int \underbrace{\left[\int [P(y,t,z|x) - P(y,t|x)P(z|x)]\,dz\right]}_{0}\log P(y,t|x)\,dy\,dt \\ = & \; D_{KL}(P(y,t,z|x)\|P(y,t|x)P(z|x)) + D_{KL}(P(y,t|x)P(z|x)\|P(y,t,z|x)) \geq 0, \end{aligned}$$

Averaging over x, and plugging in to the definition of mutual information from Equation 5.3.1, we arrive at:

$$\begin{aligned} I(Z;(Y,T)) &= \mathbb{E}_{P(x,y,t)}\mathbb{E}_{P(z|x,y,t)}\log P(y,t|z) + H(y,t) \\ &= \mathbb{E}_{P(x,y,t)}\mathbb{E}_{P(z|x)}\log P(y|t,z) + \log P(t|z) \\ &+ D_{KL}(P(y,t,z|x)\|P(y,t|x)P(z|x)) + D_{KL}(P(y,t|x)p(z|x)\|P(y,t,z|x)) + H(y,t) \\ &\geq \mathbb{E}_{P(x,y,t)}\mathbb{E}_{P(z|x)}\log P(y,t|z) + H(y,t). \end{aligned}$$

In order to implement such a decoder model in practice, we use an architecture similar to the TARnet (Johansson et al., 2016), where we replace conditioning on high dimensional covariates X with conditioning on reduced covariate Z. We can thus formulate the conditionals as,

$$\begin{aligned} P_\psi(t|z) &= \text{Bern}(\sigma_1(z)) \\ P_\theta(y|t,z) &= \mathcal{N}(\mu = \hat{\mu}, \sigma^2 = \hat{s}), \end{aligned} \tag{5.3.3}$$

with logistic function $\sigma(\cdot)$, and outcome Y given by a Gaussian distribution parameterised with a TARnet with $\hat{\mu} = tf_2(z) + (1-t)f_3(z)$. Note that the terms f_k correspond to neural networks. While distribution $P_\psi(t|z)$ is included to ensure the joint distribution over treatments, outcomes and covariates is identifiable, in practice, our goal is to approximate the effects of a given T on Y. Hence, we train our model in a teacher-forcing fashion by using the true treatment assignments T from the data, and fixing the treatments T at test time. Unlike other approaches to inferring treatment effects, the Lagrange parameter λ in CEIB allows us to adjust the degree of compression, which in this context, enables us to learn a sufficient statistic Z. In particular, adjusting λ enables us to explore a range of such representations from having a completely insufficient covariate to a completely sufficient compression of confounding.

Learning Equivalence Classes and Distribution Transfer Using the proposed architecture allows us to learn a low-dimensional compression Z of the relevant information during training. Since V_1 and V_2 are discrete latent representations of the covariate information, we make use of the Gumbel softmax reparameterisation trick (Jang et al., 2017) to draw samples Z from a categorical distribution with probabilities π. Here,

$$z = \texttt{one\_hot}(\arg\max_i [g_i + \log \pi_i]), \tag{5.3.4}$$

where $g_1, g_2, \ldots, g_k$ are samples drawn from Gumbel(0,1). The softmax function is used to approximate the arg max in Equation 5.3.4, and generate k-dimensional sample vectors $w \in \Delta^{k-1}$, where

$$w_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^{k} \exp((\log(\pi_j) + g_j)/\tau)}, i = 1, \ldots, k. \tag{5.3.5}$$

and τ is the softmax temperature parameter. By using the Gumbel softmax reparameterisation trick to obtain a discrete representation of relevant information, we can learn equivalence classes among patients based on which we can compute the SCE for each group using sufficient covariate Z via Equation 2.5.5. Specifically, during training, X_1 and X_2 are used to learn cluster assignment probabilities π for each data point. At test time, we subsequently assign an example (patient) with missing covariates to the relevant equivalence class. Computing the SCE allows us potentially to tailor or individualise treatments to specific groups based on Z rather than an entire population. This is especially important when addressing heterogeneity among patients. Based on the SCE, we can also compute the population-level effects of an intervention via the ACE from Equation 2.5.6. In the absence of the latent compression via CEIB and the discrete representation of relevant information, it would not be possible to transfer knowledge from examples with complete information to cases with incomplete information, since estimating treatment effects would require integrating over all covariates – an infeasible task in high dimensions.

5.4 Experiments

The goal of our experiments is to demonstrate the ability of CEIB to accurately infer treatment effects while learning a low-dimensional, interpretable representation of confounding in cases where covariate information is systematically missing at test time. We report the ACE and SCE values in our experiments for this purpose. In general, the lack of ground truth in real-world data makes evaluating causal inference algorithms a difficult problem. To overcome this, in our artificial experiments we consider a semi-synthetic data set where true outcomes and treatment assignments are known.

5.4.1 Infant Health and Development Program

The Infant Health and Development Program (IHDP) (Hill, 2011b; McCormick et al., 2013) is a randomised control experiment assessing the impact of educational intervention on outcomes of pre-mature, low birth weight infants born in 1984-1985. Measurements from children and their mothers were collected for studying the effects of childcare and home visits from a trained specialist on test scores. The study contains information about the children and their mothers/caregivers. Data on children includes, sex, birth weight, head circumference, health indices. Information about the mothers includes maternal age, mother's race as well as educational achievement. We denote this set of information as X. Treatments in this study T correspond to participation in IHDP child development centres, while outcomes Y correspond to the IQ-test score measured at the end of interventions. Like Hill (2011b), features and treatment assignments are extracted from the real world clinical trial, and selection bias is introduced in the data by artificially removing a non-random portion of the treatment group, in particular children with non-white mothers. In total, the resulting data set then consists of 747 subjects (139 treated, 608 control), each represented by 26 covariates measuring the properties of the child and their mother. We subsequently divide this data set into 60/10/30% training/validation/test sets. For our setup, we use encoder and decoder architectures with 3 hidden layers. Our model is trained with Adam optimiser with a learning rate of 0.001. We compare the performance of CEIB for predicting the ACE against several existing baselines: OLS-1 is a least squares regression; OLS-2 uses two separate least squares regressions to fit the treatment and control groups respectively; TARnet is a feedforward neural network from Shalit et al. (2017); KNN is a k-nearest neighbours regression; RF is a random forest; BNN is a balancing neural network (Johansson et al., 2016); BLR is a balancing linear regression (Johansson et al., 2016), and CFRW is a counterfactual regression that using the Wasserstein distance (Shalit et al., 2017). We train our model with four 3-dimensional Gaussian mixture components, although our method can be applied, without loss of generality, to any number of dimensions.

Experiment 1: In the first experiment, we compared the performance of CEIB for estimating the ACE against the baselines when using the complete set of measurements at test time. These results are shown in Table 5.1a. Evidently, CEIB outperforms existing approaches. To demonstrate that we can transfer the relevant information to cases where covariates are incomplete at test time, we artificially excluded $n = 3$ covariates that have a moderate correlation with ethnicity at test time. We compute the ACE and compare this to the performance of TARnet and CFRW also on the

Method	$\epsilon_{ACE}^{within-s}$	$\epsilon_{ACE}^{out-of-s}$
OLS-1	$.73 \pm .04$	$.94 \pm .06$
OLS-2	$.14 \pm .01$	$.31 \pm .02$
KNN	$.14 \pm .01$	$.79 \pm .05$
BLR	$.72 \pm .04$	$.93 \pm .05$
TARnet	$.26 \pm .01$	$.28 \pm .01$
BNN	$.37 \pm .03$	$.42 \pm .03$
RF	$.73 \pm .05$	$.96 \pm .06$
CEVAE	$.34 \pm .01$	$.46 \pm .02$
CFRW	$.25 \pm .01$	$.27 \pm .01$
CEIB	$\mathbf{.11 \pm .01}$	$\mathbf{.21 \pm .01}$

(a)

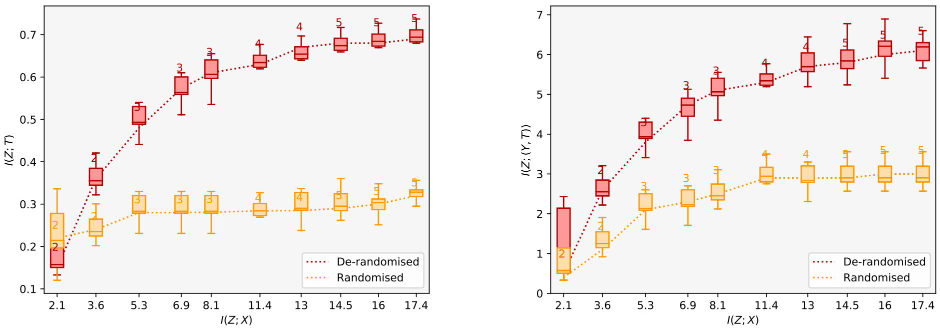
Method	$\epsilon_{ACE}^{within-s}$	$\epsilon_{ACE}^{out-of-s}$
TARnet	$.30 \pm .01$	$.34 \pm .01$
CFRW	$.28 \pm .01$	$.49 \pm .02$
CEIB	$\mathbf{.14 \pm .02}$	$\mathbf{.23 \pm .01}$

(b)

Table 5.1: (a) Within-sample and out-of-sample mean and standard errors in ACE across models on the complete IHDP data set. A smaller value indicates better performance. Bold values indicate the method with the best performance. (b) Within-sample and out-of-sample mean and standard errors in ACE across models using a reduced set of 22 covariates at test time.

reduced set of covariates (Table5.1b). If we extend this to the extreme case of removing 8 covariates at test time, the out-of-sample error in predicting the ACE increases to 0.29 +/- 0.02. Thus CEIB achieves state-of-the-art predictive performance for both in-sample and out-of-sample predictions, even with incomplete covariate information.

Experiment 2: Building on Experiment 1, we perform an analysis of the latent space of our model to assess whether we learn a sufficiently reduced covariate. We use the IHDP data set as before, but this time consider both the data before introducing selection bias (analogous to a randomised study), as well as after introducing selection bias by removing a non-random proportion of the treatment group as before (akin to a de-randomised study). We plot the information curves illustrating the number of latent dimensions required to reconstruct the output for the terms $I(Z;(Y,T))$ and $I(Z,T)$ respectively for varying values of λ. These results are shown in Figure 5.3a and 5.3b. Theoretically, we should be able to examine the shape of the curves to identify whether a sufficiently reduced covariate has been obtained. In particular, we know from Section 2.5.2 that in the case where a study is randomised, the sufficient covariate Z should have no impact on the treatment T (see Figure 2.6 where the α arm is removed). In this case, the mutual information $I(Z,T)$ should be approximately zero and the curve should remain flat for varying values of $I(Z,X)$. This result is confirmed in Figure 5.3a. The information curves in Figure 5.3b additionally demonstrate our model's ability to account for confounding when predicting the overall outcomes: when data is de-randomised, we are able to reconstruct treatment outcomes more accurately. Specifically, the point at which each of the information curves saturates is the point at which we have learnt a sufficiently reduced covariate based on which we can infer treatment effects. Overall, the results from Figures 5.3a and 5.3b highlight another benefit of using CEIB for estimating treatment outcomes: in particular, by adjusting the Lagrange parameter λ, CEIB allows for a task-dependent adjustment of the latent space. This adjustment allows one to explore a range of solutions across the information curve, from having a completely insufficient covariate to a completely sufficient compression of the covariates where the information curve saturates. In the absence of the IB objective, this is not possible. Overall, we are able to learn a low-dimensional representation that is consistent with the ethnicity confounder and account for its effects when predicting treatment outcomes.



(a)

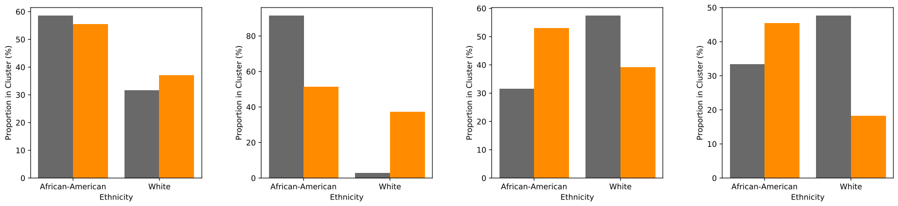
(b)

Figure 5.3: (a) Information curves for $I(Z;T)$ and (b) $I(Z;(Y,T))$ with de-randomised and randomised data respectively. When the data is randomised, the value of $I(Z;T)$ is close to zero. The differences between the curves illustrates confounding. When data is de-randomised, we are able to estimate treatment effects more accurately by accounting for this confounding.

We also analysed the discretised latent space by comparing the proportions of ethnic groups of test subjects in each cluster in the de-randomised setting. These results are shown in Figure 5.4 where we plot a hard assignment of test subjects to clusters on the basis of their ethnicity. Evidently, the clusters exhibit a clear structure with respect to ethnicity. In particular, Cluster 2 in Figure 5.4b has a significantly higher proportion of non-white members in the de-randomised setting. The discretisation also allows us to calculate the SCE for each cluster. In general, Cluster 2 tends to have a lower SCE than the other groups. This is consistent with how the data was de-randomised, since we removed a proportion of the treated instances with non-white mothers. Conditioning on this kind of information is thus crucial to be able to accurately assess the impact of educational intervention on test scores. Finally, we assess the error in estimating the ACE when varying the number of mixture components in Figure 5.5. When the number of clusters is larger, the clusters get smaller and it becomes more difficult to reliably estimate the ACE since we average over the cluster members to account for partial covariate information at test time. Here, model selection is made by observing where the error in estimating the ACE stabilises (anywhere between 4-7 mixture components).

5.4.2 Sepsis Management

We illustrate the performance CEIB on the real-world task of managing and treating sepsis. Sepsis is one of the leading causes of mortality within hospitals and treating septic patients is highly challenging, since outcomes vary with interventions and there are



(a) SCE: 4.9 (b) SCE: 2.7 (c) SCE: 4.3 (d) SCE: 4.1

Figure 5.4: Illustration of the proportion of major ethnic groups within the four clusters. Grey and orange indicate de-randomised and randomised data respectively. For better visualisation, we only report the two main clusters which include the majority of all patients. The first cluster in (a) is a neutral cluster. The second cluster in (b) shows an enrichment of information in the African-American group. Clusters 3 and 4 in (c) and (d) respectively, show an enrichment of information in the White group. Overall, the clusters exhibit a distinct structure with respect to the known ethnicity confounder. Moreover, each of the clusters is associated with different SCE values. In particular, the second cluster has a lower SCE which suggests that educational intervention for these members has less of an impact on outcomes – a result consistent with our de-randomisation strategy.

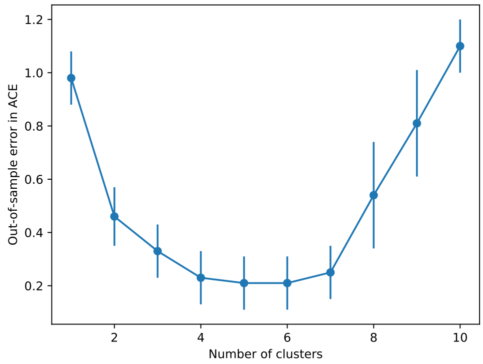
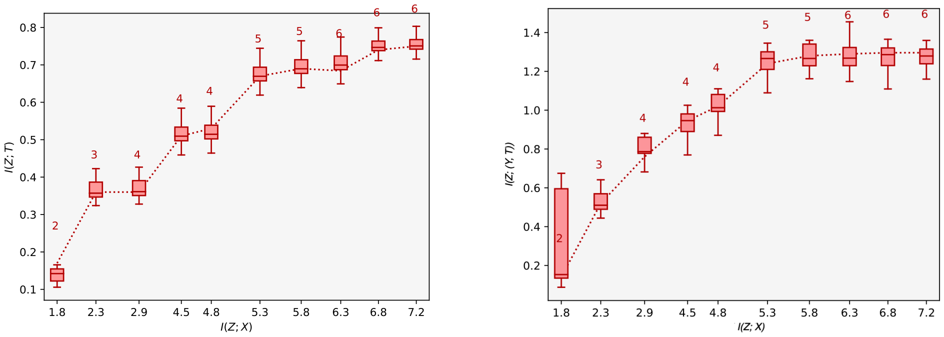


Figure 5.5: Out-of-sample error in ACE with a varying number of clusters.

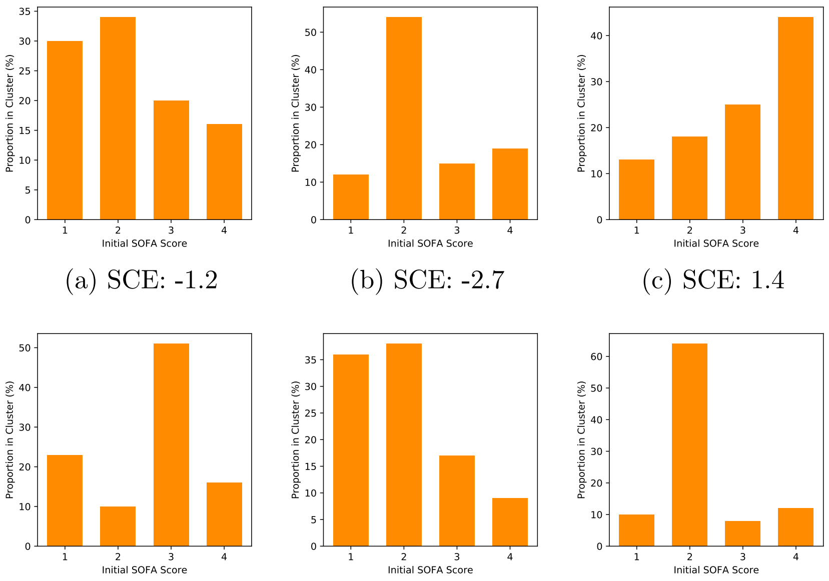
no universal treatment guidelines. For this experiment, we make use of data from the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC-III) database (Johnson et al., 2016b). We focus specifically on patients satisfying Sepsis-3 criteria (16 804 patients in total). For each patient, we have a 48-dimensional set of physiological parameters including demographics, lab values, vital signs and input/output events, where covariates are partially incomplete. We denote this set as X. Our outcomes Y correspond to the odds of mortality, while we binarise medical interventions T according to whether or not a vasopressor is administered. The data set is divided into 60/20/20% into training/validation/testing sets. We train our model with 6, 4-dimensional Gaussian mixture components and analysed the information curves and cluster compositions respectively.



(a)

(b)

Figure 5.6: Subfigures (a) and (b) illustrate the information curve $I(Z;T)$ and $I(Z;(Y,T))$ for the task of managing sepsis. We perform a sufficient reduction of the covariates to 6-dimensions and are able to approximate the ACE on the basis of this.



(a) SCE: -1.2

(b) SCE: -2.7

(c) SCE: 1.4

(d) SCE: 2.1

(e) SCE: -3.8

(f) SCE: 1.7

Figure 5.7: Proportion of initial SOFA scores in each cluster. The variation in initial SOFA scores across clusters suggests that it is a potential confounder of odds of mortality when managing and treating sepsis.

The information curves for $I(Z;T)$ and $I(Z;(Y,T))$ are shown in Figures 5.6a and 5.6b respectively. We observe that we can perform a sufficient reduction of the high-dimensional covariate information to between 4 and 6 dimensions while achieving high predictive accuracy of outcomes Y. Since there is no ground truth available for the sepsis task, we do not have access to the true confounding variables. However, we can perform an analysis on the basis of the clusters obtained over the latent space. Here, we see that we can characterise the patients in each cluster according to their initial SOFA (Sequential Organ Failure Assessment) scores. SOFA scores range between 1-

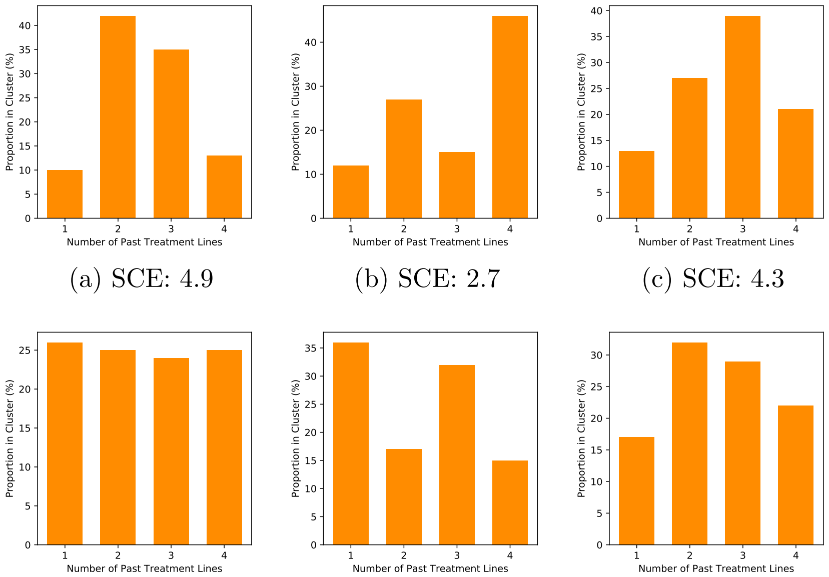
4 and are used to track a patient's stay in hospital. In Figure 5.7, we observe clear differences in cluster composition relative to the SOFA scores. Clusters 2, 5 and 6 tend to have higher proportions of patients with lower SOFA scores, while Clusters 3 and 4 have larger proportions of patients with higher SOFA scores. This result suggests that a patient's initial SOFA score is potentially a confounder when determining how to administer subsequent treatments and predicting their odds of in-hospital mortality. This is consistent with medical studies such as Medam et al. (2017); Studnek et al. (2012) where authors indicate that high initial SOFA scores were likely to impact on their overall chances of survival and treatments administered in hospital.

While we cannot quantify an error in estimating the ACE since we do not have access to the counterfactual outcomes, we can still compute the ACE for the sepsis management task. Here, we specifically observe a *negative* ACE value. This means that in general, treating patients with vasopressors reduces the chances of mortality in comparison to not treating patients with vasopressors. Overall, performing such analyses for tasks like Sepsis may shed light on what information is relevant for making predictions and reasoning about the effects of medical intervention. In turn, this may assist in establishing potential therapy guidelines for better decision-making.

