Causal Inference Methods for Supporting, Understanding, and Improving Decision-Making



Ioana Bica

Wolfson College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2022

Acknowledgements

Personal

This thesis would not have been possible without the help and support of my advisor, my colleagues, my family, and my friends to which I am extremely grateful!

First and foremost, I would like to thank my advisor, Professor Mihaela van der Schaar for her guidance throughout my DPhil and for pushing me to achieve my best. Thank you for encouraging me to go beyond my comfort zone, to think deeply about research problems, to be creative, and to think big and outside the box! I am also grateful for your research vision which has significantly shaped this thesis and for always providing me with new ideas, insights, and invaluable advice which have had a profound impact on my development as a researcher.

I would also like to thank the members of the van der Schaar Lab for the countless insightful research discussions and engaging group meetings from which I have learned so much. To begin with, I would like to thank Ahmed M. Alaa for his invaluable support and guidance throughout the first year of my DPhil. Moreover, I am also grateful to my close collaborators James Jordon and Dan Jarrett. It has been a pleasure to brainstorm research ideas with you, learn from you and, work together on several research papers! I would also like to thank my other collaborators and co-authors from the lab: Trent Kyono, Alihan Hüyük, Jeroen Berrevoets, Alicia Curth, Jinsung Yoon, Yao Zhang, Zhaozhi Qian, Alex Chan, Changhee Lee, Evgeny Saveliev, Jonathan Crabbé, Nabeel Seedat, Fergus Imrie and Kamilė Stankevičiūtė. Being able to discuss ideas and work with all of you has been one of my favorite parts of the DPhil and it has been a privilege to have such talented people as my colleagues.

I am also deeply grateful to Professor Pietro Lió for advising my undergraduate and master's research projects which motivated me to pursue a DPhil. Thank you for your unwavering support, advice and for always ensuring I was doing well!

Furthermore, I would like to thank my advisers and collaborators from DeepMind who provided me with a wonderful research internship experience from which I have learned a lot and which has helped me grow as a researcher. Thank you, Jovana Mitrović, for being an amazing and supportive host and for your invaluable advice and guidance which has contributed greatly to the success of my internship project. I am also extremely grateful to Nenad Tomašev, Brian McWilliams, Răzvan Paşcanu, Lars Buesing, and Charles Blundell for their helpful advice throughout my internship.

Moreover, I would like to thank Professor Pawan Mudigonda and Professor Victor Adrian Prisacariu for their support in helping me to continue my DPhil at the University of Oxford.

In addition, I am grateful to my DPhil viva examiners, Professor Alison Noble and Dr Danielle Belgrave for the insightful questions and conversations during my viva examination and for their suggestions for future work.

I am also grateful to my friend Malavika Nair, who has been by my side throughout my academic journey of becoming a researcher since my undergraduate studies and who has provided me with invaluable advice and support. In addition, I would also like to thank my friend Bianca Danciu for always being there for me and for always believing in me.

Last but not least, I am eternally grateful to my parents, Ion and Elena Bica, for their unconditional love, support, and encouragement. This thesis would not have been possible without all of the opportunities they have provided me with throughout my life. I am also deeply grateful to my little sister, Ana-Maria Bica, for her love and support, for taking care of me, for putting up with me, and for always being there for me.

Institutional

I am incredibly grateful to the Alan Turing Institute for offering me a Doctoral Studentship which has funded my DPhil and for providing me with the computational resources needed for my research projects.

Abstract

Causality and the ability to reason about cause-and-effect relationships are central to decision-making. This thesis contributes to the area of causal inference by proposing new machine learning methods that can be used for supporting, understanding, and improving decision-making, with a focus on the healthcare setting.

Firstly, we introduce several causal inference tools for supporting decision-making by estimating the causal effects of interventions (treatments) from observational data, such as electronic health records. We begin by addressing the under-explored problem of estimating counterfactual outcomes for continuous-valued interventions and propose a method based on generative adversarial networks that achieves state-of-the-art performance and that can help us choose both the correct treatment and dosage for each patient. Then, we shift our attention to the temporal setting where we develop a sequence-to-sequence model that uses domain adversarial training to handle time-dependent confounding and that can help us determine the best sequence of treatments for each patient. Moreover, we introduce the first method that can handle the presence of multi-cause hidden confounders in temporal data. By taking advantage of the dependencies in the treatment assignments over time, our method learns latent variables that can be used as substitutes for the hidden confounders.

Secondly, we integrate counterfactual reasoning into batch inverse reinforcement learning to develop a method for better understanding the decision-making behaviour of experts by modelling their reward functions in terms of preferences over 'what-if' (counterfactual) outcomes. We show that this allows us to obtain an interpretable parameterization of the experts' decision-making process and subsequently uncover the trade-offs and preferences associated with their actions.

Thirdly, we improve the robustness of decision-making by proposing a new model for batch imitation learning that incorporates causal structure into the learnt imitation policy. By ensuring that the imitation policy only depends on the causal parents of the actions, we learn a decision-making guideline that is robust to spurious correlations and that generalizes well to new environments.

Overall, this thesis introduces methodological advances in machine learning capable of reasoning about cause-and-effect relationships that can enable us to improve delivery of personalized care for patients, support clinical decision-making and build a more transparent account of clinical practice. We provide a discussion highlighting the challenges of incorporating such methods into practice and include suggestions for future work in this direction.

Contents

List of Figures x

List of Tables xii

List of Notation xiv

List of Abbreviations and Definitions xxii

1 Introduction 1
- 1.1 Causal Inference and Machine Learning . . . 2
- 1.2 Thesis Outline and Contributions . . . 5
- 1.3 Publications . . . 11
  - 1.3.1 Relevant Publications . . . 11
  - 1.3.2 Other Publications . . . 12

2 Background and Literature Review 14
- 2.1 From Randomized Trials to Patient Observational Data . . . 15
- 2.2 Individualized Treatment Recommendations in the Static Setting . . . 18
  - 2.2.1 Challenges in Using Observational Data for Estimating Individualized Treatment Effects . . . 20
  - 2.2.2 Modelling Choices and Methods . . . 23
  - 2.2.3 Individualized Dose-Response Estimation . . . 28
- 2.3 Individualized Effects of Time-dependent Treatments . . . 30
  - 2.3.1 Challenges in Using Observational Data for Treatment Decisions in the Time-Series Setting . . . 32
  - 2.3.2 Modelling Choices and Methods for Estimating Individualized Effects of Time-Dependent Treatments . . . 33
- 2.4 Evaluation of Causal Inference Models . . . 36

3 Estimating the Causal Effects of Continuous-valued Interventions 40
- 3.1 Introduction . . . 41
- 3.2 Related Work . . . 44
- 3.3 Problem Formulation . . . 45

3.4 SCIGAN . . . 46
3.4.1 Counterfactual Generator . . . 46
3.4.2 Counterfactual Discriminator . . . 47
3.4.3 Inference Network . . . 49
3.4.4 Theoretical Analysis . . . 49
3.5 Architecture . . . 49
3.5.1 Generator Architecture . . . 50
3.5.2 Hierarchical Discriminator Architectures . . . 50
3.6 Evaluation . . . 51
3.6.1 Experimental Setup . . . 52
3.6.2 Source of Gain . . . 53
3.6.3 Benchmarks Comparison . . . 54
3.6.4 Discrete Dosages . . . 54
3.6.5 Treatment and Dosage Selection Bias . . . 55
3.7 Conclusion . . . 55

4 Estimating the Causal Effects of Treatments in the Temporal Setting 57
4.1 Counterfactual Recurrent Network . . . 58
4.1.1 Introduction . . . 58
4.1.2 Related Work . . . 62
4.1.3 Problem Formulation . . . 63
4.1.4 Counterfactual Recurrent Network . . . 64
4.1.5 Adversarially Balanced Representation over Time . . . 66
4.1.6 Experiments . . . 68
4.1.7 Conclusion . . . 73
4.2 Time Series Deconfounder . . . 75
4.2.1 Introduction . . . 75
4.2.2 Related Work . . . 77
4.2.3 Problem Formulation . . . 79
4.2.4 Time Series Deconfounder . . . 81
4.2.5 Factor Model over Time in Practice . . . 85
4.2.6 Experiments on Synthetic Data . . . 86
4.2.7 Experiments on MIMIC III . . . 90
4.2.8 Conclusion . . . 92

5 Using Counterfactuals to Explain Sequential Decision-Making 94
5.1 Introduction . . . 95
5.2 Related Works . . . 98
5.3 Problem Formulation . . . 99
5.4 Batch Inverse Reinforcement Learning using Counterfactuals . . . 102
5.4.1 Counterfactual $\mu-$learning . . . 103
5.4.2 Finding Optimal Policy for Given Reward Weights . . . 104
5.4.3 Counterfactual Inverse Reinforcement Learning (CIRL) . . . 105
5.5 Experiments . . . 106
5.5.1 Extracting Different Types of Expert Behaviour . . . 107
5.5.2 Case Study on Real-world Dataset: MIMIC III . . . 109
5.6 Discussion . . . 110

6 Using Causal Structure to Improve the Robustness of Decision-Making 113
6.1 Introduction . . . 114
6.2 Related Works . . . 117
6.3 Problem Formalism . . . 119
6.3.1 Imitation Learning . . . 119
6.3.2 Imitation Learning from Multiple Environments . . . 120
6.4 Invariant Causal Imitation Learning for Generalization across Domains 123
6.4.1 Learning Invariant Causal Representations . . . 124
6.4.2 Matching Expert Behaviour in a Strictly Batch Setting . . . 126
6.5 Experiments . . . 127
6.5.1 Evaluation on OpenAI Gym . . . 128
6.5.2 Evaluation on MIMIC-III . . . 129
6.6 Discussion . . . 131

7 Discussion and Conclusion 133
7.1 Summary of Contributions . . . 134
7.2 Relationship to Other Publications . . . 136
7.3 Limitations and Future Directions . . . 139

Appendices

A Appendix for Chapter 3 142
A.1 Expanded Related Works . . . 143
A.2 Proofs of Theoretical Results . . . 145
A.3 Counterfactual Generator Pseudo-code . . . 148
A.4 Inference Network . . . 150
A.5 Architecture . . . 150

A.6 Single Discriminator Model . . . 151
A.7 Ablation Studies Architectures . . . 153
A.8 Dataset Descriptions . . . 153
A.9 Dosage Bias . . . 154
A.10 Benchmarks . . . 155
A.11 Metrics . . . 156
A.12 Additional Results . . . 157

B Appendix for Chapter 4 **162**
B.1 Counterfactual Recurrent Network . . . 163
B.1.1 Extended Related Work . . . 163
B.1.2 Assumptions . . . 164
B.1.3 Time-dependent Confounding . . . 165
B.1.4 Proof of Theorem 4.1.1 . . . 167
B.1.5 Training Procedure for CRN . . . 168
B.1.6 Pharmacokinetic-Pharmacodynamic Model of Tumour Growth . 170
B.1.7 Marginal Structural Models . . . 171
B.1.8 Recurrent Marginal Structural Networks . . . 172
B.1.9 Baseline RNN and Linear Model . . . 173
B.1.10 Hyperparameter Optimization for CRN . . . 174
B.1.11 Full Results for Counterfactual Prediction . . . 176
B.1.12 Test Set Generation for Evaluating Timing of Treatment . . . 177
B.1.13 Results on Factual Prediction on MIMIC III . . . 178
B.2 Time Series Deconfounder . . . 180
B.2.1 Proof for Theorem 4.2.1 . . . 180
B.2.2 Implementation Details for the Factor Model . . . 183
B.2.3 Baselines for Evaluating Factor Model . . . 184
B.2.4 Outcome Models . . . 186
B.2.5 Additional Results . . . 188
B.2.6 Discussion . . . 190

C Appendix for Chapter 5 **192**
C.1 Additional Related Works . . . 193
C.2 Counterfactual Estimation and Time-dependent Confounding . . . 193
C.3 Counterfactual μ-learning . . . 195
C.4 Finding Optimal Policy for Given Reward Weights . . . 196
C.5 Data Simulation . . . 198
C.6 Implementation Details CIRL . . . 200
C.7 Implementation Details Benchmarks . . . 200

D Appendix for Chapter 6 **202**
D.1 Structure of Environment and Interventions 203
D.2 Train Energy Based Model 204
D.3 Causal Features for Imitation using Invariant Risk Minimization 205
D.4 Algorithm 206
D.5 Experimental Details 206
D.6 Additional Experiments 211

References **214**

List of Figures

2.1 Estimating individualized treatment effects for personalized treatment recommendations. . . . 17
2.2 Training causal inference methods on observational patient data from EHRs and to estimate individualized patient outcomes under different possible treatment options. . . . 20
2.3 Using observational patient data to estimate individualized causal effects. 24
2.4 Different approaches for estimating individualized patient outcomes using machine learning. . . . 26
2.5 Causal inference methods for estimating individualized effects of time-dependent treatments need to model the patient's history in order to make counterfactual predictions of potential outcomes for possible sequences of future treatments. . . . 32

3.1 Overview of GAN framework used as part of SCIGAN for learning the distribution of the counterfactual outcomes. . . . 43
3.2 Overview of SCIGAN for the setting with two treatments (w_f/w_{cf} being the factual/counterfactual treatment). . . . 47
3.3 Architecture of our generator and discriminators. . . . 51
3.4 Comparison between SCIGAN and GANITE. . . . 54
3.5 Performance of the 4 methods on datasets with varying bias levels. . . . 55

4.1 Applicability of CRN in cancer treatment planning. . . . 61
4.2 CRN architecture. . . . 65
4.3 Training procedure for building balancing representation. . . . 67
4.4 Results for prediction of patient counterfactuals. . . . 70
4.5 TSNE embedding of the balancing representation . . . 71
4.6 Graphical factor model. . . . 82
4.7 Proposed factor model implementation. . . . 86
4.8 Predictive checks over time . . . 88
4.9 Results for deconfounding the one-step ahead estimation of treatment responses in two outcome models: (a) Marginal Structural Models (MSM) and (b) Recurrent Marginal Structural Networks (R-MSN). . . . 89

5.1 Explaining decision-making behaviour in terms of preferences over "what if" outcomes. . . . 97
5.2 Counterfactual inverse reinforcement learning (CIRL). . . . 103
5.3 Reward weights recovered by benchmarks over 10 runs. . . . 108
5.4 Radar plot of reward weights magnitude for assigning antibiotics. . . . 110
5.5 Different components part of our method for obtaining interpretable parametrizations of decision making and their limitations. . . . 111

6.1 Causal diagram for the structure of environments. . . . 116
6.2 Causal diagram illustrating temporal dependencies between causal parents of action $\{x_t^1, x_t^2\}$ and noise variables $\{x_t^3\}$. . . . 123
6.3 Block diagram of ICIL model. . . . 125
6.4 Evaluation on OpenAI gym environments. . . . 128

A.1 Overview of the single discriminator architecture. . . . 152
A.2 Architecture of the generator without multitask and discriminator without the invariant/equivariant layers used in the ablation studies. . . . 153
A.3 Performance of single vs. hierarchical discriminator when increasing the number of dosage samples (n_w) on News dataset. . . . 157
A.4 Performance of single vs. hierarchical discriminator when increasing the number of dosage samples (n_w) on TCGA dataset. . . . 158
A.5 Comparison between SCIGAN and GANITE in the discrete dosage set-up. 160
A.6 Additional performance metrics of the 4 methods on datasets with varying bias levels on TCGA dataset . . . 161

B.1 Causal graphs for 2-step time-varying exposures [14]. . . . 165
B.2 Results for prediction of patient counterfactuals for multiple steps ahead. 176
B.3 (a) Proposed factor model using a recurrent neural network with multitask output and variational dropout. (b) Alternative design without multitask output. (c) Factor model using an MLP (shared across timestep) and multitask output. . . . 185
B.4 Results for deconfounding one-step ahead estimation of treatment responses in two outcome models: (a) Marginal Structural Models (MSM) and (b) Recurrent Marginal Structural Networks (R-MSN). . . . 188

D.1 Evaluating the impact of the different loss components on overall performance for the CartPole control task. . . . 212
D.2 Train vs. test performance on CartPole. . . . 213
D.3 Robustness to increasing the number of noise variables. . . . 213

List of Tables

2.1 Different types of observational data, treatments and outcomes that can be used to train causal inference methods to estimate treatment effects. . 19
2.2 Summary of machine learning methods for estimating individualized treatment effects. 27
2.3 Summary of machine learning methods for estimating individualized treatment effects of time-varying treatments. 35

3.1 Dose response curves used to generate semi-synthetic outcomes for patient features $\mathbf{x}$. 52
3.2 Source of gain analysis for our model on the TCGA. 53
3.3 Performance of individualized treatment-dose response estimation. . . . 54

4.1 Results for recommending the correct treatment and timing of treatment. 72
4.2 Average RMSE and standard error for predicting the effect of antibiotics, vassopressors and mechanical ventilator on patient covariates. 91

5.1 Comparison of proposed method (batch, counterfactual IRL) with related works. 99
5.2 Mean reward and standard deviation for running learnt policy in the environment. 109
5.3 Average accuracy and standard deviation for action matching. 109
5.4 Accuracy on matching expert actions. 110

6.1 Comparison of our proposed ICIL method with related works. 118
6.2 Evaluation on MIMIC-III in terms of action-matching. 130

A.1 Summary description of datasets. 154
A.2 Hyperparameters search range for SCIGAN. 156
A.3 Hyperparameters used for obtaining results. 156
A.4 Performance of individualized treatment-dose response estimation on three datasets. 158
A.5 Performance of SCIGAN and the benchmarks when we increase the number of treatments to 6 and 9. 159
A.6 Sample efficiency analysis for MIMIC 159

A.7 Performance of individualized treatment-dose response when mixing treatment with no-treatment options. 161

B.1 Hyperparameter search range for RMSNs propensity networks and encoder. 173
B.2 Hyperparameter search range for RMSNs decoder. 174
B.3 Hyperparameter search range for baseline RNN model. 174
B.4 Hyperparameter search range for CRN encoder. 175
B.5 Optimal hyperparameters for the CRN encoder when different degrees of time-dependent confounding are applied in the model of tumour growth. 175
B.6 Optimal hyperparameters for the CRN decoder when different degrees of time-dependent confounding are applied in the model of tumour growth. 176
B.7 Normalized RMSE for one-step-ahead prediction of counterfactuals. . . 177
B.8 Normalized RMSE for τ-step-ahead prediction of counterfactuals. . . . 177
B.9 RMSE for τ-step-ahead prediction of factual outcomes on MIMIC III. . 179
B.10 Hyperparameter search range for the proposed factor model 184
B.11 Optimal hyperparameters for the factor model when different amounts of hidden confounding are applied in the synthetic dataset. 184
B.12 Hyperparameter search range for factor model without multitask (Figure B.3(b)). 185
B.13 Hyperparameter search range for MLP factor model. Figure B.3(c)) . . 185
B.14 Hyperparameters used for R-MSN. 188
B.15 Average RMSE $\times 10^2$ and the standard error for predicting the effect of chemotherapy and radiotherapy on the tumour volume. 189
B.16 Average RMSE $\times 10^2$ and the standard error for predicting the effect of antibiotics, vasopressors, and mechanical ventilator on white blood cell count. 190

C.1 Hyperparameters for training μ-network for estimating feature expectations. 197
C.2 Hyperparameters used for training Q-network to find optimal policy for a setting of the reward weights. 198
C.3 Hyperparameters used for training Q-network to solve the simulated environment. 199
C.4 Hyperparameter search range for Counterfactual Recurrent Network. . . 200
C.5 Hyperparameter search range for the RNN. 201
C.6 Hyperparameter search range for MLP. 201

D.1 Environment details. The random and demonstrator performances are averaged over 1,000 episodes roll-outs. 209

List of Notation

Chapter 2

$\mathcal{X}$	Feature space.
$\mathcal{W}$	Treatment space.
$\mathcal{D}$	Dosage space.
$\mathcal{Y}$	Outcome space.
$X \in \mathcal{X}$	Features (random variable).
$W \in \mathcal{W}$	Treatment (random variable).
$D \in \mathcal{D}$	Dosage (random variable).
$Y \in \mathcal{Y}$	Observed outcome (random variable).
$Y(w, d)$	Potential outcome for treatment w and dosage d.
$T(x)$	Conditional average treatment effect for features $X = x$.
f_w	Response surface for treatment w.
μ	Individualised dose-response function.
t	Timestep.
V	Baseline covariates.
X_t	Features at timestep t.
W_t	Treatment at timestep t.
Y_t	Outcome at timestep t.
H_t	History of covariates and treatments until timestep t.
$Y_{t+\tau}(w(t, t+\tau-1))$	Potential outcome observed at timestep $t+\tau$ for the possible sequence of treatments $w(t, t+\tau-1)$ given from timesteps t to $t+\tau-1$.

Chapter 3

$\mathcal{X}$	Feature space.
$\mathcal{Y}$	Outcome space.
$\mathcal{W} = \{w_1, ..., w_k\}$	Set of treatments.
$\mathcal{D}_w$	Dosage space for treatment $w \in \mathcal{W}$.
$\mathcal{T}$	Intervention space.
$\mathcal{Z}$	Noise space.
$\mathcal{H}$	Latent space.
N	Number of samples in the dataset.
$\mathbf{X} \in \mathcal{X}$	Features (random variable).
$Y : \mathcal{T} \to \mathcal{Y}$	Potential outcome function (function-valued random variable).
$T_f = (W_f, D_f)$	Factual/observed intervention (treatment-dosage pair) (random variable).
$Y_f \in \mathcal{Y}$	Outcome corresponding to the observed intervention ($Y_f = Y(W_f, D_f)$).
μ	Individualised dose-response function.
$\mathbf{G}$	Generator.
$\mathbf{Z}$	Random noise (for input to generator) (random variable).
$\hat{Y}_{cf} : \mathcal{T} \to \mathcal{Y}$	Counterfactual outcome function induced by $\mathbf{G}$.
$\mathbf{D}$	Discriminator.
$\tilde{\mathcal{D}}_w = \{D_i^w\}_{i=1}^{n_w}$	Random (finite) subset of $\mathcal{D}_w$.
n_w	Size of $\tilde{\mathcal{D}}_w$ (i.e. number of dosage levels passed to discriminator for treatment $w \in \mathcal{W}$).
$\tilde{\mathbf{Y}}_w = (D_i^w, \tilde{Y}_i^w)_{i=1}^{n_w}$	Vector of dosage-outcome pairs generated by $\mathbf{G}$ (and Y_f) using $\tilde{\mathcal{D}}_w$.
$\mathbf{D}_{\mathcal{W}}$	Treatment discriminator.
$\mathbf{D}_{\mathcal{W}}^w$	Output of treatment discriminator $\mathbf{D}_{\mathcal{W}}$ corresponding to treatment w.
$\mathbf{D}_w$	Dosage discriminator for treatment $w \in \mathcal{W}$.
$\mathbf{D}_H$	Hierarchical discriminator defined by combining $\mathbf{D}_{\mathcal{W}}$ and $\mathbf{D}_w$.
$\mathbf{I}$	Inference network.

$\mathcal{L}$	GAN loss.
$\mathcal{L}_S$	Supervised loss.
$\mathcal{L}_\mathcal{W}$	Treatment discriminator loss.
$\mathcal{L}_d$	Dosage discriminator loss.
$\mathbf{v}_1^w, \mathbf{v}_2^w, \mathbf{v}_3^w$	Set of parameters used in the data simulation.
α	Dosage selection bias used in the semi-synthetic data simulation.
κ	Treatment selection bias used in the semi-synthetic data simulation.

Chapter 4.1

$\mathcal{X}_t$	Feature space for timestep t.
$\mathcal{A} = \{A_1, \ldots, A_K\} \in \mathcal{A}$	Set of treatments for each timestep.
$\mathcal{Y}_{t+1}$	Outcome space.
$\mathcal{V}$	Baseline covariates space.
$\mathcal{D}$	Dataset.
N	Number of patients in the dataset.
t	Timestep.
$\mathbf{X}_t \in \mathcal{X}_t$	Covariates observed at timestep t.
$\mathbf{A}_t \in \mathcal{A}$	Treatment at timestep t.
$\mathbf{Y}_{t+1} \in \mathcal{Y}_{t+1}$	Outcome at timestep $t+1$.
$\mathbf{V} \in \mathcal{V}$	Baseline covariates.
$\mathbf{Y}[\bar{\mathbf{a}}]$	Potential outcome, either factual or counterfactual, for a possible course of treatment $\bar{\mathbf{a}}$.
$\bar{\mathbf{X}}_t = (\mathbf{X}_1, \ldots, \mathbf{X}_t)$	History of the patient covariates until timestep t.
$\bar{\mathbf{A}}_t = (\mathbf{A}_1, \ldots, \mathbf{A}_t)$	History of treatment assignments until timestep t.
$\bar{\mathbf{H}}_t = (\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}, \mathbf{V})$	History until timestep t of the patient covariates $\bar{\mathbf{X}}_t$, treatment assignments $\bar{\mathbf{A}}_t$ and static features $\mathbf{V}$.
$\bar{\mathbf{a}}(t, t+\tau-1)$	Possible sequence of treatments from timestep t just until before the potential outcome $\mathbf{Y}_{t+\tau}[\bar{\mathbf{a}}(t, t+\tau-1)]$ is observed.

Φ	Representation function that maps the patient history $\bar{\mathbf{H}}_t$ to a representation space $\mathcal{R}$.
θ_r	Parameters of representation function Φ.
G_a	Treatment classifier.
θ_a	Parameters of treatment classifier G_a.
G_y	Outcome predictor.
θ_y	Parameters of outcome predictor G_y.
$\mathcal{L}_{t,a}$	Treatment loss at timestep t.
$\mathcal{L}_{t,y}$	Outcome loss at timestep t.
$\mathcal{L}_t$	Overall loss at timestep t.
λ	Hyperparameter that controls the trade-off between treatment discrimination and outcome prediction.
γ_c, γ_r	Parameters controlling the amount of time-dependent confounding bias used in the synthetic data simulation.

Chapter 4.2

$\mathcal{X}_t$	Feature space for timestep t.
$\mathcal{A}_t$	Treatment space for timestep t.
$\mathcal{Y}_{t+1}$	Outcome space for time $t+1$.
$\mathcal{V}$	Baseline covariates space.
$\mathcal{D}$	Dataset.
N	Number of patients in the dataset.
t	Timestep.
k	Number of treatments.
$\mathbf{X}_t \in \mathcal{X}_t$	Covariates observed at timestep t (random variable).
$\mathbf{A}_t = [A_{t1} \dots A_{tk}] \in \mathcal{A}_t$	Treatment at timestep t (random variable).
$\mathbf{Y}_{t+1} \in \mathcal{Y}_{t+1}$	Outcome at timestep $t+1$ (random variable).
$\mathbf{Z}_t$	Latent variable used as substitute for the hidden confounders at timestep t.
$\mathbf{Y}(\bar{\mathbf{a}}_{\geq t})$	Potential outcome, either factual or counterfactual, for a possible course of treatment $\bar{\mathbf{a}}_{\geq t}$.

$\bar{\mathbf{X}}_t = (\mathbf{X}_1, \ldots, \mathbf{X}_t)$ . . History of the patient covariates until timestep t.

$\bar{\mathbf{A}}_t = (\mathbf{A}_1, \ldots, \mathbf{A}_t)$ . . History of treatment assignments until timestep t.

$\bar{\mathbf{Z}}_t = (\mathbf{Z}_1, \ldots, \mathbf{Z}_t)$ . . . History of the latent variables inferred as substitutes for the hidden confounders until timestep t.

$\bar{\mathbf{H}}_t = (\bar{\mathbf{A}}_t, \bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t)$ . . History until timestep t of the treatment assignments $\bar{\mathbf{A}}_t$, patient covariates $\bar{\mathbf{X}}_t$, and latent variables $\bar{\mathbf{Z}}_t$.

τ . . . . . Observed patient trajectory.

$\bar{\mathbf{a}}_{\geq t}$ . . . . . Possible treatment plan that starts at timestep t and consists of a sequence of treatments that ends just before the patient outcome $\mathbf{Y}(\bar{\mathbf{a}}_{\geq t})$ is observed.

$\theta_{1:k}$ . . . . . Factor model parameters.

γ_A . . . . . Parameter controlling the amount of hidden confounding applied to the treatment in the synthetic data simulation.

γ_Y . . . . . Parameter controlling the amount of hidden confounding applied to the outcome in the synthetic data simulation.

D_Z . . . . . Dimensionality of inferred hidden confounders.

Chapter 5

$\mathcal{X}$ . . . . . Feature space.

$\mathcal{A}$ . . . . . Action space (finite).

$\mathcal{H}$ . . . . . History space.

$X_t \in \mathcal{X}$ . . . . . Features observed at timestep t (random variable).

$A_t \in \mathcal{A}$ . . . . . Action taken at timestep t (random variable).

x_t . . . . . Realization of X_t.

a_t . . . . . Realization of A_t.

$h_t = (x_{0:t}, a_{0:t-1})$ . . . Realization of the history $H_t \in \mathcal{H}$ of patient observations and actions until timestep t.

$\pi : \mathcal{H} \times \mathcal{A} \to [0, 1]$ . . Stationary stochastic policy, where $\pi(a \mid h)$ indicates the probability of choosing action $a \in \mathcal{A}$ given history $h \in \mathcal{H}$ and $\sum_{a \in \mathcal{A}} \pi(a \mid h) = 1$.

$R : \mathcal{H} \times \mathcal{A} \to \mathbb{R}$ . . . Reward function where $R(h, a)$ represents the reward for taking action $a \in \mathcal{A}$ given history $h \in \mathcal{H}$.

$V : \mathcal{H} \to \mathbb{R}$	Value function of a policy π.
γ	Discount factor.
$Q : \mathcal{H} \times \mathcal{A} \to \mathbb{R}$ . . .	Action-value function.
$\mathcal{D}$	Batch observational dataset.
N	Number of patient trajectories in the dataset.
t	Timestep.
T	Maximum timestep of a trajectory.
ζ	Patient trajectory.
π_E	Expert policy.
w	Reward weights.
$\phi(h_t, a_t)$	Feature map.
$w^\star$	Reward weights of expert policy.
R^*	Reward function of expert policy.
μ^π	Feature expectation of policy π.
$Y[a]$	Potential outcome, either factual or counterfactual, for treatment (action) $a \in \mathcal{A}$.
h'	Next history after taking action a given history h.
α	Hyperparameter denoting the learning rate for the counterfactual μ-learning algorithm.
θ	Parameters of recurrent neural network used to approximate the feature expectations μ^π.
$\mathcal{L}$	Loss function used to train the recurrent neural network that approximates the feature expectations μ^π.

Chapter 6

$\mathcal{X}$	Feature space.
$\mathcal{A}$	Action space (finite).
$\mathcal{S}$	State representation space.
$\mathcal{Z}$	Noise representation space.
$\mathcal{E}$	Set of environments.
$x \in \mathcal{X}$	Observation.

$a \in \mathcal{A}$	Action.
$T \in \Delta(\mathcal{X})^{\mathcal{X}\times\mathcal{A}}$	Transition function.
$r \in \mathbb{R}^{\mathcal{X}\times\mathcal{A}}$	Reward function.
γ	Discount factor.
$e = (\mathcal{X}, \mathcal{A}, T, r, \gamma)$	Environment.
$\pi \in \Delta(\mathcal{A})^{\mathcal{X}}$	Policy.
ρ_π	Occupancy measure of policy π.
π_D	Demonstrator (expert) policy.
ρ_D	Occupancy measure of demonstrator policy π_D.
$\mathcal{D}$	Dataset of trajectories from a demonstrator policy π_D.
N	Number of trajectories in the dataset.
t	Timestep.
$\tau = (x_t, a_t, x_{t+1})_{t=0,\dots}$	Trajectory consisting of a sequence of observation, action, next observation tuples that are sampled as $a_t \sim \pi_D(\cdot \mid x_t)$ and $x_{t+1} \sim T(\cdot \mid x_t, a_t)$.
$R(\pi)$	Risk of policy π.
$\mathcal{L}$	Loss function.
$\mathcal{M}$	Family of environments.
$\mathcal{E}_{train}$	Set of training environments.
$s \in \mathcal{S}$	Invariant state representation.
$\eta^e \in \mathcal{Z}^e$	Noise terms (i.e. spurious correlations) for environment e.
$\phi : \mathcal{X} \to \mathcal{S}$	Encoder for learning the state representation s.
θ_s	Parameters of ϕ.
$\mu^e : \mathcal{X} \to \mathcal{Z}^e$	Encoder for learning the environment-specific noise terms η^e.
θ^e_η	Parameters of η^e.
$c_s : \mathcal{S} \to \lvert\mathcal{E}_{train}\rvert$	Environment classifier.
θ_c	Parameters of environment classifier c_s.
$\mathcal{L}_{inv}$	Environment classifier loss.
$g_s : \mathcal{S} \times \mathcal{A} :\to \mathcal{S}$	Transition dynamics for the state variables.
θ_{g_s}	Parameters of g_s.

$g_\eta^e : \mathcal{Z}^e \times \mathcal{A} \to \mathcal{Z}^e$ . .	Environment specific transition dynamics for the noise variables.
$\theta_{g_\eta}^e$	Parameters of g_η^e.
$\psi : \mathcal{S} \times \mathcal{Z}^e \to \mathcal{X}$ . . .	Next state decoder.
θ_ψ	Parameters of ψ.
$\mathcal{L}_{dyn}$	Transitions dynamics loss function.
$I(U, V)$	Neural information measure that approximates the mutual information between random variables U and V.
$\mathcal{L}_{mi}$	Loss which seeks to minimize the mutual information between the state representation and noise representation.
T_{θ_m}	Statistics network used for computing the mutual information.
θ_m	Parameters of the statistics network T_{θ_m}.
θ_π	Parameters of learnt policy π.
$\mathcal{L}_\pi$	Policy loss for matching the expert actions.
$E_{\bar{\theta}}(x) : \mathcal{X} \to \mathbb{R}$	Energy function.
$\bar{\theta}$	Parameters of the energy function.
$Z(\bar{\theta})$	Partition function.
$\mathcal{L}_{energy}$	Energy loss.

List of Abbreviations and Definitions

Abbreviations

ACC	Accuracy.
APR	Area under Precision-Recall curve.
AUC	Area under the Receiver Operating Characteristic curve.
BART	Bayesian Additive Regression Trees.
BC	Behaviour Cloning.
BNN	Balancing Neural Network.
BMI	Body Mass Index.
CFR	Counterfactual Regression.
CIRL	Counterfactual Inverse Reinforcement Learning.
CMGP	Causal Multi-task Gaussian Processes.
CRN	Counterfactual Recurrent Network.
DCRN	Disentangled Counterfactual Recurrent Network.
DSFN	Deep Successor Feature Network.
DPE	Dosage Policy Error.
DRNet	Dose-Response Network.
DQN	Deep Q-Network.
EBM	Energy-based Model.
EDM	Energy-based Distribution Matching.
EHR	Electronic Health Record.
ELU	Exponential Linear Unit.
FC	Fully-connected.
GAN	Generative Adversarial Network.
GANITE	Generative Adversarial Networks for inference of Individualised Treatment Effects.

GP . . . . . Gaussian Process.

GPS . . . . . Generalized Propensity Score.

GRL . . . . . Gradient Reversal Layer.

ICIL . . . . . Invariant Causal Imitation Learning.

ICU . . . . . Intensive Care Unit.

IL . . . . . Imitation Learning.

IPTW . . . . . Inverse Probability of Treatment Weighting.

IRL . . . . . Inverse Reinforcement Learning.

IRM . . . . . Invariant Risk Minimization.

ITE . . . . . Individualised Treatment Effect.

JSD . . . . . Jensen-Shannon Divergence.

KL divergence . . . . . Kullback–Leibler divergence

LSTM . . . . . Long Short-Term Memory.

LSTD . . . . . Least Squares Temporal Difference.

MB-IRL . . . . . Model-Based Inverse Reinforcement Learning.

MDP . . . . . Markov Decision Process.

MIMIC . . . . . Medical Information Mart for Intensive Care.

MINE . . . . . Mutual Information Neural Estimation.

MISE . . . . . Mean Integrated Squared Error.

MLP . . . . . Multi-Layer Perceptron.

MSE . . . . . Mean Squared Error.

MSM . . . . . Marginal Structural Model.

NN . . . . . Neural Network.

NSGP . . . . . Non-stationary Gaussian Processes.

PE . . . . . Policy Error.

PCA . . . . . Principal Component Analysis.

PK/PD . . . . . Pharmacokinetic/Pharmacodynamic.

PO . . . . . Potential Outcome.

POMDP . . . . . Partially Observable Markov Decision Process.

PPO . . . . . Proximal Policy Optimization.

RCAL	Reward-regularized Classification for Apprenticeship Learning.
RCT	Randomized Control Trial.
ReLU	Rectified Linear Unit.
RL	Reinforcement Learning.
RMSE	Root Mean Squared Error.
RMSN	Recurrent Marginal Structural Network.
RNN	Recurrent Neural Network.
SCIGAN	eStimating the effects of Continuous Interventions using Generative Adversarial Networks.
SCM	Structural Causal Model.
SITE	Similarity Preserved Individual Treatment Effects.
TARNet	Treatment-Agnostic Representation Network.
TCGA	The Cancer Genome Atlas.
TSD	Time Series Deconfounder.
TSNE	T-distributed Stochastic Neighbor Embedding.
QSP	Quantitative Systems Pharmacology.
WBC	White Blood Cell.

Definitions

Confounder	Variable affecting both the treatment assignment and the potential outcomes.
Counterfactual outcome	Potential outcome that is never observed, i.e. patient outcome under a different treatment than the one they were given.
Expert policy	Policy that aims to maximize some reward function and from which we have access to patient trajectories (sequences of observations and treatments/actions) in the form of an observational dataset.
Factual outcome	Potential outcome observed after a given treatment.
Overlap/Positivity	Common support between treatment groups, i.e. that each patient has a non-zero probability of receiving each treatment option.

Potential outcome	Patient outcome under a potential treatment.
Selection bias	Bias present in observational data induced by the fact that the treatment assignments are non-random and depend on the patient's features.
Semi-synthetic simulation	Data simulation where the covariates are taken from a real dataset, while the treatment assignments and outcomes are simulated.
Spurious correlation . . .	Mathematical relationship between variables that represents an association which does not generalize beyond the observed data, i.e. relationship that is not causal.
Synthetic simulation . . .	Data simulation where the covariates, treatment assignments and outcomes are all simulated.
Time-dependent confounder	Covariate that is affected by past treatments which then influences future treatments and outcomes.
Unconfoundedness	Assumption that there are no hidden confounders in the observational data, i.e. that we observe all of the confounders.

1 Introduction

Contents

1.1 Causal Inference and Machine Learning 2
1.2 Thesis Outline and Contributions 5
1.3 Publications 11
1.3.1 Relevant Publications 11
1.3.2 Other Publications 12

1.1 Causal Inference and Machine Learning

As human beings, we often consider cause and effect relationships when making decisions as part of our daily lives. Causality enables us to perform "what-if" reasoning and consider outcomes in counterfactual scenarios (e.g. would my headache disappear if I take an aspirin?) such that we can choose the action that would result in the desired outcome. Moreover, our ability to understand the difference between correlation and causation also plays a significant role in the way we reason and interact with the world around us.

When considering real-world applications, causal understanding and inference are fundamental for many decision-making problems: e.g. in healthcare, for determining which treatments to give to patients [1], in education, for deciding which school curriculum would be most suitable for students [2] or in public policy for choosing who would benefit from job training programs [3]. As such, building data-driven machine learning models to estimate “what-if” (counterfactual) outcomes indicating what would happen if we took a certain action given the current observations would provide invaluable decision support to these areas. The rest of this thesis focuses on causal inference in healthcare as the main application area since many problems in medicine would significantly benefit from patient-specific data-driven approaches. Nevertheless, the methods developed in this thesis are general and can be applied to any of these other application areas.

Learning causal relationships and answering causal questions from data presents significant challenges as we need to make sure we account for confounding bias and avoid learning spurious correlations [4–6]. Randomized control trials (RCTs) represent a gold standard for performing causal inference to test the efficacy of new treatments. Through randomization, RCTs evenly distribute the known and unknown factors about patients into interventional and control groups, thereby reducing the potential bias due to confounding variables. Nevertheless, clinical trials inquire large costs and often have a very small number of patients which hinders the estimation of heterogeneous treatment effects [7, 8]. Moreover, clinical trials tend to have narrow inclusion criteria, where the selected patients have a single diagnosis and straightforward treatment strategy [9]. This makes them unrealistic for modelling in-hospital situations where patients can have

commodities and are administered multiple drugs which are then adjusted over time according to the patient's response.

Alternatively, observational data, such as electronic health records (EHRs), which contain information about patient clinical features, the treatments administered to them and their outcomes, provide a more cost-effective and viable alternative to estimating the causal effects of treatments. In this context, a significant part of the statistics literature has been focused on building methods that can account for the confounding bias present in observational datasets in order to estimate average causal effects, to determine whether an intervention is beneficial, on average, for a certain population. These methods tackle both the static (cross-sectional) setting [10–13] and the longitudinal setting with treatments administered over time [14–17]. Nevertheless, by only considering average treatment effects, such methods fail to account for patient heterogeneity.

Recently, the availability of larger observational datasets from EHRs and developments in machine learning methods bring us closer to achieving the promise of personalized medicine to provide care tailored to each individual patient. Data from EHRs has already been successfully leveraged in several machine learning applications such as modelling disease progression [18], predicting disease deterioration [19] and risk factors [20, 21]. However, it is important to not only make predictions from EHRs, but also understand what the best way is to act and give treatments to different patients. This requires understanding causality and being able to quantify causal relationships using the available observational data [22, 23].

As such, there has been a growing interest in building causal inference methods based on machine learning that go beyond average treatment effects to estimate the heterogeneous effects of treatments (conditional average treatment effects) such that we can determine for each individual patient whether they would benefit or not from different treatments. So far, many of the machine learning methods for causal inference have been developed in the cross-sectional (static) setting, for the case when a binary treatment decision needs to be made, e.g. whether to treat a patient given their characteristics [1, 24–34]. Only a few methods have been built to handle additional treatment scenarios, such as categorical treatments [35], treatments with an associated dosage [36], structured

treatments [37] and combinations of treatments [38, 39]. From a methodological perspective, the main challenges these models address involve correctly handling the confounders present in observational data and modelling the treatment response functions and the missing counterfactuals.

Extending causal inference methods to the time series setting would provide even more powerful information for decision-making. By estimating treatment effects over time from EHRs we can understand how diseases evolve under different treatment plans, how individual patients respond to treatments given over time, but also the optimal timings for making the interventions. For longitudinal trajectories, the difficulty of estimating treatment effects is amplified by the presence of time-dependent confounders, covariates affected by past treatments which then influence the assignment of future treatments and several methods have been developed to address this challenge [18, 40]. Such methods for estimating the heterogeneous effects of interventions would be crucial for clinical decision support as they can provide clinicians with actionable intelligence with respect to the patient outcomes which can help them decide on the best treatment plan for each patient. Nevertheless, the area of building machine learning methods for causal inference is still in its early stages and additional methods are needed for handling the complexity of treatment options present in the clinical setting. Moreover, such methods need to undergo thorough validation and evaluation before being deployed into clinical practice.

While the development of machine learning methods has generally benefited from access to large amounts of data, such methods are also prone to learning spurious correlations instead of causal relationships [41]. Ensuring that machine learning methods for decision-making take into account causal relationships is crucial for their robustness and our ability to use them in practice. As a consequence, the field of 'causal machine learning' has recently emerged which proposes incorporating causal structure and ideas from causality into common machine learning algorithm [42–47]. As causal relationships are generally stable across different domains and data distributions, building methods that can reason about the causal structure of environments would help us obtain more robust models that are capable of generalising outside of the training distribution [46, 48].

Overall, developing machine learning methods capable of reasoning about cause-and-effect relationships and, in particular, about the effects of actions is needed to make appropriate decisions given different contexts and also to generalise well across domains.

1.2 Thesis Outline and Contributions

This thesis advances the state-of-the-art in causal inference methods for supporting, improving, and understanding decision-making. Chapter 2 provides a background and literature review for machine learning methods for causal inference from observational data. This helps put the subsequent chapters of the thesis in context, but also highlights which problems in causal inference require the development of better tools. As previously mentioned, causal inference enables us to perform "what-if" (counterfactual) reasoning, i.e. given the current history of observations, what would happen if we took a particular action or sequence of actions? Then, based on the estimated counterfactual outcomes, we can decide which intervention or sequence of interventions will result in the best outcome. In the next chapters, new methods that can estimate such counterfactual outcomes for continuous interventions in the static setting (Chapter 3) and for discrete actions taken over time, both in the presence and absence of hidden confounders (Chapter 4) are proposed. In addition, to better understand decision-making, Chapter 5 proposes a new method for obtaining interpretable explanations of sequential decision-making by integrating counterfactual reasoning into batch inverse reinforcement learning and recovering the preferences of experts over 'what-if' outcomes. Finally, Chapter 6 investigates how causal structure can be leveraged to improve the robustness of decision-making algorithms based on imitation learning to distribution shifts in the deployed environment.

Note that this thesis is presented in an integrated format, and as such is a collection of published papers. This means that each of the chapters, including the literature review in Chapter 2 and the research contributions in Chapters 3, 4, 5, 6 correspond to peer-reviewed publications. The papers are reproduced in the thesis formatting and are presented in the "camera-ready accepted format" for the corresponding peer-reviewed venues. Finally, a discussion and conclusion for the thesis will be provided in Chapter 7. The remainder of this section outlines the content and contributions of each thesis chapter.

Chapter 2: Background and Literature Review

This chapter introduces the area of causal inference for treatment effect estimation, highlights the main challenges and assumptions needed for estimating treatment effects from observational data such as electronic health records (EHRs) and provides a literature review of this area. In addition, we provide an overview of the main modelling choices of state-of-the-art machine learning methods for causal inference, developed for estimating treatment effects both in the cross-section and longitudinal settings. Moreover, the chapter also provides examples of clinical applications where such causal inference methods have been used for making treatment recommendations and discusses the main challenges for applying these methods in practice.

From this chapter, we can identify several key areas in causal inference for treatment effect estimation that have been under-explored and that require the development of methods that can better estimate counterfactual outcomes. In subsequent chapters, we propose new such methods for some of the identified areas which represents an important step in achieving the full potential of leveraging EHRs and causal inference for making individualized treatment recommendations. However, this is by no means sufficient and in Chapter 2 we also discuss additional challenges and future directions. Note that while this chapter provides an in-depth literature review of causal inference methods for estimating the effects of interventions that helps put the work in this thesis in context, each individual subsequent chapter will also have a section that discusses the most relevant related works for each proposed new method in more detail.

Chapter 3: Estimating the Causal Effects of Continuous-valued Interventions

In Chapter 3, we start to tackle some of the problems in causal inference highlighted in Chapter 2 that require better solutions. In particular, we propose a new method for estimating the effects of continuous-valued interventions (e.g treatments with an associated dosage parameter). This is an understudied problem, as most causal inference methods consider the discrete treatment case (e.g deciding whether to treat or not to treat). However, healthcare decisions often involve choosing the correct dosage of the administered treatment, where using a high dosage can lead to toxic effects, while a low

dosage can result in no effect on the patient outcome. This decision becomes increasingly important when the relationships between drug toxicity, efficacy, and heterogeneous patient features are complex.

To address this problem, we introduce SCIGAN, a novel causal inference method that can simultaneously estimate individualized dose-response curves for several different treatments. SCIGAN builds on the framework of generative adversarial networks (GANs) [49] to learn the distribution of the unobserved counterfactuals. In particular, SCIGAN consists of a counterfactual generator that generates the individualized dose-response curves conditional on the patient characteristics and a discriminator that acts on a finite set of points from each generated curve and identifies the factual outcome from the set of factual outcomes and the generated counterfactual ones. The intuition is that if a counterfactual generator and discriminator are trained adversarially, then the generator can fool the discriminator such that it will not be able to correctly identify the factual outcome by generating the potential outcomes according to their true distribution. Once the counterfactual generator is trained, it can be used to learn an inference model, using standard supervised methods, capable of estimating these counterfactuals for a new sample. We provide a theoretical justification for our approach and we propose novel neural network architectures for the counterfactual generator, discriminator and inference network. Finally, this chapter also introduces a new semi-synthetic data simulation benchmark that can be used to evaluate causal inference methods for estimating the effects of continuous interventions. Given that the counterfactuals are never observed, having appropriate evaluation benchmarks is crucial for deciding which causal inference methods to use in practice.

Chapter 4: Estimating the Causal Effects of Treatments in the Temporal Setting

In Chapter 4, we move our attention to the problem of estimating counterfactual outcomes for treatments assigned over time. Even though estimating treatment effects over time is a more challenging problem than estimating the causal effects of treatment in the static setting, it also poses unique opportunities in terms of understanding how diseases evolve under different treatment plans and how individual patients respond to medication

over time. In particular, given a patient history of clinical measurements such as vital signs, lab tests and past treatments as well as static features such as gender and age, some critical medical questions involve deciding when to start or stop a treatment plan, when is the optimal time to give the next treatment, or how to select among multiple treatments over time.

In Chapter 4.1, we introduce the Counterfactual Recurrent Network (CRN), a novel sequence-to-sequence model that can provide decision support by being able to estimate counterfactual outcomes for each patient under an intended sequence of future treatments. These counterfactual trajectories can then be used to determine which treatment plan leads to the best patient outcome. Estimating treatment effects over time is challenging due to the presence of time-dependent confounders in observational datasets. Time-dependent confounders are patient covariates that affect the treatment assignments and are themselves affected by past treatments. To handle this bias from time-dependent confounders, CRN leverages the recent advances in representation learning [50] and domain adversarial training [51] to build treatment invariant representations that can break the association between patient history and treatment assignments. Using these representations, we can then obtain reliable estimates of the counterfactual outcomes. We evaluate CRN's performance on a model of tumour growth where we show that it can achieve lower error in forecasting counterfactual outcomes and in choosing the correct treatment and timing of treatment than the previous state-of-the-art methods.

Chapter 4.2 considers the case when there may be hidden confounders in the observational temporal data available. Previous methods for estimating the effects of treatments over time, including CRN, assume that all confounders–variables affecting the treatment assignments and the potential outcomes–are observed. However, since counterfactuals are never observed, it is not possible to test for the existence of hidden confounders that could affect them and often domain knowledge is needed to assess the validity of this assumption. Estimating treatment effects over time in the presence of hidden confounders will result in biased estimates and subsequently in making erroneous treatment recommendations. To account for this problem, the Time Series Deconfounder proposes taking advantage of the dependencies in the multiple treatments assigned over time in order to infer latent

variables that can act as substitutes for the hidden confounders. These latent variables can then be used to augment observational datasets and subsequently obtain unbiased estimates of the treatment effects. We provide a theoretical analysis for the Time Series Deconfounder and show its effectiveness on both simulated and real datasets.

Chapter 5 - Using Counterfactuals to Explain Sequential Decision-Making

In Chapter 5 we aim to integrate the counterfactual estimates from Chapter 4 into batch inverse reinforcement learning (IRL) with the aim of learning inherently interpretable reward functions which can help us understand the decision-making behaviour of experts. By building such interpretable parameterisations of sequential decision-making, we can quantify and interpret the different trade-offs associated with expert actions, and ultimately capture variation in practice. For instance, modelling the reward functions of different clinical practitioners can reveal their tendencies to treat various diseases more or less aggressively, which in combination with patient outcomes has the potential to inform and drive change in clinical guidelines.

Batch IRL offers a principled way of recovering these preferences and explaining expert policies from demonstrated behaviour and satisfies the constraints of real-world decision-making where active experimentation is often impossible (e.g. in healthcare). In this chapter, we propose the first method that incorporates counterfactual reasoning into modelling decision behaviours in this batch IRL setting. As decision-makers often consider counterfactual outcomes when acting, it becomes natural to use them to explain their behaviour [52]. At each decision point, counterfactuals tell us what would happen if we took a particular action given the current history of observations and we can use methods such as the one proposed in Chapter 4.1 to estimate them. Firstly, this offers a principled approach to learning inherently interpretable reward functions by defining them as weighted sums of the possible counterfactual outcomes. Recovering these reward weights enables us to understand the cost-benefit tradeoffs associated with an expert's actions. Secondly, by estimating the effects of different actions, counterfactuals readily tackle the off-policy nature of policy evaluation in the batch setting. This alleviates the cold-start problem typical of conventional solutions for

batch IRL, and also accommodates settings where the expert policies are depending on histories of observations rather than just current states. Through experiments in both real and simulated medical environments, we illustrate the effectiveness of our proposed batch, Counterfactual Inverse Reinforcement Learning (CIRL) approach in recovering accurate and interpretable descriptions of expert behaviour.

Chapter 6 - Using Causal Structure to Improve the Robustness of Decision-Making

In Chapter 6, we turn our attention to incorporating ideas from causality into batch imitation learning to improve robustness of decision-making by learning imitation policies that generalize across environments and that are free from spurious correlations and biases. This is relevant in the healthcare setting where for instance we might have access to expert data from how doctors have treated patients from several different hospitals. We then want to leverage such data to learn a behaviour policy that would generalise and achieve similar expert performance in a new environment. This could serve as an individualized clinical guideline for treatments that can be given to different patients and would represent a valuable way of providing clinical decision support, especially in a new hospital, region, or patient demographic from which we have no access to data during training.

To learn such an imitation policy, we impose a causal structure on the different environments where we assume that the expert's actions are causally influenced only by a subset of the observations. In particular, we assume that the observations in each environment can be decomposed into (1) state representations that are invariant and that contain the causal parents of the expert's actions and (2) noise representations that may change across environments and that consist of spurious correlations and biases. Our proposed Invariant Causal Imitation Learning (ICIL) method leverages data from multiple environments to learn the causal representation that consists of the causal parents of the expert's actions. This invariant representation is then used to build an imitation policy that matches the expert's behaviour and that generalises across environments. In particular, by conditioning the imitation policy on the state (causal) representation, we learn a policy that generalizes to environments with the same shared latent structure, but with different noise distribution and dynamics. ICIL operates in the strictly batch setting and it uses

an energy-based approach to encourage the learnt imitation policy to stay within the support of the distribution of states visited by the expert's policy. We evaluate ICIL against benchmarks for batch imitation learning in control and healthcare environments where we show its ability to learn more robust decision-making policies.

1.3 Publications

1.3.1 Relevant Publications

The thesis chapters are based on the following peer-reviewed publications. Here * indicates equal contribution.