5.4.3 HIV Therapy Selection

We tested the performance on the real-world application of treating HIV. For this experiment, we use data for 15 000 patients from the EuResist database (Zazzi et al., 2012). For each patient in training, we have a 94-dimensional set of parameters including blood counts, viral load, previous treatments, adherence data, and viral mutations (genotype). During testing, the genotype is missing for a fixed subset of the patients. Our outcomes Y correspond to the log viral load, while we simplified medical interventions T to whether or not a PI is administered. The data set is divided into 60/20/20% training/validation/testing sets. We train our model with six 5-dimensional Gaussian mixture components and analysed the cluster compositions.

The results of running CEIB for HIV therapy selection are shown in Figure 5.8, where we plot the proportion of patients in each cluster on the basis of their number of past treatment lines (corresponding to the number of times a therapy combination was changed, where 4+ indicates more than 4 switches). Evidently, the clusters differ in composition and SCE scores (in terms of viral load). In general, it appears as if clusters with a higher proportion of patients that frequently switched therapy combinations have lower SCE scores. This suggests that PIs have the largest impact on those patients for whom therapies have previously failed more frequently. A possible reason is that these are also patients that frequently get treated using PI-boosted therapies as a result of previous treatment failures or other reasons for therapy switching such as side-effects. In contrast, patients that have fewer past treatment lines are less likely to receive PI-boosted therapies because they are not necessarily required. Hence, the number of past treatments is likely a confounder of interventions and outcomes. Here, CEIB enables us to learn a compact, interpretable representation of this, while simultaneously accounting for its effects when estimating treatment outcomes.



(a) SCE: 4.9

(b) SCE: 2.7

(c) SCE: 4.3

(d) SCE: 3.1

(e) SCE: 4.1

(f) SCE: 3.2

Figure 5.8: Illustration of the proportion of HIV patients in each cluster according to their number of past treatment lines [1, 2, 3, 4+] where 4+ corresponds to more having than 4 previous therapy combinations. The clusters differ in composition and in the SCE scores. In general clusters with a larger proportion of individuals who received more past therapies tend to have smaller SCEs. This suggests that PIs seem to have the largest impact on these patients. Patients frequently get treated with PI-boosted therapies as a result of previous treatment failures or other reasons for therapy switching such as side-effects. In contrast, patients with fewer past treatment lines are less likely to receive PI-boosted therapies because they are unnecessary.

5.5 Discussion

CEIB learns a low-dimensional, interpretable representation of confounding Since CEIB extracts only the information that is relevant for making predictions, it is able to learn a *low-dimensional* representation of confounding, and conditions on this representation this to make predictions. In particular, the introduction of a discrete cluster structure in the latent space allows an easier interpretation of the confounding effect. For the IHDP experiment, we are able to learn a low-dimensional representation that is consistent with the known ethnicity confounder and account for its effects when making predictions of outcomes to intervention. Similar methods such as Louizos et al. (2017) typically use a higher dimensional representation (in the order of 20 dimensions) to account for these effects and make less accurate predictions nonetheless. This is potentially a consequence of learning a poor representation of confounding. Modelling the task as an IB alleviates this problem. Analogously, for the sepsis task we identify a latent space of 6 dimensions when predicting odds of mortality, where clusters exhibit a distinct structure with respect to a patient's initial SOFA score. In all three tasks, CEIB enables us to learn a meaningful, compact representation of confounding, conditioned on which we can accurately infer treatment effects without sacrificing interpretability.

CEIB enables estimating the causal effect with incomplete covariates. Unlike previous approaches, CEIB can deal with incomplete covariate data during test time by introducing a discrete latent space. Specifically, we learn equivalence classes among patients such that the approximate the effects of treatments can be computed where data is incomplete.

CEIB makes state-of-the-art predictions of the ACE that are robust against confounding Across the IHDP dataset, we see that predictions of the ACE are more accurate than existing approaches. In the IHDP case, we see reductions in the error in estimating the ACE up to 0.58 for in-sample predictions. This performance is sustained when making out-of-sample predictions we see error reductions of between 0.04 and 0.73 in comparison with existing methods. Overall, we attribute this increase in performance directly to the fact that CEIB extracts *only the information that is relevant for making predictions about interventions*. Proxy-based approaches such as Louizos et al. (2017) do not explicitly trade off learning meaningful representations of confounding and achieving accurate predictions. In contrast, we can explicitly inspect the information curves in Figure 5.3b and adjust compression parameter λ to learn a reasonable representation of confounding. If we set λ in accordance to Figure 5.3b where $I(Z;(Y,T))$ stabilises, we require only a 4-dimensional representation to adequately account for the confounding effect Z (as shown in Figure 5.4b). This produces more accurate predictions about outcomes to interventions as a result.

5.6 Conclusion

In this chapter, we focused on the problem of learning better representations of confounding in order to account for these effects in the absence of complete information at test time. This is an important problem, particularly in healthcare, since doctors must frequently reason about the effects of therapeutic interventions in the absence of complete information, primarily because of the difficulty in acquiring certain measurements such as genotypic information. To address this question, we considered the decision-theoretic perspective of causal inference. This perspective allowed us to both reason about the effects of interventions on a population level as well as on a subgroup level, thus enabling us to account for heterogeneity among patients. Specifically, we analysed the role of a sufficient covariate in the context of the IB framework to estimate the causal effect. This included introducing a discrete latent space to facilitate the knowledge transfer to cases where information was systematically missing – a task that is otherwise infeasible in high dimensions. In doing so, we could estimate the causal effect if parts of the covariates are missing during test time, while simultaneously accounting for confounding. In contrast to previous methods, the compression parameter λ in the IB framework allowed for a task-dependent adjustment of the latent dimensionality. Our extensive experiments showed that our method outperforms state-of-the-art approaches on multiple synthetic and real world datasets. Since handling systematic missingness is a highly relevant problem in healthcare, we view this as step towards improving these systems on a larger scale.

Chapter 6

Tree Regularisation For Interpretable Machine Learning

6.1 Introduction

The previous chapters presented how causal inference can be performed to estimate causal effects and hence individualise therapies as a result. In this chapter, we shift our focus to learning more interpretable models for estimating patient outcomes. In general, the lack of interpretability of modern deep learning models is a key problem to adopting them in safety critical environments such as healthcare. Specifically, we describe how deep time-series models can be trained such that we not only accurately estimate patient outcomes, but also retain explainability via decision trees with few nodes. The majority of the work in this chapter appears in Wu et al. (2018).

Here, we seek a specific form of interpretability known as *human-simulatability.* A human-simulatable model is one in which a human user can "take in input data together with the parameters of the model and in reasonable time step through every calculation required to produce a prediction" (Lipton, 2016). For example, small decision trees with only a few nodes are easy for humans to simulate and thus understand and trust. In contrast, even simple deep models like multi-layer perceptrons with a few dozen units can have far too many parameters and connections for a human to easily step through. Deep models for sequences are even more challenging. Of course, decision trees with too many nodes are also hard to simulate. The key question we address in this chapter is: can we create deep models that are well-approximated by compact, human-simulatable models? The question of creating accurate yet human-simulatable models is an important one in healthcare and medicine, for despite advances in deep learning for clinical decision support (e.g. Che et al. (2015); Choi et al. (2016); Miotto et al. (2016)), the clinical community remains skeptical of machine learning systems (Chen & Asch, 2017). Simulatability allows clinicians to audit predictions easily. They can manually inspect changes to outputs under slightly-perturbed inputs, check substeps against their expert knowledge, and identify when predictions are made due to systemic bias in the data rather than real causes.

To address this need for interpretability, a number of works have been developed to assist in the interpretation of already-trained models. Craven & Shavlik (1996) train decision trees that mimic the predictions of a fixed, pre-trained neural network, but do not train the model itself to be simpler. Other post-hoc interpretations typically

typically evaluate the sensitivity of predictions to local perturbations of inputs or the input gradient (Adler et al., 2016; Erhan et al., 2009; Lundberg & Lee, 2016; Ribeiro et al., 2016; Selvaraju et al., 2017). In parallel, research efforts have emphasised that simple lists of (perhaps locally) important features are not sufficient: Singh et al. (2016) provide explanations in the form of programs, while Lakkaraju et al. (2016) learn decision sets and show benefits over other rule-based methods. These techniques focus on understanding already learned models, rather than finding models that are more interpretable. Frequently however, deep models have multiple optima with similar predictive accuracy, so one might hope to find more interpretable models of equal predictive accuracy. However, the field of *optimising* deep models for interpretability remains nascent. Efforts such as Ross et al. (2017a) focus on penalising input sensitivity to less relevant features. This approach may expose a list of relevant features, is not necessarily sufficient to *simulate* the prediction. Alternatively, methods for model-compression (e.g. Balan et al. (2015); Han et al. (2015); Hinton et al. (2015)) try to learn smaller models that perform similarly to large, black-box models, while edge and node regularisation techniques are frequently used to improve prediction accuracy (Drucker & Le Cun, 1992; Ochiai et al., 2017). Sometimes, these regularisations—which all smooth or simplify decision boundaries—can have the effect of also improving interpretability. However, there is no guarantee that these regularisations improve interpretability.

We instead take steps toward *optimising* deep models for human-simulatability via a new model complexity penalty function we call *tree regularisation.* Tree regularisation favours models whose decision boundaries can be well-approximated by small decision-trees, thus penalising models that would require many calculations to simulate predictions. We first demonstrate how this technique can be used to train simple multi-layer perceptrons to have tree-like decision boundaries. We then focus on the time-series tasks of predicting outcomes for patients with HIV and sepsis, and show that gated recurrent unit (GRU) models trained with strong tree-regularisation reach a high-accuracy-at-low-complexity sweet spot that is not possible with any strength of L1 or L2 regularisation. Prediction quality can be further boosted by training new hybrid models – GRU-HMMs – which explain the residuals of interpretable discrete HMMs via tree-regularised GRUs. We further show that the approximate decision trees for our tree-regularised deep models are useful for human simulation and interpretability.

6.2 Background and Problem Setup

We consider supervised learning tasks given datasets of N labeled examples, where each example (indexed by n) has an input feature vector of covariates x_n and a target outcomes vector y_n. We shall assume the targets y_n are binary, though it is simple to extend to other types. When modeling time series, each example sequence n contains $\mathtt{T_n}$ time steps indexed by $\mathtt{t}$, which each have a feature vector $x_{n\mathtt{t}}$ and an output $y_{n\mathtt{t}}$. Formally, we write: $x_n = [x_{n1} \ldots x_{n\mathtt{T_n}}]$ and $y_n = [y_{n1} \ldots y_{n\mathtt{T_n}}]$. Here, each vector $x_{n\mathtt{t}}$ could be the set of vital measurements for a patient n at time $\mathtt{t}$, and the value $y_{n\mathtt{t}}$ could be the prediction about the patient's outcomes for instance, if the patient became septic at time $\mathtt{t}$. Since the focus of this chapter is on developing interpretable deep neural network models for drawing inferences in healthcare, we briefly describe neural networks and time-series modelling via recurrent neural networks in what follows.

6.2.1 Standard Neural Networks

We have already seen in Chapters 3 and 4 that we can use MLPs to make a set of predictions in a supervised learning setting. Here, an MLP allows us to predict $\hat{y}_n$ of the target y_n via a function $\hat{y}_n(x_n, W)$, where the vector W represents all parameters of the network. Given a data set $\{(x_n, y_n)\}$, the primary goal is to learn the parameters W that minimise the objective,

$$\min_W \lambda\Psi(W) + \sum_{n=1}^{N} \mathrm{loss}(y_n, \hat{y}_n(x_n, W)). \tag{6.2.1}$$

When target outcomes y_n are binary, the binary cross entropy serves as an effective choice for the loss function. The regularisation term $\Psi(W)$ can represent L1 or L2 penalties (e.g. Drucker & Le Cun (1992); Ochiai et al. (2017)) or the new regularisation approach we introduce in this chapter.

6.2.2 Recurrent Neural Networks with Gated Recurrent Units

A recurrent network (RNN) takes as input an arbitrary length sequence $x_n = [x_{n\mathtt{1}} \dots x_{n\mathtt{T_n}}]$ and produces a "hidden state" sequence $h_n = [h_{n\mathtt{1}} \dots h_{n\mathtt{T_n}}]$ of the same length as the input. Each hidden state vector at time step $\mathtt{t}$ represents a location in a (possibly low-dimensional) "state space" with K dimensions: $h_{n\mathtt{t}} \in \mathbb{R}^K$. RNNs perform sequential *nonlinear* embedding of the form $h_{n\mathtt{t}} = f(x_{n\mathtt{t}}, h_{n\mathtt{t}-1})$ in hope that the state space location $h_{n\mathtt{t}}$ is a useful summary statistic for making predictions of the target $y_{n\mathtt{t}}$ at time step $\mathtt{t}$. Crucially, this embedding is done by repeatedly applying the same transition function, thus producing the state at time $\mathtt{t}$ from both the input covariates and the previous hidden state: $h_\mathtt{t} = f(x_\mathtt{t}, h_{\mathtt{t}-1})$. Many different variants of the transition function architecture f have been proposed to solve the challenge of capturing long-term dependencies. Here, we use gated recurrent units (GRUs) (Cho et al., 2014), which are simpler than other alternatives such as long short-term memory units (LSTMs) (Hochreiter & Schmidhuber, 1997). While GRUs are convenient, any differentiable RNN architecture is compatible with the new tree-regularisation approach we introduce in this chapter.

In what follows, we describe the evolution of a single GRU sequence, dropping the sequence index n for readability. The GRU transition function f produces the state vector $h_\mathtt{t} = [h_{\mathtt{t1}} \dots h_{\mathtt{tK}}]$ from a previous state $h_{\mathtt{t}-1}$ and an input vector $x_\mathtt{t}$, via the following feed-forward architecture:

$$\begin{aligned}
\text{output state}: h_{\mathtt{tk}} &= (1 - z_{\mathtt{tk}})h_{\mathtt{t-1,k}} + z_{\mathtt{t,k}}\tilde{h}_{\mathtt{tk}} \\
\text{candidate state}: \tilde{h}_{\mathtt{tk}} &= \tanh(V_\mathtt{k}^h x_\mathtt{t} + U_\mathtt{k}^h(r_\mathtt{t} \odot h_{\mathtt{t}-1})) \\
\text{update gate}: z_{\mathtt{tk}} &= \sigma(V_\mathtt{k}^z x_\mathtt{t} + U_\mathtt{k}^z h_{\mathtt{t}-1}) \\
\text{reset gate}: r_{\mathtt{tk}} &= \sigma(V_\mathtt{k}^r x_\mathtt{t} + U_\mathtt{k}^r h_{\mathtt{t}-1})
\end{aligned} \tag{6.2.2}$$

The internal network nodes include candidate state gates $\tilde{h}$, update gates z and reset gates r which have the same cardinality as the state vector h. Reset gates allow the network to forget past state vectors when set near zero via the logistic sigmoid nonlinearity $\sigma(\cdot)$. Update gates allow the network to either pass along the previous state vector unchanged or use the new candidate state vector instead. This architecture is diagrammed in Figure 6.1.

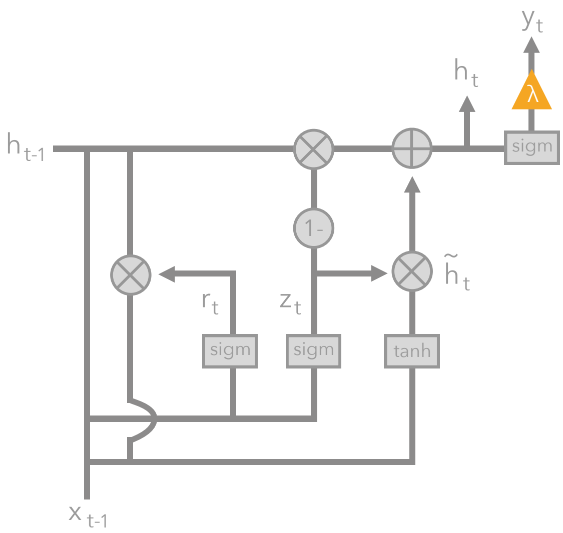


Figure 6.1: Diagram of gated recurrent unit (GRU) used for each timestep our neural time-series model. The orange triangle indicates the predicted output $\hat{y}_{\mathtt{t}}$ at time $\mathtt{t}$.

The predicted probability of the binary label $y_{\mathtt{t}}$ for time $\mathtt{t}$ is a sigmoid transformation of the state at time $\mathtt{t}$:

$$\hat{y}_{\mathtt{t}} = \sigma(w^T h_{\mathtt{t}}). \tag{6.2.4}$$

Here, weight vector $w \in \mathbb{R}^K$ represents the parameters of this output layer. We denote the parameters for the entire GRU-RNN model as $W = (w, U, V)$, concatenating all component parameters. We can subsequently train GRU-RNN time-series models (hereafter often just called GRUs) via the following loss minimisation objective:

$$\min_W \lambda \Psi(W) + \sum_{n=1}^{N} \sum_{\mathtt{t}=1}^{\mathtt{T_n}} \text{loss}(y_{n\mathtt{t}}, \hat{y}_{n\mathtt{t}}(x_n, W)), \tag{6.2.5}$$

where again $\Psi(W)$ defines a regularisation cost and $\lambda > 0$ defines the strength of regularisation.

6.3 Tree Regularisation of Deep Models

We now propose a novel *tree regularisation* function $\Omega(W)$ for the parameters of a differentiable model which attempts to penalise models whose predictions are not easily *simulatable*. Of course, it is difficult to measure "simulatability" directly for an arbitrary network, so we take inspiration from decision trees. Our chosen method has two stages: first, find a single binary decision tree which accurately reproduces the network's thresholded binary predictions $\hat{y}_n$ given input x_n. Second, measure the complexity of this decision tree as the output of $\Omega(W)$. We measure complexity as the *average decision path length*—the average number of decision nodes that must be touched to make a prediction for an input example x_n. We compute the *average* with respect to some designated reference dataset of example inputs $D = \{x_n\}$ from the training set. While many ways to measure complexity exist, we find average path length is most relevant to our notion of *simulatability*. Remember that for us, human simulation requires stepping

Algorithm 2 Average-Path-Length Cost Function

```
Require:
    $\hat{y}(\cdot, W)$ : binary prediction function, with parameters $W$
    $D = \{x_n\}_{n=1}^{N}$ : reference dataset with $N$ examples
function $\Omega(W)$
    tree ← TrainTree($\{x_n, \hat{y}(x_n, W)\}$)
    return $\frac{1}{N} \sum_n$ PathLength(tree, $x_n$)
```

through every calculation required to make a prediction. Average path length exactly counts the number of true-or-false boolean calculations needed to make an average prediction, assuming the model is a decision tree. Total number of nodes could be used as a metric, but might penalise more accurate trees that have short paths for most examples but need more involved logic for few outliers.

Our true-average-path-length cost function $\Omega(W)$ is detailed in Algorithm 2. It requires two subroutines, TrainTree and PathLength. TrainTree trains a binary decision tree to accurately reproduce the provided labeled examples $\{x_n, \hat{y}_n\}$. We use the `DecisionTree` module distributed in Python's scikit-learn (Pedregosa et al., 2011) with post-pruning to simplify the tree. These trees can give probabilistic predictions at each leaf.[1] Next, PathLength counts how many nodes are needed to make a specific input to an output node in the provided decision tree. In our evaluations, we will apply our average-decision-tree-path-length regularisation, or simply "tree regularisation," to several neural models. Algorithm 2 defines our average-path-length cost function $\Omega(W)$, which can be plugged into the abstract regularisation term $\Psi(W)$ in the objectives in Equations 6.2.1 and 6.2.5.

6.3.1 Making the Decision-Tree Loss Differentiable

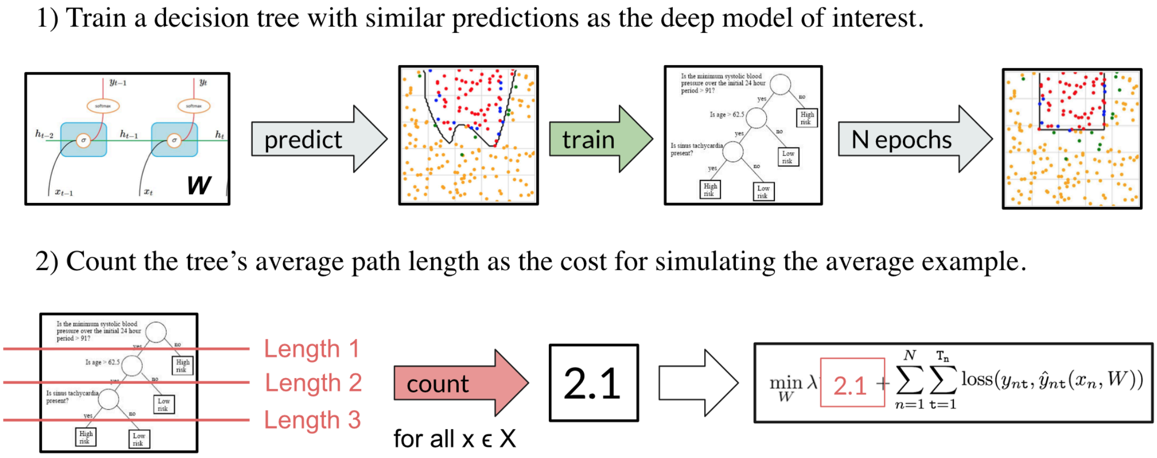
Training decision trees is not differentiable, and thus $\Omega(W)$ as defined in Algorithm 2 is not differentiable with respect to the network parameters W (unlike standard regularisers such as the L1 or L2 norm). While one could resort to derivative-free optimisation techniques (Audet & Kokkolaras, 2016), gradient descent has been an extremely fast and robust way of training networks.

A key technical contribution of our work is introducing and training a *surrogate* regularisation function $\hat{\Omega}(W) : \text{supp}(W) \rightarrow \mathbb{R}_+$ to map each candidate neural model parameter vector W to an *estimate* of the average-path-length. Our approximate function $\hat{\Omega}$ is implemented as a standalone multi-layer perceptron network and is thus *differentiable.* Let vector ξ of size k denote the parameters of this chosen MLP approximator. We can train $\hat{\Omega}$ to be a good estimator by minimising a squared error loss function:

$$\min_{\xi} \textstyle\sum_{j=1}^{J} (\Omega(W_j) - \hat{\Omega}(W_j, \xi))^2 + \epsilon ||\xi||_2^2, \tag{6.3.1}$$

where W_j are the *entire* set of parameters for our model, $\epsilon > 0$ is a regularisation strength, and we assume we have a dataset of J known parameter vectors and their associated true path-lengths: $\{W_j, \Omega(W_j)\}_{j=1}^{J}$. This dataset can be assembled using the candidate W vectors obtained while training our target neural model $\hat{y}(\cdot, W)$, as

[1] Complete decision-tree training details are provided in the appendix.



1) Train a decision tree with similar predictions as the deep model of interest.

2) Count the tree's average path length as the cost for simulating the average example.

W
predict
train
N epochs
Is the minimum systolic blood pressure over the initial 24 hour period > 91?
Is age > 62.5
Is sinus tachycardia present?
yes
no
High risk
Low risk
Length 1
Length 2
Length 3
count
for all x ϵ X
2.1
min λ 2.1 + Σ Σ loss(y_nt, ŷ_nt(x_n, W))

Figure 6.2: Overview of tree regularisation procedure.

well as by evaluating $\Omega(W)$ for randomly generated W. Importantly, one can train the surrogate function $\hat{\Omega}$ in parallel with our network. Experimentally, we show evidence that our surrogate predictor $\hat{\Omega}(\cdot)$ tracks the true average path length as we train the target predictor $\hat{y}(\cdot, W)$ (See appendix for details).

6.3.2 Training the Surrogate Loss

Even moderately-sized GRUs can have parameter vectors W with thousands of dimensions. Our labeled dataset for surrogate training – $\{W_j, \Omega(W_j)\}_{j=1}^{J}$—will only have one W_j example from each target network training iteration. Thus, in early iterations, we will have only few examples from which to learn a good surrogate function $\hat{\Omega}(W)$. We resolve this challenge via *augmenting* our training set with additional examples: We randomly sample weight vectors W and calculate the true average path length $\Omega(W)$, and we also perform several random restarts on the unregularised GRU and use those weights in our training set.

A second challenge occurs later in training: as the model parameters W shift away from their initial values, those early parameters may not be as relevant in characterising the current decision function of the GRU. To address this, for each epoch, we use examples only from the past E epochs (in addition to augmentation), where in practice, E is empirically chosen. Using examples from a fixed window of epochs also speeds up training. The appendix shows a comparison of the importance of these heuristics for efficient and accurate training—empirically, data augmentation for stabilising surrogate training allows us to scale to GRUs with 100s of nodes. GRUs of this size are sufficient for many real problems, such as those we encounter in healthcare domains.

Typically, we use $J = 50$ labeled pairs for surrogate training for toy datasets and $J = 100$ for real world datasets in our experiments. Optimisation of our surrogate objective is done via gradient descent. We use Autograd to compute gradients of the loss in Equation. 6.3.1 with respect to ξ, then use Adam to compute descent directions with step sizes set to 0.01 for toy datasets and 0.001 for healthcare datasets.

An overview of the overall training procedure is provided in Figure 6.2. Importantly, the second step of approximating the true average path length is performed via the surrogate network described in this section. The overall procedure may thus be summarised as follows: first a tree is trained to mimic a deep model's predictions. Next the cost of simulating the average example is computed as the average path length. The surrogate MLP is subsequently used to approximate the tree's predicted path length. Given a fixed surrogate MLP, the model parameters W may be optimised via gradient descent. Subsequently, given a fixed set of model parameters W, we can find the best surrogate MLP. The overall training procedure alternates between these two stages successively.

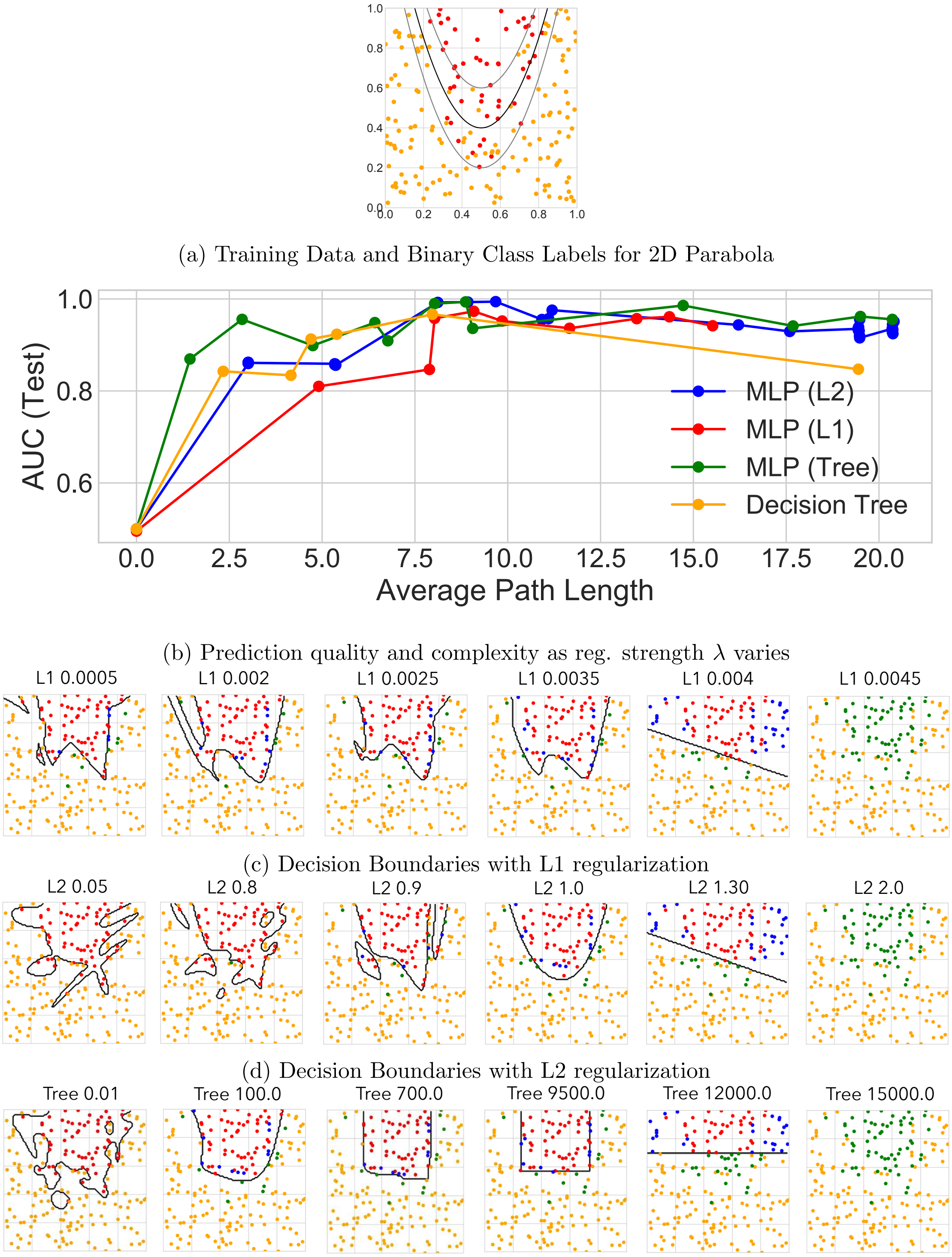
6.4 Tree-Regularised MLPs: A Demonstration

While time-series models are the main focus of this work, we first demonstrate tree regularisation on a simple binary classification task to build intuition. We call this task the 2D Parabola problem, because as Figure 6.3(a) shows, the training data consists of 2D input points whose two-class decision boundary is roughly shaped like a parabola. The true decision function is defined by $y = 5 * (x - 0.5)^2 + 0.4$. We sampled 500 input points x_n uniformly within the unit square $[0, 1] \times [0, 1]$ and labeled those above the decision function as positive. To make it easy for models to overfit, we flipped 10% of the points in a region near the boundary. A random 30% were held out for testing.

For the classifier $\hat{y}$, we train a 3-layer MLP with 100 first layer nodes, 100 second layer nodes, and 10 third layer nodes. This MLP is intentionally overly expressive to encourage overfitting and expose the impact of different forms of regularisation: our proposed tree regularisation $\Psi(W) = \hat{\Omega}(W)$ and two baselines: an L2 penalty on the weights $\Psi(W) = ||W||_2$, and an L1 penalty on the weights $\Psi(W) = ||W||_1$. For each regularisation function, we train models at many different regularisation strengths λ chosen to explore the full range of decision boundary complexities possible under each technique. For our tree regularisation, we model our surrogate $\hat{\Omega}(W)$ with a 1-hidden layer MLP with 25 units. We find this simple architecture works well, but certainly more complex MLPs could could be used on more complex problems. The objective in Equation 6.2.1 was optimised via Adam gradient descent (Kingma & Ba, 2014) using a batch size of 100 and a learning rate of 1e-3 for 250 epochs, and hyperparameters were set via cross validation using grid search (see appendix for full experimental details).

Figure 6.3 (b) shows the each trained model as a single point in a 2D fitness space: the x-axis measures model complexity via our average-path-length metric, and the y-axis measures AUC prediction performance. These results show that simple L1 or L2 regularisation does *not* produce models with both small node count and good predictions at *any* value of the regularisation strength λ. As expected, large λ values for L1 and L2 only produce far-too-simple linear decision boundaries with poor accuracies. In contrast, our proposed tree regularisation directly optimizes the MLP to have simple tree-like boundaries at high λ values which can still yield good predictions.

The lower panes of Figure 6.3 shows these boundaries. Our tree regularisation is uniquely able to create axis-aligned functions, because decision trees prefer functions that are axis-aligned splits. These axis-aligned functions require very few nodes but are more effective than L1 and L2 counterparts. The L1 boundary is more sharp, whereas the L2 is more round.



(a) Training Data and Binary Class Labels for 2D Parabola

(b) Prediction quality and complexity as reg. strength λ varies

(c) Decision Boundaries with L1 regularization

(d) Decision Boundaries with L2 regularization

(e) Decision Boundaries Tree regularization

Figure 6.3: *2D Parabola task*: (a) Each training data point in 2D space, overlaid with true parabolic class boundary. (b): Each method's prediction quality (AUC) and complexity (path length) metrics, across range of regularisation strength λ. In the small path length regime between 0 and 5, tree regularisation produces models with higher AUC than L1 or L2. (c-e): Decision boundaries (black lines) have qualitatively different shapes for different regularisation schemes, as regularisation strength λ increases. We colour predictions as true positive (red), true negative (yellow), false negative (green), and false positive (blue).

6.5 Tree-Regularised Time-Series Models

We now evaluate our tree-regularisation approach on time-series models. We focus on GRU-RNN models, with some later experiments on new hybrid GRU-HMM models. As with the MLP, each regularisation technique (tree, L2, L1) can be applied to the output node of the GRU across a range of strength parameters λ. Importantly, Algorithm 2 can compute the average-decision-tree-path-length for any fixed deep model given its parameters, and can hence be used to measure decision boundary complexity under any regularisation, including L1 or L2. This means that when training any model, we can track both the predictive performance (as measured by area-under-the-ROC-curve (AUC); higher values mean better predictions), as well as the complexity of the decision tree required to explain each model (as measured by our average path length metric; lower values mean more interpretable models). We also show results for a baseline standalone decision tree classifier without any associated deep model, sweeping a range of parameters controlling leaf size to explore how this baseline trades off path length and prediction quality. Further details of our experimental protocol, as well as more extensive results with additional baselines are provided in the appendix.

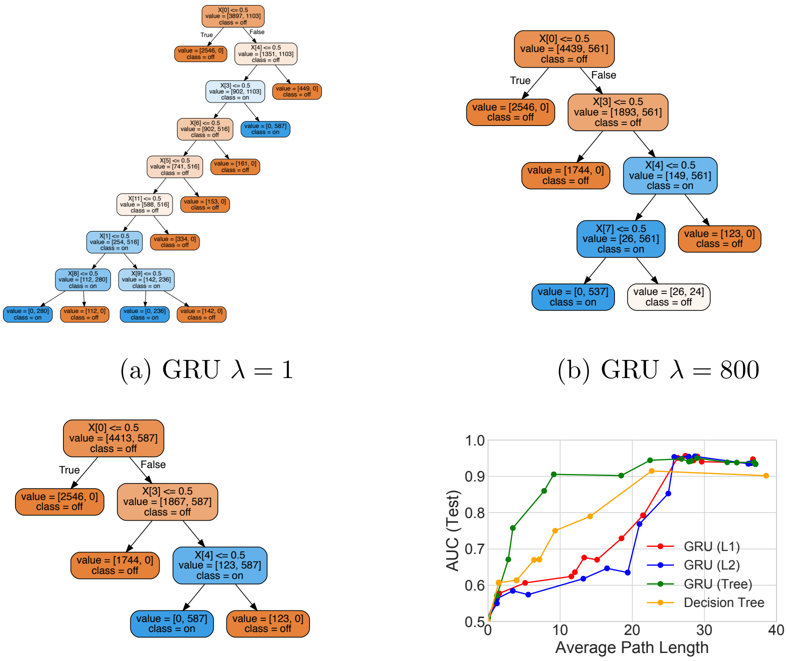
6.5.1 Synthetic Task: Signal-and-noise HMM

We generated a toy dataset of $N = 100$ sequences, each with 50 time steps. Each time step has a data vector $x_{n\texttt{t}}$ of 14 binary features and a single binary output label $y_{n\texttt{t}}$. The data comes from two separate HMM processes. First, a "signal" HMM generates the first 7 data dimensions from 5 well-separated states. Second, an independent "noise" HMM generates the remaining 7 data dimensions from a different set of 5 states. Each timestep's output label $y_{n\texttt{t}}$ is produced by a rule involving *both* the signal data and the signal hidden state: the target is 1 at time step $\texttt{t}$ only if both the first signal state is active and the first observation is turned on. We deliberately designed the generation process so that neither logistic regression with x as features nor an RNN model that makes predictions from hidden states alone can perfectly separate this data. These results are shown in Figure 6.4.

6.5.2 Real-World Tasks: Predicting Patient Outcomes for HIV and Sepsis

We tested our approach on several real tasks: predicting medical outcomes of hospitalised septic patients, predicting HIV therapy outcomes, as well as the task of identifying stop phonemes in English speech recordings. To normalise scales, we independently standardised features x via z-scoring.

- Sepsis Critical Care: We study time-series data for 11 786 septic ICU patients from the public MIMIC III dataset (Johnson et al., 2016a). We observe at each hour $\texttt{t}$ a data vector $x_{n\texttt{t}}$ of 35 vital signs and lab results as well as a label vector $y_{n\texttt{t}}$ of 5 binary outcomes. Hourly data $x_{n\texttt{t}}$ measures continuous features such as respiration rate (RR), blood oxygen levels (paO$_2$), fluid levels, and more. Hourly binary labels $y_{n\texttt{t}}$ include whether the patient died in hospital and if mechanical ventilation was applied. Models are trained to predict all 5 output dimensions concurrently from one shared embedding. The average sequence length is 15 hours. 7 070 patients are used in training, 1 769 for validation, and 294 for test.



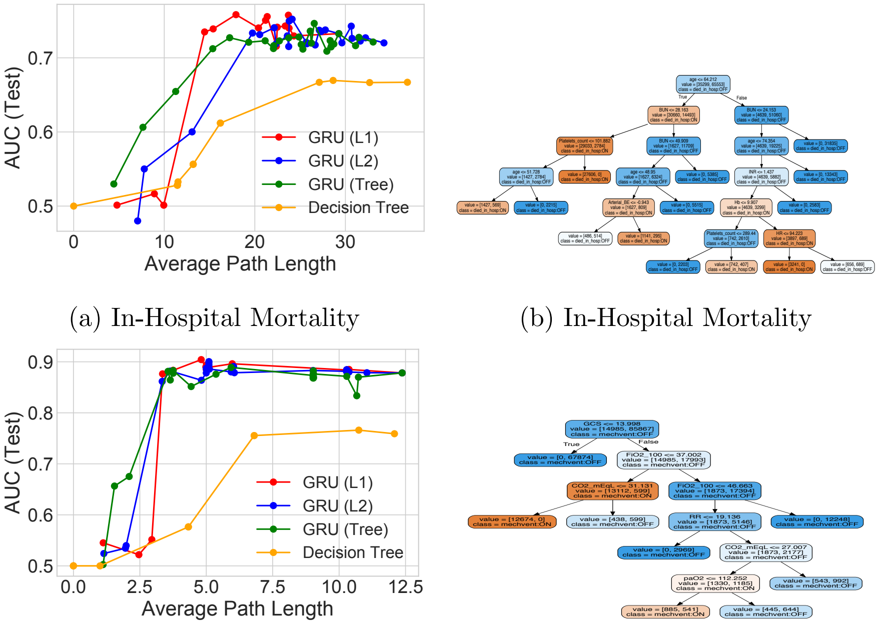
(a) GRU $\lambda = 1$

(b) GRU $\lambda = 800$

(c) GRU $\lambda = 1\,000$

(d) GRU

Figure 6.4: *Toy Signal-and-Noise HMM Task:* (a)-(c) Decision trees trained to mimic predictions of GRU models with 25 hidden states at different regularisation strengths λ; as expected, increasing λ decreases the size of the learned trees (see supplement for more trees). Decision tree (c) suggests the model learns to predict positive output (blue) if and only if "$x[0] == 1$ and $x[3] == 1$ and $x[4] == 0$", which is consistent with the true rule we used to generate labels: assign positive label only if first dimension is on ($x[0] == 1$) and first state is active (emission probabilities for this state: [.5 .5 .5 .5 0 ...]). (d) Tree-regularised GRU models reach a sweet spot of small path lengths yet high AUC predictions that alternatives cannot reach at any tested value of λ.



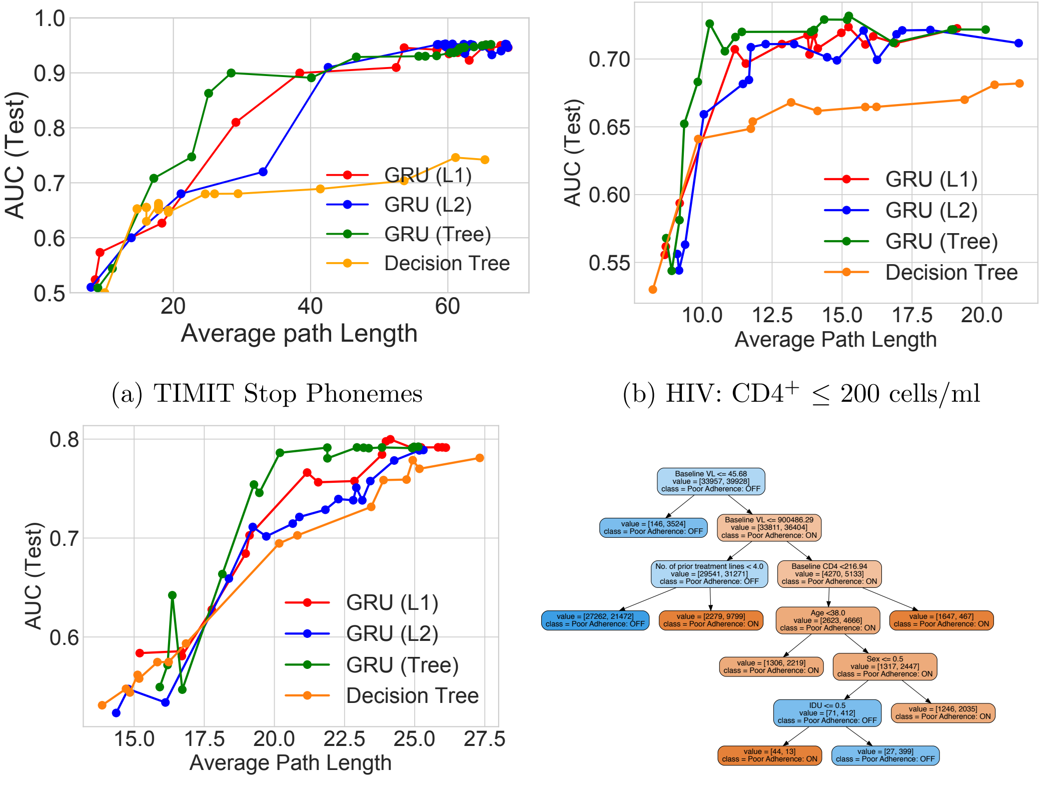
(a) In-Hospital Mortality

(b) In-Hospital Mortality

(c) Mechanical Ventilation

(d) Mechanical Ventilation

Figure 6.5: *Sepsis task:* Study of different regularisations for GRU model with 100 states, trained to jointly predict 5 binary outcomes for ICU patients. Panels (a) and (c) show AUC vs. path length for 2 of the 5 outcomes (remainder in the supplement); in both cases, tree-regularisation provides higher AUC in the target regime of low-complexity decision trees. Panels (b) and (d) show proxy trees for the tree-regularised GRU ($\lambda = 2\,000$); these were found interpretable by an ICU clinician (see main text).



(a) TIMIT Stop Phonemes

(b) HIV: CD4$^+$ $\leq$ 200 cells/ml

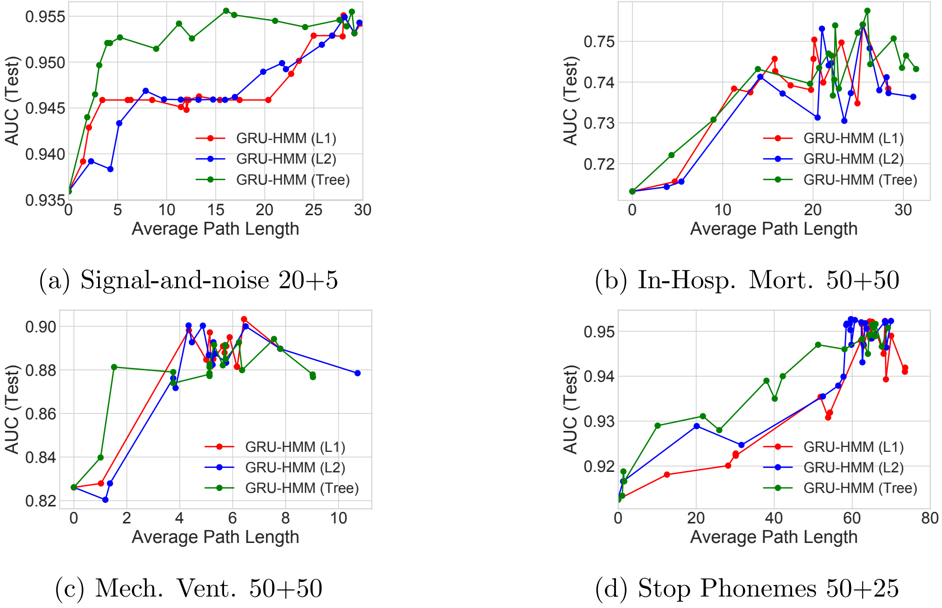
(c) HIV Therapy Adherence

(d) HIV Therapy Adherence

Figure 6.6: *TIMIT and HIV tasks:* Study of different regularisation techniques for GRU model with 75 states. Panels (a)-(c) are tradeoff curves showing how AUC predictive power and decision-tree complexity evolve with increasing regularisation strength under L1, L2 or tree regularisation on both TIMIT and HIV tasks. The GRU is trained to jointly predict 15 binary outcomes for HIV, of which 2 are shown here in Panels (b) - (c). The GRU's decision tree proxy for HIV Adherence is shown in (d).