- **Ioana Bica**, Ahmed M. Alaa, Craig Lambert, and Mihaela van der Schaar. "From real world patient data to individualized treatment effects using machine learning: Current and future methods to address underlying challenges". In *Clinical Pharmacology & Therapeutics*, 2020. [53] (Chapter 2)

- **Ioana Bica***, James Jordon*, and Mihaela van der Schaar. "Estimating the Effects of Continuous-valued Interventions using Generative Adversarial Networks". In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. [54] (Chapter 3)

- **Ioana Bica**, Ahmed M. Alaa, James Jordon, and Mihaela van der Schaar. "Estimating Counterfactual Treatment Outcomes over Time through Adversarially Balanced Representations". In *International Conference on Learning Representation (ICLR)*, 2020. [**Selected as spotlight (top 6% of submissions)**] [55] (Chapter 4.1)

- **Ioana Bica**, Ahmed M. Alaa, and Mihaela van der Schaar. "Time Series Deconfounder: Estimating Treatment Effects over Time in the Presence of Hidden Confounders". In *International Conference on Machine Learning (ICML)*, 2020. [56] (Chapter 4.2)

- **Ioana Bica**, Daniel Jarrett, Alihan Hüyük, and Mihaela van der Schaar. "Learning "What-if" Explanations for Sequential Decision-Making". In *International Conference on Learning Representation (ICLR)*, 2021. [57] (Chapter 5)

- **Ioana Bica**⋆, Daniel Jarrett⋆, and Mihaela van der Schaar. "Invariant Causal Imitation Learning for Generalizable Policies". In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. [58] (Chapter 6)

1.3.2 Other Publications

Moreover, during my doctoral studies, I have also contributed to the following papers within the topic of causal inference and machine learning for healthcare.

Further contributions to the area of causal inference:

- Jeroen Berrevoets, James Jordon, **Ioana Bica**, and Mihaela van der Schaar. "Organ-ITE: Optimal transplant donor organ offering using an individual treatment effect". In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. [59]

- Can Xu⋆, Ahmed M. Alaa⋆, **Ioana Bica**, Brent D. Ershoff, Maxime Cannesson, and Mihaela van der Schaar. "Learning Matching Representations for Individualized Organ Transplantation Allocation". In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021. [60]

- Zhaozhi Qian, Yao Zhang, **Ioana Bica**, Angela Wood, and Mihaela van der Schaar. "SyncTwin: Treatment Effect Estimation with Longitudinal Outcomes". In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. [61]

- Jeroen Berrevoets, Alicia Curth, **Ioana Bica**, Eoin McKinney, and Mihaela van der Schaar. "Disentangled Counterfactual Recurrent Networks for Treatment Effect Inference over Time." arXiv preprint, 2021. [62]

- Trent Kyono⋆, **Ioana Bica**⋆, Zhaozhi Qian, and Mihaela van der Schaar. "Selecting Treatment Effects Models for Domain Adaptation Using Causal Knowledge." arXiv preprint, 2021. [63]

- William R.Zame, **Ioana Bica**, Cong Shen, Alicia Curth, Hyun-Suk Lee, Stuart Bailey, James Weatherall, David Wright, Frank Bretz, and Mihaela van der Schaar. "Machine learning for clinical trials in the era of COVID-19". In *Statistics in Biopharmaceutical Research*, 2020. [64]

Contributions to machine learning for healthcare:

- Pedro Baqui*, **Ioana Bica***, Valerio Marra, Ari Ercole, and Mihaela van der Schaar. "Ethnic and regional variations in hospital mortality from COVID-19 in Brazil: a cross-sectional observational study". In *The Lancet Global Health no. 8*, 2020. [65]

- Daniel Jarrett*, **Ioana Bica***, and Mihaela van der Schaar. "Strictly batch imitation learning by energy-based distribution matching". In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. [66]

- Daniel Jarrett*, Jinsung Yoon*, **Ioana Bica**, Zhaozhi Qian, Ari Ercole, and Mihaela van der Schaar. "Clairvoyance: A Pipeline Toolkit for Medical Time Series". In *International Conference on Learning Representation (ICLR)*, 2021. [67]

- Alex J. Chan, **Ioana Bica**, Alihan Hüyük, Daniel Jarrett, and Mihaela van der Schaar. "The Medkit-Learn (ing) Environment: Medical Decision Modelling through Simulation". In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021. [68]

- Daniel Jarrett, **Ioana Bica**, and Mihaela van der Schaar. "Time-series Generation by Contrastive Imitation". In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. [69]

These papers will not be covered in detail in this thesis. However, in Section 7.2 we will outline some of them and discuss the relationship with the rest of the work presented in this thesis.

2

Background and Literature Review

Contents

2.1 From Randomized Trials to Patient Observational Data 15
2.2 Individualized Treatment Recommendations in the Static Setting 18
2.2.1 Challenges in Using Observational Data for Estimating Individualized Treatment Effects 20
2.2.2 Modelling Choices and Methods 23
2.2.3 Individualized Dose-Response Estimation 28
2.3 Individualized Effects of Time-dependent Treatments 30
2.3.1 Challenges in Using Observational Data for Treatment Decisions in the Time-Series Setting 32
2.3.2 Modelling Choices and Methods for Estimating Individualized Effects of Time-Dependent Treatments 33
2.4 Evaluation of Causal Inference Models 36

Publication Included

- **Ioana Bica**, Ahmed M Alaa, Craig Lambert, and Mihaela van der Schaar. "From real-world patient data to individualized treatment effects using machine learning: Current and future methods to address underlying challenges". In *Clinical Pharmacology & Therapeutics*, 2020. Link: https://ascpt.onlinelibrary.wiley.com/doi/full/10.1002/cpt.1907. [Accepted]

Abstract

Clinical decision making needs to be supported by evidence that treatments are beneficial to individual patients. While randomized control trials are the gold standard for testing and introducing new drugs, due to the focus on specific questions with respect to establishing efficacy and safety versus standard treatment, they do not provide a full characterisation of the heterogeneity in the final intended treatment population. Conversely, real-world observational data, such as electronic health records contain large amounts of clinical information about heterogeneous patients and their response to treatments. In this chapter, we introduce the main opportunities and challenges in using observational data for training machine learning methods to estimate individualized treatment effects and make treatment recommendations. We describe the modelling choices of state-of-the-art machine learning for causal inference methods developed for estimating treatment effects both in the cross-sectional and longitudinal settings. Additionally, we discuss the limitations of existing methods and highlight future research directions that could lead to achieving the full potential of leveraging electronic health records and machine learning for making individualized treatment recommendations.

2.1 From Randomized Trials to Patient Observational Data

Decision-making in the healthcare domain has largely depended on statistical and longitudinal approaches utilising data from randomized clinical trials (RCTs) for determining the effects of treatments [70, 71]. Regulators currently make decisions for drug approvals

based on RCTs and prescribing physicians use these to determine the treatment approaches to be used [70, 71]. Through randomization, well-designed RCTs evenly distribute the known and unknown factors about patients into interventional and control groups, thereby reducing the potential bias due to confounding variables. However, external validity is an issue for RCTs, as findings sometimes fail to generalize beyond the study population [72–74]. This may be due to the narrow inclusion criteria in RCT compared to the real world, where historically, population restrictions with respect to disease severity, comorbidities, elderly patients and ethnic minorities can be underrepresented [75]. While there is increasing awareness of this issue and global regulatory authorities are encouraging wider inclusion criteria in clinical trials, it remains an issue that is unlikely to be solved by RCTs and associated integrated and model-based analyses alone [76].

Nevertheless, when drugs are FDA-approved after the clinical trials stage, they start being administered to a much larger and varied population of patients. Each patient's unique medical characteristics—including comorbidities, lab results, prior treatments received, as well as the various adverse or beneficial effects observed—are stored in electronic health records (EHRs). EHRs have already been widely adopted in U.S. hospitals [77]. While data from EHRs have less regulatory acceptance compared to data from RCTs in addition to weaknesses such as informative missingness and observational biases, it facilitates the quantity, accessibility, and heterogeneity of offline observational data for a variety of diseases and patients. We refer the readers to Beaulieu-Jones et al. [22] for a detailed review of the advantages and limitations of using real-world evidence, such as EHRs, to infer treatment effects. Estimating individualized treatment effects from EHR data represents a thriving area of research, in which machine learning methods are primed to take centre stage.

While data from EHRs has already been successfully leveraged in several predictive machine learning applications in healthcare [18–21], it is important to also be able to use such data to understand what the best way is to act and give treatments. In this context, machine learning methods for causal inference can be used to learn the effects of treatments from observational data and subsequently provide clinicians with actionable intelligence for making treatment decisions. We illustrate in Figure 2.1 how learning

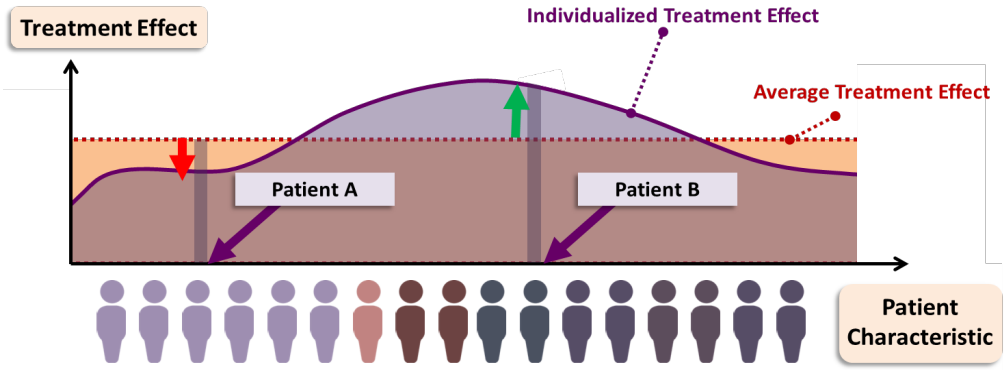


Figure 2.1: By using causal inference methods to estimate individualized treatment effects, we can understand which patients benefit from certain medication and which patient do not, thus enabling us to make personalized treatment recommendations. We describe machine learning methods for causal inference, which are capable of estimating individualized treatment effects.

individualized treatment effects allow us to identify which patients benefit most and least from certain treatments.

In this chapter, we describe the recent progress that has been made in using observational data for building data-driven methods for making treatment recommendations. Our focus is on causal inference method for both the static (cross-sectional) setting and the longitudinal setting (where patient history and treatment timing are taken into account). We describe the challenges associated with learning from observational data, such as confounding bias, as well as the modelling choices used by machine learning methods to handle them in both settings. We illustrate the benefits of various methods and discuss the limitations and challenges of different approaches. We also note that this research is still in the early stages and more methods are needed to achieve the full potential of learning from observational data as well as for validating machine learning methods for causal inference. As such, this chapter highlights several areas in causal inference that would benefit from the development of better machine learning methods. In subsequent chapters, we focus on building some of these methods.

Moreover, we note that relatively little work has been done at the interplay of combining experimental data from RCTs with observational data from EHRs or using existing Pharmacometric and Quantitative Systems Pharmacology (QSP) approaches to provide priors for machine learning models for causal inference (or machine learning driving pharmacology). Developing methods capable of integrating data from RCT,

EHR and Pharmacokinetic/Pharmacodynamic (PK/PD) models also provides an exciting opportunity for real progress.

2.2 Individualized Treatment Recommendations in the Static Setting

What Can Be Done Using Machine Learning and What Kind of Patient Observational Data Is Needed?

We first consider estimating individualized treatment effects in the cross-sectional or static setting, i.e. when one single treatment decision needs to be made based on the one-time measurement (current) patient features. The information about patients collected in EHRs can be used by data-driven methods to inform clinical decision making. However, building machine learning methods to extract actionable intelligence from observational patient data involves causal inference, which goes beyond standard supervised learning methods for prediction. In this section, we describe what is possible to achieve using current state-of-the-art machine learning methods for causal inference. In the static (cross-sectional) setting, the aim is to make a one-time treatment decision. We will use observational data containing clinical information such as age, gender, comorbidities, symptoms and lab measurements collected before the treatment decision is made. Conditioned on this patient data, machine learning methods can be used to estimate the effects of single or multiple treatments on various patient outcomes. In each setting, the observational data needs to contain the corresponding information about the type of treatments and patient outcomes to be able to train the data-driven models.

The standard scenario involves deciding whether or not to treat a patient. In this case, we need observational data about patients that have received the treatment, as well as patients that have not received the treatment and their corresponding outcome. The outcome of interest can be, for instance, one-year mortality (binary outcome), a specific disease (categorical outcome) or the value of a specific lab measurement (continuous outcome) that is observed after the treatment decision is made. Using such observational data, machine learning methods can be trained to estimate a patient's outcome—both when they receive the treatment, as well as when they do not receive the treatment. By

Type of patient data	Examples
Patient clinical data	age, gender, commodities, genetic information, lab measurement, vital signs, symptoms, BMI
Type of treatments	binary treatment (treat or no treat) [26, 29, 31], categorical treatment (one treatment out of multiple possible treatment options) [35], multiple treatments assigned simultaneously [78], single treatment or multiple treatments with associated dosage [54]
Patient outcomes	single outcome (e..g survival probability), multiple outcomes (e.g. probability of recovery and associated side effects of treatment)
Treatment Decisions	optimal single treatment [26, 35], optimal combinations of treatments [78], optimal dosage by trading off added benefits of treatments and side effects [54]

Table 2.1: Different types of patient observational data, treatments and patient outcomes that can be used to train causal inference methods to estimate treatment effects in the static setting. We also included references to specific machine learning methods that can handle the different types of treatments present in observational data.

comparing these two outcomes, physicians can then decide whether the treatment will benefit the patient or not. It is also possible to estimate multiple outcomes such as patient recovery and associated side effects of treatments—in which case the treatment decision can be a trade-off between the different outcomes.

Additional realistic treatment scenarios involve choosing one treatment out of multiple treatment options or deciding the optimal combination of treatment. In the case of multiple treatments, interactions between treatments can also be considered. Moreover, treatments often have an associated dosage which is critical for assessing the therapeutic effect of the treatment on a patient. Causal inference methods can be also be used to estimate individualized dose-response curves for patients and thus lead to decisions about optimal treatment dosage. To summarize, we provide in Table 2.1 some examples of treatment decisions that machine learning methods can be used for and references to machine learning methods that can handle the different types of treatments present in observational data. While we note that these methods are general and can be applied

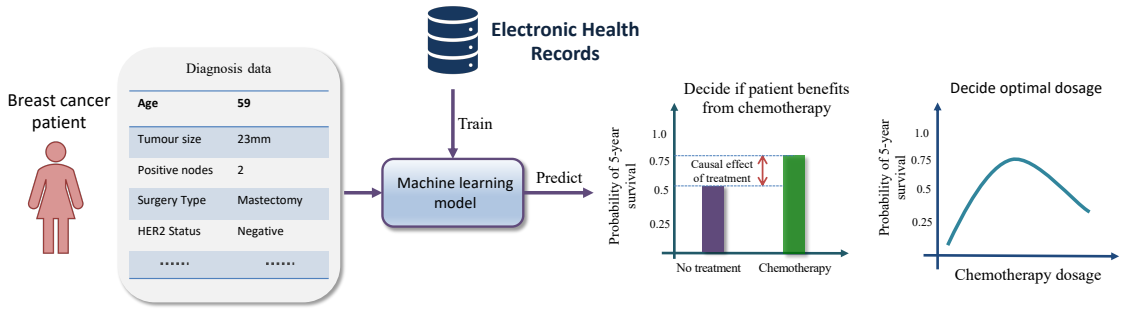


Figure 2.2: Causal inference methods can be trained on observational patient data from EHRs and and then used to estimate individualized patient outcomes under possible treatment options. We illustrate how these estimates can be used to decide, at diagnosis time, the chemotherapy is beneficial, and the optimal dosage for a patient with breast cancer.

to various types of patient characteristics in observational data, in Table 2.1 we also provide some examples of patient clinical data and patient outcomes that can be used for training these machine learning methods.

Figure 2.2 shows an illustrative example, highlighting how causal inference methods could be used to inform treatment decisions for patients with breast cancer. The EHR for a breast cancer patient will contain, at diagnosis, data about the tumour such as tumor size, number of positive nodes, ER status, genetic information etc., as well as clinical features describing the patient's overall health state including comorbidities and lab measurements. Using this diagnosis information, we can estimate the causal effect of various treatment alternatives (post- surgery), e.g. chemotherapy and radiotherapy, on a patient's outcome of interest. Some possible patient outcomes are 5 or 10-year survival, probability of relapse, side effects, or some other measurable feature of interest associated with the patient's health. Based on the estimated individualized treatment responses and individualized dose-response curves, physicians can then choose the best treatment and the optimal dosage for each patient.

2.2.1 Challenges in Using Observational Data for Estimating Individualized Treatment Effects

To estimate individualized treatment effects, machine learning methods need access to observational data containing information about how patients with different characteristics have responded to treatments. By learning based on how other patients have responded to the treatment, these methods can be trained to estimate potential outcomes (i.e. patient

outcomes under different treatment options) [79–81] conditioned on the patient's clinical state. Trained models can then be applied to estimate potential outcomes for new patients.

Nevertheless, learning from observational data poses several challenges and requires strong assumptions that need to be rigorously assessed before performing causal inference. While in RCTs the treatments are assigned randomly, in observational data the treatment is often highly dependent on the patient's clinical characteristics. For instance, cancer patients with more aggressive tumors are more likely to receive more extensive treatments but are also more prone to experience worse outcomes despite the treatment intervention [82]. This induces treatment selection bias in the observational data. Without adjusting for the selection bias, we can incorrectly conclude that the extensive treatment is harmful to the patients with aggressive tumors. Additionally, the selection bias also creates covariate shift in the distribution of patients receiving different treatments. For instance, when sicker patients receive more extensive treatments, the distribution of patients receiving extensive treatments will be different from the distribution of patients receiving non-extensive treatments. Standard supervised learning methods in this context will be biased towards the treatment assignments present in observational data and will not be able to generalize well in estimating patient outcomes for other possible treatment options.

Moreover, to be able to perform causal inference using observational patient data, the following assumptions need to hold: overlap (a.k.a. positivity), as well as no hidden confounders (a.k.a. unconfoundedness) [10, 83]. Overlap means that there is common support between the treatment groups, i.e that each patient has a non-zero probability of receiving each treatment option. If we do not observe treatment alternatives in similar contexts, it is not possible to make reliable treatment recommendations. For instance, if diabetic patients always receive the same treatment, we cannot estimate the effects of other possible treatments on these patients. In an observational dataset, overlap can be evaluated by computing treatment probabilities for each patient. If these are bounded away from 0 and 1 for each patient, then there is overlap between the treatment groups. Overlap becomes more problematic when we have high dimensional features for the patients. Several methods have been developed to identify regions in observational studies where all treatment groups are well-represented [84–86]. The current state-of-the-art

method for assessing overlap, developed by Johansson et al. [86] identifies overlap regions by finding minimum value sets [87] of the observational distribution that are subject to coverage constraints by estimating treatment assignment propensities. The proposed algorithm, OverRule [86] outputs a rule-based characterization of overlap between treated populations where causal conclusions from observational data are valid and is applied on a dataset with post-surgical patients containing information about the effect of opioid prescriptions on future misuse [88].

The unconfoundedness assumption ensures that we observe all the confounders, i.e. all the variables affecting the treatment assignment and the outcome. This is needed to be able to correctly adjust for the confounders when estimating individualized treatment effects. Nevertheless, having no hidden confounders is a strong assumption, and cannot be tested in practice: since counterfactuals are never observed, it is not possible to test for the existence of hidden confounders that might affect them. In this context, domain knowledge becomes vital to assess the validity of this assumption. In particular, elicitation from doctors or key opinion leaders may be needed to check if all counfounding variables are included in the observational data [22]. Additionally, prior knowledge of causal links between patient characteristics, treatments and outcomes can also be leveraged to identify potential unobserved confounders [89, 90].

Furthermore, sensitivity analyses can be used to evaluate the potential impact that an unmeasured confounder could have on the estimation of treatment effect [91–93]. For instance, Franks et al. [93] propose factorizing the joint distribution of factual (observed) and counterfactual (unobserved) patient outcomes, into separate factors that are nonparametrically identified from the data and factors that cannot be uniquely identified from the data. This factorization enables quantifying the dependence between that partially observable patient outcomes, treatments and patient features. Thus, sensitivity analyses can be used to determine the suitability of using treatment effects methods for an observational dataset of interest. Alternatively, several methods have been proposed to make use of proxy variables [94–98] or the presence of multiple treatments [56, 99–101] to infer latent variables that can act as substitutes for the hidden confounders. Wang and Blei [99] observed that the dependencies in the assignment of multiple treatments

can be used to infer latent variables that render the treatments independent and thus be used as substitutes of potential hidden confounders present in the observational dataset. Zhang et al. [101] extended the work of [99] and developed the Medical Deconfounder, a method for reliably estimating the individualized effects of multiple treatment assignments on patient's lab tests. In this case latent variables are used to correct for unobserved confounders, such as body mass index, that affect the patient's medication and clinical outcome of interest. The Medical Deconfounder was applied to a cohort of patients with type 2 diabetes mellitus with the aim of determining causality and the magnitude of effect of their medications on their hemoglobin A1c (HbA1c) levels. It was similarly applied to a cohort of patients with potassium disorders to determine the effect of their medications on potassium levels. Zhang et al. [101] illustrate that without adjusting for hidden confounders in EHRs, standard methods would otherwise spuriously identify non-causal treatments e.g. non-causal treatments for managing hemoglobin A1c (HbA1c) levels in diabetic patients.

In summary, while machine learning methodology has been developed to handle the covariate shift between different treatment groups, it is crucial to assess the validity of the overlap and unconfoundedness assumptions before training causal inference methods on the observational data. If these assumptions fail to hold, then the observational data would not be suitable for making personalized treatment recommendations.

2.2.2 Modelling Choices and Methods for Individualized Treatment Effects

For simplicity, consider the problem of binary treatment assignment, which is the most studied one in the causal inference literature. In this case, the observational data consists of patient features $X_i \in \mathcal{X}$, assigned treatment $W_i \in \{0, 1\}$ and observed outcome $Y_i \in \mathbb{R}$ for each patient i. The potential outcomes framework [79, 80] allows us to define two potential outcomes for the patient: $Y_i^{(0)}$ and $Y_i^{(1)}$ representing the patient's outcomes when they have not received the treatment ($W_i = 0$) and when they have received the treatment ($W_i = 1$) respectively. The *factual* patient outcome $Y_i = W_i Y_i^{(1)} + (1 - W_i) Y_i^{(0)}$ is the observed outcome, while the *counterfactual* outcome is unobserved, and represents

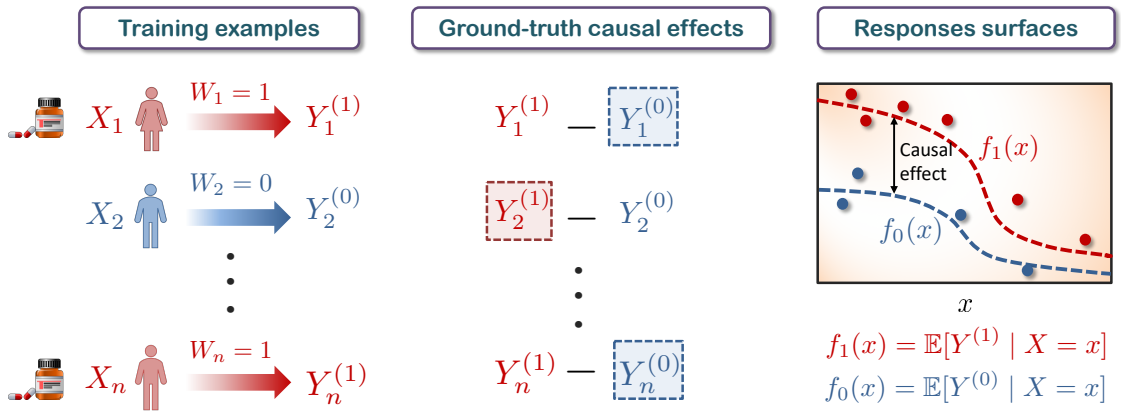


Figure 2.3: Using observational patient data to estimate individualized causal effects. (Left) The observational patient data (training data) contains information about patient features X_i, factual treatment W_i and factual outcome Y_i. (Middle) The ground truth causal effect of the treatment is defined as the difference between receiving the treatment $Y_i^{(1)}$ and not receiving the treatment $Y_i^{(0)}$. We can only observe the factual outcomes; the counterfactual one (indicated by dotted lines) is never observed. (Right) The observational data points for patients that have received treatment (red) and patients that have not received the treatment (blue) can be used to learn response surfaces for each treatment option and thus estimate the causal effect of the treatment.

what would have happened to the patient under a different treatment option than the one assigned in the observational dataset. The individualized treatment effect is defined as the difference in potential outcomes $T(x) = \mathbb{E}[Y^{(1)} - Y^{(0)} \mid X_i = x]$.

The fundamental challenge of causal inference is that the counterfactuals are never observed. This makes causal inference significantly different from standard supervised learning. In supervised learning, all data points are labelled and a machine learning model can be trained to map the input features to the label. In causal inference, the label needed for assessing the benefit of treatment is given by the difference in potential outcomes $T(x) = \mathbb{E}[Y^{(1)} - Y^{(0)} \mid X_i = x]$. However, we only have access to part of this outcome, namely the factual outcome of the patient; the counterfactual outcome, needed for computing the individualized treatment effect cannot be observed.

As illustrated in Figure 2.3, the observational (training) dataset only contains information about the factual treatment and factual patient outcome. However, computing the ground-truth causal effects requires access to both factual and counterfactual outcomes. To solve this problem, machine learning methods are trained to estimate both response surfaces $f_0(x) = \mathbb{E}[Y^{(0)} \mid X = x]$ and $f_1(x) = \mathbb{E}[Y^{(1)} \mid X = x]$ using the observational data. By learning f_0 and f_1, these data-driven methods can estimate *both* potential

outcomes conditioned on the patient's features, and thus can be used to compute the individualized treatment effect. Several modelling alternatives exist for estimating the treatment response functions. Moreover, as previously described, the treatment assignment mechanism creates selection bias in the observational dataset, which induces a covariate shift between the treated and untreated patient distributions. An additional modelling choice comes from handling this distributional mismatch between the training and testing distributions. Thus, machine learning methods for estimating individualized treatment effects are characterized in terms of how they model the treatments (in terms of the response surfaces), as well as how they handle selection bias. We describe each in turn.

There are three options for using the observational data in a supervised learning setting to estimate the treatment response surfaces. One option is to fit a single regression model where the treatment assignment variable is added to the feature space $f_0(x) = f(x, 0)$ and $f_1(x) = f(x, 1)$. Such a supervised model shares all the training samples, but it is less flexible in terms of modelling the response surfaces. Another option fits two separate regression models for the treated and untreated patients: $f_0(x)$ and $f_1(x)$. While this provides more flexibility in modelling the response surfaces, it has the disadvantage of splitting the training samples between the two supervised regression models. A third option—adopted by many of the recent machine learning methods—involves having a shared structure between the two response functions, while at the same time fitting separate outcome models for the different treatments. The latter two options allow for model structures with "outcome-specific" hyperparameters. We illustrate in Figure 2.4 these approaches; note that in this figure, the second and third options mentioned in this paragraph fit under the approach on the right.