- HIV Therapy Outcome (HIV): We use the EuResist Integrated Database (Zazzi et al., 2012) for 53 236 patients diagnosed with HIV. We consider 4-6 month intervals (corresponding to hospital visits) as time steps. Each data vector x_{nt} has 40 features, including blood counts, viral load measurements and lab results. Each output vector y_{nt} has 15 binary labels, including whether a therapy was successful in reducing viral load to below detection limits, if therapy caused CD4 blood cell counts to drop to dangerous levels (indicating AIDS), or if the patient suffered adherence issues to medication. The average sequence length is 14 steps. 37 618 patients are used for training; 7 986 for testing, and 7 632 for validation.

- Phonetic Speech (TIMIT): We have recordings of 630 speakers of eight major dialects of American English reading ten phonetically rich sentences (Garofolo et al., 1993). Each sentence contains time-aligned transcriptions of 60 phonemes. We focus on distinguishing stop phonemes (those that stop the flow of air, such as “b” or “g”) from non-stops. Each time step has one binary label y_{nt} indicating if a stop phoneme occurs or not. Each input x_{nt} has 26 continuous features:



(a) Signal-and-noise 20+5

(b) In-Hosp. Mort. 50+50

(c) Mech. Vent. 50+50

(d) Stop Phonemes 50+25

Figure 6.7: Prediction quality (AUC) vs. complexity (path length) for the GRU-HMM over a range of regularisation strengths λ. Subtitles show the number of HMM states and GRU states. See earlier figures to compare these GRU-HMM numbers to simpler GRU and decision tree baselines.

the acoustic signal's Mel-frequency cepstral coefficients and derivatives. There are 6 303 sequences, split into 3 697 for training, 925 for validation, and 1 681 for testing. The average length is 614.

6.5.3 Results

The major conclusions of our experiments comparing GRUs with various regularisations are outlined below.

Tree-regularised models have fewer nodes than other forms of regularisation. Across tasks, we see that in the target regime of small decision trees (low average-path lengths), our proposed tree-regularisation achieves higher prediction quality (higher AUCs). In the signal-and-noise HMM task, tree regularisation (green line in Figure 6.4(d)) achieves AUC values near 0.9 when its trees have an average path length of 10. Similar models with L1 or L2 regularisation reach this AUC only with trees that are nearly double in complexity (path length over 25). On the Sepsis task (Figure 6.5) we see AUC gains of 0.05-0.1 at path lengths of 2-10. On the TIMIT task (Figure 6.6a), we see AUC gains of 0.05-0.1 at path lengths of 20-30. Finally, on the HIV CD4 blood cell count task in Figure 6.6b, we see AUC differences of between 0.03 and 0.15 for path lengths of 10-15. The HIV adherence task in Figure 6.6d has AUC gains of between 0.03 and 0.05 in the path length range of 19 to 25 while at smaller paths all methods are quite poor, indicating the problem's difficulty. Overall, these AUC gains are particularly useful in determining how to administer subsequent HIV therapies.

We emphasise that our tree-regularisation usually achieves a sweet spot of high

AUCs at short path lengths not possible with standalone decision trees (orange lines), L1-regularised deep models (red lines) or L2-regularised deep models (blue lines). In unshown experiments, we also tested elastic net regularisation (Zou & Hastie, 2005), a linear combination of L1 and L2 penalties. We found elastic nets to follow the same trend lines as L1 and L2, with no visible differences. In domains where human-simulatability is required, increases in prediction accuracy in the small-complexity regime can mean the difference between models that provide value on a task and models that are unusable, either because performance is too poor or predictions are uninterpretable.

Our learned decision tree proxies are interpretable. Across all tasks, the decision trees which mimic the predictions of tree-regularised deep models are small enough to simulate by hand (path length $\leq$ 25) and help users grasp the model's nonlinear prediction logic. Intuitively, the trees for our synthetic task in Figure 6.4(a)-(c) decrease in size as the strength λ increases. The logic of these trees also matches the true labelling process: even the simplest tree (c) checks a relevant subset of input dimensions necessary to verify that both the first state and the first output dimension are active.

In Figure 6.5, we show decision tree proxies for our deep models on two sepsis prediction tasks: mortality and need for ventilation. We consulted a clinical expert on sepsis treatment, who noted that the trees helped him understand what the models might be doing and thus determine if he would trust the deep model. For example, he said that using FiO_2, RR, CO_2 and paO_2 to predict need for mechanical ventilation (Figure 6.5d) was sensible, as these all measure breathing quality. In contrast, the in-hospital mortality tree (Figure 6.5b) predicts that some young patients with no organ failure have high mortality rates while other young patients with organ failure have low mortality. These counter-intuitive results led to hypotheses about how uncaptured variables impact the training process. Such reasoning would not be possible from simple sensitivity analyses of the deep model.

Finally, we have verified that the decision tree proxies of our tree-regularised deep models of the HIV task in Figure 6.6d are interpretable for understanding why a patient has trouble adhering to a prescription; that is, taking drugs regularly as directed. Our clinical collaborators confirm that the baseline viral load and number of prior treatment lines, which are prominent attributes for the decisions in Figure 6.6d, are useful predictors of a patient with adherence issues. Several medical studies (Langford et al., 2007; Socias et al., 2011) suggest that patients with higher baseline viral loads tend to have faster disease progression, and hence have to take several drug cocktails to combat resistance. Juggling many drugs typically makes it difficult for these patients to adhere as directed. We hope interpretable predictive models for adherence could help assess a patient's overall prognosis (Paterson et al., 2000) and offer opportunities for intervention (e.g. with alternative single-tablet regimens).

Decision trees trained to mimic deep models make faithful predictions. Across datasets, we find that each tree-regularised deep time-series model has predictions that agree with its corresponding decision tree proxy in about 85-90% of test examples. Table 1 shows exact fidelity scores for each dataset. Thus, the simulatable paths of the decision tree will be trustworthy in a majority of cases.

Dataset	**Fidelity**
signal-and-noise HMM	0.88
SEPSIS (In-Hospital Mortality)	0.81
SEPSIS (90-Day Mortality)	0.88
SEPSIS (Mech. Vent.)	0.90
SEPSIS (Median Vaso.)	0.92
SEPSIS (Max Vaso.)	0.93
HIV (CD4$^+$ below 200)	0.84
HIV (Therapy Success)	0.88
HIV (Mortality)	0.93
HIV (Poor Adherence)	0.90
HIV (AIDS Onset)	0.93
TIMIT	0.85

Table 6.1: Fidelity of predictions from our trained deep GRU-RNN and its corresponding decision tree. Fidelity is defined as the percentage of test examples on which the prediction made by a tree agrees with the deep model (Craven & Shavlik, 1996). We used 20 hidden GRU states for signal-and-noise task, 50 states for all others.

Practical runtimes for tree regularisation are less than twice that of simpler L2. While our tree-regularised GRU with 10 states takes 3977 seconds per epoch on TIMIT, a similar L2-regularised GRU takes 2116 seconds per epoch. Thus, our new method has cost less than twice the baseline *even when the surrogate is serially computed.* Because the surrogate $\hat{\Omega}(W)$ will in general be a much smaller model than the predictor $\hat{y}(x, W)$, we expect one could get faster per-epoch times by parallelising the creation of $(W, \Omega(W))$ training pairs and the training of the surrogate $\hat{\Omega}(W)$. Additionally, 3977 seconds includes the time needed to train the surrogate. In practice, we do this sparingly, only once every 25 epochs, yielding an amortised per-epoch cost of 2191 seconds (more runtime results are in the appendix).

Decision trees are stable over multiple optimisation runs. When tree regularisation is strong (high λ), the decision trees trained to match the predictions of deep models are stable. For both signal-and-noise and sepsis tasks, multiple runs from different random restarts have nearly identical tree shape and size, perhaps differing by a few nodes. This stability is crucial to building trust in our method. On the signal-and-noise task ($\lambda = 7000$), 7 of 10 independent runs with random initialisations resulted in trees of exactly the same structure, and the others closely resembled those sharing the same subtrees and features (more details in the appendix).

The deep residual GRU-HMM achieves high AUC with less complexity. So far, we have focused on regularising standard deep models, such as MLPs or GRUs. Another option is to use a deep model as a residual on another model that is already interpretable: for example, discrete HMMs partition time steps into clusters, each of which can be inspected, but its predictions might have limited accuracy. In Figure 6.7, we show the performance of jointly training a *GRU-HMM*, a new model which combines an HMM with a tree-regularised GRU to improve its predictions (details and further results in the appendix). Here, the ideal path length is zero, indicating only the HMM makes predictions. For small average-path-lengths, the GRU-HMM improves the original HMM's predictions *and* has simulatability gains over earlier GRUs. On the mechanical ventilation task, the GRU-HMM requires an average path length of only 28 to reach AUC of 0.88, while the GRU alone with the same number of states requires a path length of 60 to reach the same AUC. This suggests that jointly-trained deep residual models may provide even better interpretability.

6.6 Conclusion

We have introduced a novel tree-regularisation technique that encourages the complex decision boundaries of any differentiable model to be well-approximated by human-simulatable functions, allowing domain experts to quickly understand and approximately *compute* what the more complex model is doing. In general, our training procedure is robust and efficient; across three complex, real-world domains – HIV treatment, sepsis treatment, and human speech processing – our tree-regularised models provide gains in prediction accuracy in the regime of simpler, approximately human-simulatable models. Overall, our tree-regularisation method can be viewed as a step towards building explainable models in healthcare that we can trust.

Chapter 7

Conclusion

The fundamental question we addressed in this thesis is: *What is the effect of a therapeutic intervention on a patient?* We studied this question in the context of three broadly related themes namely causal inference, decision-making and interpretability: each of the methods we proposed may be viewed as a means of identifying a suitable representation of confounding in order to infer treatment effects or as a means of gaining interpretability for decision-making. We summarise our contributions below.

Learning suitable representations of measured confounding. Our work in Chapter 3 showed how to combine non-parametric methods with parametric methods that explicitly build a causal model for learning a treatment policy. Both of these methods used different representations to capture measured confounding, based on which we could subsequently learn a treatment policy. In the parametric approach, this information was summarised in terms of a belief state representation, that we conditioned on to infer treatment effects to estimate a policy; the non-parametric approach used a kernel-similarity score to perform matching for this purpose. In Chapter 4, we adapted this approach to combine both representations into a modified belief state representation such that we could forward simulate counterfactuals to deduce treatment effects. The information bottleneck approach in Chapter 5, provided an alternative, information-theoretic perspective to this problem: we used the idea of the relevance of information to perform a sufficient reduction of the measured covariates and learn a constrained representation of confounding; conditioned on this reduced representation of confounding, we could subsequently infer treatment effects.

Interpretable medical decision-making. Both mixture model approaches in Chapters 3 and 4 could be used to assess the efficacy of treatment policies. Specifically, our model presented in Chapter 4 used online planning to reason about what would happen if we actively intervened at a time point in a particular way. By forward simulating the potential scenarios of performing certain interventions over a particular horizon length, we were able to step through predictions about how a patient's particular state of health may evolve and what outcomes can be observed at each particular time point. In high-stake domains such as medicine, this simulatability can guide or be used to audit treatment recommendations, and provide a complementary context to a simple set of guidelines or recommendations. In Chapter 5, we explicitly learned an interpretable, low-dimensional disentangled representation of confounding using the information bottleneck principle. By adjusting the degree of compression in the bottleneck, we could explore a range of such representations from which could sample and infer treatment effects. Finally, in Chapter 6, we explicitly developed a regularisation mechanism that encouraged the decision boundaries of deep models to be approximated by small decision trees that could be stepped through, allowing domain experts to understand why a particular model makes its predictions.

7.1 Limitations and Future Work

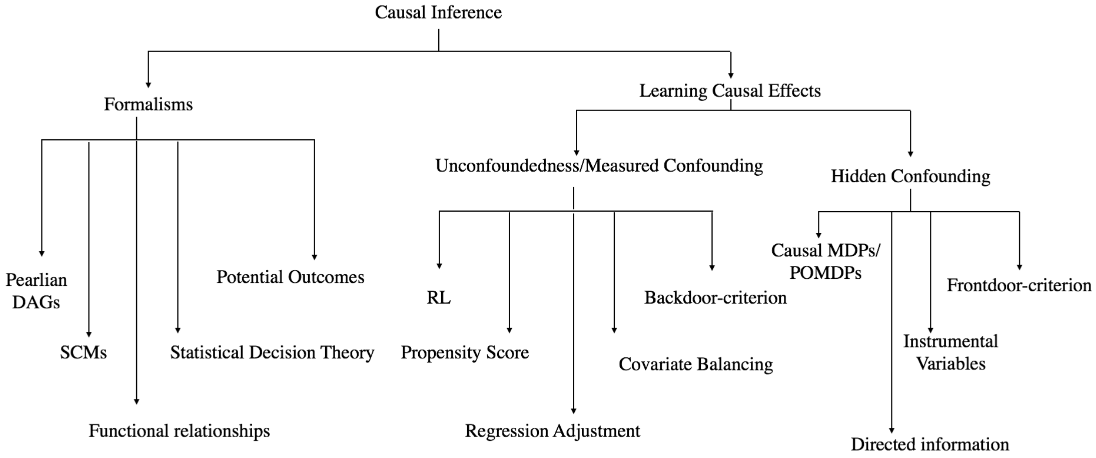


Figure 7.1: Summary of causal inference frameworks and methods for estimating causal effects under measured and hidden confounding. The work in our thesis only addressed measured confounding. Possible extensions to scenarios with hidden confounding include the use of directed information in our RL models and instrumental variables.

The Limitations of Off-Policy Evaluation. In Chapters 3 and 4 of this thesis, we presented two mixture-of-expert approaches for both therapy selection and counterfactual reasoning. Both of these approaches were evaluated quantitatively using the off-policy evaluation strategies described in Chapter 2 as well as qualitatively in terms of medical guidelines. As we have already discussed previously, the results of off-policy evaluation are dependent on the degree of overlap between the clinical data policy and the learnt policy. In both these chapters we demonstrated that there is indeed an adequate overlap in this the case by examining the distribution of non-zero importance weights when performing off-policy evaluation. However, when applying these methods of evaluation to other clinical settings, this may not necessarily be the case. These results can also be sensitive to the way in which the behaviour policy is computed. As a result, a certain degree of caution needs to be taken when performing off-policy evaluation. One way of overcoming this issue would be to use the proposed mixture-of-experts framework not only as a tool for estimating a suitable treatment policy, but also as a means of evaluating existing policies. That is, one could combine parametric and non-parametric estimators to estimate the value of a particular policy, such that the error of this estimate is minimised.

Exploring Alternative Back-Off Strategies. Both of the models in Chapters 3 and 4 combine parametric and non-parametric approaches to infer treatment effects and produce better treatment policies. Future work could explore alternative ways to design the back-off strategy from kernel to model-based methods and the connections between the regularisation afforded by non-parametric dynamical system models such as kPOMDPs or PSRs. The goal of such an extension could be to develop a more accurate method for off-policy evaluation, that is robust to the choice of representation.

Accounting for the Influence of Hidden or Unmeasured Confounding. An important assumption we made across all our models was the fact that confounding factors may be measured. In reality of course, there will however, be many confounders that are unmeasured or that we do not necessarily know about, which together influence the predictions we can make about a patient's outcomes to treatment. Accounting for such confounding however requires additional assumptions and tests. In particular, Pearl (2009) introduces the front-door criterion as a general test for identifying causal effects with hidden confounding. Equivalently, in an RL setting it no longer suffices to use a simple MDP or POMDP structure as a causal model, since in these cases unobserved confounders will influence the actions, observations and rewards. Many approaches have been proposed to deal with unobserved confounding in a non-RL setting. However, these techniques require strict mathematical assumptions, and in practice we do not always know whether data will meet these assumptions. As an alternative, it may be helpful to consider the notion of *directed information* when calculating the value functions of a particular policy. The overall idea here would be to rewrite the transition, observation and reward functions of a POMDP in terms of directed information where we explicitly *causally condition* on previous states and actions. In a similar vein, it may be interesting to adapt the information bottleneck method that we developed in Chapter 5 to use the concept of directed information too when learning a low-dimensional representation of confounding.

Another interesting extension of the model in Chapter 5, considers using instrumental variables for which treatment is never applied, to estimate the average causal effect defined only on the subpopulation of patients that are treated. Theoretically, this quantity should not depend on learning a sufficiently reduced covariate, and one could thus use it to verify theoretically whether the model in Chapter 5 learns an adequate representation of confounding. In general, instrumental variables are typically employed in scenarios where hidden confounding is prevalent and could thus also provide a suitable means of extending the model to such a case. A general summary of methods that are applicable for estimating causal effects with unmeasured confounding is shown in Figure 7.1.

Extensions of tree regularisation for local explainability, and to domains that are not prima facie interpretable. Finally our work in Chapter 6 focused on using tree-regularisation to learn a model that can be interpreted and explained globally in terms of a single decision tree (per gradient step). However, in healthcare particularly, it may be of relevance to apply tree regularisation to local, example-specific approximations of a loss (Ribeiro et al., 2016) or to representation learning tasks (encouraging embeddings with simple boundaries). We could also continue to explore ways in which to improve the stability of such models and identify ways to apply this approach to situations where the inputs are not inherently interpretable e.g. for medical image analysis.

Appendix A

Details for Policy Mixing Models

A.1 The History Alignment Kernel

The history alignment kernel first constructs a *resistance mutations kernel* to quantify the pairwise similarities between different therapy combinations. Formally, the kernel may be defined as follows. Let denote the set of different drug groups, and $u_{a\xi}$ and $u_{a'\xi}$ be the sets of resistance-relevant mutations for the drugs occurring in drug group $\xi \in \Xi$ of the therapies a and a', respectively. The pairwise similarity between the drug-ξ mutations of the drug combinations a and a' is then calculated using the Jaccard index:

$$sim_{\xi}(a, a') = \frac{|u_{a\xi} \bigcap u_{a'\xi}|}{|u_{a\xi} \bigcup u_{a'\xi}|}, \tag{A.1.1}$$

where $|\cdot|$ denotes set cardinality. We then derive the similarity $k_m(a, a')$ between the therapies a and a' by averaging the similarities of their corresponding drug groups:

$$k_m(a, a') = \sum_{\xi \in \Xi} \frac{sim_{\xi}(a, a')}{|\Xi|}. \tag{A.1.2}$$

The resistance mutations kernel is subsequently used together with the Needleman Wunsch algorithm to deduce a history alignment kernel over patient histories. This kernel can subsequently be used to perform non-parametric policy learning.

A.2 Sensitivity to Choice of Reward Function

We investigated the performance of the mixture-of-experts approach against the benchmarks described in the experimentation section of Chapter 3 with different reward criteria for the HIV therapy selection task. We tested three alternative formulations of reward functions wherein, (a) a higher weight is placed on CD4$^+$ counts than viral load, (b) a higher weight is placed on the absolute number of mutations than both the CD4$^+$ counts and viral load. These reward functions are given as follows:
(a)

$$r_{\mathtt{t}} = \begin{cases} -0.6 \log V_{\mathtt{t}} + 0.7 \log C_{\mathtt{t}} - 0.2|M_{\mathtt{t}}|, & \text{if } V_{\mathtt{t}} \text{ is above detection} \\ 5 + 0.7 \log C_{\mathtt{t}} - 0.2|M_{\mathtt{t}}|, & \text{if } V_{\mathtt{t}} \text{ is below detection,} \end{cases}$$

(b)

$$r_{\mathtt{t}} = \begin{cases} -0.7 \log V_t + 0.6 \log C_{\mathtt{t}} - 0.8|M_t|, & \text{if } V_{\mathtt{t}} \text{ is above detection} \\ 5 + 0.6 \log C_{\mathtt{t}} - 0.8|M_{\mathtt{t}}|, & \text{if } V_{\mathtt{t}} \text{ is below detection,} \end{cases}$$

where $V_{\mathtt{t}}$ is the viral load (in copies/mL), $C_{\mathtt{t}}$ is the CD4$^+$ count (in cells/mL) and $|M_{\mathtt{t}}|$ is the number of mutations at time $\mathtt{t}$.

	DR	**WIS**	**IS**
Random	–8.42 ± 2.68	–10.43 ± 4.17	–10.74 ± 4.16
LT kernel	9.47 ± 1.62	7.34 ± 3.79	8.71 ± 3.65
POMDP	3.57 ± 1.31	3.82 ± 2.15	3.68 ± 2.12
Mixture-of-Experts	**10.51 ± 1.20**	**11.23 ± 2.10**	**11.11 ± 1.99**

Table A.1: Performance comparison of mixture-of-experts vs. baselines for HIV therapy selection across 3 000 held-out patients using a POMDP model with 30 states using reward criterion (a) ($\gamma = 0.98$). The mixture-of-experts still produces the largest immune response while reducing the viral load, regardless of whether a larger weight is given to CD4$^+$ or V_t ($\gamma = 0.98$).

Our setup was identical to that described in the experimentation section in Chapter 3, where the reward criterion was replaced by (a) and (b) respectively. We tested the performance of the mixture-of-experts with the alternative reward criteria on the same held-out set of 3 000 patients from the EIDB as before. These results are shown in the following Tables A.1 and A.2 respectively.

	DR	**WIS**	**IS**
Random	–12.88 ± 6.42	–13.65 ± 7.46	–13.50 ± 7.16
LT kernel	**4.71 ± 4.63**	5.27 ± 4.74	4.02 ± 6.14
POMDP	1.97 ± 4.51	4.19 ± 4.22	**7.18 ± 4.69**
Mixture-of-Experts	4.02 ± 3.31	**6.29 ± 3.01**	6.96 ± 4.81

Table A.2: Performance comparison of mixture-of-experts vs. baselines for HIV therapy selection across 3 000 held-out patients using a POMDP model with 30 states using reward criterion (b) ($\gamma = 0.98$). Performance of the mixture-of-experts has significantly higher variance when placing a higher weight on the number of mutations. Evidently in this case, the mixture-of-experts does not always lead to the best immune response.

The results show the performance of the mixture-of-experts approach varies depending on the relative weightings of immune response indicators, however placing a heavier negative weight on the absolute number of mutations results in higher variance in the results across all policies, including the other baseline policies. In this case, the mixture-of-experts does not always lead to the best immune response. Since the number of mutations at each point is highly dependent on the number of strains infecting a patient at a time and past exposure to drugs, they can fluctuate considerably across patients. Normalising these counts or incorporating the mutations into a risk score (indicating the likelihood of resistance), rather than including an absolute count in the reward function may overcome this issue. Importantly however, all the methods appear to be sensitive to this choice.

A.2.1 Policy performance across unpruned treatment space

In addition to the policy evaluation results presented in Chapter 3, we performed similar evaluation experiments using the unpruned distribution treatments as actions to learn a POMDP model. The performance results from the EuResist and SHCS test data sets are presented in Tables A.3 and A.4 respectively. A higher value indicates a better performing treatment policy.

	DR	**IS**	**WIS**
Random	–7.27 ± 2.19	–8.39 ± 2.64	–8.19 ± 2.71
ST kernel	1.74 ±1.15	1.39 ±1.78	1.27 ±1.62
LT kernel	9.11 ± 1.57	8.18 ± 1.29	6.64 ± 1.72
POMDP	5.31 ± 2.30	4.79 ± 2.51	6.84 ± 2.10
Mixture-of-experts	**8.3 ± 1.12**	**12.86 ± 1.49**	**11.80 ± 1.10**

Table A.3: Off-policy evaluation using importance sampling, weighted importance sampling and doubly robust methods for different therapy selection models across EuResist test set using an unpruned treatment distribution of treatments ($\gamma = 0.98$).

	DR	**IS**	**WIS**
Random	–7.64 ± 2.19	–7.57 ± 3.67	–7.21 ± 2.11
ST kernel	1.26 ±1.19	2.35 ±1.17	2.19 ±1.30
LT kernel	6.72 ± 1.69	7.16 ± 1.63	6.89 ± 1.87
POMDP	4.86 ± 1.71	5.81 ± 2.96	5.21 ± 2.10
Mixture-of-experts	**6.79 ± 2.72**	**7.59 ± 2.94**	**7.26 ± 2.63**

Table A.4: Off-policy evaluation using importance sampling, weighted importance sampling and doubly robust methods for different therapy selection models across SHCS test set using an unpruned treatment distribution of treatments ($\gamma = 0.98$).

Evidently in both cases, the mixture-of-experts approach still outperforms its kernel and model-based counterparts, but the performance gains are not as pronounced, particularly for the SHCS data set. In the unpruned setting, there may be very few samples containing rare therapy combinations in the data set which pushes these importance weights to 0 when performing off-policy evaluation. A consequence of this is that a larger portion of the data set remains unused for off-policy evaluation. It also means that the policies learned using the unpruned space of actions are more difficult to trust.

Appendix B

Extended Results on Tree Regularisation

B.1 Details for Decision-Tree Training

Training decision trees with post-pruning. Our average path length function $\Omega(W)$ for determining the complexity of a deep model with parameters W – defined in Chapter 6 in Algorithm 2 – assumes that we have a robust, black-box way to train binary decision-trees called TRAINTREE given a labeled dataset $\{x_n, \hat{y}_n\}$. For this we use the DecisionTree module distributed in Python's sci-kit learn, which optimises information gain with Gini impurity. The specific syntax we use (for reproducibility) is:

```
tree = DecisionTree(min_sample_count=5)
tree.fit(x_train, y_train)
tree = prune_tree(tree, x_valid, y_valid)
```

The provided keyword options force the tree to have at least 5 examples from the training set in every leaf. We found that tuning hyperparameters of the TRAINTREE subprocedure, such as the minimum size of a leaf node, to be important for making useful trees.

Generally, the runtime cost of sklearn's fitting procedure scales superlinearly with the number of examples N and linearly with the number of features F – a total complexity of $O(FN\log(N))$. In practice, we found that with $N = 1000$ examples, $F = 10$ features, tree construction takes 15.3 microseconds.

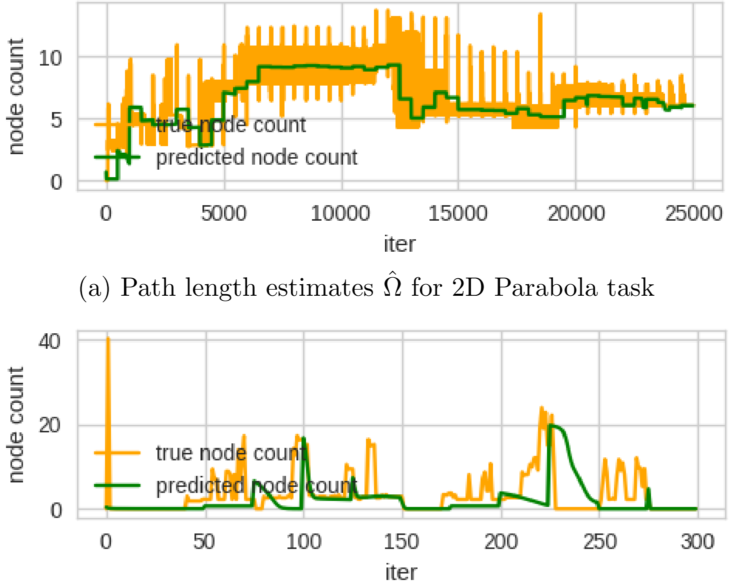
The pruning procedure is a heuristic to create simpler trees, summarised in algorithm 3. After TRAINTREE delivers a working decision tree, we propose iteratively removing each remaining leaf node, accepting the proposal if the squared prediction error on a validation set improves. This pruning removes sub-trees that don't generalise to unseen data.

Sanity check: Surrogate path length closely follow true path length. Fig. B.1 shows that our surrogate predictor $\hat{\Omega}(\cdot)$ tracks the true average path length as we train the target predictor $\hat{y}(\cdot, W)$ on several different datasets.

Sensitivity to different choices for surrogate training. In Fig. B.2, we show sample learning curves for variations of methods for approximating the average path

Algorithm 3 Post-pruning for training decision trees.

```
Require:
    $T$ : initial decision tree
    ErrOnVal$(\cdot)$ : squared error on validation data
        ErrOnVal$(T) \triangleq \sum_{n=1}^{N}(T(x_n) - y_n)^2$
procedure PruneTree( $T$, $err$ )
    $e \leftarrow$ ErrOnVal$(T)$.
    for node $n \in$ SortLeafToRoot$(T.nodes)$ do
        $T' \leftarrow$ RemoveNode$(T, n)$
        $e_{new} \leftarrow$ ErrOnVal$(T')$
        if $e_{new} < e$ then $T \leftarrow T'$
    Return $T$
```



(a) Path length estimates $\hat{\Omega}$ for 2D Parabola task

(b) Path length estimates $\hat{\Omega}$ for Signal-and-noise HMM task

Figure B.1: True average path lengths (yellow) and surrogate estimates $\hat{\Omega}$ (green) across many iterations of network parameter training iterations.

length (also called "node count") in a decision tree. In blue is the true value. Each of the other 3 lines use the same surrogate model: an MLP with 25 hidden nodes. Increasing its capacity too much, i.e. 100 hidden nodes, leads to overfitting where the surrogate is able to predict the average path length extremely well for a small number of iterations, while the performance quickly decays. With an MLP of the right capacity, four additional tricks: (1) weight augmentation, (2) random restarts with an unregularised model, (3) fixed window of data, and (4) surrogate retraining greatly improve the accuracy of the average path length predictions.

Normally, if our differentiable model is a GRU, we compile examples using the GRU weights at every batch and calculate the true average path length. This dataset is used to train the surrogate model. If examples are very sparse, surrogate predictions may

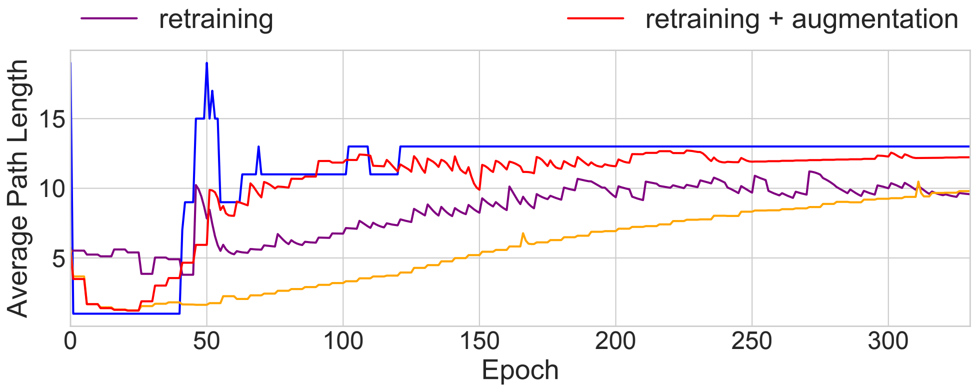


Figure B.2: This figure shows the effects of weight augmentation and retraining. The blue line is the true average path length of the decision tree at each epoch. All other lines show predicted path lengths using the surrogate MLP. By randomly sampling weights and intermittently retraining the surrogate, we significantly improve the ability of the surrogate model to track the changes in the ground truth.

be unstable. Augmentation addresses this by randomly sampling weight vectors and computing the average path length to artificially create a larger dataset. Early epochs are especially problematic when it comes to lacking data. In addition to augmentation, we use random restarts to separately train unregularised GRUs (each with different weight initialisations) to grow a dataset of weight vectors prior to training the regularised model.

As the GRU parameters take steps away from their initial values, our examples from those early epochs no longer describe the current state of the model. Retraining and a fixed window of data address this by re-learning the surrogate function at a fixed frequency using examples only from the last J epochs. In practice, both the augmentation size, the retraining frequency, and J are functions of the learning rate and the dataset size. See table B.1 for exact numbers.

B.2 Experimental Protocol

See table B.1 for model hyperparameters for each dataset. For standard recurrent models such as HMM or GRU, the decision trees were trained on the input data and the predictions of the model's output node. For our deep residual GRU-HMM, the decision trees were trained on the predictions on the GRU's output node only. For both synthetic and real-world datasets, our surrogate to the tree loss is a multilayer perceptron with 1 hidden layer of 25 nodes. For each dataset, when we investigated several regularisation strengths (λ), we initialise the model weights using the same random seed. We use the Adam algorithm (Kingma & Ba, 2014) for all optimisation.

Dataset	Total Num. Sequences	Avg. seq. length	Learning Rate	Batch size	Minimum Leaf Sample	Post-pruned	Epochs (Model)	Epochs (Surrogate)	Retraining Freq.	J
parabola	n/a	n/a	1e-2	32	0	N	250	500	100	n/a
signal-and-noise HMM	100	50	1e-2	10	25	Y	300	1000	50	50
HIV	53 236	14	1e-3	256	1 000	Y	300	5000	25	100
SEPSIS	11 786	15	1e-3	256	1 000	Y	300	5000	25	100
TIMIT	6 303	614	1e-3	256	5 000	Y	200	5000	25	100

Table B.1: Dataset summaries and training parameters used in our experiments.

B.2.1 2D Parabola

Dataset generation. The training data consists of 2D input points whose two-class decision boundary is roughly shaped like a parabola. The true decision function is defined by $y = 5 * (x - 0.5)^2 + 0.4$. We sampled all 200 input points x_n uniformly within the unit square $[0, 1] \times [0, 1]$ and labeled those above the decision function as positive. To add randomness, we flipped 10% of the points in the region near the boundary between $y = 5 * (x - 0.5)^2 + 0.2$ and $y = 5 * (x - 0.5)^2 + 0.6$.

Regularisation strengths. Tested values of regularisation strength parameter λ: 0.1, 0.5, 1, 5, 10, 25, 50, 75, 100, 250, 500, 750, 1 000, 2 500, 5 000, 7 500, 10 000, 25 000, 50 000, 75 000, 100 000

B.2.2 Signal-and-noise HMM

Dataset generation The transition and emission matrices describing the generative process used to create the signal-and-noise HMM are shown in Fig. B.3. The output y_n at every timestep is created by concatenating a one-hot vector of an emitted state and the 7-dimensional binary input vector. We emphasize that to output 1, the HMM must be in state 1 and the first input feature must be 1.

$$\begin{pmatrix} .5 & .5 & .5 & .5 & 0 & 0 & 0 \\ .5 & .5 & .5 & .5 & .5 & 0 & 0 \\ .5 & .5 & .5 & 0 & .5 & 0 & 0 \\ .5 & .5 & .5 & 0 & 0 & .5 & 0 \\ .5 & .5 & .5 & 0 & 0 & 0 & .5 \end{pmatrix}$$

(a)

$$\begin{pmatrix} .7 & .3 & 0 & 0 & 0 \\ .5 & .25 & .25 & 0 & 0 \\ 0 & .25 & .5 & .25 & 0 \\ 0 & 0 & .25 & .25 & .5 \\ 0 & 0 & 0 & .5 & .5 \end{pmatrix}$$

(b)

$$\begin{pmatrix} .5 & .5 & .5 & 0 & 0 & 0 & 0 \\ 0 & .5 & .5 & .5 & 0 & 0 & 0 \\ 0 & 0 & .5 & .5 & .5 & 0 & 0 \\ 0 & 0 & 0 & .5 & .5 & .5 & 0 \\ 0 & 0 & 0 & 0 & .5 & .5 & .5 \end{pmatrix}$$

(c)

$$\begin{pmatrix} .2 & .2 & .2 & .2 & .2 \\ .2 & .2 & .2 & .2 & .2 \\ .2 & .2 & .2 & .2 & .2 \\ .2 & .2 & .2 & .2 & .2 \\ .2 & .2 & .2 & .2 & .2 \end{pmatrix}$$

(d)

Figure B.3: Emission (5 states vs 7 features) and transition probabilities for the signal HMM (a, b) and noise HMM (c, d).

Training Details for Synthetic Data With synthetic datasets, we explore (1, 5, 6, 10, 15, 20) GRU nodes, (5, 6, 20) HMM states, and GRU-HMMs with 5 HMM states and (1, 5, 10, 15) GRU nodes.

B.2.3 Sepsis Training Details

We explore (1, 5, 6, 10, 11, 15, 20, 25, 26, 30, 35, 50, 51, 55, 60, 75, 100) GRU nodes, (5, 6, 10, 11, 15, 20, 25, 26, 30, 35, 50, 51, 55, 60, 75, 100) HMM states, and GRU-HMMs with (5, 10, 25, 50) HMM states and (1, 5, 10, 25, 50) GRU nodes. The input features are z-scored prior to training.

B.2.4 HIV Training Details

We explore (1, 5, 6, 10, 11, 15, 20, 25, 26, 30, 35, 50, 51, 55, 60, 75) GRU nodes, (5, 6, 10, 11, 15, 20, 25, 26, 30, 35, 50, 51, 55, 60, 75) HMM states, and GRU-HMMs with (5, 10, 25) HMM states and (1, 5, 10, 25, 50) GRU nodes.

B.2.5 TIMIT Training Details

We explore (1, 5, 6, 10, 11, 15, 20, 25, 26, 30, 35, 50, 51, 55, 60, 75) GRU nodes, (5, 6, 10, 11, 15, 20, 25, 26, 30, 35, 50, 51, 55, 60, 75) HMM states, and GRU-HMMs with (5, 10, 25) HMM states and (1, 5, 10, 25, 50) GRU nodes. Like Sepsis, the input features are z-scored prior to training.

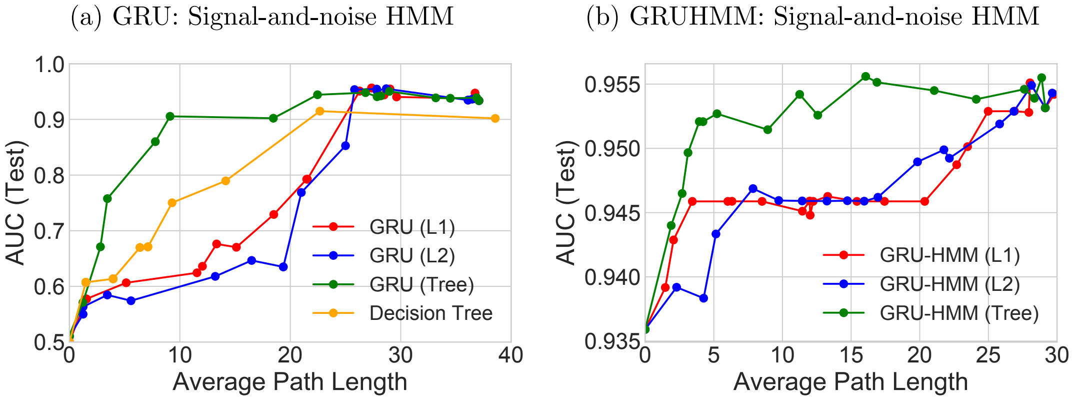


Figure B.4: Performance and complexity trade-offs using L1, L2, and Tree regularisation on (a) GRU and (b) GRU-HMM performance on the Signal-and-noise HMM dataset. Note the differences in scale.

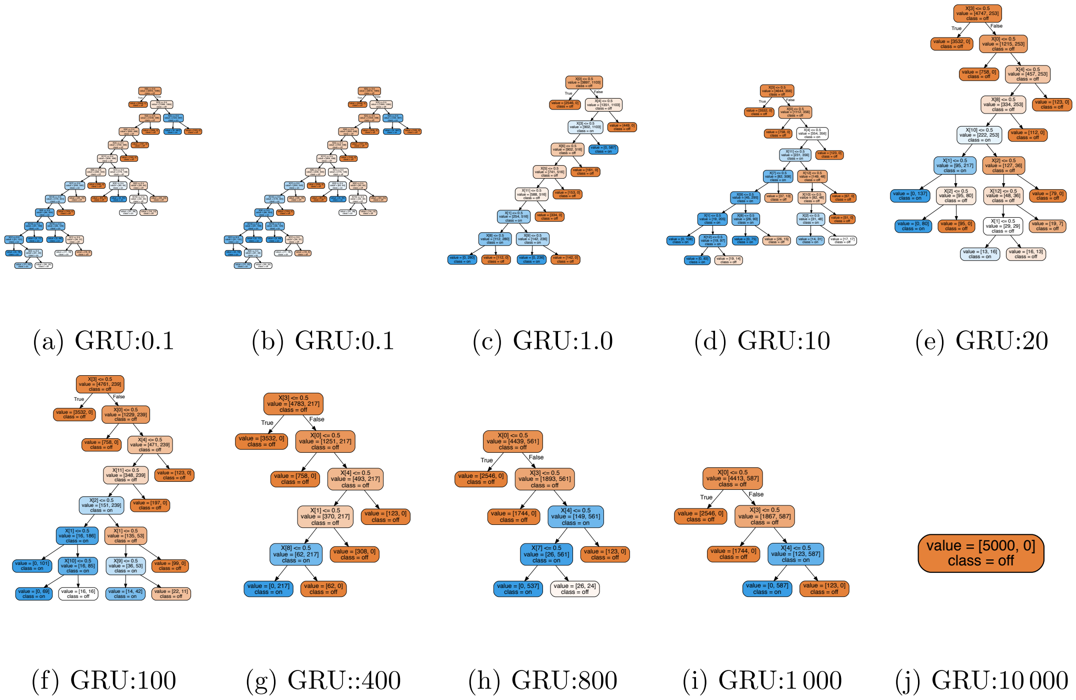
B.3 Extended Results

For signal-to-noise HMM, Sepsis, and TIMIT, we first show expanded versions of the fitness trace plots and the tree visualisations. For Sepsis and HIV, we show the additional output dimensions not in the paper.

We also include tables of the test AUC performance for our synthetic and real data sets over a vast array of parameter settings (GRU node counts, HMM state counts, regularisation strengths). Consistent with the common wisdom of training deep models, we found that larger models, with regularisation, tended to perform the best.

B.4 GRU-HMM: Deep Residual Timeseries Model

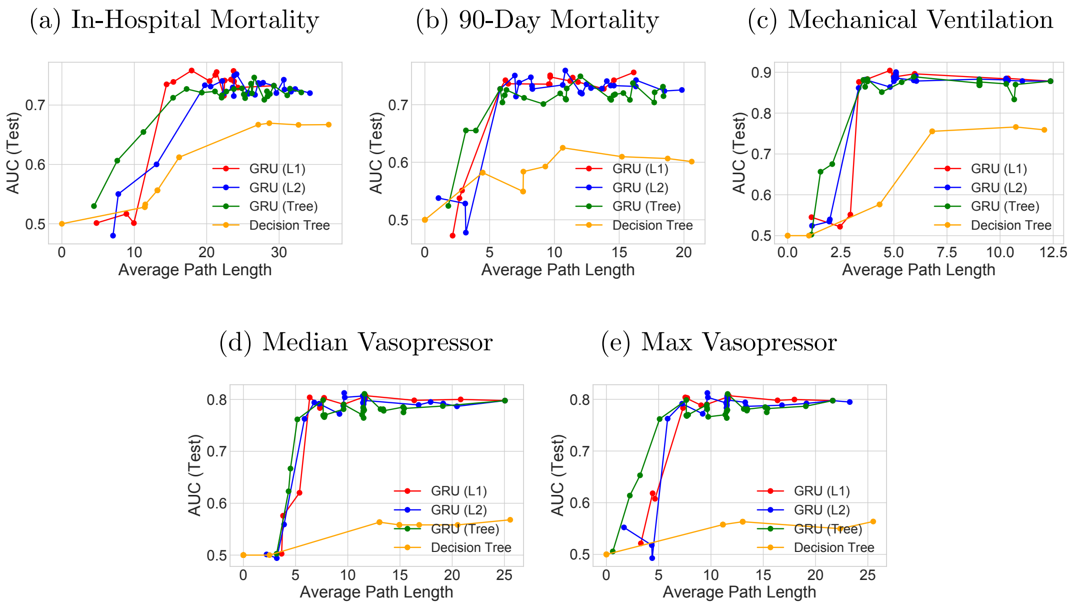
Hidden Markov Model For our purposes, Hidden Markov Models (HMMs) can be viewed as stochastic RNNs which can be interpreted as probabilistic generative models. In this work, we consider an HMM to generate a latent variable sequence $z = [z_1, \dots z_T]$ via a Markov chain, where each latent indicates one of K possible discrete states: $z_t \in \{1, ..., K\}$. This state sequence is then used to jointly produce the "data" x_t and "outcomes" y_t observed at each timestep. The joint distribution over



(a) GRU:0.1 (b) GRU:0.1 (c) GRU:1.0 (d) GRU:10 (e) GRU:20

(f) GRU:100 (g) GRU::400 (h) GRU:800 (i) GRU:1 000 (j) GRU:10 000

Figure B.5: Decision trees trained under varying tree regularisation strengths for GRU models on the signal-and-noise HMM dataset dataset. As the tree regularisation increases, the number of nodes collapses to a single one. If we focus on (h), we see that the tree resembles the ground truth data-generating function quite closely.