In terms of handling the treatment selection bias, a common approach is to use the propensity score [10] (i.e. the probability that a patient receives a treatment) to perform inverse probability of treatment weighting of the loss in the prediction models—for matching patients, or as part of regularization schemes (e.g. propensity-dropout). A different approach involves building a balancing representation that makes the treated and control distributions more similar. Balancing representations, also known as treatment invariant representations, are representations of the patient covariates that have equal

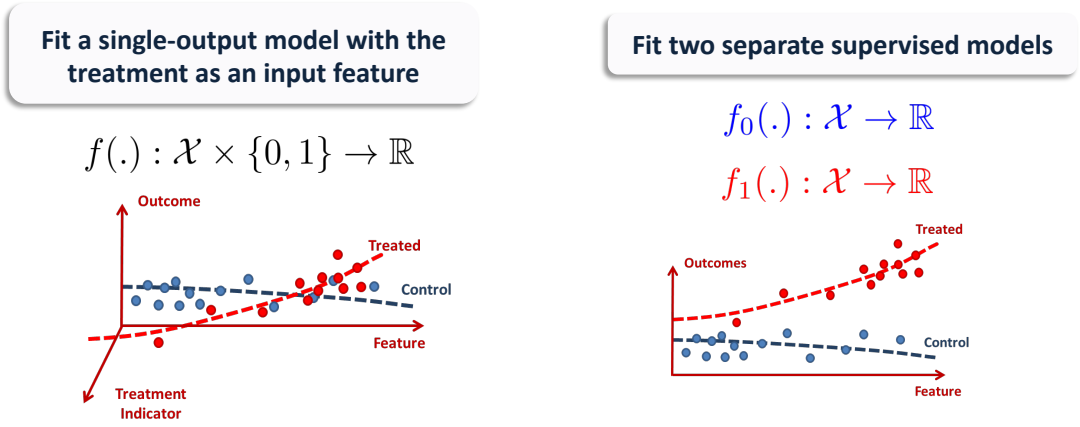


Figure 2.4: Different approaches for estimating individualized patient outcomes using machine learning. Methods that use the approach on the left are less flexible, as they include the treatment as an input feature and thus have the same outcome model for both treated and untreated patients. On the other hand, methods using the approach on the right fit different models for the different treatments and thus provide more flexibility in estimating patient outcomes. Such an approach allows for model structures with “outcome-specific" hyperparameters.

probabilities under the possible treatment options. Conditioning on the balancing representations to estimate the patient outcomes effectively reduces the selection bias present in observational datasets [102].

Many methods have been developed for estimating individualized treatment effects, each using different ways of modelling response surfaces and handling selection bias. Bayesian Additive Regression Trees (BART) [24, 103], Causal Forest [1, 27] and Bayesian Causal Forest [104] use tree-based estimators. Balancing Neural Networks (BNN) [29], Treatment-Agnostic Representation Network (TARNet) [102], Counterfactual Regression (CFR) [102], Similarity Preserved Individual Treatment Effect (SITE) [28] are based on neural networks and build balancing representations of the patients’ features. Deep Counterfactual Networks with Propensity Dropout [25] are also based on neural networks and use a regularization scheme based on the propensity score. MTDL-KNN [78] is a multi-task deep learning model that builds patient representations predictive of their assigned treatment and uses matching based on K-Nearest Neighbours to estimate patient outcomes. Causal Multi-task Gaussian Processes (CMGP) [26] and Non-stationary Gaussian Processes (NSGP) [31] model the data distribution through Gaussian Processes. Conversely, GANITE [35] use the generative adversarial framework [49] for modelling that observational data distribution and estimate counterfactual outcomes. Table 2.2

Method	Using outcome-specific hyperparameters	Method for handling selection bias	Machine learning model
BART [24, 103]	No	None	Regression Trees
Causal Forest [1, 27]	No	Propensity-guided splitting rule	Random Forests
CMGP, NSGP [26, 31]	Yes	Propensity-aware hyperparameter tuning	Multi-task Gaussian Processes
BNN [29]	No	Representation balancing	Neural Network
TARNet [102]	Yes	None	Multi-task Neural Network
CFR [102], SITE [28]	Yes	Representation balancing	Multi-task Neural Network
MTDL-KNN [78]	Yes	Matching based on representation learning	Multi-task Neural Network
PD-NN [25]	Yes	Propensity-based regularization	Multi-task Neural Network
GANITE [35]	Yes	None	Generative Adversarial Network

Table 2.2: Summary of machine learning methods for estimating individualized treatment effects. The modelling choices for the causal inference methods in the cross-sectional setting are mainly defined by the way they model the treatment response surfaces and by the way they handle the treatment selection bias.

provides a summary of the current state-of-the-art methods for estimating treatment effects and highlights their design choices with respect to using outcome-specific hyperparameters, handling the selection bias and the type of machine learning model used.

Several of these methods allow for the computation of *confidence intervals* for the potential outcome estimates: CMGP, NSGP, Causal Forest and GANITE. In addition, the Causal Forests, CMGP and NSGP models have built-in variable selection, which allows us to understand which patient features are relevant for the patient's response to treatment. The methods described, except for GANITE, were designed for binary treatment assignments. Nevertheless, by comparing treatments in pairs, such methods can also be adapted to the case of multiple treatment options. GANITE, on the other hand, can be directly used to estimate potential outcomes for multiple treatments.

Alaa and van der Schaar [31, 105] developed a unifying theory highlighting the *fundamental limits* of learning individualized treatment effects from observational data, as well as how they can be achieved. In light of the theory, [31] provides several guidelines for the algorithmic design choice depending on the observational data available. Thus, [31] shows that when limited amounts of observational data are available (small sample regime) it is vital to handle the selection bias and it is better to share training data between the response surfaces, by either fitting a single regression model for the two response surfaces with the treatment as an input feature, or by using multi-task learning to estimate

the response surfaces. The causal inference methods that are most appropriate in this case are CMGP, NSGP, CFR and SITE. Conversely, when the observational dataset has enough samples (large sample regime), selection bias no longer represents a serious problem and the model structure plays a more crucial role—that is, the way in which outcomes are modeled and how hyperparameter tuning is performed. Thus, using outcome-specific hyperparameters become more important. The methods that are more appropriate in this case are TARNet and GANITE. These theoretical results are essential for understanding which algorithms to use depending on the patient observational data available for training.

These methods for causal inference in the static setting have been designed to be general and applicable to various types of observational datasets. They are usually evaluated on semi-synthetic datasets with real patient features and simulated outcomes. The outcomes need to be simulated to be able to have access to the counterfactuals. Alaa and van der Schaar [26] evaluate the CMGP model on estimating the survival benefits of using Left Ventricular Assistance Devices on patients awaiting a heart donor. Moreover, Wendling et al. [106] evaluate BART [24, 103] and Causal Forest [1, 27] on estimating the impact of the volume of transcatheter aortic valve replacement on 30-day hospital readmission [107], as well as the effect of endovascular and surgical revascularization therapy for critical limb ischemia on 30-day hospital readmission [108]. Alternatively, [78] also evaluates the ability of their method to match the factual treatment recommendation present in the dataset. For this purpose, they use a heart failure dataset and asses the effect of the following medication: ACEI/ARB, beta-blockers and aldosterone antagonist on 1-year hospital readmission.

2.2.3 Individualized Dose-Response Estimation

Most of the methods developed for estimating individualized treatment effects from observational data focus on the binary or categorical treatment settings, and very few methods consider the impact of the associated *treatment dosage* on the patient's response. Nevertheless, when making treatment recommendations, the specific dosage chosen for each treatment also has a critical role in the decision. Observational patient data about how different patients have responded to different dosages can also be used to train

machine learning methods for estimating individualized *dose-response curves*. Naturally, being able to estimate such curves for multiple treatments can aid us in determining the best treatment-dosage pair for each patient, as well as to obtain therapeutic indices of drugs tailored to the heterogeneous patient features. This becomes increasingly important when the relationships between drug toxicity, efficacy and heterogeneous patient features are complex.

In this setting, we need observational data consisting of patient features $X_i \in \mathcal{X}$, assigned treatment $W_i \in \mathcal{W}$, assigned dosage $D_i \in \mathcal{D}$ and observed outcome $Y_i \in \mathbb{R}$ for each patient i. The potential outcomes framework [79, 80] can also be extended to dose-response estimation. Thus, for each treatment-dosage pair (w, d) let $Y_i(w, d) \in \mathcal{Y}$ (e.g. 1-year survival probability) be the potential outcome for patient i. The aim of machine learning methods is to derive unbiased estimates of the potential outcomes $\mu((w, d), \mathbf{x}) = \mathbb{E}[Y(w, d)|X = x]$, for each treatment dosage pair (w, d) and for any patient covariates X. $\mu(\cdot)$ represents the individualised dose-response function.

As before, we again face the problem of only observing the factual treatment type W_i and dosage D_i, as well as the factual patient outcome $Y_i = \mathbb{I}_{\{W_i=w,D_i=d)\}}Y(w, d)$ for a patient i. However, in the dose-response setting this problem is exacerbated by the fact that the number of counterfactual outcomes being potentially infinite. Thus, it is no longer possible to estimate a response surface for each possible treatment-dosage outcome. Moreover, observational data will contain not only *treatment* assignment bias, but also *dosage* assignment bias. Due to the continuous nature of the dosage parameter, standard methods for adjusting for the treatment selection bias in the categorical treatment setting no longer apply—or at least need to be significantly adjusted to also handle the bias in the dosage assignment.

Several methods developed for estimating individualized dose-response curves make use of the *generalized propensity score* [109–111], which represents the probability of the treatment-dosage assignment given the patient's characteristics. The propensity score is used to weight individual samples in the regression method for estimating the outcome. Such methods require correct specification of the generalized propensity score model, and can also be numerically unstable due to extreme propensity weights. More

recently, Schwab et al. [36] developed a deep learning model for this problem, named the Dose-Response Network (DRNets). DRNet proposes subdividing the dosage interval for each treatment into equally sized sub-intervals, and then using multi-task learning for estimating patient outcomes for each sub-interval. However, this architecture restricts the shape of the dose-response function; moreover, while DRNet can be augmented to handle the treatment selection bias, it does not handle the dosage selection bias. Bica et al. [54] propose SCIGAN, a model that uses generative adversarial networks to learn the data distribution of the counterfactual outcomes and thus generate individualized dose-response curves. SCIGAN does not place any restrictions on the form of the treatment-dose response functions, and is capable of estimating patient outcomes for multiple treatments, each with an associated dosage parameter. These methods have also been evaluated on semi-synthetic medical datasets with simulated outcomes and have considered the effect of antibiotic dosage on patient outcomes [54] or the effect of dosage in cancer treatments such as chemotherapy, radiotherapy and immunotherapy on patient relapse [36, 54].

We note that, despite its importance, the dose-response problem has been less studied in the causal inference literature. To be able to achieve the full potential of machine learning methods for learning individualized dose-response curves, additional theory is needed to understand which model structures learn more efficiently, how to best handle treatment and dosage selection bias, as well as how to best select models in this context. The challenge in a static setting is recognized but the dose-response problem using longitudinal data is even more complex.

2.3 Individualized Effects of Time-dependent Treatments

How Can Machine Learning Help to Make Treatment Decisions in the Longitudinal Setting?

EHRs also contain information about *time-dependent* patient covariates and treatment strategies, including how treatment plans are changed over time due to adverse effects or drug resistance. Learning treatment effects in the cross-sectional setting can be used to understand what is the optimal time to administer a treatment, when the treatment regime needs to be stopped, or in which order treatments should be given to obtain the

best patient response. Answering these questions involves modelling a patient's history and how the patient's features are affected by treatments in a longitudinal setting. By estimating counterfactual trajectories under different treatment plans in the future, we can then choose the sequence and timing of treatments that lead to the best patient outcome. Machine learning methods for estimating treatment effects over time can therefore be used to support decision making in the temporal setting.

Besides baseline patient features such as gender and genetic information, longitudinal information about the patient is collected and added to the their EHR for each follow-up event. This information may include lab test, specific treatments administered, the patient's performance status while undergoing treatment, etc. While the patient is receiving particular treatments, it is possible that the treatment's toxicity becomes too high (for instance, in the case of chemotherapy) or the patient starts developing resistance to the treatment. In this case, machine learning methods can be used to estimate what would happen to the patient if the treatment strategy were changed. The counterfactual outcomes that depend on the patient's history can be used to decide what is the best future treatment plan. The counterfactual predictions can be made in a dynamic fashion, by modelling and updating the patient's state over time to take into account all past medical events. At each time point, we can assess whether the treatment strategy chosen for the patient is beneficial or if it needs to be modified to prevent adverse outcomes.

In addition to estimating single or multiple patient outcomes, in the longitudinal setting an outcome of interest can either be observed immediately after receiving the treatment (at the next timestep), or after a planned sequence of treatments. In terms of treatment options, we can model time-dependent binary treatment applications (i.e. treating or not treating at a specific time), multiple combinations of treatments, as well as treatment dosages. Machine learning methods for estimating individualized treatment effects in these scenarios can be used not only for choosing the best sequences of treatments, but also for choosing their best timing. In Figure 2.5 we illustrate such an example for a breast cancer patient.

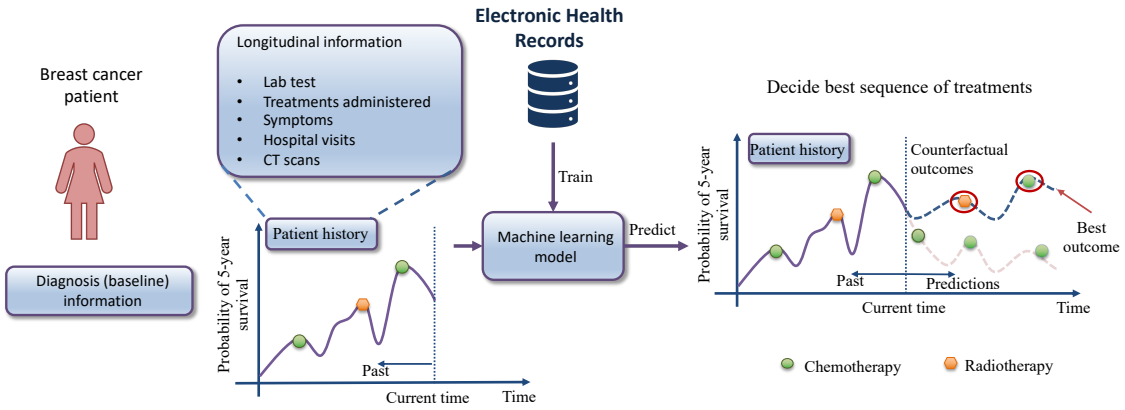


Figure 2.5: Causal inference methods for estimating individualized effects of time-dependent treatments need to model the patient's history in order to make counterfactual predictions of potential outcomes for possible sequences of future treatments. We illustrate an example of how such predictions can be used to decide the best timing and order of chemotherapy and radiotherapy treatments for a patient with breast cancer.

2.3.1 Challenges in Using Observational Data for Treatment Decisions in the Time-Series Setting

Estimating treatment effects in the longitudinal setting becomes more challenging than in the cross-sectional setting due to the presence of *time-dependent confounders*: patient covariates that are affected by past treatments and which then influence future treatments. For instance, consider for type II diabetic patients that we want to estimate the effect of an antiglycemic drug on the risk of experiencing a cardiac event. Sustained high blood glucose levels for a patient will result in an escalation of the treatment, while also leading to an increased risk of cardiac events. Thus, blood glucose levels are confounders that are part of the causal path to the risk of the cardiac event. Without adjusting for the bias introduced by these confounders, we may incorrectly conclude that the antiglycemic drug is harmful. Since the blood glucose levels vary over time and are affected by past treatments (e.g. an effective antiglycemic drug will lower the blood glucose levels) it is considered a time-varying confounder.

In the longitudinal setting, we once again only observe factual patient trajectories, but we need to estimate counterfactuals in order to make treatment decisions. To be able to identify the counterfactual patient outcomes from observational data, we also need the overlap and unconfoundedness assumptions to hold. In this case, overlap means that at

each timestep we assume that each treatment option has a non-zero probability of being administered. The overlap assumption can be assessed by computing the probability of a patient for receiving each possible treatment at each time point. Moreover, the unconfoundedness assumption in the sequential setting means that at each timestep we observe all variables affecting the patient's treatment and outcome. Validating this assumption requires domain expert knowledge, performing sensitivity analysis [91–93], or estimating substitutes for the hidden confounders [56]. For instance, Bica et al. [56] takes advantage of the sequential assignment of multiple treatments in order to estimate the effect of time-varying treatments in the presence of hidden confounders. They apply their model to improve the estimation of the effect of antibiotics, mechanical ventilator and vasopressors on a patient's white blood cell count, blood pressure and oxygen saturation.

2.3.2 Modelling Choices and Methods for Estimating Individualized Effects of Time-Dependent Treatments

In the setting of estimating the effects of time-varying treatments, we consider that for each patient i we observed time-dependent patient features $X_{i,t} \in \mathcal{X}_t$, treatments administered $W_{i,t} \in \mathcal{W}_t$, and outcomes $Y_{i,t+1} \in \mathbb{R}$ for T_i timesteps. The dependence on time is indicated by the subscript t. Note that while we provide notation for the discrete time-setting, this framework can be easily extended to continuous time. Moreover, the patient can also have baseline covariates $V_i \in \mathcal{V}$ such as genetic information, age, and gender.

The potential outcomes framework proposed by [79, 80] for the cross-sectional setting has been extended by [81] to account for time-varying treatments. Thus, let $Y(\bar{w})$ be the potential outcomes, either factual or counterfactual, for each possible course of treatment $\bar{w}$. In this setting, $\bar{w}$ can represent either an individual treatment application, or a possible sequence of future treatments.

Let $H_{i,t} = (V_i, X_{i,0}, W_{i,0}, \ldots, X_{i,t-1}, W_{i,t-1}, X_{i,t}) = (V_i, X_{i,0:t}, W_{i,0:t-1})$ be the history of patient i until time t. The history $H_{i,t}$ contains the baseline variables, the evolution of the time-dependent features and the treatment assignments until time t. We are interested in characterising the distribution of potential outcomes for any

patient history H_t:

$$\mathbb{E}[Y_{t+\tau}(w(t, t+\tau-1))|H_t] \tag{2.1}$$

where $w(t, t+\tau-1) = [w_t, \dots w_{t+\tau-1}]$ represents a possible sequence of treatments from timestep t just until before the potential outcome $Y_{t+\tau}$ is observed.

Modelling choices in the estimation of treatment effects over time involve handling the time-dependent confounding bias, handling discrete-time or continuous-time interventions, and modelling the progression of one or multiple patient covariates of interest.

The main methods for handling the time-dependent confounding bias were originally proposed in the epidemiology literature and include the g-computation formula, g-estimation of structural nested mean models and inverse probability of weighting estimation of marginal structural models [14, 81, 112, 113]. These methods are based on strong theoretical foundations for removing the time-dependent confounding bias. However, they were originally developed with prediction models based on linear/logistic regression which make strong assumptions about the underlying data structure, i.e that the outcome is a linear function of the history of patient covariates and treatments. Additionally, to handle the changes in the patient covariates and treatments over time, these prediction models based on linear/logistic regression also make strong assumptions about which of the previous timesteps affect the current one and also require different sets of coefficients to be estimated at each timestep. Consequently, such methods are not capable of handling complex non-linear relationships and long-term dependencies between the history of patient covariates, treatments and potential outcomes.

Several methods based on Bayesian non-parametrics have been proposed to estimate the effects of longitudinal treatment assignments in discrete-time [114] and continuous-time settings [40, 115, 116]. These models based on Gaussian Processes also make strong assumptions about how the treatments and patient features interact. As a result, the heterogeneous treatment effects resulting from baseline variables (e.g. genetic and demographic information) are either omitted [40] or incorporated as a linear component [114]. Moreover, [40, 114] only model how treatments affect the progression of a single patient covariate of interest. This poses severe limitations in the medical context where

Method	Method for handling time-dependent confounders	Machine learning model for estimating outcomes	Time
MSMs [14]	Inverse probability of treatment weighting	Linear regression model	Discrete
Xu et al. [114]	G-computation formula	Bayesian Nonparametrics	Continuous
Schulam et al. [40]	Modelling treatment assignments using marked point process	Gaussian Process model	Continuous
Soleimani et al. [116]	Model treatment effects using Linear Time Invariant Dynamical System	Multi-output Gaussian Process	Continuous
RMSNs [117]	Inverse probability of treatment weighting	Sequence-to-sequence RNN model	Discrete
CRN [55]	Balancing representations	Sequence-to-sequence RNN model	Discrete

Table 2.3: Summary of machine learning methods for estimating individualized treatment effects of time-varying treatments. The modelling choices for the causal inference methods for estimating the effects of time-varying treatments are defined by the way they model they handle the bias from time-dependent confounders and the time series model used for estimating the outcomes.

the patient features interact with each other, and where treatments can affect multiple patient features at the same time (for instance, when treatments have side effects).

Lim et al. [117] propose Recurrent Marginal Structural Networks (RMSNs), which can estimate treatment effects in the discrete-time setting and can incorporate baseline patient features as well as model the progression of multiple clinical covariates and treatments of interest. RMSNs improve on standard Marginal Structural Models by using recurrent neural networks (RNNs) to estimate the inverse probability of treatment weights and the counterfactual treatment outcomes. Through inverse probability of treatment weighting, these models create a pseudo-population where, at each timestep, the probability of treatment does not depend on the time-varying confounders. However, RMSNs are not robust to model miss-specification in computing the sequential treatment probabilities and can also be numerically unstable due to extreme weights. Bica et al. [55] introduce the Counterfactual Recurrent Network (CRN), which builds balancing representations to remove the time-dependent confounding bias, thereby overcoming the problems of inverse probability of treatment weighting. CRN estimates individualized treatment effects in the discrete-time setting and allows for flexible modelling of the patient's baseline information and progression of multiple clinical variables and treatments of interest. We provide in Table 2.3 a comparison of the causal inference methods for estimating the effects of time-varying treatments.

Although these methods have been developed to be general and applicable to various types of observational datasets from EHRs, we describe several example applications

that these methods have been used for in practice. To begin with, Xu et al. [114] consider patients with sepsis in the intensive care unit (ICU) and evaluate the ability of their model to estimate the individualized effect of intermittent hemodialysis and continuous renal replacement therapy on creatine levels, a measure of kidney deterioration. Schulam and Saria [40] consider the same clinical scenario to estimate the effect of intermittent hemodialysis, continuous venovenous hemofiltration, and continuous venovenous hemodialysis on creatine levels. Alternatively, Bica et al. [55] also use observational data from the ICU and evaluate the ability of their proposed CRN model to estimate the personalized effect of antibiotics on white blood cell count. Note that, on real datasets, these methods are usually evaluated on their ability to estimate the factual patient. However, as we discuss in Section 2.4, this is not enough for a thorough validation of these methods.

These models for performing counterfactual estimation of future treatment outcomes can be used to answer critical medical questions such as deciding when to give treatments, when to start and stop treatment regimes, as well as how to select from multiple treatments over time. However, the estimation of treatment effects over time has been less studied in the causal inference literature compared to the estimation of the causal effects of binary treatments in the static setting. As such, this setting of estimating the effects of time-dependent treatments also requires the development of better methods for handling the time-dependent confounding bias, additional theoretical understanding and more principled methods of performing model selection.

2.4 Evaluation of Causal Inference Models

Evaluating causal inference methods and deploying them in practice still represents a challenge. In standard supervised learning, one can choose the best performing method by computing the prediction error on a validation dataset, and then selecting the model with the lowest error. Since we do not observe counterfactuals, for causal inference models, it is not possible to compute the error on estimating the ground truth causal effects on real data.

An initial step for validating causal inference methods involves using synthetic data. To create a synthetic observational dataset, parametric functions, such as polynomials

or exponentials are usually used to generate patient outcomes under different treatment options [26, 54]. Confounding bias is introduced by assigning treatments according to Bernoulli or Categorical random variables, with parameters depending on the patient characteristics [35, 54]. In this synthetic set-up, the causal inference methods can indeed be evaluated based on their error on estimating all potential outcomes and the ground truth causal effects. The sensitivity of the causal inference methods to various degrees of confounding bias can also be assessed by changing the parameters of the probability distributions used to assign treatments for the patients in the observational data generated for training [54]. While this allows us to perform a preliminary evaluation and assess the robustness of the machine learning methods for estimating individualized treatment effects, synthetic data generated in this way will not capture the complexities of real-world data.

Taking this a step further, Pharmacokinetic/Pharmacodynamic (PK/PD) modelling could provide a more realistic approach of evaluating and selecting among causal inference methods. For instance, [117] proposed using a PK/PD model for non-small cell lung cancer which models the evolution of tumour growth under the combined effects of chemotherapy and radiotherapy [118]. This PK/PD model has been used to obtain counterfactual outcomes and thus evaluate causal inference methods for estimating treatment effects over time.

Furthermore, for each observational dataset, several causal inference methods may be appropriate to use and in fact, no model will perform best on all datasets [119]. As discussed in Section 2.3, different modelling choices for the causal inference methods will be important depending on the size of the observational dataset available for training [31]. The theoretical guidelines provided by Alaa and van der Schaar [31] and validation through synthetic data enables us to narrow down the choice of suitable causal inference methods for the observational data available. Nevertheless, data-driven validation procedures, similar to cross-validation for standard machine learning tasks [120] are needed to find the most suitable causal inference method for a given real-world study. We discuss further possible ways of performing internal model evaluation (selecting the causal inference model that performs the best on a held-out test set of the available

observational data) and external model evaluation (assessing causal inference methods once they have been deployed in practice).

For internal model evaluation, Alaa and van der Schaar [119] propose a method for assessing causal inference models by using *influence functions* (a technique in robust statistics and efficiency theory [121, 122]) to estimate the loss of machine learning models for causal inference without requiring counterfactual data. Model selection is an essential first step before deploying any causal inference method in practice. The method proposed by Alaa and van der Schaar [119] allows us to approximate the error of causal inference methods on a held-out test set and subsequently enables us to select the best performing model for an observational dataset of interest. While this method has been developed for the cross-sectional setting, future work could extend it to the setting of time-varying treatments.

Once a model has been deployed in practice, external model evaluation is needed. In this context, the machine learning method for causal inference can be evaluated by measuring whether the patient outcomes have been improved when assigning treatments according to the recommendation of the causal inference model.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	From real world patient data to individualized treatment effects using machine learning: Current and future methods to address underlying challenges
Publication Status	☑ Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Ioana Bica, Ahmed M. Alaa, Craig Lambert, and Mihaela van der Schaar. "From real world patient data to individualized treatment effects using machine learning: Current and future methods to address underlying challenges". In *Clinical Pharmacology & Therapeutics*, 2020.

Student Confirmation

Student Name:	Ioana Bica		
Contribution to the Paper	- I was the lead author responsible for conceptualizing the paper, structuring and writing the original draft. - All authors contributed to the development of the paper through multiple discussions. - All authors reviewed the final draft.		
Signature 		Date	30/03/2022

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Mihaela van der Schaar		
Supervisor comments I agree that the candidate has made a substantial contribution to the publication and that the above description is accurate.		
Signature 	Date	30/03/2022

This completed form should be included in the thesis, at the end of the relevant chapter.

3

Estimating the Causal Effects of Continuous-valued Interventions

Contents

3.1 Introduction 41
3.2 Related Work 44
3.3 Problem Formulation 45
3.4 SCIGAN 46
3.4.1 Counterfactual Generator 46
3.4.2 Counterfactual Discriminator 47
3.4.3 Inference Network 49
3.4.4 Theoretical Analysis 49
3.5 Architecture 49
3.5.1 Generator Architecture 50
3.5.2 Hierarchical Discriminator Architectures 50
3.6 Evaluation 51
3.6.1 Experimental Setup 52
3.6.2 Source of Gain 53
3.6.3 Benchmarks Comparison 54
3.6.4 Discrete Dosages 54
3.6.5 Treatment and Dosage Selection Bias 55
3.7 Conclusion 55

Publication Included

- **Ioana Bica***, James Jordon*, and Mihaela van der Schaar. "Estimating the Effects of Continuous-valued Interventions using Generative Adversarial Networks". In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. Link: https://arxiv.org/abs/2002.12326. [Accepted]

Abstract

While much attention has been given to the problem of estimating the effect of discrete interventions from observational data, relatively little work has been done in the setting of continuous-valued interventions, such as treatments associated with a dosage parameter. In this chapter, we tackle this problem by building on a modification of the generative adversarial networks (GANs) framework. Our model, SCIGAN, is flexible and capable of simultaneously estimating counterfactual outcomes for several different continuous interventions. The key idea is to use a significantly modified GAN model to learn to generate counterfactual outcomes, which can then be used to learn an inference model, using standard supervised methods, capable of estimating these counterfactuals for a new sample. To address the challenges presented by shifting to continuous interventions, we propose a novel architecture for our discriminator - we build a hierarchical discriminator that leverages the structure of the continuous intervention setting. Moreover, we provide theoretical results to support our use of the GAN framework and of the hierarchical discriminator. In the experiments section, we introduce a new semi-synthetic data simulation for use in the continuous intervention setting and demonstrate improvements over the existing benchmark models.