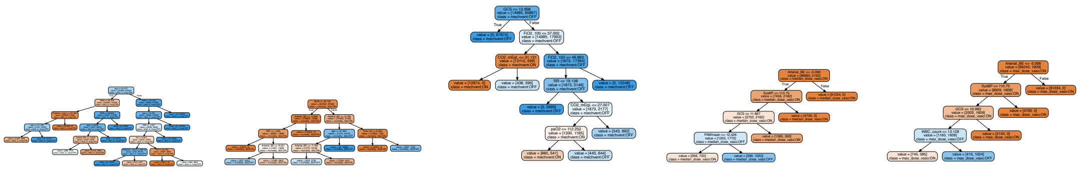
(a) In-Hospital Mortality (b) 90-Day Mortality (c) Mechanical Ventilation

(d) Median Vasopressor (e) Max Vasopressor

Figure B.6: Performance and complexity trade-offs using L1, L2, and Tree regularization on GRU performance on the Sepsis dataset.

Model	AUC (Test)	Average Path Length	Parameter Count
logreg	0.91832	17.302	6
decision tree	0.92050	29.4424	-
hmm (5)	0.93591	25.5736	71
hmm (20)	0.94177	27.2784	581
gru (1)	0.65049	1.8876	29
gru (5)	0.94812	26.304	205
gru (6)	0.94883	27.2118	264
gru (10)	0.94962	28.563	560
gru (15)	0.93982	30.7172	1 065
gru (20)	0.93368	37.0844	1 720
grutree (20/10.0)	0.94226	28.1850	1 720
grutree (20/200.0)	0.94806	26.8140	1 720
grutree (20/7 000.0)	0.94431	22.4646	1 720
grutree (20/9 000.0)	0.90555	9.1127	1 720
grutree (20/10 000.0)	0.82770	3.4400	1 720
gruhmm (5/1)	0.95146	18.2202	100
gruhmm (5/5)	0.95584	27.258	276
gruhmm (5/10)	0.95773	30.9624	631
gruhmm (5/15)	0.94857	36.7188	1 136
gruhmmtree (5/15/1.0)	0.95382	24.115	1 136
gruhmmtree (5/15/10.0)	0.95180	16.883	1 136
gruhmmtree (5/15/50.0)	0.95258	12.573	1 136
gruhmmtree (5/15/200.0)	0.95145	8.926	1 136
gruhmmtree (5/15/500.0)	0.95769	5.231	1 136
gruhmmtree (5/15/900.0)	0.95708	3.942	1 136
gruhmmtree (5/15/2 000.0)	0.95648	2.694	1 136
gruhmmtree (5/15/5 000.0)	0.95399	1.896	1 136
gruhmmtree (5/15/7 000.0)	0.93591	0.000	1 136

Table B.2: Performance metrics across models on the signal-and-noise HMM dataset. The parameter count is included as a measure of the model capacity.



(a) In-Hospital Mortality (b) 90-Day Mortality (c) Mechanical Ventilation (d) Median Vasopressor (e) Max Vasopressor

Figure B.7: Decision trees trained using $\lambda = 800.0$ for a GRU model using Sepsis. The 5 output dimensions are jointly trained.

z, x, y factorizes as:

$$p(z, y) = \pi_0(z_0) \prod_{\mathtt{t}=1}^{T} p(z_{\mathtt{t}} | z_{\mathtt{t}-1}, A) \cdot p(x_{\mathtt{t}} | z_{\mathtt{t}}, \phi) \mathrm{Bern}(y_{\mathtt{t}} | \sigma(\sum_k w_k \delta_k(z_{\mathtt{t}}))), \tag{B.4.1}$$

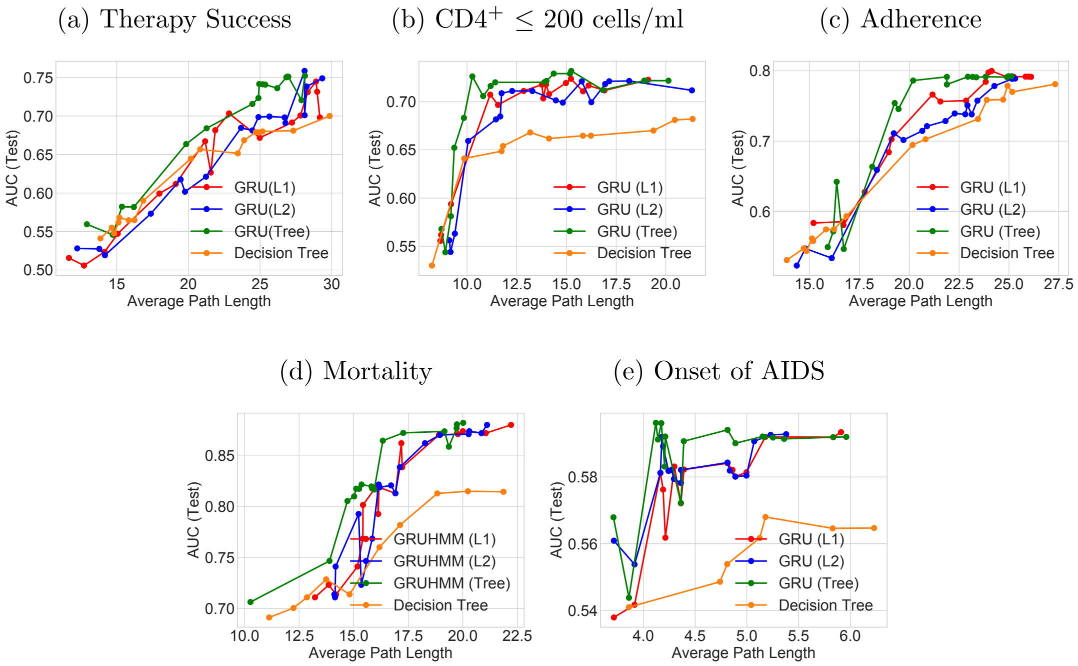
where A is a transition matrix such that $A_{i,j} = \Pr(z_{\mathtt{t}} = i | z_{\mathtt{t}-1} = j)$, $\pi_0 = p(z_0)$ is the initial state distribution, $\{\phi_k\}_{k=1}^{K}$ are the emission parameters that generate data. We

Model	In-Hospital Mortality	90-Day Mortality	Mechanical Ventilation	Median Vasopressor	Max Vasopressor	Total Average Path Length	Parameter Count
logreg	0.6980	0.6986	0.8242	0.7392	0.7392	32.489	180
decision tree	0.7017	0.7016	0.8509	0.7439	0.7427	76.242	-
hmm (5)	0.7128	0.7095	0.6979	0.7295	0.7290	35.125	405
hmm (10)	0.7227	0.7297	0.8237	0.7409	0.7405	57.629	860
hmm (15)	0.7216	0.7282	0.8188	0.7346	0.7341	61.832	1 365
hmm (20)	0.7233	0.7350	0.8218	0.7371	0.7364	62.353	1 920
hmm (25)	0.7147	0.7321	0.8089	0.7313	0.7310	63.415	2 525
hmm (30)	0.7164	0.7297	0.8099	0.7316	0.7311	65.164	3 180
hmm (35)	0.7177	0.7237	0.8095	0.7201	0.7195	65.474	3 885
hmm (50)	0.7267	0.7357	0.8373	0.7335	0.7328	66.317	6 300
hmm (75)	0.7254	0.7361	0.8059	0.7434	0.7430	72.553	11 325
hmm (100)	0.7294	0.7354	0.8129	0.7408	0.7403	80.415	17 600
gru (1)	0.3897	0.6400	0.4761	0.7414	0.7411	31.816	117
gru (5)	0.7357	0.7296	0.8795	0.7866	0.7862	45.395	645
gru (10)	0.7488	0.7445	0.8892	0.7983	0.7979	58.102	1 440
gru (15)	0.7529	0.7450	0.8912	0.8020	0.8021	61.025	2 385
gru (20)	0.7535	0.7497	0.8887	0.8018	0.8017	61.214	3 480
gru (25)	0.7578	0.7486	0.8902	0.8113	0.8114	62.029	4 725
gru (30)	0.7602	0.7508	0.8927	0.8063	0.8061	72.854	6 120
gru (35)	0.7522	0.7483	0.8900	0.8095	0.8091	74.091	7 665
gru (50)	0.7431	0.7390	0.8895	0.8054	0.8051	76.543	13 200
gru (75)	0.7408	0.7239	0.8837	0.8006	0.8000	87.422	25 425
gru (100)	0.7325	0.7273	0.8781	0.7977	0.7975	94.161	41 400
grutree (100/0.01)	0.7276	0.7314	0.8776	0.7873	0.7867	91.797	41 400
grutree (100/1.0)	0.7147	0.7040	0.8741	0.7812	0.7810	82.019	41 400
grutree (100/8.0)	0.7232	0.7203	0.8763	0.7845	0.7840	73.767	41 400
grutree (100/20.0)	0.7123	0.7085	0.8733	0.7813	0.7813	65.035	41 400
grutree (100/70.0)	0.7360	0.7376	0.8813	0.7988	0.7986	61.012	41 400
grutree (100/300.0)	0.7210	0.7197	0.8681	0.7676	0.7678	54.177	41 400
grutree (100/2 000.0)	0.7230	0.7167	0.8335	0.7616	0.7619	48.206	41 400
grutree (100/5 000.0)	0.6546	0.6552	0.6752	0.6668	0.6530	26.085	41 400
grutree (100/7 000.0)	0.6063	0.6554	0.6565	0.6230	0.6138	20.214	41 400
grutree (100/8 000.0)	0.5298	0.5242	0.5025	0.5026	0.5057	13.383	41 400
gruhmm (1/5)	0.4222	0.6472	0.4678	0.7478	0.7477	41.583	722
gruhmm (1/10)	0.4007	0.6295	0.4730	0.7418	0.7419	61.041	1 517
gruhmm (1/25)	0.4019	0.6207	0.4773	0.7353	0.7352	65.955	4 802
gruhmm (1/50)	0.3999	0.6162	0.4772	0.7120	0.7121	70.534	13 277
gruhmm (5/5)	0.7430	0.7372	0.8798	0.8009	0.8006	47.639	1 050
gruhmm (5/10)	0.7408	0.7320	0.8819	0.7991	0.7988	63.627	1 845
gruhmm (5/25)	0.7365	0.7279	0.8776	0.7955	0.7952	68.215	5 130
gruhmm (5/50)	0.7222	0.7107	0.8660	0.7814	0.7811	71.572	13 605
gruhmm (10/5)	0.7468	0.7467	0.8949	0.8098	0.8097	50.902	1 505
gruhmm (10/10)	0.7490	0.7478	0.8958	0.8098	0.8096	63.522	2 300
gruhmm (10/25)	0.7422	0.7407	0.8916	0.8055	0.8054	70.919	5 585
gruhmm (10/50)	0.7254	0.7221	0.8824	0.7903	0.7903	71.297	14 060
gruhmm (25/5)	0.7580	0.7568	0.8941	0.8236	0.8235	51.794	3 170
gruhmm (25/10)	0.7592	0.7563	0.8945	0.8225	0.8225	64.223	3 965
gruhmm (25/25)	0.7525	0.7508	0.8912	0.8186	0.8184	72.480	7 250
gruhmm (25/50)	0.7604	0.7583	0.8954	0.8106	0.8103	79.127	11 025
gruhmm (50/5)	0.7655	0.7592	0.9006	0.8228	0.8226	64.229	6 945
gruhmm (50/10)	0.7648	0.7568	0.9003	0.8220	0.8219	69.281	7 740
gruhmm (50/25)	0.7600	0.7555	0.8981	0.8205	0.8203	85.503	11 025
gruhmm (50/50)	0.7412	0.7373	0.8910	0.8056	0.8055	101.637	19 500
gruhmmtree (50/50/0.5)	0.7432	0.7492	0.879	0.7854	0.7849	84.188	19 500
gruhmmtree (50/50/20.0)	0.7435	0.747	0.8826	0.7914	0.7906	77.815	19 500
gruhmmtree (50/50/50.0)	0.7384	0.7548	0.8914	0.7922	0.7918	71.719	19 500
gruhmmtree (50/50/200.0	0.747	0.7502	0.8767	0.7832	0.7824	69.715	19 500
gruhmmtree (50/50/300.0)	0.7539	0.7623	0.8942	0.8092	0.8091	66.9	19 500
gruhmmtree (50/50/600.0	0.7435	0.7453	0.8821	0.7909	0.7905	63.703	19 500
gruhmmtree (50/50/1 000.0)	0.7575	0.7502	0.8739	0.7882	0.7873	60.949	19 500
gruhmmtree (50/50/3 000.0)	0.7396	0.7484	0.8926	0.8013	0.8011	54.751	19 500
gruhmmtree (50/50/4 000.0)	0.7432	0.7511	0.8915	0.802	0.8024	44.868	19 500
gruhmmtree (50/50/7 000.0)	0.7308	0.7477	0.8813	0.7881	0.7882	27.836	19 500
gruhmmtree (50/50/9 000.0)	0.7132	0.7319	0.8261	0.7301	0.7299	0.0	19 500

Table B.3: Performance metrics for multi-dimensional classification on a held-out portion of the Sepsis dataset. *Total Average Path Length* refers to the summed average path lengths across the 5 output dimensions. Refer to Fig. B.6 for average-path-lengths split across dimensions.

can then apply the same objective as above for training.

GRU-HMM: Modeling the residuals of an HMM. We now consider an additional model, the GRU-HMM, designed for interpretability. The idea is to use a GRU to to model the residual errors when predicting the binary target via the HMM belief


(a) Therapy Success

(b) CD4$^+ \leq 200$ cells/ml

(c) Adherence

(d) Mortality

(e) Onset of AIDS

Figure B.8: Performance and complexity trade-offs using L1, L2, and Tree regularisation on GRU for the HIV dataset. The 5 outputs shown here were trained jointly.

states. We can further penalise the complexity of the GRU predictions via our tree regularisation, so that higher-quality predictions do not come at the price of a much less interpretable model.

We train the deep residual model on the same suite of synthetic and real world datasets. See Tables B.2, B.3, B.5 for a comparison of GRU-HMM with vanilla GRU and HMM models under different regularisation and expressiveness parameters. We can see that across the datasets, deep residual models perform around 1% better than their vanilla equivalents with roughly the same number of model parameters.

By nature of being a residual model, decision trees were trained only on the GRU output node, leaving the HMM unconstrained. See Figure B.10 for a pictoral representation. Similar to what we did for GRU models, Figures B.4b, B.11 compare model performance as the λ parameter for L1, L2, and Tree regularisation increase. We can see a similar albeit less pronounced effect where Tree regularization dominates other methods in low node count regions. It is important to notice the range of the AUC axis in these figures, where the worst the residual model can performance is the HMM-only AUC. Figure B.12 show the regularised trees produced by the GRU-HMM. Although they share some structure with Figure B.7, there are important distinctions that encourage us to conclude that the GRU in a residual models performs a different role than when trained alone.

Model	Poor Adherence	Mortality	CD4^+ Count ≤ 200	Therapy Success	Total Average Path Length	Parameter Count
logreg	0.6884	0.7031	0.5741	0.6092	38.942	1155
decision tree	0.7100	0.7601	0.5937	0.6286	62.150	-
hmm (5)	0.7106	0.7611	0.6012	0.6265	41.864	865
hmm (10)	0.7287	0.7627	0.6237	0.6409	46.309	1780
hmm (25)	0.7243	0.7627	0.6327	0.6384	56.159	4825
hmm (50)	0.7181	0.7639	0.6412	0.6370	69.014	10900
hmm (75)	0.7244	0.7661	0.6294	0.6518	70.476	18225
hmm (100)	0.7261	0.7657	0.6287	0.6524	71.159	26800
gru (5)	0.6457	0.6814	0.6695	0.6834	58.347	1310
gru (25)	0.7516	0.7986	0.7073	0.6991	60.072	8050
gru (50)	0.7011	0.8290	0.6995	0.7054	67.513	19850
gru (75)	0.7623	0.8514	0.7117	0.7490	64.870	35400
gru (100)	0.7340	0.8216	0.6981	0.7235	67.183	54700
grutree (100/0.01)	0.7176	0.7948	0.7046	0.6803	91.020	54700
grutree (100/1.0)	0.7134	0.7997	0.7138	0.6892	86.774	54700
grutree (100/20.0)	0.7157	0.8066	0.7216	0.7114	76.025	54700
grutree (100/70.0)	0.7485	0.8210	0.7413	0.7060	68.952	54700
grutree (100/300.0)	0.7251	0.8178	0.7264	0.6746	54.058	54700
grutree (100/2 000.0)	0.7030	0.8169	0.6342	0.6627	49.839	54700
grutree (100/5 000.0)	0.6549	0.7582	0.6142	0.6352	23.895	54700
grutree (100/7 000.0)	0.6167	0.7524	0.5740	0.5634	15.283	54700
grutree (100/8 000.0)	0.5874	0.7412	0.5003	0.5027	7.391	54700
gruhmm (5/5)	0.6430	0.6647	0.5418	0.6479	67.619	2175
gruhmm (5/10)	0.6708	0.6720	0.5879	0.6517	72.137	3090
gruhmm (5/25)	0.6951	0.6981	0.6476	0.6955	68.200	6135
gruhmm (5/50)	0.6810	0.7002	0.6760	0.7114	71.518	12210
gruhmm (10/5)	0.7018	0.7147	0.7049	0.7208	64.852	3635
gruhmm (10/10)	0.7190	0.7378	0.7136	0.7578	73.252	4550
gruhmm (10/25)	0.7264	0.7457	0.7217	0.7951	70.884	7595
gruhmm (10/50)	0.7570	0.7522	0.7224	0.8234	69.726	13670
gruhmm (25/10)	0.7462	0.7861	0.7152	0.8217	68.241	9830
gruhmm (25/25)	0.7435	0.8102	0.7425	0.8186	79.261	12875
gruhmm (25/50)	0.7484	0.7714	0.7501	0.8006	76.174	18950
gruhmm (50/10)	0.7437	0.7668	0.7813	0.8260	70.081	21630
gruhmm (50/25)	0.7380	0.7557	0.7824	0.8215	88.617	24675
gruhmm (50/50)	0.7317	0.7684	0.7920	0.8007	97.864	30750
gruhmmtree (50/50/0.5)	0.7432	0.7692	0.8790	0.7804	73.168	30750
gruhmmtree (50/50/50.0)	0.7426	0.8152	0.8914	0.7979	67.729	30750
gruhmmtree (50/50/200.0	0.7461	0.8308	0.8767	0.8032	59.025	30750
gruhmmtree (50/50/600.0	0.7467	0.8820	0.8821	0.8293	52.128	30750
gruhmmtree (50/50/1 000.0)	0.7375	0.8951	0.8739	0.7882	48.247	30750
gruhmmtree (50/50/4 000.0)	0.7242	0.8461	0.8515	0.8030	14.868	30750
gruhmmtree (50/50/7 000.0)	0.7280	0.8462	0.8313	0.7484	1.836	30750

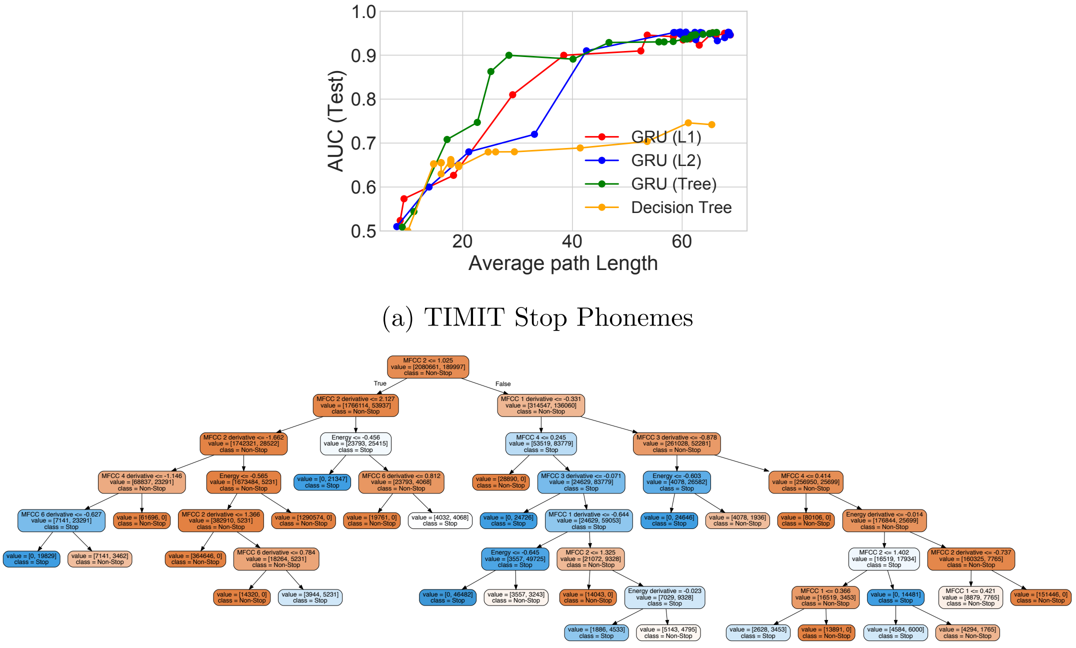
Table B.4: Performance metrics for multi-dimensional classification on a held-out portion of the HIV dataset. *Total Average Path Length* refers to the summed average path lengths across the output dimensions.