3.1 Introduction

Estimating the personalised effects of interventions is crucial for decision making in many domains such as medicine, education, public policy and advertising. Such domains have a wealth of observational data available. Most of the methods developed in the causal inference literature focus on learning the counterfactual outcomes of discrete

interventions, such as binary or categorical treatments[1] [1, 25–27, 31, 32, 35, 53, 123]. Unfortunately, in many cases, deciding how to intervene involves not only deciding which intervention to make (e.g. whether to treat cancer with radiotherapy, chemotherapy or surgery) but also deciding on the value of some continuous parameter associated with intervening (e.g. the dosage of radiotherapy to be administered). In medicine there are many examples of treatments that are associated with a continuous dosage parameter (such as vasopressors [124]). In the medical setting, using a high dosage for a treatment can lead to toxic effects while using a low dosage can result in no effect on the patient outcome [64, 125]. In other domains, there are many examples of continuous interventions, such as the duration of an education or job training program, the frequency of an exposure or the price used in an advert. Naturally, being able to estimate the effect of these continuous interventions will aid in the decision making process.

Learning from observational data already presents significant challenges when there is only a single intervention (and thus the decision is binary - whether to intervene or not). As explained in [126], in an observational dataset, only the factual outcome is present - the "counterfactual" outcomes are not observed. This problem is exacerbated in the setting of continuous interventions where the number of counterfactuals is no longer even finite. Moreover, the decision to intervene is non-random and instead is assigned according to the features associated with each sample. Due to the continuous nature of the interventions, adjusting for selection bias is significantly more complex than for binary (or even multiple) interventions. Thus, standard methods for adjusting for selection bias for discrete treatments cannot be easily extended to handle bias in the continuous setting.

We propose SCIGAN (eStimating the effects of Continuous Interventions using GANs). We build on the GAN framework of [49] to learn the distribution of the unobserved counterfactuals. GANs have already been used in GANITE [35] to generate the unobserved counterfactual outcomes for discrete interventions. The intuition is that if a counterfactual generator and discriminator are trained adversarially, then the generator can fool the discriminator (i.e. the discriminator will not be able to correctly identify the factual outcome) by generating counterfactuals according to their true

[1]For ease of exposition, we will sometimes refer to interventions as treatments and to the associated continuous parameter as the dosage throughout the chapter.

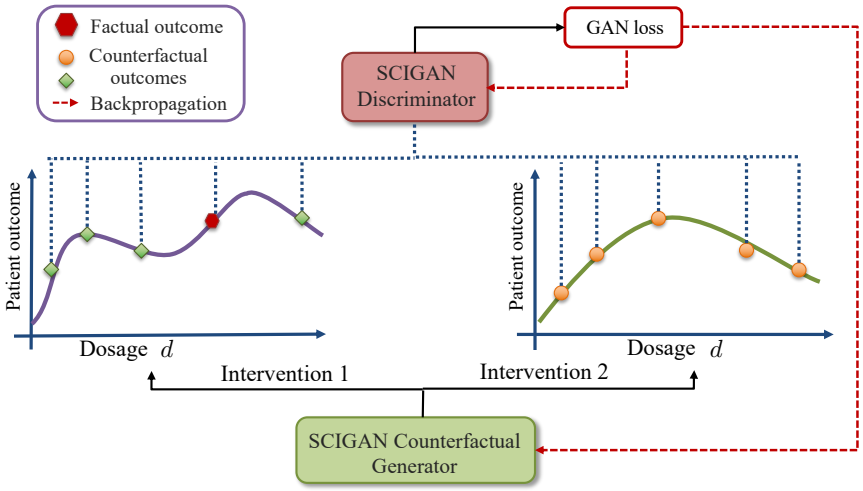


Figure 3.1: Overview of GAN framework used as part of SCIGAN for learning the distribution of the counterfactual outcomes.

distribution. Unfortunately, no theoretical work was provided in [35] to back up this intuition. A key contribution of this paper is to provide theoretical results that justify using the GAN framework to learn to generate counterfactual outcomes; these results also apply to GANITE.

GANITE itself presents a significant modification to the original GAN framework - rather than the discriminator discriminating between entirely real or entirely fake samples, the discriminator is attempting to identify the real component from a vector containing the real (factual) outcome from the dataset and the fake (counterfactual) outcomes generated by the generator. SCIGAN inherits this key difference from a standard GAN. However, beyond our theoretical contribution, we propose significant changes to the generator and discriminator in order to tackle the more complex problem of estimating outcomes of continuous interventions.

We define a discriminator that acts on a finite set of points from each generated response curve (rather than on entire curves), as shown in Fig. 3.1. We draw on ideas from [127] to ensure that our discriminator acts as a function of a set, rather than of a vector. Moreover, we focus on the setting of interventions *each* with an associated continuous parameter. In this setting, we propose a *hierarchical* discriminator which breaks down the job of the discriminator into determining the factual intervention and determining the factual parameter using separate networks. We show in the experiments section that this approach significantly improves performance and is more stable than

using a single network discriminator. We also model the generator as a multi-task deep network capable of taking a continuous parameter as an input; this gives us the flexibility to learn heterogeneous response curves for the different interventions.

Our contributions in this chapter are 4-fold: (1) we propose SCIGAN, a significantly modified GAN framework, capable of estimating outcomes for continuous and many-level-discrete interventions, (2) we provide theoretical justification for both the use of a GAN framework and a hierarchical discriminator, (3) we propose novel architectures for each of our networks, (4) we propose a new semi-synthetic data simulation for use in the continuous intervention setting. We show, using semi-synthetic experiments, that our model outperforms existing benchmarks.

3.2 Related Work

Methods for estimating the outcomes of treatments with a continuous dosage from observational data make use of the generalized propensity score (GPS) [109–111] or build on top of balancing methods for multiple treatments. [36] developed a neural network based method to estimate counterfactuals for multiple treatments and continuous dosages. The proposed Dose Response networks (DRNets) in [36] consist of a three level architecture with shared layers for all treatments, multi-task layers for each treatment and additional multi-task layers for dosage sub-intervals. Specifically, for each treatment w, the dosage interval $[a_w, b_w]$ is subdivided into E *equally* sized sub-intervals and a multi-task head is added for each sub-interval. This is an extension of the architecture in [102]. However, the main advantage of using multi-task heads for dosage intervals would be the added flexibility in the model to learn potentially very different functions over different regions of the dosage interval. DRNets do not determine these intervals dynamically and thus much of this flexibility is lost. Our approach (using GANs to generate counterfactuals) fundamentally differs from DRNets (supervised learning with bias-adjustment) and we demonstrate experimentally that SCIGAN outperforms GPS and DRNets.

For a discussion of works that address treatment-response estimation without a dosage parameter, see Appendix A.1. Note that for such methods we cannot treat the dosage as

an input due to the bias associated with its assignment. In Appendix A.1, we also describe the relationships between our work (causal inference for continuous interventions) and policy optimization with continuous treatments.

3.3 Problem Formulation

We consider receiving observations of the form $(\mathbf{x}^i, t_f^i, y_f^i)$ for $i = 1, ..., N$, where, for each i, these are independent realizations of the random variables $(\mathbf{X}, T_f, Y_f)$. We refer to $\mathbf{X}$ as the feature vector lying in some feature space $\mathcal{X}$, containing pre-treatment covariates (such as age, weight and lab test results). The treatment random variable, T_f, is in fact a pair of values $T_f = (W_f, D_f)$ where $W_f \in \mathcal{W}$ corresponds to the *type* of treatment being administered (e.g. chemotherapy or radiotherapy) which lies in the discrete space of k treatments, $\mathcal{W} = \{w_1, ..., w_k\}$, and D_f corresponds to the *dosage* of the treatment (e.g. number of cycles, amount of chemotherapy, intensity of radiotherapy), which, for a given treatment w lies in the corresponding treatment's dosage space, $\mathcal{D}_w$ (e.g. the interval $[0, 1]$). We define the set of all treatment-dosage pairs to be $\mathcal{T} = \{(w, d) : w \in \mathcal{W}, d \in \mathcal{D}_w\}$.

Following Rubin's potential outcome framework [128], we assume that for all treatment-dosage pairs, (w, d), there is a potential outcome $Y(w, d) \in \mathcal{Y}$ (e.g. 1-year survival probability). The *observed* outcome is then defined to be $Y_f = Y(W_f, D_f)$. We will refer to the unobserved (potential) outcomes as *counterfactuals*.

The goal is to derive *unbiased* estimates of the potential outcomes for a given set of input covariates:

$$\mu(t, \mathbf{x}) = \mathbb{E}[Y(t)|\mathbf{X} = \mathbf{x}] \tag{3.1}$$

for each $t \in \mathcal{T}$, $\mathbf{x} \in \mathcal{X}$. We refer to $\mu(\cdot)$ as the individualised dose-response function. In order to ensure that this quantity is equal to $\mathbb{E}[Y|\mathbf{X} = \mathbf{x}, T = t]$ and that the dose-response function is identifiable from the observational data, we require the following two assumptions.

Assumption 3.3.1. *(Unconfoundedness) The treatment assignment, T_f, and potential outcomes, $Y(w, d)$, are conditionally independent given the covariates $\mathbf{X}$, i.e. $\{Y(w, d)|w \in \mathcal{W}, d \in \mathcal{D}_w\} \perp\!\!\!\perp T_f|\mathbf{X}$.*

Assumption 3.3.2. *(Overlap)* $\forall \mathbf{x} \in \mathcal{X}$ *such that* $p(\mathbf{x}) > 0$*, we have* $1 > p(t|\mathbf{x}) > 0$ *for each* $t \in \mathcal{T}$*.*

3.4 SCIGAN

We propose estimating μ by first training a generator to generate response curves for each sample *within* the training dataset. The learned generator can then be used to train an inference network using standard supervised methods. We build on the idea presented in [35] that uses a modified GAN framework to generate potential outcomes conditional on the observed features, treatment and factual outcome. Several changes must be made to both the generator and discriminator architectures and learning paradigms in order to produce a model capable of handling the dose-response setting.

3.4.1 Counterfactual Generator

Our generator, $\mathbf{G} : \mathcal{X} \times \mathcal{T} \times \mathcal{Y} \times \mathcal{Z} \to \mathcal{Y}^{\mathcal{T}}$ takes features, $\mathbf{x} \in \mathcal{X}$, factual outcome, $y_f \in \mathcal{Y}$, received treatment and dosage, $t_f = (w_f, d_f) \in \mathcal{T}$, and some noise, $\mathbf{z} \in \mathcal{Z}$ (typically multivariate uniform or Gaussian), as inputs. The output will be a dose-response curve for each treatment (as shown in Fig. 3.1), so that the output is a function from $\mathcal{T}$ to $\mathcal{Y}$, i.e. $\mathbf{G}(\mathbf{x}, t_f, y_f, \mathbf{z})(\cdot) : \mathcal{T} \to \mathcal{Y}$. We can then write

$$\hat{y}_{cf}(t) = \mathbf{G}(\mathbf{x}, t_f, y_f, \mathbf{z})(t) \tag{3.2}$$

as our generated counterfactual outcome for treatment-dosage pair t. We write $\hat{Y}_{cf}(t) = \mathbf{G}(\mathbf{X}, T_f, Y_f, \mathbf{Z})(t)$ (i.e. the random variable induced by $\mathbf{G}$).

While the job of the counterfactual generator is to generate outcomes for the treatment-dosage pairs which were *not* observed, [35] demonstrated that the performance of the counterfactual generator is improved by adding a supervised loss term that regularises its output for the factual treatment (in our case treatment-dosage pair). We define the supervised loss, $\mathcal{L}_S$, to be

$$\mathcal{L}_S(\mathbf{G}) = \mathbb{E}\left[\left(Y_f - \mathbf{G}(\mathbf{X}, T_f, Y_f, \mathbf{Z})(T_f) \right)^2 \right], \tag{3.3}$$

where the expectation is taken over $\mathbf{X}, T_f, Y_f$ and $\mathbf{Z}$.

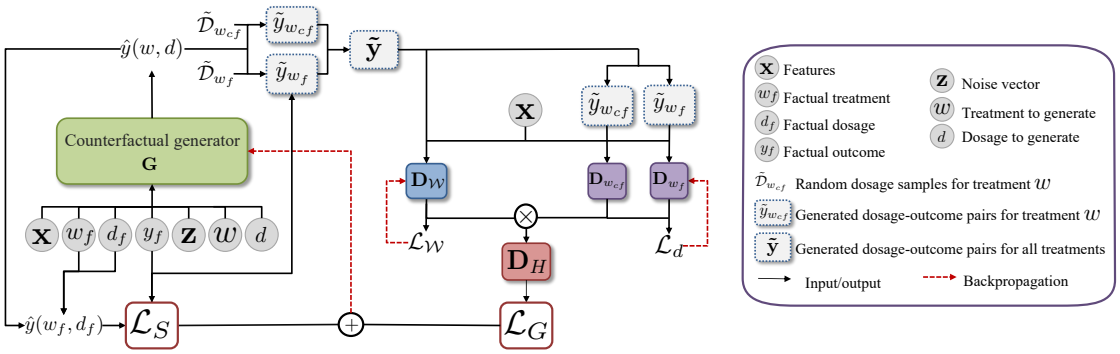


Figure 3.2: Overview of our model for the setting with two treatments (w_f/w_{cf} being the factual/counterfactual treatment). The generator is used to generate an output for each dosage level in each $\tilde{\mathcal{D}}_w$, these outcomes together with the factual outcome, y_f, are used to create the set of dosage-outcome pairs, $\tilde{\mathbf{y}}$, which is passed to the treatment discriminator. Each dosage discriminator receives only the part of $\tilde{\mathbf{y}}$ corresponding to that treatment, i.e. $\tilde{\mathbf{y}}_w$. These discriminators are combined (Eq. 3.7) to define $\mathbf{D}_H$ which gives feedback to the generator.

3.4.2 Counterfactual Discriminator

As noted in Section 3.1, our discriminator will act on a random set of points from each of the generated dose-response curves. Similar to [35], we define a discriminator, $\mathbf{D}$, that will attempt to pick out the factual treatment-dosage pair from among the (random set of) generated ones. To handle the complexity arising from multiple treatments, we break down our discriminator into two distinct models: a treatment discriminator and a dosage discriminator.

Formally, let $n_w \in \mathbb{Z}^+$ be the number of dosage levels we will compare for treatment $w \in \mathcal{W}$[2]. For each $w \in \mathcal{W}$, let $\tilde{\mathcal{D}}_w = \{D_1^w, ..., D_{n_w}^w\}$ be a random subset[3] of $\mathcal{D}_w$ of size n_w, where for the factual treatment, W_f, $\tilde{\mathcal{D}}_{W_f}$ contains $n_{W_f} - 1$ random elements along with D_f. We define $\tilde{\mathbf{Y}}_w = (D_i^w, \tilde{Y}_i^w)_{i=1}^{n_w} \in (\mathcal{D}_w \times \mathcal{Y})^{n_w}$ to be the vector of dosage-outcome pairs for treatment w where

$$\tilde{Y}_i^w = \begin{cases} Y_f \text{ if } W_f = w \text{ and } D_f = D_i^w \\ \hat{Y}_{cf}(w, D_i^w) \text{ else} \end{cases} \tag{3.4}$$

and $\tilde{\mathbf{Y}} = (\tilde{\mathbf{Y}}_w)_{w \in \mathcal{W}}$. We will write $d_j^w, \tilde{\mathbf{y}}_w$ and $\tilde{\mathbf{y}}$ to denote realisations of $D_j^w, \tilde{\mathbf{Y}}_w$ and $\tilde{\mathbf{Y}}$.

The treatment discriminator, $\mathbf{D}_{\mathcal{W}} : \mathbf{X} \times \prod_{w \in \mathcal{W}} (\mathcal{D}_w \times \mathcal{Y})^{n_w} \to [0,1]^k$, takes the features, $\mathbf{x}$, and generated potential outcomes, $\tilde{\mathbf{y}}$, and outputs a probability for each

[2] In practice we set all n_w to be the same. The default setting is 5 in the experiments.

[3] In practice, when $\mathcal{D}_w = [0,1]$, each D_j^w is sampled independently and uniformly from $[0,1]$. Note that for each training iteration, $\tilde{\mathcal{D}}_w$ is resampled (see Section 3.1).

treatment, $w_1, ..., w_k$. Writing $\mathbf{D}_{\mathcal{W}}^w$ to denote the output of $\mathbf{D}_{\mathcal{W}}$ corresponding to treatment w, we define the loss, $\mathcal{L}_{\mathcal{W}}$, to be

$$\mathcal{L}_{\mathcal{W}}(\mathbf{D}_{\mathcal{W}}; \mathbf{G}) = -\mathbb{E}\Bigg[\sum_{w \in \mathcal{W}} \mathbb{I}_{\{W_f = w\}} \log \mathbf{D}_{\mathcal{W}}^w(\mathbf{X}, \tilde{\mathbf{Y}}) + \mathbb{I}_{\{W_f \neq w\}} \log(1 - \mathbf{D}_{\mathcal{W}}^w(\mathbf{X}, \tilde{\mathbf{Y}}))\Bigg], \tag{3.5}$$

where the expectation is taken over $\mathbf{X}, W_f, D_f, \tilde{\mathbf{Y}}$ and $\{\tilde{\mathcal{D}}_w\}_{w \in \mathcal{W}}$.

Then, for each $w \in \mathcal{W}$, the dosage discriminator, $\mathbf{D}_w : \mathcal{X} \times (\mathcal{D}_w \times \mathcal{Y})^{n_w} \to [0,1]^{n_w}$, is a map that takes the features, $\mathbf{x}$, and generated potential outcomes, $\tilde{\mathbf{y}}_w$, corresponding to treatment w and outputs a probability for each dosage level, $d_1^w, ..., d_{n_w}^w$, in a given realisation of $\tilde{\mathcal{D}}_w$. Writing $\mathbf{D}_w^j$ to denote the output of $\mathbf{D}_w$ corresponding to dosage level D_j^w, we define the loss of each dosage discriminator to be

$$\mathcal{L}_d(\mathbf{D}_w; \mathbf{G}) = -\mathbb{E}\Bigg[\mathbb{I}_{\{W_f = w\}} \sum_{j=1}^{n_w} \mathbb{I}_{\{D_f = D_j^w\}} \log \mathbf{D}_w^j(\mathbf{X}, \tilde{\mathbf{Y}}_w) + \mathbb{I}_{\{D_f \neq D_j^w\}} \log(1 - \mathbf{D}_w^j(\mathbf{X}, \tilde{\mathbf{Y}}_w))\Bigg], \tag{3.6}$$

where the expectation is taken over $\mathbf{X}, \tilde{\mathcal{D}}_w, \tilde{\mathbf{Y}}_w, W_f$ and D_f. The $\mathbb{I}_{\{W_f = w\}}$ term ensures that only samples for which the factual treatment is w are used to train dosage discriminator $\mathbf{D}_w$ (otherwise there would be no factual dosage for that sample).

We define the overall discriminator $\mathbf{D}_H : \mathcal{X} \times \prod_{w \in \mathcal{W}} (\mathcal{D}_w \times Y)^{n_w} \to [0,1]^{\sum n_w}$ by defining its output corresponding to the treatment-dosage pair (w, d_j^w) as

$$\mathbf{D}_H^{w,j}(\mathbf{x}, \tilde{\mathbf{y}}) = \mathbf{D}_{\mathcal{W}}^w(\mathbf{x}, \tilde{\mathbf{y}}) \times \mathbf{D}_w^j(\mathbf{x}, \tilde{\mathbf{y}}_w)\,. \tag{3.7}$$

Instead of the standard GAN minimax game, the generator and discriminators are trained according to the minimax game defined by seeking $\mathbf{G}^*$, $\mathbf{D}_H^*$ that solve:

$$\mathbf{G}^* = \arg\min_{\mathbf{G}} \mathcal{L}(\mathbf{D}_H^*; \mathbf{G}) + \lambda \mathcal{L}_S(\mathbf{G}) \qquad \mathbf{D}_H^{*\,w,j} = \mathbf{D}_{\mathcal{W}}^{*\,w} \times \mathbf{D}_w^{*\,j} \tag{3.8}$$

$$\mathbf{D}_{\mathcal{W}}^* = \arg\min_{\mathbf{D}_{\mathcal{W}}} \mathcal{L}_{\mathcal{W}}(\mathbf{D}_{\mathcal{W}}; \mathbf{G}^*) \qquad \mathbf{D}_w^* = \arg\min_{\mathbf{D}_w} \mathcal{L}_d(\mathbf{D}_w; \mathbf{G}^*), \forall w \in \mathcal{W} \tag{3.9}$$

Fig. 3.2 depicts our generator and hierarchical discriminator. Pseudo-code for our algorithm can be found in Appendix A.3.

3.4.3 Inference Network

Once we have learned the counterfactual generator, we can use it only to access (generated) dose-response curves for all samples in the dataset. To generate dose-response curves for a new sample we use the counterfactual generator along with the original data to train an inference network, $\mathbf{I} : \mathcal{X} \times \mathcal{T} \rightarrow \mathcal{Y}$. Details of the loss and pseudo-code can be found in Appendix A.4.

3.4.4 Theoretical Analysis

We now state our key theorem: the game defined by Eqs. (3.8-3.9) results in our hierarchical GAN learning counterfactuals that agree (in marginal distribution) with the true data.

Theorem 3.4.1. *The global minimum of* $\mathcal{L}(\mathbf{D}_H^*; \mathbf{G}) + \lambda \mathcal{L}_S(\mathbf{G})$ *subject to* ${\mathbf{D}_H^*}^{w,j} = {\mathbf{D}_{\mathcal{W}}^*}^{w} \times {\mathbf{D}_w^*}^{j}$, $\mathbf{D}_{\mathcal{W}}^* = \arg\min_{\mathbf{D}_{\mathcal{W}}} \mathcal{L}_{\mathcal{W}}(\mathbf{D}_{\mathcal{W}}; \mathbf{G}^*)$, $\mathbf{D}_w^* = \arg\min_{\mathbf{D}_w} \mathcal{L}_d(\mathbf{D}_w; \mathbf{G}^*), \forall w \in \mathcal{W}$ *is achieved if and only if for all* $\tilde{\mathcal{D}}_w$, *for all* $w, w' \in \mathcal{W}$ *and for all* $d \in \tilde{\mathcal{D}}_w$, $d' \in \tilde{\mathcal{D}}_{w'}$

$$p_{w,d}(\mathbf{y}|\mathbf{x}) = p_{w',d'}(\mathbf{y}|\mathbf{x}) \tag{3.10}$$

which in turn implies that for any $(w, d) \in \mathcal{T}$ *we have that the generated counterfactual for outcome* (w, d) *for any sample (that was not assigned* (w, d)*) has the same (marginal) distribution (conditional on the features) as the true marginal distribution for that outcome.*

Proof. The proof and intermediate results can be found in Appendix A.2. □

3.5 Architecture

In this section, we describe in detail the novel architectures that we adopt to model each of the functions $\mathbf{G}, \mathbf{D}, \mathbf{D}_{\mathcal{W}}, \mathbf{D}_{w_1}, ..., \mathbf{D}_{w_k}$ which draws from the ideas in [127]. The inference network, $\mathbf{I}$, has the same architecture as the generator, but does not receive w_f, d_f, y_f or $\mathbf{z}$ as inputs.

3.5.1 Generator Architecture

We adopt a multi-task deep learning model for $\mathbf{G}$ by defining a function $g : \mathcal{X} \times \mathcal{T} \times \mathcal{Y} \times \mathcal{Z} \to \mathcal{H}$ for some latent space $\mathcal{H}$ (typically $\mathbb{R}^l$ for some l) and then for each treatment $w \in \mathcal{W}$ we introduce a multitask "head", $g_w : \mathcal{H} \times \mathcal{D}_w \to \mathcal{Y}$ taking inputs from $\mathcal{H}$ *and* a dosage, d, to produce an outcome $\hat{y}(w, d) \in \mathcal{Y}$. Given observations, $(\mathbf{x}, t_f, y_f)$, a noise vector $\mathbf{z}$, and a target treatment-dosage pair, $t = (w, d)$, we define

$$\mathbf{G}(\mathbf{x}, t_f, y_f, \mathbf{z})(t) = g_w(g(\mathbf{x}, t_f, y_f, \mathbf{z}), d)\,. \tag{3.11}$$

Each of $g, g_{w_1}, ..., g_{w_k}$ are fully connected networks. A figure of our generator architecture is given in Figure 3.3(a).

3.5.2 Hierarchical Discriminator Architectures

In order to ensure the discriminators act as functions of sets, we use ideas from [127] to create permutation invariant and permutation equivariant[4] networks. Zaheer et al. [127] provide several possible building blocks to use to construct invariant and equivariant deep networks. The basic building block we will use for invariant functions will be a layer of the form:

$$f_{inv}(\mathbf{u}) = \sigma(\mathbf{1}_b \mathbf{1}_m^T(\phi(u_1), ..., \phi(u_m)))\,, \tag{3.12}$$

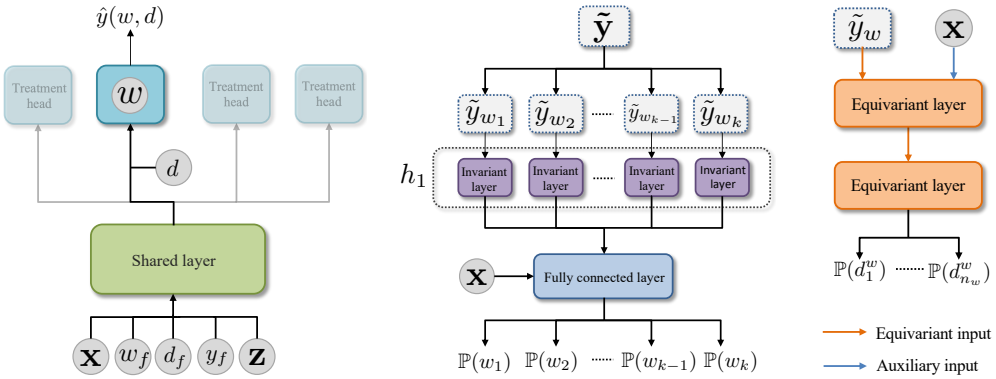
where $\mathbf{1}_l$ is a vector of 1s of dimension l, ϕ is any function $\phi : \mathcal{U} \to \mathbb{R}^q$ for some q (in this paper we use a standard fully connected layer) and σ is some non-linearity. The basic building block for equivariant functions is defined in terms of equivariance input, $\mathbf{u}$, and auxiliary input, $\mathbf{v}$, by:

$$f_{equi}(\mathbf{u}, \mathbf{v}) = \sigma(\lambda \mathbf{I}_m \mathbf{u} + \gamma(\mathbf{1}_m \mathbf{1}_m^T)\mathbf{u} + (\mathbf{1}_m \Theta^T)\mathbf{v})\,, \tag{3.13}$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix, λ and γ are scalar parameters and Θ is a vector of weights.

In the case of the hierarchical discriminator, we want the treatment discriminator, $\mathbf{D}_{\mathcal{W}}$, to be permutation invariant with respect to $\tilde{\mathbf{y}}_w$ for each treatment. To achieve

[4] Definitions can be found in [127] and in Appendix A.5.



(a) Generator **(b)** Treatment Discriminator **(c)** Dosage Discriminator

Figure 3.3: Architecture of our generator and discriminators.

this we define $h_1 : \prod_{w \in \mathcal{W}} (\mathcal{D}_w \times \mathcal{Y})^{n_w} \to \mathcal{H}_H$ and require that h_1 be permutation invariant w.r.t. each of the spaces $(\mathcal{D}_w \times \mathcal{Y})^{n_w}$. We concatenate the output of h_1 with the features $\mathbf{x}$ and pass these through a fully connected network $h_2 : \mathcal{X} \times \mathcal{H}_H \to [0,1]^k$ so that $\mathbf{D}_{\mathcal{W}}(\mathbf{x}, \tilde{\mathbf{y}}) = h_2(\mathbf{x}, h_1(\tilde{\mathbf{y}}))$.

To construct h_1, we concatenate the outputs of several invariant layers of the form given in Eq. (3.12) that each individually act on the spaces $(\mathcal{D}_w \times \mathcal{Y})^{n_w}$. That is, for each treatment, $w \in \mathcal{W}$ we define a map $h_{inv}^w : (\mathcal{D}_w \times \mathcal{Y})^{n_w} \to \mathcal{H}_H^w$ by substituting $\tilde{\mathbf{y}}_w$ for $\mathbf{u}$ in Eq. (3.12). We then define $\mathcal{H}_H = \prod_{w \in \mathcal{W}} \mathcal{H}_H^w$ and $h_1(\tilde{\mathbf{y}}) = (h_{inv}^{w_1}(\tilde{\mathbf{y}}_{w_1}), ..., h_{inv}^{w_k}(\tilde{\mathbf{y}}_{w_k}))$.

We want each dosage discriminator, $\mathbf{D}_w$, to be permutation equivariant with respect to $\tilde{\mathbf{y}}_w$. To achieve this each $\mathbf{D}_w$ will consist of two layers of the form given in Eq. (3.13) with the equivariance input, $\mathbf{u}$, to the first layer being $\tilde{\mathbf{y}}_w$ and to the second layer being the output of the first layer and the auxiliary input, $\mathbf{v}$, to the first layer being the features, $\mathbf{x}$, and then no auxiliary input to the second layer.

Diagrams depicting the architectures of the treatment discriminator and dosage discriminators can be found in Fig. 3.3(b) and Fig. 3.3(c) respectively.

3.6 Evaluation

The nature of the treatment-effects estimation problem in even the binary treatments setting does not allow for meaningful evaluation on real-world datasets due to the inability to observe the counterfactuals. While there are well-established benchmark synthetic

models for use in the binary (or multiple) case, no such models exist for the dosage setting. We propose our own semi-synthetic data simulation to evaluate our model against several benchmarks.

3.6.1 Experimental Setup

Semi-synthetic data generation: We simulate data as follows. We obtain features, $\mathbf{x}$, from a real dataset (in this paper we use TCGA [129], News [29, 36]) and MIMIC III [130])[5]. We consider 3 treatments each accompanied by a dosage. Each treatment, w, is associated with a set of parameters, $\mathbf{v}_1^w, \mathbf{v}_2^w, \mathbf{v}_3^w$. For each run of the experiment, these parameters are sampled randomly by sampling a vector, $\mathbf{u}_i^w$, from $\mathcal{N}(\mathbf{0}, \mathbf{1})$ and then setting $\mathbf{v}_i^w = \mathbf{u}_i^w / ||\mathbf{u}_i^w||$ where $||\cdot||$ is Euclidean norm. The shape of the response curve for each treatment, $f_w(\mathbf{x}, d)$, is given in Table 3.1, along with a closed-form expression for the optimal dosage. We add $\epsilon \sim \mathcal{N}(0, 0.2)$ noise to the outcomes.

We assign interventions by sampling a dosage, d_w, for each treatment from a beta distribution, $d_w|\mathbf{x} \sim \text{Beta}(\alpha, \beta_w)$. $\alpha \geq 1$ controls the dosage selection bias ($\alpha = 1$ gives the uniform distribution - see Appendix A.9). $\beta_w = \frac{\alpha - 1}{d_w^*} + 2 - \alpha$, where d_w^* is the optimal dosage[6] for treatment w. This setting of β_w ensures that the mode of $\text{Beta}(\alpha, \beta_w)$ is d_w^*. We then assign a treatment according to $w_f|\mathbf{x} \sim \text{Categorical}(\text{softmax}(\kappa f(\mathbf{x}, d_w))$ where increasing κ increases selection bias, and $\kappa = 0$ leads to random assignments. The factual intervention is given by (w_f, d_{w_f}). Unless otherwise specified, we set $\kappa = 2$ and $\alpha = 2$.

Treatment	Dose-Response	Optimal dosage
1	$f_1(\mathbf{x}, d) = C((\mathbf{v}_1^1)^T\mathbf{x} + 12(\mathbf{v}_2^1)^T\mathbf{x}d - 12(\mathbf{v}_3^1)^T\mathbf{x}d^2)$	$d_1^* = \frac{(\mathbf{v}_2^1)^T\mathbf{x}}{2(\mathbf{v}_3^1)^T\mathbf{x}}$
2	$f_2(\mathbf{x}, d) = C((\mathbf{v}_1^2)^T\mathbf{x} + \sin(\pi(\frac{\mathbf{v}_2^{2T}\mathbf{x}}{\mathbf{v}_3^{2T}\mathbf{x}})d))$	$d_2^* = \frac{(\mathbf{v}_3^2)^T\mathbf{x}}{2(\mathbf{v}_2^2)^T\mathbf{x}}$
3	$f_3(\mathbf{x}, d) = C((\mathbf{v}_1^3)^T\mathbf{x} + 12d(d-b)^2$, where $b = 0.75\frac{(\mathbf{v}_2^3)^T\mathbf{x}}{(\mathbf{v}_3^3)^T\mathbf{x}})$	$\frac{b}{3}$ if $b \geq 0.75$ 1 if $b < 0.75$

Table 3.1: Dose response curves used to generate semi-synthetic outcomes for patient features $\mathbf{x}$. In the experiments, we set $C = 10$. $\mathbf{v}_1^w, \mathbf{v}_2^w, \mathbf{v}_3^w$ are parameters associated with each treatment w.

Benchmarks: We compare against Generalized Propensity Score (GPS) [109] and Dose Reponse Networks (DRNet) [36] (the standard model and with Wasserstein

[5]Details of each dataset can be found in Appendix A.8

[6]For symmetry, if $d_w^* = 0$, we sample d_w from $1 - \text{Beta}(\alpha, \beta_w)$ where β_w is set as though $d_w^* = 1$.

regularization (DRN-W)). As a baseline, we compare against a standard multilayer perceptron (MLP) that takes patient features, treatment and dosage as input and estimates the patient outcome and a multitask variant (MLP-M) that has a designated head for each treatment. See Appendix A.10 for details of the benchmark models and their hyperparameter optimisation. For metrics, we use Mean Integrated Squared Error (MISE), Dosage Policy Error (DPE) and Policy Error (PE) [36, 131]. For further details, see Appendix A.11.

3.6.2 Source of Gain

Before comparing against the benchmarks, we investigate how each component of our model affects performance. We start with a baseline model in which both the generator and discriminator consist of a single fully connected network. One at a time, we add in the following components (cumulatively until we reach our full model): (1) the supervised loss in Eq. 3.3 (+ $\mathcal{L}_S$), (2) multitask heads in the generator (+ Multitask), (3) hierarchical discriminator (+ Hierarchical) and (4) invariance/equivariance layers in the treatment and dosage discriminators (+Inv/Eqv). We report the results in Table 3.2 for TCGA for all 3 error metrics (MISE, DPE and PE), computed over 30 runs.

	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$
Baseline	4.18 ± 0.32	2.06 ± 0.16	1.93 ± 0.12
+ $\mathcal{L}_S$	3.37 ± 0.11	1.14 ± 0.05	0.84 ± 0.05
+ Multitask	3.15 ± 0.12	0.85 ± 0.05	0.67 ± 0.05
+ Hierchical	2.54 ± 0.05	0.36 ± 0.05	0.45 ± 0.05
+ Inv/Eqv	1.89 ± 0.05	0.31 ± 0.05	0.25 ± 0.05

Table 3.2: Source of gain analysis for our model on the TCGA. Metrics are reported as Mean $\pm$ Std.

The addition of each component results in improved performance, with the final row (our full model) demonstrating the best performance across both datasets and for all metrics. In Appendix A.12 we further compare our hierarchical discriminator with a single network discriminator by investigating both models sensitivity to the hyperparameter n_w. Details of the single discriminator can be found in Appendix A.6. Architectures for other components of the ablation studies can be found in Appendix A.7.

3.6.3 Benchmarks Comparison

We now compare SCIGAN[7] against the benchmarks on our 3 semi-synthetic datasets. For MIMIC, due to the low number of samples available, we use two treatments - 2 and 3. We report $\sqrt{\text{MISE}}$ and $\sqrt{\text{PE}}$ in Table 3.3, $\sqrt{\text{DPE}}$ is given in Appendix A.12. We see that SCIGAN demonstrates a statistically significant improvement over every benchmark across all 3 datasets. In Appendix A.12 we compare SCIGAN with DRNET and GPS for an increasing number of treatments.

Method	TCGA		News		MIMIC	
	$\sqrt{\text{MISE}}$	$\sqrt{\text{PE}}$	$\sqrt{\text{MISE}}$	$\sqrt{\text{PE}}$	$\sqrt{\text{MISE}}$	$\sqrt{\text{PE}}$
SCIGAN	$\mathbf{1.89 \pm 0.05}$	$\mathbf{0.25 \pm 0.05}$	$\mathbf{3.71 \pm 0.05}$	$\mathbf{3.90 \pm 0.05}$	$\mathbf{2.09 \pm 0.12}$	$\mathbf{0.32 \pm 0.05}$
DRNet	3.64 ± 0.12	0.67 ± 0.05	4.98 ± 0.12	4.17 ± 0.11	4.45 ± 0.12	1.44 ± 0.05
DRN-W	3.71 ± 0.12	0.63 ± 0.05	5.07 ± 0.12	4.56 ± 0.12	4.47 ± 0.12	1.37 ± 0.05
GPS	4.83 ± 0.01	1.60 ± 0.01	6.97 ± 0.01	24.1 ± 0.05	7.39 ± 0.00	20.2 ± 0.01
MLP-M	3.96 ± 0.12	1.20 ± 0.05	5.17 ± 0.12	5.82 ± 0.16	4.97 ± 0.16	1.59 ± 0.05
MLP	4.31 ± 0.05	0.97 ± 0.05	5.48 ± 0.16	6.45 ± 0.21	5.34 ± 0.16	1.65 ± 0.05

Table 3.3: Performance of individualized treatment-dose response estimation. Bold indicates the method with the best performance for each dataset. Metrics are reported as Mean $\pm$ Std.

3.6.4 Discrete Dosages

In this experiment, we investigate the discrete dosage setting. Details of the experimental setup can be found in Appendix A.12. We report Mean Squared Error (MSE) of SCIGAN and GANITE in Fig. 3.4 where we vary the number of discrete dosages from 3 to 30. We see that GANITE is incapable of handling more than 7 discrete dosages, whereas the hierarchical discriminator together with the invariant and equivariant layers allow SCIGAN to maintain performance as the number of dosages increases. Importantly, this demonstrates SCIGAN's wide-ranging applicability in both discrete and continuous settings.

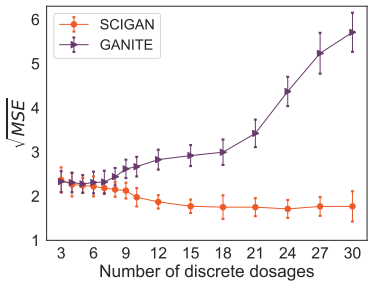
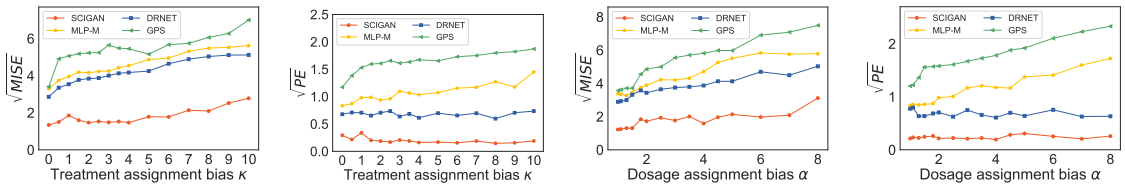


Figure 3.4: Comparison between SCIGAN and GANITE.

[7]The implementation of SCIGAN can be found at https://github.com/vanderschaarlab/mlforhealthlabpub and at https://github.com/ioanabica/SCIGAN.

3.6.5 Treatment and Dosage Selection Bias

Finally, we assess each model's robustness to treatment and dosage bias. We report $\sqrt{\text{MISE}}$ and $\sqrt{\text{PE}}$ on TCGA here. For the other metrics see Appendix A.12. Fig. 3.5(a) shows the performance of the 4 methods for κ between 0 (no bias) and 10 (strong bias). Fig. 3.5(b) shows the performance for α between 1 (no bias) and 8 (strong bias). SCIGAN shows consistent performance, significantly outperforming the benchmarks for all κ and α.



(a) Treatment bias **(b)** Treatment bias **(c)** Dosage bias **(d)** Dosage bias

Figure 3.5: Performance of the 4 methods on datasets with varying bias levels.

3.7 Conclusion

In this chapter, we proposed a novel framework for estimating response curves for continuous interventions from observational data. We provided theoretical justification for our use of a modified GAN framework, which introduced a novel hierarchical discriminator. While our approach is very flexible (it accepts multiple treatments potentially with/without dosage parameters), one limitation is that SCIGAN needs at least a few thousand training samples, as is generally the case with neural networks, and GANs in particular. As for future research, we used out-of-the-box methods for the invariant and equivariant layers but feel that more can be done to make these layers as expressive as possible for this model. Moreover, another important research direction is to extend this work to the temporal causal inference setting [14, 55, 56, 117] and estimate counterfactual outcomes for sequences of continuous interventions.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Estimating the Effects of Continuous-valued Interventions using Generative Adversarial Networks
Publication Status	☑ Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Ioana Bica*, James Jordon*, and Mihaela van der Schaar. "Estimating the Effects of Continuous-valued Interventions using Generative Adversarial Networks". In *Advances in Neural Information Processing Systems (NeurIPS),* 2020.

Student Confirmation

Student Name:	Ioana Bica		
Contribution to the Paper	- The main idea of the paper that involved the design of the generative adversarial framework for continuous interventions and in particular of the hierarchical discriminator was the product of a brainstorming discussion between James and myself. - James determined how to use DeepSets to realize the discriminators as appropriate functions. - I wrote all of the code and produced all of the results for the paper. - James conducted all of the theoretical analysis in the paper. - The paper was written jointly by James and me, where I took more responsibility for the experimental section and James for the theoretical section. - Both James and I performed the experimental design and developed the semi-synthetic benchmark used for all experiments. - All authors contributed to the development of the paper through multiple discussions. - All authors reviewed the final draft.		
Signature 		Date	30/03/2022

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Mihaela van der Schaar		
Supervisor comments I agree that the candidate has made a substantial contribution to the publication and that the above description is accurate.		
Signature 	Date	30/03/2022

This completed form should be included in the thesis, at the end of the relevant chapter.

4

Estimating the Causal Effects of Treatments in the Temporal Setting

Contents

4.1 Counterfactual Recurrent Network 58
4.1.1 Introduction 58
4.1.2 Related Work 62
4.1.3 Problem Formulation 63
4.1.4 Counterfactual Recurrent Network 64
4.1.5 Adversarially Balanced Representation over Time 66
4.1.6 Experiments 68
4.1.7 Conclusion 73
4.2 Time Series Deconfounder 75
4.2.1 Introduction 75
4.2.2 Related Work 77
4.2.3 Problem Formulation 79
4.2.4 Time Series Deconfounder 81
4.2.5 Factor Model over Time in Practice 85
4.2.6 Experiments on Synthetic Data 86
4.2.7 Experiments on MIMIC III 90
4.2.8 Conclusion 92

4.1 Estimating Counterfactual Treatment Outcomes over Time through Adversarially Balanced Representations

Publication Included

- **Ioana Bica**, Ahmed M. Alaa, James Jordon, and Mihaela van der Schaar. "Estimating counterfactual treatment outcomes over time through adversarially balanced representations". In *International Conference on Learning Representations (ICLR)*, 2020. Link: https://openreview.net/pdf?id=BJg866NFvB. [Accepted, , Selected as spotlight (top 6% of submissions)]

Abstract

Identifying when to give treatments to patients and how to select among multiple treatments over time are important medical problems with a few existing solutions. In this chapter, we introduce the Counterfactual Recurrent Network (CRN), a novel sequence-to-sequence model that leverages the increasingly available patient observational data to estimate treatment effects over time and answer such medical questions. To handle the bias from time-varying confounders, covariates affecting the treatment assignment policy in the observational data, CRN uses domain adversarial training to build balancing representations of the patient history. At each timestep, CRN constructs a treatment invariant representation which removes the association between patient history and treatment assignments and thus can be reliably used for making counterfactual predictions. On a simulated model of tumour growth, with varying degree of time-dependent confounding, we show how our model achieves lower error in estimating counterfactuals and in choosing the correct treatment and timing of treatment than current state-of-the-art methods.

4.1.1 Introduction

As clinical decision-makers are often faced with the problem of choosing between treatment alternatives for patients, reliably estimating their effects is paramount. While clinical trials represent the gold standard for causal inference, they are expensive, have

a few patients and narrow inclusion criteria [9]. Leveraging the increasingly available observational data about patients, such as electronic health records, represents a more viable alternative for estimating treatment effects.

A large number of methods have been proposed for performing causal inference using observational data in the static setting [26, 28, 29, 31, 35, 102, 132] and only a few methods address the longitudinal setting [40, 114–117]. However, estimating the effects of treatments over time poses unique opportunities such as understanding how diseases evolve under different treatment plans, how individual patients respond to medication over time, but also which are optimal timings for assigning treatments, thus providing new tools to improve clinical decision support systems.

The biggest challenge when estimating the effects of time-dependent treatments from observational data involves correctly handling the time-dependent confounders: patient covariates that are affected by past treatments which then influence future treatments and outcomes [133]. For instance, consider that treatment A is given when a certain patient covariate (e.g. white blood cell count) has been outside of normal range values for several consecutive timesteps. Suppose also that this patient covariate was itself affected by the past administration of treatment B. If these patients are more likely to die, without adjusting for the time-dependent confounding (e.g. the changes in the white blood cell count over time), we will incorrectly conclude that treatment A is harmful to patients. Moreover, estimating the effect of a different sequence of treatments on the patient outcome would require not only adjusting for the bias at the current step (in treatment A), but also for the bias introduced by the previous application of treatment B.

Existing methods for causal inference in the static setting cannot be applied in this longitudinal setting since they are designed to handle the cross-sectional set-up, where the treatment and outcome depend only on a static value of the patient covariates. If we consider again the above example, these methods would not be able to model how the changes in patient covariates over time affect the assignment of treatments and they would also not be able to estimate the effect of a sequence of treatments on the patient outcome (e.g. sequential application of treatment A followed by treatment B). Different models

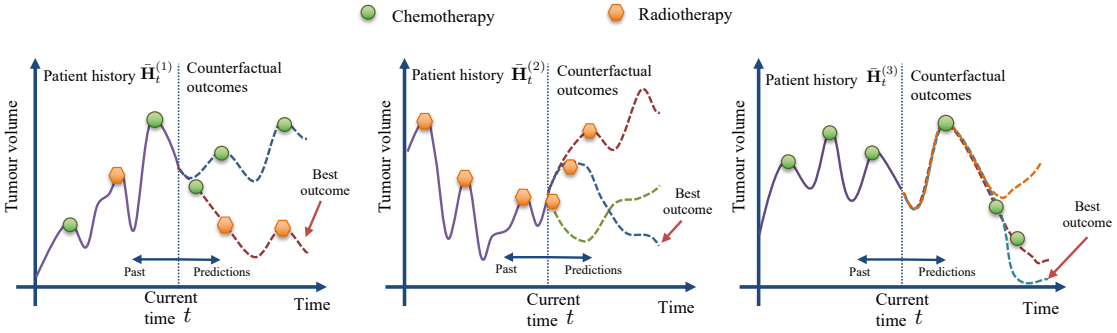
that can handle these temporal dependencies in the observational data and varying-length patient histories are needed for estimating treatment effects over time.

Time-dependent confounders are present in observational data because doctors follow policies: the history of the patients' covariates and the patients' response to past treatments are used to decide future treatments [134]. The direct use of supervised learning methods will be biased by the treatment policies present in the observational data and will not be able to correctly estimate counterfactuals for different treatment assignment policies.

Standard methods for adjusting for time-varying confounding and estimating the effects of time-varying exposures are based on ideas from epidemiology. The most widely used among these are Marginal Structural Models (MSMs) [14, 134] which use the inverse probability of treatment weighting (IPTW) to adjust for the time-dependent confounding bias. Through IPTW, MSMs create a pseudo-population where the probability of treatment does not depend on the time-varying confounders. However, MSMs are not robust to model misspecification in computing the IPTWs. MSMs can also give high-variance estimates due to extreme weights; computing the IPTW involves dividing by probability of assigning a treatment conditional on patient history which can be numerically unstable if the probability is small.

We introduce the Counterfactual Recurrent Network (CRN), a novel sequence-to-sequence architecture for estimating treatment effects over time. CRN leverages recent advances in representation learning [50] and domain adversarial training [51] to overcome the problems of existing methods for causal inference over time. Our main contributions are as follows.

Treatment invariant representations over time. CRN constructs treatment invariant representations at each timestep in order to break the association between patient history and treatment assignment and thus removes the bias from time-dependent confounders. For this, CRN uses domain adversarial training [51, 135, 136] to trade-off between building this balancing representation and predicting patient outcomes. We show that these representations remove the bias from time-varying confounders and can be reliably used for estimating counterfactual outcomes. This represents the first work that introduces ideas from domain adaptation to the area of estimating treatment effects over time. In



(a) Decide treatment plan (b) Decide optimal time of treatment (c) Decide when to stop treatment

Figure 4.1: Applicability of CRN in cancer treatment planning. We illustrate 3 patients with different covariate and treatment histories $\bar{\mathbf{H}}_t$. For a current time t, CRN can predict counterfactual trajectories (the coloured dashed branches) for planned treatments in the future. Through the counterfactual predictions, we can decide which treatment plan results in the best patient outcome (in this case, the lowest tumour volume). This way, CRN can be used to perform all of the following: choose optimal treatments (a), find timing when treatment is most effective (b) decide when to stop treatment (c).

addition, by building balancing representations, we propose a novel way of removing the bias introduced by time-varying confounders.

Counterfactual estimation of future outcomes. To estimate counterfactual outcomes for treatment plans (and not just single treatments), we integrate the domain adversarial training procedure as part of a sequence-to-sequence architecture. CRN consists of an encoder network which builds treatment invariant representations of the patient history that are used to initialize the decoder. The decoder network estimates outcomes under an intended sequence of future treatments, while also updating the balanced representation. By performing counterfactual estimation of future treatment outcomes, CRN can be used to answer critical medical questions such as deciding when to give treatments to patients, when to start and stop treatment regimes, and also how to select from multiple treatments over time. We illustrate in Figure 4.1 the applicability of our method in choosing optimal cancer treatments.

In our experiments, we evaluate CRN in a realistic set-up using a model of tumour growth [118]. We show that CRN achieves better performance in predicting counterfactual outcomes, but also in choosing the right treatment and timing of treatment than current state-of-the-art methods.

4.1.2 Related Work

We focus on methods for estimating treatment effects over time and for building balancing representations for causal inference. A more in-depth review of related work is in Appendix B.1.1.

Treatment effects over time. Standard methods for estimating the effects of time-varying exposures were first developed in the epidemiology literature and include the g-computation formula, Structural Nested Models and Marginal Structural Models (MSMs) [14, 81, 112, 113]. Originally, these methods have used predictors performing logistic/linear regression which makes them unsuitable for handling complex time-dependencies [134, 137, 138]. To address these limitations, methods that use Bayesian non-parametrics or recurrent neural networks as part of these frameworks have been proposed. [114, 115, 117].

To begin with, [114] use Gaussian processes to model discrete patient outcomes as a generalized mixed-effects model and uses the g-computation method to handle time-varying confounders. [116] extend the approach in [114] to the continuous time-setting and model treatment responses using linear time-invariant dynamical systems. [115] use Dirichlet and Gaussian processes to model the observational data and estimate the IPTW in Marginal Structural Models. [40] build upon work from [139, 140] and use marked point processes and Gaussian processes to learn causal effects in continuous-time data. These Bayesian non-parametric methods make strong assumptions about model structure and consequently cannot handle well heterogeneous treatment effects arising from baseline variables [40, 116] and multiple treatment outcomes [40, 114].

The work most related to ours is the one of [117] which improves on the standard MSMs by using recurrent neural networks (RNNs) to estimate the inverse probability of treatment weights (IPTWs). [117] introduces Recurrent Marginal Structural Networks (RMSNs) which also use a sequence-to-sequence deep learning architecture to forecast treatment responses in a similar fashion to our model. However, RMSNs require training additional RNNs to estimate the propensity weights and does not overcome the fundamental problems with IPTWs, such as the high-variance of the weights. Conversely, CRN

takes advantage of the recent advances in machine learning, in particular, representation learning to propose a novel way of handling time-varying confounders.

Balancing representations for treatment effect estimation. Balancing the distribution of control and treated groups has been used for counterfactual estimation in the static setting. The methods proposed in the static setting for balancing representations are based on using discrepancy measures in the representation space between treated and untreated patients, which do not generalize to multiple treatments [28, 29, 102, 132]. Moreover, due to the sequential assignment of treatments in the longitudinal setting, and due to the change of patient covariates over time according to previous treatments, the methods for the static setting are not directly applicable to the time-varying setting [134, 141].

4.1.3 Problem Formulation

Consider an observational dataset $\mathcal{D} = \left\{\{\mathbf{x}_t^{(i)}, \mathbf{a}_t^{(i)}, \mathbf{y}_{t+1}^{(i)}\}_{t=1}^{T^{(i)}} \cup \{\mathbf{v}^{(i)}\}\right\}_{i=1}^{N}$ consisting of information about N independent patients. For each patient (i), we observe time-dependent covariates $\mathbf{X}_t^{(i)} \in \mathcal{X}_t$, treatment received $\mathbf{A}_t^{(i)} \in \{A_1, \dots A_K\} = \mathcal{A}$ and outcomes $\mathbf{Y}_{t+1}^{(i)} \in \mathcal{Y}_{t+1}$ for $T^{(i)}$ discrete timesteps. The patient can also have baseline covariates $\mathbf{V}^{(i)} \in \mathcal{V}$ such as gender and genetic information. Note that the outcome $\mathbf{Y}_{t+1}^{(i)}$ will be part of the observed covariates $\mathbf{X}_{t+1}^{(i)}$. For simplicity, the patient superscript (i) will be omitted unless explicitly needed.