B.5 Runtime comparisons

Table B.6 shows the wall time for training one epoch of each of the models presented in this paper using each of the datasets. Please note that the wall times for GRU-TREE and GRU-HMM-TREE include the cost of surrogate training. If the retraining frequency is small, then the amortised cost should be small.

B.6 Extended Stability Tests

In the paper, we noted that decision trees are stable over multiple run. Here, we show that using the signal-and-noise HMM dataset, 10 independent runs with random initialisations and $\lambda = 1000.0$ produce either the same or comparable trees. Additionally, we show that with weak regularisation ($\lambda = 0.01$), the variability of the learned decision trees is high. Figures B.16, B.15 include examples of such trees on the signal-and-noise



(a) TIMIT Stop Phonemes

(b) GRU:500

Figure B.9: (a) Performance and complexity trade-offs using L1, L2, and Tree regularisation for GRU models on TIMIT. (b) Decision tree trained using $\lambda = 500.0$ tree regularization on GRU.

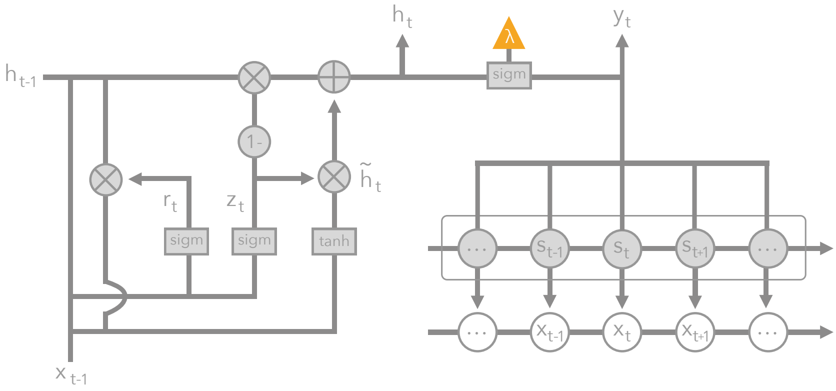
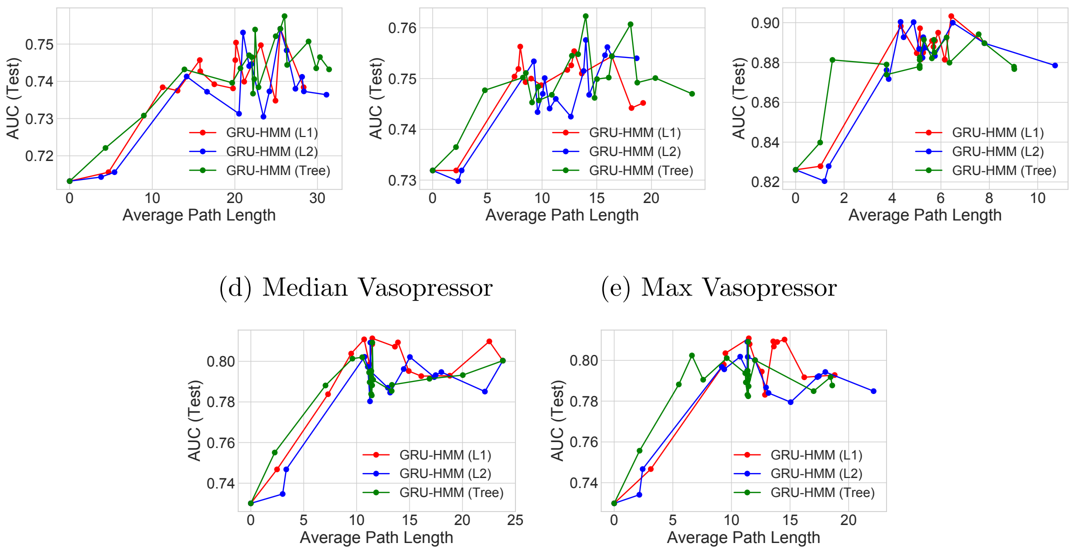


Figure B.10: Deep residual model: GRU-HMM. The orange triangle indicates the output used in surrogate training for tree regularization.

dataset. Similar results are found for real-world datasets.

Model	AUC	Average Path Length	Parameter Count
logreg	0.7747	23.460	27
decision tree	0.8668	59.2061	-
hmm (5)	0.8900	51.911	295
hmm (10)	0.8981	56.273	640
hmm (25)	0.9129	57.602	1 975
hmm (50)	0.9189	63.752	5 200
hmm (75)	0.9251	71.473	9 675
gru (1)	0.9169	42.602	86
gru (5)	0.9451	49.275	490
gru (10)	0.9509	60.079	1 130
gru (25)	0.9547	62.051	3 950
gru (50)	0.9578	64.957	11 650
gru (75)	0.9620	68.998	23 100
gruhmm (1/5)	0.9419	54.9723	381
gruhmm (1/10)	0.9535	53.5642	726
gruhmm (1/25)	0.9636	57.3290	2601
gruhmm (5/5)	0.9569	55.9531	785
gruhmm (5/10)	0.9575	57.6199	1 130
gruhmm (5/25)	0.9603	59.9925	2 465
gruhmm (10/5)	0.9626	57.0652	1 425
gruhmm (10/10)	0.9641	60.7877	1 770
gruhmm (10/25)	0.9651	61.0018	3 105
gruhmm (25/5)	0.9635	57.5288	4 245
gruhmm (25/10)	0.9657	60.5212	4 590
gruhmm (25/25)	0.9663	65.0161	5 925
gruhmm (50/5)	0.9676	62.2378	11 945
gruhmm (50/10)	0.9679	65.1191	12 290
gruhmm (50/25)	0.9685	67.4301	13 625
grutree (75/0.01)	0.9517	66.2801	23 100
grutree (75/0.1)	0.9466	62.4316	23 100
grutree (75/0.5)	0.9367	60.8764	23 100
grutree (75/2.0)	0.9311	58.3659	23 100
grutree (75/5.0)	0.9302	55.7588	23 100
grutree (75/10.0)	0.9288	46.6616	23 100
grutree (75/100.0)	0.8911	40.1123	23 100
grutree (75/500.0)	0.8998	28.4240	23 100
grutree (75/700.0)	0.8628	25.136	23 100
grutree (75/800.0)	0.7471	22.6671	23 100
grutree (75/1 000.0)	0.7082	17.1523	23 100
grutree (75/6 000.0)	0.5441	11.1108	23 100
grutree (75/7 000.0)	0.5088	8.9910	23 100
gruhmmtree (50/25/0.1)	0.9507	69.1110	13 625
gruhmmtree (50/25/1.0)	0.9465	67.5773	13 625
gruhmmtree (50/25/6.0)	0.9515	65.1494	13 625
gruhmmtree (50/25/20.0)	0.9449	64.0072	13 625
gruhmmtree (50/25/30.0)	0.9482	62.5406	13 625
gruhmmtree (50/25/70.0)	0.9460	58.0111	13 625
gruhmmtree (50/25/100.0)	0.9470	51.2417	13 625
gruhmmtree (50/25/500.0)	0.9401	42.1882	13 625
gruhmmtree (50/25/700.0)	0.9352	40.1281	13 625
gruhmmtree (50/25/1 000.0)	0.9390	38.0072	13 625
gruhmmtree (50/25/3 000.0)	0.9280	25.9120	13 625
gruhmmtree (50/25/4 000.0)	0.9311	21.7170	13 625
gruhmmtree (50/25/7 000.0)	0.9290	10.1122	13 625
gruhmmtree (50/25/9 000.0)	0.9134	1.0563	13 625
gruhmmtree (50/25/10 000.0)	0.9125	0.0000	13 625

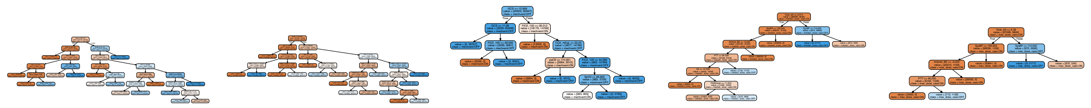
Table B.5: Performance metrics across models on a held-out portion of the TIMIT dataset.



(a) In-Hospital Mortality (b) 90-Day Mortality (c) Mechanical Ventilation

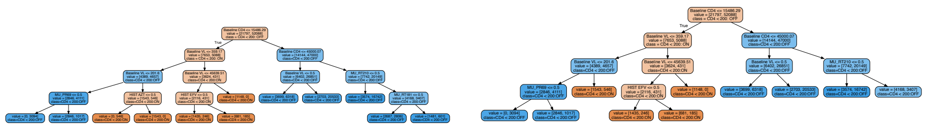
(d) Median Vasopressor (e) Max Vasopressor

Figure B.11: Performance and complexity trade-offs using L1, L2, and Tree regularization on GRU-HMM performance on the Sepsis dataset.



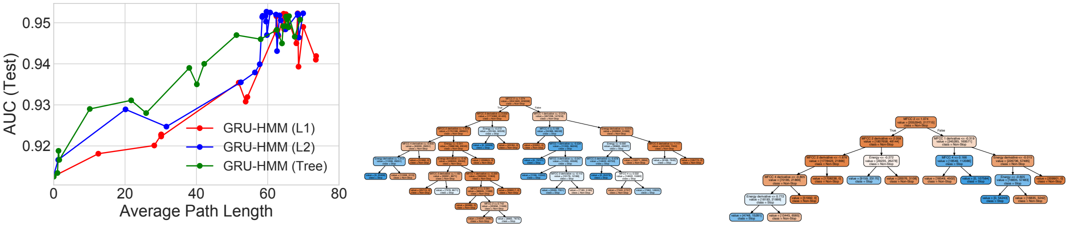
(a) In-Hospital Mortality (b) 90-Day Mortality (c) Mechanical Ventilation (d) Median Vasopressor (e) Max Vasopressor

Figure B.12: Decision trees trained using Tree regularization ($\lambda = 2000.0$) from GRU-HMM predictions on the Sepsis dataset.



(a) GRU-HMM: CD4$^+$ $\leq$ 200 cells/ml (b) GRU-HMM: CD4$^+$ $\leq$ 200 cells/ml

Figure B.13: *HIV task:* Study of different regularisation techniques for GRU-HMM model with 75 GRU nodes and 25 HMM states, trained to predict whether CD4+ $\leq$ 200 cells/ml. (a) Example decision tree for $\lambda = 1000.0$. (b) Example decision tree for $\lambda = 3000.0$. The tree in (b) is slightly smaller than the tree in (a) as a result of the regularisation.



(a) GRU-HMM: Stop vs Non-Stop (b) GRU-HMM: Stop vs Non-Stop (c) GRU-HMM: Stop vs Non-Stop

Figure B.14: *TIMIT task:* Study of different regularisation techniques for GRU-HMM model with 75 GRU nodes and 25 HMM states, trained to predict STOP phonemes. (a) Tradeoff curves showing how AUC predictive power and decision-tree complexity evolve with increasing regularisation strength under L1, L2, or Tree regularisation. (b) Example decision tree for $\lambda = 3000.0$. (c) Example decision tree for $\lambda = 7000.0$. When comparing with figure B.9b, this tree is significantly smaller, suggesting that the GRU performs a different role in the residual model.

Dataset	Model	Epoch Time (Sec.)
Signal-and-noise HMM	HMM	16.66 ± 2.53
Signal-and-noise HMM	GRU	30.48 ± 1.92
Signal-and-noise HMM	GRU-HMM	50.40 ± 5.56
Signal-and-noise HMM	GRU-TREE	43.83 ± 3.84
Signal-and-noise HMM	GRU-HMM-TREE	73.24 ± 7.86
SEPSIS	HMM	589.80 ± 24.11
SEPSIS	GRU	822.27 ± 11.17
SEPSIS	GRU-HMM	$1\,666.98 \pm 147.00$
SEPSIS	GRU-TREE	$2\,015.15 \pm 388.12$
SEPSIS	GRU-HMM-TREE	$2\,443.66 \pm 351.22$
TIMIT	HMM	$1\,668.96 \pm 126.96$
TIMIT	GRU	$2\,116.83 \pm 438.83$
TIMIT	GRU-HMM	3207.16 ± 651.85
TIMIT	GRU-TREE	$3\,977.01 \pm 812.11$
TIMIT	GRU-HMM-TREE	$4\,601.44 \pm 805.88$

Table B.6: Training time for recurrent models measured against all datasets used in this paper. Epoch time denotes the number of seconds it took for a single pass through all the training data. The epoch times for GRU-TREE and GRU-HMM-TREE include surrogate training expenses. If we retrain sparsely, then the cost of surrogate training is amortized and the epoch time for GRU and GRU-TREE, GRU-HMM and GRU-HMM-TREE are approximately the same. To measure epoch time, we used 10 HMM states, 10 GRU states, and 5 of each for GRU-HMM models. We trained the surrogate model for 5000 epochs. These tests were run on a single Intel Core i5 CPU.

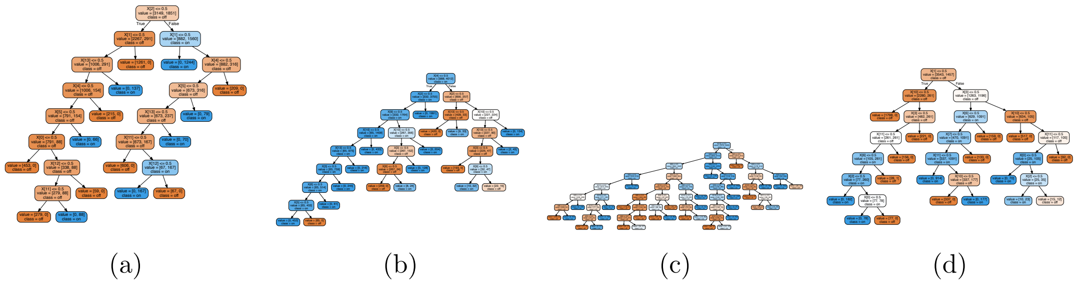
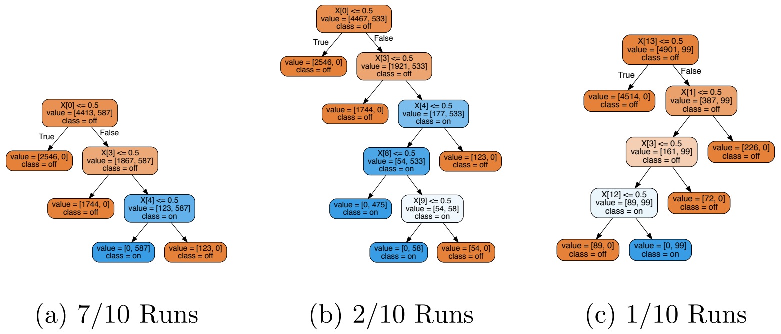

(a) (b) (c) (d)

Figure B.15: Decision trees from 10 independent runs on the signal-and-noise HMM dataset with $\lambda = 0.01$. With low regularisation, the variance in tree size and shape is high.



(a) 7/10 Runs (b) 2/10 Runs (c) 1/10 Runs

Figure B.16: Decision trees from 10 independent runs on the signal-and-noise HMM dataset with $\lambda = 1000.0$. Seven of the ten runs resulted in a tree of the same structure. The other three trees are similar, often having additional subtrees but sharing the same splits and features.

References

Alberto Abadie & Guido W Imbens. Bias-corrected matching estimators for average treatment effects. *Journal of Business & Economic Statistics*, 29(1):1–11, 2011.

Philip Adler, Casey Falk, Sorelle A Friedler, Gabriel Rybeck, Carlos Scheidegger, Brandon Smith & Suresh Venkatasubramanian. Auditing black-box models for indirect influence. In *ICDM*, 2016.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon & Kevin. Murphy. Deep Variational Information Bottleneck. *ArXiv e-prints*, December 2016.

André Altmann, Niko Beerenwinkel, Tobias Sing, Igor Savenkov, Martin Däumer, Rolf Kaiser, Soo-Yon Rhee, W Jeffrey Fessel, Robert W Shafer & Thomas Lengauer. Improved prediction of response to antiretroviral combination therapy using the genetic barrier to drug resistance. *Antiviral therapy*, 12(2):169, 2007.

Charles Audet & Michael Kokkolaras. *Blackbox and derivative-free optimization: theory, algorithms and applications*. Springer, 2016.

Peter C Austin. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate behavioral research*, 46(3):399–424, 2011.

Anoop Korattikara Balan, Vivek Rathod, Kevin P Murphy & Max Welling. Bayesian dark knowledge. In *NIPS*, 2015.

Elias Bareinboim. Causal reinforcement learning. NeurIPS 2018 Workshop on Causal Learning, December 2018. URL: `https://sites.google.com/view/nips2018causallearning/home`.

Sascha O Becker & Andrea Ichino. Estimation of average treatment effects based on propensity scores. *The stata journal*, 2(4):358–377, 2002.

Niko Beerenwinkel, Nicholas Eriksson & Bernd Sturmfels. Conjunctive bayesian networks. *Bernoulli*, pages 893–909, 2007.

Richard Bellman. Dynamic programming and stochastic control processes. *Information and control*, 1(3):228–239, 1958.

Steffen Bickel, Jasmina Bogojeska, Thomas Lengauer & Tobias Scheffer. Multi-task learning for hiv therapy screening. In *Proceedings of the 25th international conference on Machine learning*, pages 56–63. ACM, 2008.

Oleh Bodilovskyi & Anton Popov. Blood oxygen saturation alarm level analysis during mechanical lung ventilation. In *Signal Processing Symposium (SPS), 2013*, pages 1–4.

IEEE, 2013.

Jasmina Bogojeska, Daniel Stöckel, Maurizio Zazzi, Rolf Kaiser, Francesca Incardona, Michal Rosen-Zvi & Thomas Lengauer. History-alignment models for bias-aware prediction of virological response to hiv combination therapy. In *AISTATS*, pages 118–126, 2012.

Byron Boots, Geoffrey Gordon & Arthur Gretton. Hilbert space embeddings of predictive state representations. *arXiv preprint arXiv:1309.6819*, 2013.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard & Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research*, 14(1):3207–3260, 2013.

Donald A Brand, Patricia A Patrick, Jeffrey T Berger, Mediha Ibrahim, Ajsza Matela, Shweta Upadhyay & Peter Spiegler. Intensity of vasopressor therapy for septic shock and the risk of in-hospital death. *Journal of pain and symptom management*, 53(5): 938–943, 2017.

Charles J. Carpenter, David A. Cooper, Margaret A. Fischl et al. Antiretroviral therapy in adults in 2000: Updated recommendations of the international antiviral society usa-panel. *JAMA*, 283(3):381–390, 2000. URL: `+http://dx.doi.org/10.1001/jama.283.3.381`.

Charles J Carpenter, Margaret A. Fischl, Scott M. Hammer et al. Antiretroviral therapy for hiv infection in 1996: Recommendations of the international antiviral society usa-panel. *JAMA*, 276(2):146–154, 1996. URL: `+http://dx.doi.org/10.1001/jama.1996.03540020068031`.

Charles J. Carpenter, Margaret A. Fischl, Scott M. Hammer et al. Antiretroviral therapy for hiv infection in 1998: Updated recommendations of the international antiviral society usa-panel. *JAMA*, 280(1):78–86, 1998. URL: `+http://dx.doi.org/10.1001/jama.280.1.78`.

Chris K Carter & Robert Kohn. On gibbs sampling for state space models. *Biometrika*, 81(3):541–553, 1994.

Heining Cham & Stephen G West. Propensity score analysis with missing data. *Psychological methods*, 21(3):427, 2016.

Clive R Charig, David R Webb, Stephen Richard Payne & John E Wickham. Comparison of treatment of renal calculi by open surgery, percutaneous nephrolithotomy, and extracorporeal shockwave lithotripsy. *Br Med J (Clin Res Ed)*, 292(6524):879–882, 1986.

Zhengping Che, David Kale, Wenzhe Li, Mohammad Taha Bahadori & Yan Liu. Deep computational phenotyping. In *KDD*, 2015.

Yevgen Chebotar, Karol Hausman, Marvin Zhang, Gaurav Sukhatme, Stefan Schaal & Sergey Levine. Combining model-based and model-free updates for trajectory-centric reinforcement learning. *arXiv preprint arXiv:1703.03078*, 2017.

Gal Chechik, Amir Globerson, Naftali Tishby & Yair Weiss. Information bottleneck for gaussian variables. In *Journal of Machine Learning Research*, pages 165–188, 2005.

Jonathan H Chen & Steven M Asch. Machine learning and prediction in medicineâ[U+0080] [U+0094]beyond the peak of inflated expectations. *N Engl J Med*, 376(26):2507–2509, 2017.

Kyunghyun Cho, Bart van Merriënboer Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares Holger Schwenk & Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMLNP*, 2014. URL: `http://emnlp2014.org/papers/pdf/EMNLP2014179.pdf`.

Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart & Jimeng Sun. Doctor AI: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, 2016.

Thomas M Cover & Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

Mark Craven & Jude W Shavlik. Extracting tree-structured representations of trained networks. In *NIPS*, 1996.

A Philip Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(1):1–15, 1979a.

A Philip Dawid. Some misleading arguments involving conditional independence. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(2):249–252, 1979b. ISSN 00359246. URL: `http://www.jstor.org/stable/2985039`.

A Philip Dawid. Influence diagrams for causal modelling and inference. *International Statistical Review*, 70(2):161–189, 2002.

A Philip Dawid. Fundamentals of statistical causality. 2007a.

A Philip Dawid. Fundamentals of statistical causality. Technical report, Department of Statistical Science, University College London, 2007b.

A Philip Dawid. Beware of the dag! In *Causality: objectives and assessment*, pages 59–86, 2010.

A Philip Dawid. *The decision-theoretic approach to causal inference*. Wiley Online Library, 2012.

Erik De Clercq. Anti-hiv drugs: 25 compounds approved within 25 years after the discovery of hiv. *International journal of antimicrobial agents*, 33(4):307–320, 2009.