We adopt the potential outcomes framework proposed by [79, 80] and extended by [81] to account for time-varying treatments. Let $\mathbf{Y}[\bar{\mathbf{a}}]$ be the potential outcomes, either factual or counterfactual, for each possible course of treatment $\bar{\mathbf{a}}$. Let $\bar{\mathbf{H}}_t = (\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}, \mathbf{V})$ represent the history of the patient covariates $\bar{\mathbf{X}}_t = (\mathbf{X}_1, \dots, \mathbf{X}_t)$, treatment assignments $\bar{\mathbf{A}}_t = (\mathbf{A}_1, \dots, \mathbf{A}_t)$ and static features $\mathbf{V}$. We want to estimate:

$$\mathbb{E}(\mathbf{Y}_{t+\tau}[\bar{\mathbf{a}}(t, t+\tau-1)]|\bar{\mathbf{H}}_t), \tag{4.1}$$

where $\bar{\mathbf{a}}(t, t+\tau-1) = [\mathbf{a}_t, \dots \mathbf{a}_{t+\tau-1}]$ represents a possible sequence of treatments from timestep t just until before the potential outcome $\mathbf{Y}_{t+\tau}$ is observed. We make the standard assumptions [14, 117] needed to identify the treatment effects: consistency, positivity and no hidden confounders (sequential strong ignorability). See Appendix B.1.2 for more more details.

4.1.4 Counterfactual Recurrent Network

The observational data can be used to train a supervised learning model to forecast: $\mathbb{E}(\mathbf{Y}_{t+\tau} \mid \bar{\mathbf{A}}(t, t+\tau-1) = \bar{\mathbf{a}}(t, t+\tau-1), \bar{\mathbf{H}}_t)$. However, without adjusting for the bias introduced by time-varying confounders, this model cannot be reliably used for making causal predictions [14, 40, 81]. The Counterfactual Recurrent Network (CRN) removes this bias through domain adversarial training and estimates the counterfactual outcomes $\mathbb{E}(\mathbf{Y}_{t+\tau}[\bar{\mathbf{a}}(t, t+\tau-1)]|\bar{\mathbf{H}}_t)$, for any intended future treatment assignment $\bar{\mathbf{a}}(t, t+\tau-1)$.

Balancing representations. The history $\bar{\mathbf{H}}_t = (\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}, \mathbf{V})$ of the patient contains the time-varying confounders $\bar{\mathbf{X}}_t$ which bias the treatment assignment $\mathbf{A}_t \in \{A_1, \dots A_K\}$ in the observational dataset. Inverse probability of treatment weighting, as performed by MSMs, creates a pseudo-population where the probability of treatment $\mathbf{A}_t$ does not depend on the time-varying confounders [14]. In this chapter, we propose instead building a representation of the history $\bar{\mathbf{H}}_t$ that is not predictive of the treatment $\mathbf{A}_t$. This way, we remove the association between history, containing the time-varying confounders $\bar{\mathbf{X}}_t$, and current treatment $\mathbf{A}_t$. [142] shows that in this case, the estimation of counterfactual treatment outcomes is unbiased. See Appendix B.1.3 for details and for an example of a causal graph with time-dependent confounders.

Let Φ be the representation function that maps the patient history $\bar{\mathbf{H}}_t$ to a representation space $\mathcal{R}$. To obtain unbiased treatment effects, Φ needs to construct treatment invariant representations such that $P(\Phi(\bar{\mathbf{H}}_t) \mid \mathbf{A}_t = A_1) = \cdots = P(\Phi(\bar{\mathbf{H}}_t) \mid \mathbf{A}_t = A_K)$. To achieve this and to estimate counterfactual outcomes under a planned sequence of treatments, we integrate the domain adversarial training framework proposed by [51] and extended by [136] to the multi-domain learning setting, into a sequence-to-sequence architecture. In our case, the different treatments at each timestep are considered the different domains. Note that the novelty here comes from the use of domain adversarial training to handle the bias from the time-dependent confounders, rather than the use of sequence-to-sequence models, which have already been applied to forecast treatment responses [117]. Figure 4.2 illustrates our model architecture.

Encoder. The encoder network uses an RNN, with LSTM unit [143], to process the history of treatments $\bar{\mathbf{A}}_{t-1}$, covariates $\bar{\mathbf{X}}_t$ and baseline features $\mathbf{V}$ to build a treatment

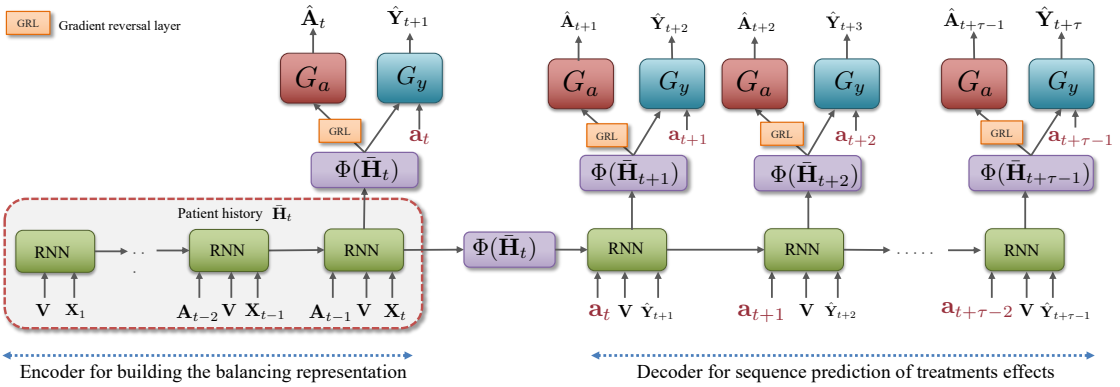


Figure 4.2: CRN architecture. Encoder builds representation $\Phi(\bar{\mathbf{H}}_t)$ that maximizes loss of treatment classifier G_a and minimizes loss of outcome predictor G_y. $\Phi(\bar{\mathbf{H}}_t)$ is used to initialize the decoder, which continues to update it to predict counterfactual outcomes of a sequence of future treatments.

invariant representation $\Phi(\bar{\mathbf{H}}_t)$, but also to predict one-step-ahead outcomes $\mathbf{Y}_{t+1}$. To achieve this, the encoder network aims to maximize the loss of the treatment classifier G_a and minimize the loss of the outcome predictor network G_y. This way, the balanced representation $\Phi(\bar{\mathbf{H}}_t)$ is not predictive of the assigned treatment $\mathbf{A}_t$, but is discriminative enough to estimate the outcome $\mathbf{Y}_{t+1}$. To train this model using gradient descent, we use the Gradient Reversal Layer [51].

Decoder. The decoder network uses the balanced representation computed by the encoder to initialize the state of an RNN that predicts the counterfactual outcomes for a sequence of future treatments. During training, the decoder uses as input the outcomes from the observational data $(\mathbf{Y}_{t+1}, \dots \mathbf{Y}_{t+\tau-1})$, the static patient features $\mathbf{V}$ and the intended sequence of treatments $\bar{\mathbf{a}}(t, t+\tau-1)$. The decoder is trained in a similar way to the encoder to update the balanced representation and to estimate the outcomes. During testing, we do not have access to ground-truth outcomes; thus, the outcomes predicted by the decoder $(\hat{\mathbf{Y}}_{t+1}, \dots \hat{\mathbf{Y}}_{t+\tau-1})$ are auto-regressively used instead as inputs. By running the decoder with different treatment settings, and by auto-regressively feeding back the outcomes, we can determine when to start and end different treatments, which is the optimal time to give the treatment and which treatments to give over time to obtain the best patient outcomes.

The representation $\Phi(\bar{\mathbf{H}}_t)$ is built by applying a fully connected layer, with Exponential Linear Unit (ELU) activation to the output of the LSTM. The treatment classifier G_a

and the predictor network G_y consist of a hidden layer each, also with ELU activation. The output layer of G_a uses softmax activation, while the output layer of G_y uses linear activation for continuous predictions. For categorical outcomes, softmax activation can be used. We follow an approach similar to [117] and we split the encoder and decoder training into separate steps. See Appendix B.1.5 for details.

The encoder and decoder networks use variational dropout [144] such that the CRN can also give uncertainty intervals for the treatment outcomes. This is particularity important in the estimation of treatment effects, since the model predictions should only be used when they have high confidence. Our model can also be modified to allow for irregular samplings of observations by using a PhasedLSTM [145].

4.1.5 Adversarially Balanced Representation over Time

At each timestep t, let the K different possible treatments $\mathbf{A}_t \in \{A_1, \dots A_K\}$ represent our domains. As described in Section 4.1.4, to remove the bias from time-dependent confounders, we build a representation of history $\bar{\mathbf{H}}_t$ that is invariant across treatments: $P(\Phi(\bar{\mathbf{H}}_t) \mid A_1) = \cdots = P(\Phi(\bar{\mathbf{H}}_t) \mid A_K)$.

This requirement can be enforced by minimizing the distance in the distribution of $\Phi(\bar{\mathbf{H}}_t)$ between any two pairs of treatments. [146, 147], propose measuring the disparity between distributions based on their separability by a discriminatively-trained classifier. Let the symmetric hypothesis class $\mathcal{H}$ consist of the set of symmetric multiclass classifiers, such as neural network architectures. The $\mathcal{H}$-divergence between all pairs of two distributions is defined in terms of the capacity of the hypothesis class $\mathcal{H}$ to discriminate between examples from the multiple distributions. Empirically, minimizing the $\mathcal{H}-$divergence involves building a representation where examples from the multiple domains are as indistinguishable as possible [135, 136, 147]. [51] use this idea to propose an adversarial framework for domain adaptation involving building a representation which achieves maximum error on a domain classifier and minimum error on an outcome predictor. Similarly, in our case, we use domain adversarial training to build a representation of the patient history $\Phi(\bar{\mathbf{H}}_t)$ that is both invariant to the treatment given at timestep t, $\mathbf{A}_t$ and that achieves low error in estimating the outcome $\mathbf{Y}_{t+1}$.

Let $G_a(\Phi(\bar{\mathbf{H}}_t);\theta_a)$ be the treatment classifier with parameters θ_a and let $G_a^j(\Phi(\bar{\mathbf{H}}_t);\theta_a)$ be the output corresponding to treatment A_j. Let $G_y(\Phi(\bar{\mathbf{H}}_t);\theta_y)$ be the predictor network with parameters θ_y. The representation function Φ is parameterized by the parameters θ_r in the RNN: $\Phi(\bar{\mathbf{H}}_t;\theta_r)$. Figure 4.3 shows the adversarial training procedure used.

For timestep t and patient (i), let $\mathcal{L}_{t,a}^{(i)}(\theta_r,\theta_a)$ be the treatment (domain) loss and let $\mathcal{L}_{t,y}^{(i)}(\theta_r,\theta_y)$ the outcome loss, defined as follows:

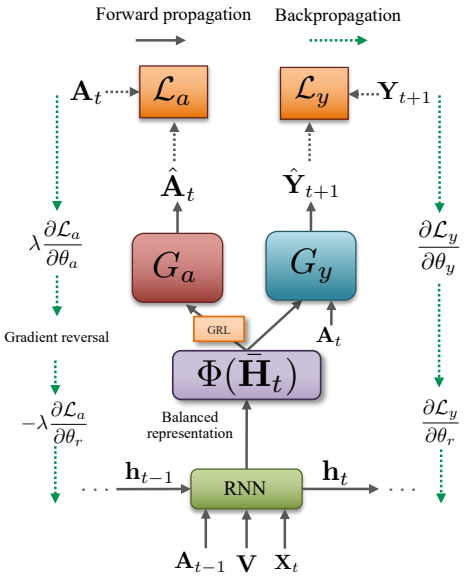


Figure 4.3: Training procedure for building balancing representation.

$$\mathcal{L}_{t,a}^{(i)}(\theta_r,\theta_a) = -\sum_{j=1}^{K} \mathbb{I}_{\{\mathbf{a}_t^{(i)}=a_j\}} \log(G_a^j(\Phi(\bar{\mathbf{H}}_t;\theta_r);\theta_a)) \tag{4.2}$$

$$\mathcal{L}_{t,y}^{(i)}(\theta_r,\theta_y) = \|\mathbf{Y}_{t+1}^{(i)} - (G_y(\Phi(\bar{\mathbf{H}}_t;\theta_r),\theta_y))\|^2. \tag{4.3}$$

If the outcome is binary, the cross-entropy loss can be used instead for $\mathcal{L}_{t,y}$. To build treatment invariant representations and to also estimate patient outcomes, we aim to maximize treatment loss and minimize outcome loss.

Thus, the overall loss $\mathcal{L}_{t,y}^{(i)}$ at timestep t is given by:

$$\mathcal{L}_t^{(i)}(\theta_r,\theta_y,\theta_a) = \sum_{i=1}^{N} \mathcal{L}_{t,y}^{(i)}(\theta_r,\theta_y) - \lambda\mathcal{L}_{t,a}^{(i)}(\theta_r,\theta_a), \tag{4.4}$$

where the hyperparameter λ controls this trade-off between domain discrimination and outcome prediction. We use the standard procedure for training domain adversarial networks from [51] and we start off with an initial value for λ and use an exponentially increasing schedule during training. To train the model using backpropagation, we use the Gradient Reversal Layer (GRL) [51]. For more details about the training procedure, see Appendix B.1.5.

By using the objective $\mathcal{L}_t^{(i)}(\theta_r,\theta_y,\theta_a)$, we reach the saddle point $(\hat{\theta}_r,\hat{\theta}_y,\hat{\theta}_a)$ that achieves the equilibrium between domain discrimination and outcome estimation.

$$(\hat{\theta}_r,\hat{\theta}_y) = \arg\min_{\theta_r,\theta_y} \mathcal{L}_t^{(i)}(\theta_r,\theta_y,\hat{\theta}_a) \qquad \hat{\theta}_a = \arg\max_{\theta_a} \mathcal{L}_t^{(i)}(\hat{\theta}_r,\hat{\theta}_y,\theta_a). \tag{4.5}$$

The result stated in Theorem 4.1.1 proves that the treatment (domain) loss part of our objective (from equation 4.2) aims to remove the time-dependent confounding bias.

Theorem 4.1.1. *Let $t \in \{1, 2, \dots\}$. For each $j = 1, ..., K$, let P_j denote the distribution of $\bar{\mathbf{H}}_t$ conditional on $\mathbf{A}_t = A_j$ and let P_j^{Φ} denote the distribution of $\Phi(\bar{\mathbf{H}}_t)$ conditional on $\mathbf{A}_t = A_j$. Let G_a^j denote the output of G_a corresponding to treatment A_j. Then the minimax game defined by*

$$\min_{\Phi} \max_{G_a} \sum_{j=1}^{K} \mathbb{E}_{\bar{\mathbf{H}}_t \sim P_j} \Big[\log(G_a^j(\Phi(\bar{\mathbf{H}}_t); \theta_a)) \Big] \quad \textit{subject to} \quad \sum_{j=1}^{K} G_a^j(\Phi(\bar{\mathbf{H}}_t)) = 1 \tag{4.6}$$

has a global minimum which is attained if and only if $P_1^{\Phi} = P_2^{\Phi} = ... = P_K^{\Phi}$, i.e. when the learned representations are invariant across all treatments.

Proof. This result is restatement of the one in [135]. For details, see Appendix B.1.4. □

A good representation allows us to obtain a low error in estimating counterfactuals for all treatments, while at the same time to minimize the $\mathcal{H}$-divergence between induced marginal distributions of all the domains. We use an algorithm that directly minimizes a combination of the $\mathcal{H}-$divergence and the empirical training margin.

4.1.6 Experiments

In real datasets, counterfactual outcomes and the degree of time-dependent confounding are not known [40, 117]. To validate the CRN[1], we evaluate it on a Pharmacokinetic-Pharmacodynamic model of tumour growth [118], which uses a state-of-the-art bio-mathematical model to simulate the combined effects of chemotherapy and radiotherapy in lung cancer patients. The same model was used by [117] to evaluate RMSNs.

Model of tumour growth The volume of tumour t days after diagnosis is modelled as follows:

$$V(t+1) = \Big(1 + \underbrace{\rho \log\Big(\frac{K}{V(t)}\Big)}_{\text{Tumor growth}} - \underbrace{\beta_c C(t)}_{\text{Chemotherapy}} - \underbrace{\big(\alpha_r d(t) + \beta_r d(t)^2\big)}_{\text{Radiotherapy}} + \underbrace{e_t}_{\text{Noise}}\Big) V(t) \tag{4.7}$$

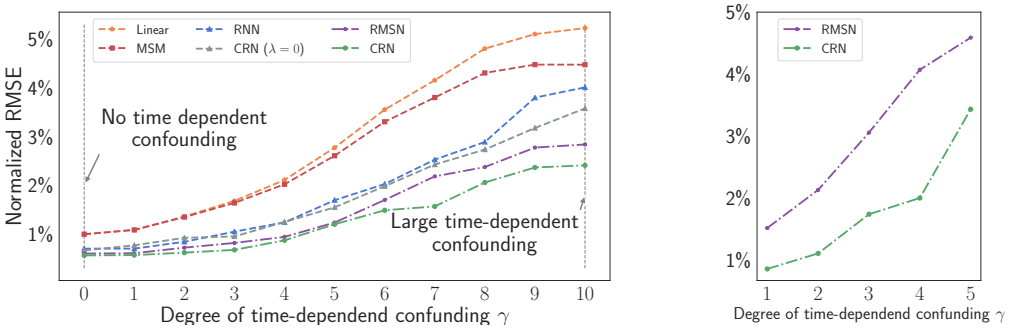
[1]The implementation of the model can be found at https://github.com/vanderschaarlab/mlforhealthlabpub and at https://github.com/ioanabica/Counterfactual-Recurrent-Network.

where $K, \rho, \beta_c, \alpha_r, \beta_r, e_t$ are sampled as described in [118]. To incorporate heterogeneity in patient responses, the prior means for β_c and α_r are adjusted to create patient subgroups, which are used as baseline features. The chemotherapy concentration $C(t)$ and radiotherapy dose $d(t)$ are modelled as described in Appendix B.1.6. Time-varying confounding is introduced by modelling chemotherapy and radiotherapy assignment as Bernoulli random variables, with probabilities p_c and p_r depending on the tumour diameter: $p_c(t) = \sigma\Big(\frac{\gamma_c}{D_{\max}}(\bar{D}(t) - \delta_c)\Big)$ and $p_r(t) = \sigma\Big(\frac{\gamma_r}{D_{\max}}(\bar{D}(t) - \delta_r)\Big)$ where $\bar{D}(t)$ is the average diameter over the last 15 days, $D_{\max} = 13$cm, $\sigma(\cdot)$ is the sigmoid and $\delta_c = \delta_r = D_{\max}/2$. The amount of time-dependent confounding is controlled through γ_c, γ_r; the higher $\gamma_\star$ is, the more important the history is in assigning treatments. At each timestep, there are four treatment options: no treatment, chemotherapy, radiotherapy, combined chemotherapy and radiotherapy. For details about data simulation, see Appendix B.1.6.

Benchmarks: The following benchmarks are used for performance comparison. Marginal Structural Models (MSMs) [14], which use logistic regression for estimating the IPTWs and linear regression for prediction (see Appendix B.1.7 for details). We also compare against the Recurrent Marginal Structural Networks (RMSNs) [117], which is the current state-of-the-art model in estimating treatment responses. RMSNs use RNNs to estimate the IPTWs and the patient outcomes (details in Appendix B.1.8). To show that standard supervised learning models do not handle the time-varying confounders we compare against an RNN and a linear regression model, which receive as input treatments and covariates to predict the outcome (see Appendix B.1.9 for details). Our model architecture follows the description in Sections 4.1.4 and 4.1.5, with full training details and hyperparameter optimization in Appendix B.1.10. To show the importance of adversarial training, we also benchmark against CRN ($\lambda = 0$) a model with the same architecture, but with $\lambda = 0$, i.e our model architecture without adversarial training.

Evaluate Models on Counterfactual Predictions

Previous methods focused on evaluating the error only for factual outcomes (observed patient outcomes) [117]. However, to build decision support systems, we need to evaluate how well the models estimate the counterfactual outcomes, i.e patient outcomes under



(a) One-step ahead prediction

(b) Five-step ahead prediction

Figure 4.4: Results for prediction of patient counterfactuals.

alternative treatment options. The parameters γ_c and γ_r control the treatment assignment policy, i.e. the degree of time-dependent confounding present in the data. We evaluate the benchmarks under different degrees of time-dependent confounding by setting $\gamma = \gamma_c = \gamma_r$. For each γ we simulate 10000 patients for training, 1000 for validation (hyperparameter tuning) and 1000 for out-of-sample testing. For the patients in the test set, for each time t, we also simulate counterfactuals $\mathbf{Y}_{t+1}$, represented by tumour volume $V(t+1)$, under all possible treatment options.

Figure 4.4 (a) shows the normalized root mean squared error (RMSE) for one-step ahead estimation of counterfactuals with varying degree of time-dependent confounding γ. The RMSE is normalized by the maximum tumour volume: $V_{max} = 1150\text{cm}^3$. The linear and MSM models provide a baseline for performance as they achieve the highest RMSE. While the use of IPTW in MSMs helps when γ increases, using linear modelling has severe limitations. When there is no time-dependent confounding, the machine learning methods achieve similar performance, close to 0.6% RMSE. As the bias in the dataset increases, the harder it becomes for the RNN and the CRN ($\lambda = 0$) to generalize to estimate outcomes of treatments not matching the training policy. When $\gamma = 10$, CRN improves by 48.1% on the same model architecture without domain adversarial training CRN ($\lambda = 0$).

Our proposed model achieves the lowest RMSE across all values of γ. Compared to RMSNs, CRN improves by $\sim 17\%$ when $\gamma > 6$. To highlight the gains of our method even for smaller γ, Figure 4.4 (b) shows the RMSE for five-step ahead prediction (with counterfactuals generated as described in Section 4.1.6 and Appendix B.1.12). RMSNs also use a decoder for sequence prediction. However, RMSNs require training additional

RNNs to estimate the IPTW, which are used to weight each sample during the decoder training. For τ-step ahead prediction, IPTW involves multiplying τ weights which can result in high variance. The results in Figure 4.4 (b) show the problems with using IPTW to handle the time-dependent confounding bias. See Appendix B.1.11 for more results on multi-step ahead prediction.

Balancing representation: To evaluate whether the CRN has indeed learnt treatment invariant represenations, for $\gamma = 5$, we illustrate in Figure 4.5 the T-SNE embeddings of the balancing representations $\Phi(\bar{\mathbf{H}}_t)$ built by the CRN encoder for test patients. We color each point by the treatment $\mathbf{A}_t \in \{$ no treatment, chemotherapy, radiotherapy, combined chemotherapy and radiotherapy $\}$ received at timestep t to highlight the invariance of $\Phi(\bar{\mathbf{H}}_t)$ across the different treatments. In Figure 4.5, we also show $\Phi(\bar{\mathbf{H}}_t)$ only for chemotherapy and radiotherapy for better understanding.

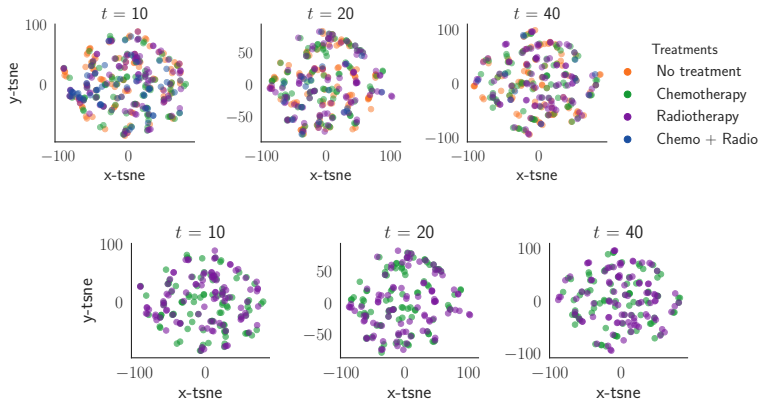


Figure 4.5: TSNE embedding of the balancing representation $\Phi(\bar{\mathbf{H}}_t)$ learnt by the CRN encoder at different timesteps t. Note that $\Phi(\bar{\mathbf{H}}_t)$ is not predictive of treatment $\mathbf{A}_t$ given at timestep t.

Evaluate Recommending the Right Treatment and Timing of Treatment

Evaluating the models just in terms of the RMSE on counterfactual estimation is also not enough for assessing their reliability when used as part of decision support systems. In this section we assess how well the models can select the correct treatment and timing of treatment for several forecasting horizons τ. We generate test sets consisting of 1000 patients where for each horizon τ and for each time t in a patient's trajectory, there are τ options for giving chemotherapy at one of $t, \dots t + \tau - 1$ and τ options for giving radiotherapy at one of $t, \dots t + \tau - 1$. At the rest of the future timesteps, no treatment is

		$\gamma_c = 5, \gamma_r = 5$			$\gamma_c = 5, \gamma_r = 0$			$\gamma_c = 0, \gamma_r = 5$		
	τ	CRN	RMSN	MSM	CRN	RMSN	MSM	CRN	RMSN	MSM
Normalized RMSE	3	**2.43%**	3.16%	6.75%	**1.08%**	1.35%	3.68%	**1.54%**	1.59%	3.23%
	4	**2.83%**	3.95%	7.65%	**1.21%**	1.81%	3.84%	**1.81%**	2.25%	3.52%
	5	**3.18%**	4.37%	7.95%	**1.33%**	2.13%	3.91%	**2.03%**	2.71%	3.63%
	6	**3.51%**	5.61%	8.19%	**1.42%**	2.41%	3.97%	**2.23%**	2.73%	3.71%
	7	**3.93%**	6.21%	8.52%	**1.53%**	2.43%	4.04%	**2.43%**	2.88%	3.79%
Treatment Accuracy	3	**83.1%**	75.3%	73.9%	**83.2%**	78.6%	77.1%	**92.9%**	87.3%	74.9%
	4	**82.5%**	74.1%	68.5%	**81.3%**	77.7%	73.9%	**85.7%**	83.8%	74.1%
	5	**73.5%**	72.7%	63.2%	**78.3%**	77.2%	72.3%	**83.8%**	82.1%	72.8%
	6	**69.4%**	66.7%	62.7%	**79.5%**	76.3%	71.8%	**78.6%**	69.7%	64.5%
	7	**71.2%**	68.8%	62.4%	**72.7%**	71.8%	71.6%	**71.9%**	69.3%	61.2%
Treatment Timing Accuracy	3	**79.6%**	78.1%	67.6%	**80.5%**	76.8%	77.5%	**79.8%**	75.7%	60.6%
	4	**73.9%**	70.3%	63.1%	**79.0%**	77.2%	73.4%	**75.4%**	71.4%	58.2%
	5	**69.8%**	68.6%	62.4%	**78.3%**	73.3%	63.6%	**66.9%**	31.3%	29.5%
	6	**66.9%**	66.2%	62.6%	**73.5%**	72.1%	63.9%	**65.8%**	24.2%	15.5%
	7	**64.5%**	63.6%	62.2%	**70.6%**	57.4%	44.2%	**63.9%**	25.6%	12.5%

Table 4.1: Results for recommending the correct treatment and timing of treatment.

applied. These 2τ treatment plans are assessed in terms of the tumour volume outcome $\mathbf{Y}_{t+\tau}$. We select the treatment (chemotherapy or radiotherapy) that achieves lowest $\mathbf{Y}_{t+\tau}$, and within the correct treatment the timing with lowest $\mathbf{Y}_{t+\tau}$. We also compute the normalized RMSE for predicting $\mathbf{Y}_{t+\tau}$. See Appendix B.1.12 for more details about the test set. The models are evaluated for 3 settings of γ_c and γ_r.

Table 4.1 shows the results for this evaluation set-up. The treatment accuracy denotes the percentage of patients for which the correct treatment was selected, while the treatment timing accuracy is the percentage for which the correct timing was selected. Note that when $\gamma_c = 0$ and $\gamma_r = 5$, RMSN and MSM select the wrong treatment timing for projection horizons $\tau > 4$. CRN performs similarly among the different policies present in the observational data and achieve the lowest RMSE and highest accuracy in selecting the correct treatment and timing of treatment.