Steven G Deeks, Sharon R Lewin & Diane V Havlir. The end of aids: Hiv infection as a chronic disease. *The Lancet*, 382(9903):1525–1533, 2013.

Rajeev H Dehejia & Sadek Wahba. Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. *Journal of the American statistical Association*, 94(448):1053–1062, 1999.

Marc Deisenroth & Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

Finale Doshi-Velez & Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Harris Drucker & Yann Le Cun. Improving generalization performance using double backpropagation. *IEEE Transactions on Neural Networks*, 3(6):991–997, 1992.

Otis Dudley Duncan. *Introduction to Structural Equation Models*. Studies in Population. Academic Press, San Diego, 1975. URL: `http://www.sciencedirect.com/science/article/pii/B9780122241505500022`.

Dumitru Erhan, Yoshua Bengio, Aaron Courville & Pascal Vincent. Visualizing higher-layer features of a deep network. Technical Report 1341, Department of Computer Science and Operations Research, University of Montreal, 2009.

Damien Ernst, Guy-Bart Stan, Jorge Goncalves & Louis Wehenkel. Clinical data based optimal sti strategies for hiv: a reinforcement learning approach. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 667–672. IEEE, 2006.

Eric O Freed. Hiv-1 and the host cell: an intimate association. *Trends in microbiology*, 12(4):170–177, 2004.

Kenji Fukumizu, Le Song & Arthur Gretton. Kernel bayes' rule. In *Advances in neural information processing systems*, pages 1737–1745, 2011.

Kenji Fukumizu, Le Song & Arthur Gretton. Kernel bayes' rule: Bayesian inference with positive definite kernels. *Journal of Machine Learning Research*, 14(1):3753–3783, 2013.

Yarin Gal, Rowan Thomas McAllister & Carl Edward Rasmussen. Improving pilco with bayesian neural network dynamics models. In *Data-Efficient Machine Learning workshop*, volume 951, page 2016, 2016.

John S Garofolo et al. TIMIT acoustic-phonetic continuous speech corpus. *Linguistic Data Consortium*, 10(5), 1993.

Samuel J Gershman. Reinforcement learning and causal models. *The Oxford handbook of causal reasoning*, page 295, 2017.

Arthur S Goldberger. Structural equation methods in the social sciences. *Econometrica: Journal of the Econometric Society*, pages 979–1001, 1972.

Sander Greenland & Timothy Lash. Bias analysis. *Modern Epidemiology*, pages 345 – 380, 2008.

Steffen Grünewälder, Luca Baldassarre, Massimiliano Pontil, Arthur Gretton & Guy Lever. Modeling transition dynamics in mdps with rkhs embeddings of conditional distributions. *CoRR, abs/1112.4722*, 2011.

Huldrych F. Günthard, Judith A. Aberg, Joseph J. Eron et al. Antiretroviral treatment of adult hiv infection: 2014 recommendations of the international antiviral society usa-panel. *JAMA*, 312(4):410–425, 2014. URL: `+http://dx.doi.org/10.1001/jama.2014.8722`.

Huldrych F. Günthard, Judith A. Aberg, Joseph J. Eron & et al. Antiretroviral treatment of adult hiv infection: 2014 recommendations of the international antiviral society–usa panel. *JAMA*, 312(4):410–425, 2014. URL: +http://dx.doi.org/10.1001/jama.2014.8722.

Huldrych F. Günthard, Vincent Calvez, Roger Paredes et al. Hiv drug resistance 2018: Recommendations of the international antiviral society usa-panel. *Clinical Infectious Diseases*, 2016.

Huldrych F Günthard, Michael S Saag, Benson Constance A et al. Antiretroviral drugs for treatment and prevention of hiv infection in adults: 2016 recommendations of the international antiviral societyâ€“usa panel. *JAMA*, 316(2):191–210, 2016. URL: +http://dx.doi.org/10.1001/jama.2016.8900.

Ruocheng Guo, Lu Cheng, Jundong Li, P Richard Hahn & Huan Liu. A survey of learning causality with data: Problems and methods. *arXiv preprint arXiv:1809.09337*, 2018.

Jihane Hajj, Natalie Blaine, Jola Salavaci & Douglas Jacoby. The centrality of sepsis : A review on incidence, mortality, and cost of care. In *Healthcare*, volume 6, page 90. Multidisciplinary Digital Publishing Institute, 2018.

Scott M. Hammer, Joseph J. Eron, Peter Reiss et al. Antiretroviral treatment of adult hiv infection: 2008 recommendations of the international antiviral society usa-panel. *JAMA*, 300(5):555–570, 2008. URL: +http://dx.doi.org/10.1001/jama.300.5.555.

Scott M. Hammer, Michael S. Saag, Mauro Schechter et al. Treatment for adult hiv infection: 2006 recommendations of the international antiviral society usa-panel. *JAMA*, 296(7):827–843, 2006. URL: +http://dx.doi.org/10.1001/jama.296.7.827.

Song Han, Jeff Pool, John Tran & William Dally. Learning both weights and connections for efficient neural network. In *NIPS*, 2015.

Miguel A Hernán & James M Robins. Estimating causal effects from epidemiological data. *Journal of Epidemiology & Community Health*, 60(7):578–586, 2006a.

Miguel A Hernán & James M Robins. Instruments for causal inference: an epidemiologist's dream? *Epidemiology*, pages 360–372, 2006b.

Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011a.

Jennifer L. Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011b.

Geoffrey Hinton, Oriol Vinyals & Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Keisuke Hirano, Guido W Imbens & Geert Ridder. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica*, 71(4):1161–1189, 2003.

Sepp Hochreiter & Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Daniel G Horvitz & Donovan J Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47 (260):663–685, 1952.

Guido W Imbens. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and statistics*, 86(1):4–29, 2004.

E. Jang, S. Gu & B. Poole. Categorical Reparameterization with Gumbel-Softmax. *International Conference on Learning Representations (ICLR)*, 2017.

Nan Jiang & Lihong Li. Doubly robust off-policy evaluation for reinforcement learning. *arXiv preprint arXiv:1511.03722*, 2015.

Harry Joe. Relative entropy measures of multivariate dependence. *Journal of the American Statistical Association*, 84(405):157–164, 1989.

Fredrik D. Johansson, Uri Shalit & David Sontag. Learning representations for counterfactual inference. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 3020–3029. JMLR.org, 2016.

Alistair EW Johnson, Tom J Pollard, Lu Shen, L H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi & Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3, 2016a.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi & Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016b.

Michael I Jordan & Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

Leslie Pack Kaelbling, Michael L Littman & Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

Herman Kahn & Andy W Marshall. Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953.

Nathan Kallus, Xiaojie Mao & Madeleine Udell. Causal inference with noisy and missing covariates via matrix factorization. In *Advances in Neural Information Processing Systems*, pages 6921–6932, 2018.

Diederik P Kingma & Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende & Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3581–3589, 2014.

Diederik P Kingma & Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Achim Klenke. *Probability theory: a comprehensive course*. Springer Science & Business

Media, 2013.

Daphne Koller & Nir Friedman. *Probabilistic graphical models: principles and techniques.* MIT press, 2009.

Manabu Kuroki & Judea Pearl. Measurement bias and effect restoration in causal inference. *Biometrika*, 101(2):423–437, 2014.

Stephen L. Morgan & Chris Winship. *Counterfactuals and Causal Inference: Methods and Principles for Social Research.* 01 2007. ISBN 9781139465908.

Himabindu Lakkaraju, Stephen H Bach & Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *KDD*, 2016.

Robert J LaLonde. Evaluating the econometric evaluations of training programs with experimental data. *The American economic review*, pages 604–620, 1986.

Simone E. Langford, Jintanat Ananworanich & David A. Cooper. Predictors of disease progression in hiv infection: a review. *AIDS Research and Therapy*, 4(1):11, May 2007. ISSN 1742-6405. URL: `https://doi.org/10.1186/1742-6405-4-11`.

Zachary C. Lipton. The mythos of model interpretability. In *ICML Workshop on Human Interpretability in Machine Learning*, 2016.

Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel & Max Welling. Causal effect inference with deep latent-variable models. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6446–6456. Curran Associates, Inc., 2017.

Scott Lundberg & Su-In Lee. An unexpected unity among methods for interpreting model predictions. *arXiv preprint arXiv:1611.07478*, 2016.

Marina Lusic & Robert F Siliciano. Nuclear landscape of hiv-1 infection and integration. *Nature Reviews Microbiology*, 15(2):69, 2017.

David JC MacKay. *Information theory, inference and learning algorithms.* Cambridge university press, 2003.

D Maclaurin, D Duvenaud, M Johnson & RP Adams. Autograd: Reverse-mode differentiation of native python. `http://github.com/HIPS/autograd`, 2015.

Louis M Mansky & Howard M Temin. Lower in vivo mutation rate of human immunodeficiency virus type 1 than that predicted from the fidelity of purified reverse transcriptase. *Journal of virology*, 69(8):5087–5094, 1995.

A. Marco, F. Berkenkamp, P. Hennig, A. P. Schoellig, A. Krause, S. Schaal & S. Trimpe. Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with bayesian optimization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1557–1563, May 2017.

Marie C. McCormick, Jeanne Brooks-Gunn & Stephen L. Buka. Infant health and development program, phase iv, 2001-2004 [united states]. 2013.

Sophie Medam, Laurent Zieleskiewicz, Gary Duclos, Karine Baumstarck, Anderson Loundou, Julie Alingrin, Emmanuelle Hammad, Coralie Vigne, François Antonini

& Marc Leone. *Medicine*, 96(50), 12 2017.

Riccardo Miotto, Li Li, Brian A Kidd & Joel T Dudley. Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific Reports*, 6(26094), 2016.

Stephen L Morgan & Christopher Winship. *Counterfactuals and causal inference.* Cambridge University Press, 2015.

Lena M Napolitano. Sepsis 2018: definitions and guideline changes. *Surgical infections*, 19(2):117–125, 2018.

Saul B Needleman & Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

Yu Nishiyama, Abdeslam Boularias, Arthur Gretton & Kenji Fukumizu. Hilbert space embeddings of pomdps. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 644–653. AUAI Press, 2012.

Tsubasa Ochiai, Shigeki Matsuda, Hideyuki Watanabe & Shigeru Katagiri. Automatic node selection for deep neural networks using group lasso regularization. In *ICASSP*, 2017.

Dirk Ormoneit & Śaunak Sen. Kernel-based reinforcement learning. *Machine learning*, 49(2-3):161–178, 2002.

Sonali Parbhoo. *A Reinforcement Learning Design for HIV Clinical Trials.* University of the Witwatersrand, Faculty of Science, School of Computer Science, 2014.

Sonali Parbhoo, Jasmina Bogojeska, Maurizio Zazzi, Karin J. Metzner, Roger Kouyos, Huldrych Günthard, Finale Doshi-Velez & Volker Roth. A unified kernel and model-based learning approach to hiv therapy selection. *Arxiv preprint*, 2019.

Sonali Parbhoo, Jasmina Bogojeska, Maurizio Zazzi, Volker Roth & Finale Doshi-Velez. Combining kernel and model-based learning for HIV therapy selection. *In Proceedings of the AMIA Summit on Clinical Research Informatics (CRI)*, 2017.

Sonali Parbhoo, Omer Gottesman, Andrew Slavin Ross, Matthieu Komorowski, Aldo Faisal, Isabella Bon, Volker Roth & Finale Doshi-Velez. Improving counterfactual reasoning with kernelised dynamic mixing models. *PloS one*, 13(11):e0205839, 2018a.

Sonali Parbhoo, Mario Wieser & Volker Roth. Causal deep information bottleneck. *arXiv preprint arXiv:1807.02326*, 2018b.

David L Paterson, Susan Swindells et al. Adherence to protease inhibitor therapy and outcomes in patients with HIV infection. *Annals of Internal Medicine*, 133(1):21–30, 2000.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

Judea Pearl. *Causality.* Cambridge university press, 2009.

Judea Pearl. The do-calculus revisited. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 3–11. AUAI Press, 2012a.

Judea Pearl. On measurement bias in causal inference. *arXiv preprint arXiv:1203.3504*, 2012b.

Judea Pearl. The seven tools of causal inference with reflections on machine learning. Technical report, 2018.

Judea Pearl et al. Causal inference in statistics: An overview. *Statistics surveys*, 3: 96–146, 2009.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Xuefeng Peng, Yi Ding, David Wihl, Omer Gottesman, Matthieu Komorowski, Li-wei H. Lehman, Andrew Ross, Aldo Faisal & Finale Doshi-Velez. Improving sepsis treatment strategies by combining deep and kernel-based reinforcement learning. 01 2019.

J. Peters, D. Janzing & B. Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA, 2017.

Joelle Pineau, Arthur Guez, Robert Vincent, Gabriella Panuccio & Massimo Avoli. Treating epilepsy via adaptive neurostimulation: a reinforcement learning approach. *International journal of neural systems*, 19(04):227–240, 2009.

Gizem Polat, Rustem Anil Ugan, Elif Cadirci & Zekai Halici. Sepsis and septic shock: Current treatment strategies and new approaches. 2017.

Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80, 2000.

Mattia CF Prosperi, Michal Rosen-Zvi, André Altmann, Maurizio Zazzi, Simona Di Giambenedetto, Rolf Kaiser, Eugen Schülter, Daniel Struck, Peter Sloot, David A Van De Vijver et al. Antiretroviral therapy optimisation without genotype resistance testing: a perspective on treatment history based models. *PloS one*, 5(10):e13753, 2010.

Matthew Rabinowitz et al. Accurate prediction of hiv-1 drug response from the reverse transcriptase and protease amino acid sequences using sparse models created by convex optimization. *Bioinformatics*, 22(5):541–549, 2005.

Aniruddh Raghu, Matthieu Komorowski, Leo Anthony Celi, Peter Szolovits & Marzyeh Ghassemi. Continuous state-space models for optimal sepsis treatment-a deep reinforcement learning approach. *arXiv preprint arXiv:1705.08422*, 2017.

AD Revell, D Wang, R Harrigan, RL Hamers, AMJ Wensing, F Dewolf, M Nelson, A-M Geretti & BA Larder. Modelling response to hiv therapy without a genotype: an argument for viral load monitoring in resource-limited settings. *Journal of antimicrobial chemotherapy*, page dkq032, 2010.

Mélanie Rey. *Copula models in machine learning*. PhD thesis, University_of_Basel, 2015.

Mélanie Rey & Volker Roth. Meta-gaussian information bottleneck. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou & Kilian Q. Weinberger, editors, *NIPS*, pages 1925–1933, 2012.

Danilo Jimenez Rezende, Shakir Mohamed & Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Eric P. Xing & Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR.

Marco Tulio Ribeiro, Sameer Singh & Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *KDD*, 2016.

David L Robertson, JP Anderson, JA Bradac, JK Carr, B Foley, RK Funkhouser, F Gao, BH Hahn, ML Kalish, C Kuiken et al. Hiv-1 nomenclature proposal. *Science*, 288(5463):55–55, 2000.

Michal Rosen-Zvi, Andre Altmann, Mattia Prosperi, Ehud Aharoni, Hani Neuvirth, Anders Sönnerborg, Eugen Schülter, Daniel Struck, Yardena Peres, Francesca Incardona et al. Selecting anti-hiv therapies based on a variety of genomic and clinical factors. *Bioinformatics*, 24(13):i399–i406, 2008.

Paul R Rosenbaum & Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Paul R Rosenbaum & Donald B Rubin. Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American statistical Association*, 79(387):516–524, 1984.

Andrew Ross, Michael C Hughes & Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *IJCAI*, 2017a.

Andrew Ross, Isaac Lage & Finale Doshi-Velez. The neural lasso: Local linear sparsity for interpretable explanations. In *Workshop on Transparent and Interpretable Machine Learning in Safety Critical Environments, 31st Conference on Neural Information Processing Systems*, 2017b.

Stéphane Ross, Joelle Pineau, Brahim Chaib-draa & Pierre Kreitmann. A bayesian approach for learning and planning in partially observable markov decision processes. *Journal of Machine Learning Research*, 12(May):1729–1770, 2011.

Stéphane Ross, Joelle Pineau, Sébastien Paquet & Brahim Chaib-Draa. Online planning algorithms for pomdps. *Journal of Artificial Intelligence Research*, 32:663–704, 2008.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

Donald B Rubin. Bayesian inference for causal effects: The role of randomization. *The Annals of statistics*, pages 34–58, 1978.

Reuven Y Rubinstein. Simulation and the monte carlo method. Technical report, 1981.

Peter Schulam & Suchi Saria. Reliable decision support using counterfactual models. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1697–1708. Curran Associates, Inc., 2017.

Rebecca M. Seibert et al. A model for predicting lung cancer response to therapy. *International Journal of Radiation Oncology, Biology, Physics*, 2007.

Jasjeet S Sekhon. The Neyman-Rubin model of causal inference and estimation via matching methods. 2008.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh & Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391v3*, 2017.

Uri Shalit, Fredrik D. Johansson & David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In Doina Precup & Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3076–3085, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Manu Shankar-Hari, Michael Ambler, Viyaasan Mahalingasivam, Andrew Jones, Kathryn Rowan & Gordon D Rubenfeld. Evidence for a causal link between sepsis and long-term mortality: a systematic review of epidemiologic studies. *Critical care (London, England)*, 20, 2016. URL: `https://www.ncbi.nlm.nih.gov/pubmed/27075205`.

Edward H Simpson. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 13(2):238–241, 1951.

Mervyn Singer, Clifford S. Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R. Bernard, Jean-Daniel Chiche, Craig M. Coopersmith, Richard S. Hotchkiss, Mitchell M. Levy, John C. Marshall, Greg S. Martin, Steven M. Opal, Gordon D. Rubenfeld, Tom van der Poll, Jean-Louis Vincent & Derek C. Angus. The third international consensus definitions for sepsis and septic shock (sepsis-3). *JAMA*, 315(8):801–810, 02 2016.

Mervyn Singer et. al. The third international consensus definitions for sepsis and septic shock (sepsis-3). *Jama*, 315(8):801–810, 2016.

Sameer Singh, Marco Tulio Ribeiro & Carlos Guestrin. Programs as black-box explanations. *arXiv preprint arXiv:1611.07579*, 2016.

Yashik Singh. Machine learning to improve the effectiveness of anrs in predicting hiv drug resistance. *Healthcare informatics research*, 23(4):271–276, 2017.

M Eugenia Socias et al. Acute retroviral syndrome and high baseline viral load are predictors of rapid HIV progression among untreated Argentinean seroconverters. *Journal of the International AIDS Society*, 14(1):40, 2011. URL: `http://dx.doi.org/10.1186/1758-2652-14-40`.

Yang Song, Jun Zhu & Yong Ren. Kernel bayesian inference with posterior regularization. In *Advances in Neural Information Processing Systems*, pages 4763–4771, 2016.

Jerzy Splawa-Neyman. Sur les applications de la théorie des probabilités aux experiences agricoles: Essai des principes. *Roczniki Nauk Rolniczych*, 10:1–51, 1923.

Jerzy Splawa-Neyman. On the application of probability theory to agricultural experiments. essay on principles. section 9. *Statistical Science*, 5(4):465–472, 1990.

Jonathan R Studnek, Melanie R Artho, Craymon L Garner Jr & Alan E Jones. The impact of emergency medical services on the ed care of severe sepsis. *The American journal of emergency medicine*, 30(1):51–56, 2012.

Somnuek Sungkanuparph et. al. Options for a second-line antiretroviral regimen for hiv type 1-infected patients whose initial regimen of a fixed-dose combination of stavudine, lamivudine, and nevirapine fails. *Clinical Infectious Diseases*, 44(3):447–452, 2007.

Richard S Sutton & Andrew G Barto. *Introduction to reinforcement learning*, volume 135. Cambridge: MIT Press, 1998.

Erik Talvitie. Model regularization for stable sample rollouts. In *UAI*, pages 780–789, 2014.

Erik Talvitie. Self-correcting models for model-based reinforcement learning. In *AAAI*, pages 2597–2603, 2017.

Philip Thomas & Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148, 2016.

Melanie A. Thompson, Judith A. Aberg, Pedro Cahn et al. Antiretroviral treatment of adult hiv infection: 2010 recommendations of the international antiviral society usa-panel. *JAMA*, 304(3):321–333, 2010. URL: +http://dx.doi.org/10.1001/jama.2010.1004.

Melanie A. Thompson, Judith A. Aberg, Hoy Jennifer F. et al. Antiretroviral treatment of adult hiv infection: 2012 recommendations of the international antiviral society usa-panel. *JAMA*, 308(4):387–402, 2012. URL: +http://dx.doi.org/10.1001/jama.2012.7961.

Naftali Tishby, Fernando C Pereira & William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Naftali Tishby & Noga Zaslavsky. Deep learning and the information bottleneck principle. *CoRR*, abs/1503.02406, 2015.

UNAIDS. AIDS by the numbers, 2015. URL: http://www.unaids.org/en/resources/documents/2015/AIDS_by_the_numbers_2015.

Théophane. Weber et al. Imagination-augmented agents for deep reinforcement learning. *arXiv preprint arXiv:1707.06203*, 2017.

Aleksander Wieczorek, Mario Wieser, Damian Murezzan & Volker Roth. Learning Sparse Latent Representations with the Deep Copula Information Bottleneck. *International Conference on Learning Representations (ICLR)*, 2018.

Mike Wu, Michael C Hughes, Sonali Parbhoo, Maurizio Zazzi, Volker Roth & Finale Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Patrick G. Yeni, Scott M. Hammer, Charles J. Carpenter et al. Antiretroviral treatment for adult hiv infection in 2002: Updated recommendations of the international antiviral society usa-panel. *JAMA*, 288(2):222–235, 2002. URL: +http://dx.doi.org/10.1001/jama.288.2.222.

Patrick G. Yeni, Scott M. Hammer, Martin S. Hirsch et al. Treatment for adult hiv infection: 2004 recommendations of the international antiviral society usa-panel. *JAMA*, 292(2):251–265, 2004. URL: `+http://dx.doi.org/10.1001/jama.292.2.251`.

Maurizzio Zazzi et al. Predicting response to antiretroviral treatment by machine learning: The euresist project. *Intervirology*, 55(2):123–127, 1 2012. ISSN 0300-5526.

Yufan Zhao, Michael R Kosorok & Donglin Zeng. Reinforcement learning design for cancer clinical trials. *Statistics in medicine*, 28(26):3294–3315, 2009.

Hui Zou & Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2): 301–320, 2005.