In Appendix B.1.13 we also show the applicability of the CRN in more complex medical scenarios involving real data. We provide experimental results based on the Medical Information Mart for Intensive Care (MIMIC III) database [148] consisting of electronic health records from patients in the ICU.

4.1.7 Conclusion

Despite its wide applicability, the problem of causal inference for time-dependent treatments has been relatively less studied compared to problem of causal inference in the static setting. Both new methods and theory are necessary to be able to harness the full potential of observational data for learning individualized effects of complex treatment scenarios. Further work in this direction is needed for proposing alternative methods for handling time-dependent confounders, for modelling combinations of treatments assigned over time or for estimating the individualized effects of time-dependent treatments with associated dosage.

In this chapter, we introduced the Counterfactual Recurrent Network (CRN), a model that estimates individualized effects of treatments over time using a novel way of handling the bias from time-dependent confounders through adversarial training. Using a model of tumour growth, we validated CRN in realistic medical scenarios and we showed improvements over existing state-of-the-art methods. We also showed the applicability of CRN on a real dataset consisting of patient electronic health records. The counterfactual predictions of CRN have the potential to be used as part of clinical decision support systems to address relevant medical challenges involving selecting the best treatments for patients over time, identify optimal treatment timings but also when the treatment is no longer needed. In future work, we will aim to build better balancing representations and to provide theoretical guarantees for the expected error on the counterfactuals.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Estimating Counterfactual Treatment Outcomes over Time through Adversarially Balanced Representations
Publication Status	☑ Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Ioana Bica, Ahmed M. Alaa, James Jordon, and Mihaela van der Schaar. "Estimating Counterfactual Treatment Outcomes over Time through Adversarially Balanced Representations". In *International Conference on Learning Representation (ICLR)*, 2020.

Student Confirmation

Student Name:	Ioana Bica		
Contribution to the Paper	- I was the lead author responsible for conceptualizing the paper, developing and implementing the method, designing the experimental analysis, obtaining the results, structuring and writing the original draft. - All authors contributed to the development of the paper through multiple discussions. - All authors reviewed the final draft.		
Signature 		Date	30/03/2022

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Mihaela van der Schaar		
Supervisor comments I agree that the candidate has made a substantial contribution to the publication and that the above description is accurate.		
Signature 	Date	30/03/2022

This completed form should be included in the thesis, at the end of the relevant chapter.

4.2 Time Series Deconfounder: Estimating Treatment Effects over Time in the Presence of Hidden Confounders

Publication Included

- **Ioana Bica**, Ahmed M. Alaa, and Mihaela van der Schaar. "Time Series Deconfounder: Estimating treatment effects over time in the presence of hidden confounders". In *International Conference on Machine Learning*, 2020. Link: http://proceedings.mlr.press/v119/bica20a. [Accepted]

Abstract

The estimation of treatment effects is a pervasive problem in medicine. Existing methods for estimating treatment effects from longitudinal observational data assume that there are no hidden confounders, an assumption that is not testable in practice and, if it does not hold, leads to biased estimates. In this chapter, we develop the Time Series Deconfounder, a method that leverages the assignment of multiple treatments over time to enable the estimation of treatment effects in the presence of multi-cause hidden confounders. The Time Series Deconfounder uses a novel recurrent neural network architecture with multitask output to build a factor model over time and infer latent variables that render the assigned treatments conditionally independent; then, it performs causal inference using these latent variables that act as substitutes for the multi-cause unobserved confounders. We provide a theoretical analysis for obtaining unbiased causal effects of time-varying exposures using the Time Series Deconfounder. Using both simulated and real data we show the effectiveness of our method in deconfounding the estimation of treatment responses over time.

4.2.1 Introduction

Forecasting the patient's response to treatments assigned over time represents a crucial problem in the medical domain. The increasing availability of observational data makes it possible to learn individualized treatment responses from longitudinal disease trajectories

containing information about patient covariates and treatment assignments [14, 40, 55, 81, 117]. Existing methods for estimating individualized treatment effects over time assume that all confounders—variables affecting the treatment assignments and the potential outcomes—are observed, an assumption which is not testable in practice[2] and probably not true in many situations.

To understand why the presence of hidden confounders introduces bias, consider the problem of estimating treatment effects for patients with cancer. They are often prescribed multiple treatments at the same time, including chemotherapy, radiotherapy and/or immunotherapy based on their tumor characteristics. These treatments are adjusted if the tumor size changes. The treatment strategy is also changed as the patient starts to develop drug resistance [149] or the toxicity levels of the drugs increase [150]. Drug resistance and toxicity levels are multi-cause confounders since they affect not only the multiple causes (treatments)[3], but also the patient outcome (e.g. mortality, risk factors). However, drug resistance and toxicity may not be observed and, even if observed, may not be recorded in the electronic health records. Estimating, for instance, the effect of chemotherapy on the cancer progression in the patient without accounting for the dependence on drug resistance and toxicity levels (hidden confounders) will produce biased results.

Wang and Blei [99] developed theory for deconfounding—adjusting for the bias introduced by the presence of hidden confounders in observational datasets—in the *static* causal inference setting and noted that the existence of multiple causes makes this task easier. [99] observed that the dependencies in the assignment of multiple causes can be used to infer latent variables that render the causes independent and act as substitutes for the hidden confounders.

In this chapter, we propose the Time Series Deconfounder, a method that enables the unbiased estimation of treatment responses *over time* in the presence of hidden confounders, by taking advantage of the dependencies in the sequential assignment of multiple treatments. We draw from the main idea in [99], but note that the estimation of hidden confounders in the longitudinal setting is significantly more complex than

[2]Since counterfactuals are never observed, it is not possible to test for the existence of hidden confounders that could affect them.

[3]Causes and treatments are used interchangeably throughout the chapter.

in the static setting, not just because the hidden confounders may vary over time but in particular because the hidden confounders may be affected by previous treatments and covariates. Thus, standard latent variable models are no longer applicable, as they cannot capture these time dependencies.

The Time Series Deconfounder relies on building a factor model *over time* to obtain latent variables which, together with the observed variables render the assigned causes conditionally independent. Through theoretical analysis we show that these latent variables can act as substitutes for the multi-cause unobserved confounders and can be used to satisfy the sequential strong ignorability condition in the potential outcomes framework for time-varying exposures [81] and obtain unbiased estimates of individualized treatment responses, using weaker assumptions than standard methods. Following our theory, we propose a novel deep learning architecture, based on a recurrent neural network with multi-task outputs and variational dropout, to build such a factor model and infer the substitutes for the hidden confounders in practice.

The Time Series Deconfounder shifts the need for observing all multi-cause confounders (untestable condition) to constructing a good factor model over time (testable condition). To assess how well the factor model captures the distribution of assigned treatments, we extend the use of predictive checks [99, 128] to the temporal setting and compute p-values at each timestep. We perform experiments on a simulated dataset where we control the amount of hidden confounding applied and on a real dataset with patients in the ICU [148] to show how the Time Series Deconfounder allows us to deconfound the estimation of treatment responses in longitudinal data. To the best of our knowledge, this represents the first method for learning latent variables that can act as substitutes for the unobserved confounders in the time series setting.

4.2.2 Related Work

Previous methods for causal inference mostly focused on the static setting [24, 26, 31, 32, 35, 54, 102, 151], and less attention has been given to the time series setting. We discuss methods for estimating treatment effects over time, as well as methods for inferring substitute hidden confounders in the static setting.

Potential outcomes for time-varying treatment assignments. Standard methods for performing counterfactual inference in longitudinal data are found in the epidemiology literature and include the g-computation formula, g-estimation of structural nested mean models, and inverse probability of treatment weighting of marginal structural models [14, 81, 113]. Alternatively, [117] improves on the standard marginal structural models by using recurrent neural networks to estimate the propensity weights and treatment responses, while [55] proposes using balancing representations to handle the time-dependent confounding bias when estimating treatment effects over time. Despite the wide applicability of these methods in forecasting treatment responses, they are all based on the assumption that there are no hidden confounders. Our chapter proposes a method for deconfounding such outcome models, by inferring substitutes for the hidden confounders which can lead to unbiased estimates of the potential outcomes.

The potential outcomes framework has been extended to the continuous-time setting by [139]. Several methods have been proposed for estimating treatment responses in continuous time [40, 116], again assuming that there are no hidden confounders. Here, we focus on deconfounding the estimation of treatment responses in the discrete-time setting.

Sensitivity analysis methods that evaluate the potential impact that an unmeasured confounder could have on the estimation of treatment effects have also been developed [91, 92, 115]. However, these methods assess the suitability of applying existing tools, rather than propose a direct solution for handling the presence of hidden confounders.

Latent variable models for estimating hidden confounders. The most similar work to ours is the one of [99], who proposed the deconfounder, an algorithm that infers latent variables that act as substitutes for the hidden confounders and then performs causal inference in the static multi-cause setting. The deconfounder involves finding a good factor model of the assigned causes which can be used to estimate substitutes for the hidden confounders. Then, the deconfounder fits an outcome model for estimating the causal effects using the inferred latent variables. Our chapter extends the theory for the deconfounder to the time-varying treatments setting and shows how the inferred latent variables can lead to sequential strong ignorability. To estimate the substitute confounders, [99] used standard factor models [152, 153], which are only applicable in

the static setting. To build a factor model over time, we propose an RNN architecture with multitask output and variational dropout.

Several other methods have been proposed for taking advantage of the multiplicity of assigned treatments in the static setting and capturing shared latent confounding [100, 154, 155]. These works are based on Pearl's causal framework [156] and use structural equation models. Alternative methods for dealing with hidden confounders in the static setting use proxy variables as noisy substitutes for the confounders [94, 97, 98].

A different line of research involves performing causal discovery in the presence of hidden confounders [157]. In this context, several methods have been proposed to perform causal graphical model structure learning with latent variables [158–160]. However, in this chapter, we are not aiming to discover causal relationships between patient covariates over time. Instead, we improve existing methods for estimating the individualized effects of time-dependent treatments by accounting for multi-cause unobserved confounders.

4.2.3 Problem Formulation

Let $\mathbf{X}_t^{(i)} \in \mathcal{X}_t$ be the time-dependent covariates, $\mathbf{A}_t^{(i)} = [A_{t1}^{(i)} \dots A_{tk}^{(i)}] \in \mathcal{A}_t$ be the possible assignment of k treatments (causes) at timestep t and let $\mathbf{Y}_{t+1}^{(i)} \in \mathcal{Y}_t$ be the observed outcomes for patient (i). Treatments can be either binary and/or continuous. Static features, such as genetic information, do not change our theory, and, for simplicity, we assume they are part of the observed covariates.

The observational data for patient (i), also known as the patient trajectory, consists of realizations of the previously described random variables $\tau^{(i)} = \{\mathbf{x}_t^{(i)}, \mathbf{a}_t^{(i)}, \mathbf{y}_{t+1}^{(i)}\}_{t=1}^{T^{(i)}}$, with samples collected for $T^{(i)}$ discrete and regular timesteps. Electronic health records consist of data for N independent patients $\mathcal{D} = \{\tau^{(i)}\}_{i=1}^{N}$. For simplicity, we omit the patient superscript (i) unless it is explicitly needed.

We leverage the potential outcomes framework proposed by [80] and [79], and extended by [81] to take into account time-varying treatments. Let $\mathbf{Y}(\bar{\mathbf{a}})$ be the potential outcome, either factual or counterfactual, for each possible course of treatment $\bar{\mathbf{a}}$.

Let $\bar{\mathbf{A}}_t = (\mathbf{A}_1, \dots, \mathbf{A}_t) \in \bar{\mathcal{A}}_t$ be the history of treatments and $\bar{\mathbf{X}}_t = (\mathbf{X}_1, \dots, \mathbf{X}_t) \in \bar{\mathcal{X}}_t$ the history of covariates until timestep t. For each patient, we want to estimate

individualized treatment effects, i.e. potential outcomes conditional on the patient history of covariates and treatments:

$$\mathbb{E}[\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t], \tag{4.8}$$

for any possible treatment plan $\bar{\mathbf{a}}_{\geq t}$ that starts at timestep t and consists of a sequence of treatments that ends just before the patient outcome $\mathbf{Y}$ is observed. The observational data can be used to fit a regression model to estimate $\mathbb{E}[\mathbf{Y} \mid \bar{\mathbf{a}}_{\geq t}, \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t]$. Under certain assumptions, these estimates are unbiased so that $\mathbb{E}[\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}] = \mathbb{E}[\mathbf{Y} \mid \bar{\mathbf{a}}_{\geq t}, \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t]$. These conditions include Assumptions 1 and 2, which are standard in the literature and can be tested in practice [81].

Assumption 4.2.1. ***Consistency***. *If $\bar{\mathbf{A}}_{\geq t} = \bar{\mathbf{a}}_{\geq t}$, then the potential outcomes for following the treatment plan $\bar{\mathbf{a}}_{\geq t}$ is the same as the observed (factual) outcome $\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) = \mathbf{Y}$.*

Assumption 4.2.2. ***Positivity (Overlap)*** *[110]: If $P(\bar{\mathbf{A}}_{t-1} = \bar{\mathbf{a}}_{t-1}, \bar{\mathbf{X}}_t = \bar{\mathbf{x}}_t) \neq 0$ then $P(\mathbf{A}_t = \mathbf{a}_t \mid \bar{\mathbf{A}}_{t-1} = \bar{\mathbf{a}}_{t-1}, \bar{\mathbf{X}}_t = \bar{\mathbf{x}}_t) > 0$ for all $\mathbf{a}_t$.*

The positivity assumption means that at each timestep t, each treatment has a non-zero probability of being given to the patient. This assumption is testable in practice.

In addition, existing methods also assume *sequential strong ignorability*:

$$\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \perp\!\!\!\perp \mathbf{A}_t \mid \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t, \tag{4.9}$$

for all possible treatment plans $\bar{\mathbf{a}}_{\geq t}$ and for all $t \in \{0, \ldots, T\}$. This condition holds if there are no hidden confounders and it cannot be tested in practice. To understand why this is the case, note that the sequential strong ignorability assumption requires the conditional independence of the treatments with all of the potential outcomes, both factual and counterfactual, conditional on the patient history. Since the counterfactuals are never observed, it is not possible to test for this conditional independence. In this chapter, we assume that there are hidden confounders. Consequently, using standard methods for computing $\mathbb{E}[\mathbf{Y} \mid \bar{\mathbf{a}}_{\geq t}, \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t]$ from the dataset will result in biased estimates since the

hidden confounders introduce a dependence between the treatments at each timestep and the potential outcomes ($\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \not\perp\!\!\!\perp \mathbf{A}_t \mid \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t$) and therefore:

$$\mathbb{E}[\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t] \neq \mathbb{E}[\mathbf{Y} \mid \bar{\mathbf{a}}_{\geq t}, \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t]. \tag{4.10}$$

By extending the method proposed by [99], we take advantage of the multiple treatment assignments at each timestep to infer a sequence of latent variables $\bar{\mathbf{Z}}_t = (\mathbf{Z}_1, \ldots, \mathbf{Z}_t) \in \bar{\mathcal{Z}}_t$ that can be used as substitutes for the unobserved confounders. We will then show how $\bar{\mathbf{Z}}_t$ can be used to estimate the treatment effects over time.

4.2.4 Time Series Deconfounder

The idea behind the Time Series Deconfounder is that multi-cause confounders introduce dependencies between the treatments. As treatment assignments change over time we infer substitutes for the hidden confounders that take advantage of the patient history to capture these dependencies.

Factor Model

The Time Series Deconfounder builds a factor model to capture the distribution of the causes (treatments) over time. At time t, the factor model constructs the latent variable $\mathbf{z}_t = g(\bar{\mathbf{h}}_{t-1})$, where $\bar{\mathbf{h}}_{t-1} = (\bar{\mathbf{a}}_{t-1}, \bar{\mathbf{x}}_{t-1}, \bar{\mathbf{z}}_{t-1})$ is the realization of history $\bar{\mathbf{H}}_{t-1}$. The latent variable $\mathbf{z}_t$, together with the observed patient covariates $\mathbf{x}_t$, render the assigned treatments conditionally independent:

$$p(a_{t1}, \ldots, a_{tk} \mid \mathbf{z}_t, \mathbf{x}_t) = \prod_{j=1}^{k} p(a_{tj} \mid \mathbf{z}_t, \mathbf{x}_t). \tag{4.11}$$

Figure 4.6(a) illustrates the corresponding graphical model for timestep t. The factor model of the assigned treatments is built as a latent variable model with joint distribution:

$$p(\theta_{1:k}, \bar{\mathbf{x}}_T, \bar{\mathbf{z}}_T, \bar{\mathbf{a}}_T) = p(\theta_{1:k}) p(\bar{\mathbf{x}}_T) \cdot \prod_{t=1}^{T} \Big(p(\mathbf{z}_t \mid \bar{\mathbf{h}}_{t-1}) \prod_{j=1}^{k} p(a_{tj} \mid \mathbf{z}_t, \mathbf{x}_t, \theta_j) \Big), \tag{4.12}$$

where $\theta_{1:k}$ are parameters. The distribution of the treatments $p(\bar{\mathbf{a}}_T)$ is the corresponding marginal. Notice that we do not assume that, in the observational data, the patient covariates $\mathbf{x}_t$ at timestep t are independent of the patient history. The graphical factor

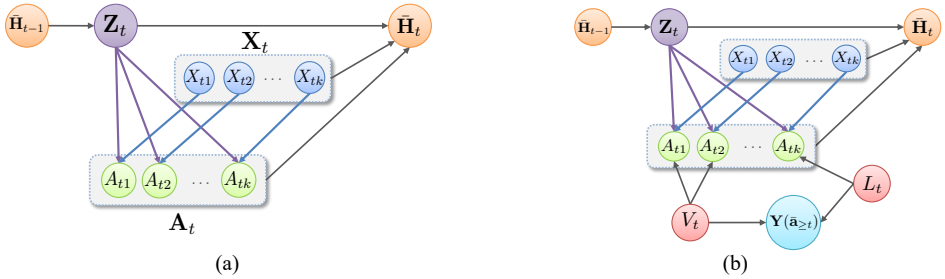


Figure 4.6: (a) Graphical factor model. Each $\mathbf{Z}_t$ is built as a function of the history, such that, with $\mathbf{X}_t$, it renders the assigned causes conditionally independent: $p(a_{t1}, \ldots, a_{tk} \mid \mathbf{z}_t, \mathbf{x}_t) = \prod_{j=1}^{k} p(a_{tj} \mid \mathbf{z}_t, \mathbf{x}_t)$. The variables can be connected to $\mathbf{Y}(\bar{\mathbf{a}}_{\geq t})$ in any way. (b) Graphical model explanation for why this factor model construction ensures that $\mathbf{Z}_t$ captures all of the multi-cause hidden confounders.

model shows how the latent variables $\mathbf{z}_t$ that can act as substitutes for the multi-cause unobserved confounders are built. As we will see in Section 4.2.4 these latent variables will be used as part of an outcome model that estimates the potential outcomes $\mathbf{Y}(\bar{\mathbf{a}}_{\geq t})$.

By taking advantage of the dependencies between the multiple treatment assignments, the factor model allows us to infer the sequence of latent variables $\bar{\mathbf{Z}}_t$ that can be used to render the assigned causes conditionally independent. Through this factor model construction and under correct model specifications, we can rule out the existence of other multi-cause confounders that are not captured by $\mathbf{Z}_t$. To understand why this is the case, consider the graphical model in Figure 4.6(b). By contradiction, assume that there exists another multi-cause confounder V_t not captured by $\mathbf{Z}_t$. Then, by d-separation the conditional independence between the assigned causes given $\mathbf{Z}_t$ and $\mathbf{X}_t$ does not hold anymore. This argument cannot be used for single-cause confounders, such as L_t, which are only affecting one of the causes and the potential outcomes. Thus, we assume sequential single strong ignorability (no hidden single cause confounders).

Assumption 4.2.3. ***Sequential single strong ignorability:***

$$\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \perp\!\!\!\perp A_{tj} \mid \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}, \tag{4.13}$$

for all $\bar{\mathbf{a}}_{\geq t}$*, for all* $t \in \{0, \ldots, T\}$*, and for all* $j \in \{1, \ldots, k\}$.

Causal inference relies on assumptions. Existing methods for estimating treatment effects over time assume that there are no multi-cause and no single-cause hidden confounders. Here, we make the *weaker* assumption that there are no single-cause

hidden confounders. While this assumption is also untestable in practice, as the number of treatments increases for each timestep, it becomes increasingly weaker: the more treatments we observe, the more likely it becomes for a hidden confounder to affect multiple of the these treatments rather than a single one of them.

Theorem 4.2.1. *If the distribution of the assigned causes* $p(\bar{\mathbf{a}}_T)$ *can be written as the factor model* $p(\theta_{1:k}, \bar{\mathbf{x}}_T, \bar{\mathbf{z}}_T, \bar{\mathbf{a}}_T)$*, we obtain sequential ignorable treatment assignment:*

$$\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \perp\!\!\!\perp (A_{t1}, \ldots, A_{tk}) \mid \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t, \tag{4.14}$$

for all $\bar{\mathbf{a}}_{\geq t}$ *and for all* $t \in \{0, \ldots, T\}$.

Theorem 4.2.1 is proved by leveraging Assumption 4.2.3, the fact that the latent variables $\mathbf{Z}_t$ are inferred without knowledge of the potential outcomes $\mathbf{Y}(\bar{\mathbf{a}}_{\geq t})$ and the fact that the causes $(A_{t1}, \ldots, A_{tk})$ are jointly independent given $\mathbf{Z}_t$ and $\mathbf{X}_t$. The result means that, at each timestep, the variables $\bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t, \bar{\mathbf{A}}_{t-1}$ contain all of the dependencies between the potential outcomes $\mathbf{Y}(\bar{\mathbf{a}}_{\geq t})$ and the assigned causes $\mathbf{A}_t$. See Appendix B.2.1 for the full proof.

As discussed in [99], the substitute confounders $\mathbf{Z}_t$ also need to satisfy positivity (Assumption 2), i.e. if $P(\bar{\mathbf{A}}_{t-1} = \bar{\mathbf{a}}_{t-1}, \bar{\mathbf{Z}}_t = \bar{\mathbf{z}}_t, \bar{\mathbf{X}}_t = \bar{\mathbf{x}}_t) \neq 0$ then $P(\mathbf{A}_t = \mathbf{a}_t \mid \bar{\mathbf{A}}_{t-1} = \bar{\mathbf{a}}_{t-1}, \bar{\mathbf{Z}}_t = \bar{\mathbf{z}}_t, \bar{\mathbf{X}}_t = \bar{\mathbf{x}}_t) > 0$ for all $\mathbf{a}_t$. After fitting the factor model, this can be tested [81]. When positivity is limited, the outcome model estimates of treatment responses will also have high variance. In practice, positivity can be enforced by setting the dimensionality of $\mathbf{Z}_t$ to be smaller than the number of treatments [99].

Predictive Checks over Time: The theory holds if the fitted factor model captures well the distribution of assigned treatments. This condition can be assessed by extending predictive model checking [128] to the time-series setting. We compute p-values over time to evaluate how similar the distribution of the treatments learned by the factor model is with the distribution of the treatments in a validation set of patients. At each timestep t, for the patients in the validation set, we obtain M replicas of their treatment assignments $\{\mathbf{a}_{t,\text{rep}}^{(i)}\}_{i=1}^{M}$ by sampling from the factor model. The replicated treatment assignments are compared with the actual treatment assignments, $\mathbf{a}_{t,\text{val}}$, using the test statistic $T(\mathbf{a}_t)$:

$$T(\mathbf{a}_t) = \mathbb{E}_Z[\log\, p(\mathbf{a}_t \mid Z_t, X_t)], \tag{4.15}$$

which is related to the marginal log likelihood [99]. The predictive p-value for timestep t is computed as follows: $\frac{1}{M}\sum_{i=1}^{M}\mathbf{1}\left(T\left(\mathbf{a}_{t,\text{rep}}^{(i)}\right) < T\left(\mathbf{a}_{t,\text{val}}\right)\right)$, where $\mathbf{1}(\cdot)$ represents the indicator function. If the model captures well the distribution of the assigned causes, then the test statistics for the treatment replicas are similar to the test statistics for the treatments in the validation set, which makes 0.5 the ideal $p-$value in this case.

Outcome Model

If the factor model passes the predictive checks, the Time Series Deconfounder fits an outcome model [14, 117] to estimate treatment effects over time. After sampling the sequence of latent variables $\hat{\bar{\mathbf{Z}}}_t = (\hat{\mathbf{Z}}_1, \ldots, \hat{\mathbf{Z}}_t)$ from the factor model, the outcome model can be used to estimate $\mathbb{E}[\mathbf{Y} \mid \bar{\mathbf{a}}_{\geq t}, \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t, \hat{\bar{\mathbf{Z}}}_t] = \mathbb{E}[\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t, \hat{\bar{\mathbf{Z}}}_t]$. To compute uncertainty estimates of the potential outcomes, we can sample $\hat{\bar{\mathbf{Z}}}_t$ repeatedly and then fit an outcome model for each sample to obtain multiple point estimates of $\mathbf{Y}(\bar{\mathbf{a}}_{\geq t})$. The variance of these point estimates will represent the uncertainty of the Time Series Deconfounder. D'Amour [161] raised some concerns about identifiability of the mean potential outcomes using the deconfounder framework in [99] in the static setting and illustrated some pathological examples where identifiability might not hold.[4] In practical settings, the outcome estimates from the Time Series Deconfounder are identifiable, as supported by the experimental results in Sections 4.2.6 and 4.2.7. Nevertheless, when identifiability represents an issue, the uncertainty in the potential outcomes can be used to assess the reliability of the Time Series Deconfounder. In particular, the variance in the potential outcomes indicates how the finite observational data inform the estimation of substitutes for the hidden confounders and subsequently the treatment outcomes of interest. When the treatment effects are non-identifiable, the estimates of the Time Series Deconfounder will have high variance.

By using this framework to estimate substitutes for the hidden confounders we are trading off confounding bias for estimation variance [99]. The treatment effects computed without accounting for the hidden confounders will inevitably be biased. Alternatively,

[4] See [99] for a longer discussion addressing the concerns in [161].

using the latent variables from the factor model will result in unbiased, but higher variance estimates of treatment effects.

4.2.5 Factor Model over Time in Practice

Since we are dealing with time-varying treatments, we cannot use standard factor models, such as PCA [152] or Deep Exponential Families [153], as they can only be applied in the static setting. Using the theory developed for the factor model over time we introduce a practical implementation based on a recurrent neural network (RNN) with multitask output and variational dropout as illustrated in Figure 4.7. The recurrent part of the model infers the latent variables $\mathbf{Z}_t$ such that they depend on the patient history:

$$\mathbf{Z}_1 = \text{RNN}(\mathbf{L}), \qquad \mathbf{Z}_t = \text{RNN}(\bar{\mathbf{Z}}_{t-1}, \bar{\mathbf{X}}_{t-1}, \bar{\mathbf{A}}_{t-1}, \mathbf{L}), \tag{4.16}$$

where $\mathbf{L}$ consists of randomly initialized parameters that are trained with the rest of the parameters in the RNN. The size of the RNN output is D_Z and this specifies the size of the latent variables that are inferred as substitutes for the hidden confounders. In our experiments, we use an LSTM unit [143] as part of the RNN. Moreover, to infer the assigned treatments at timestep t, $\mathbf{A}_t = [A_{t1}, \ldots, A_{tk}]$ such that they are conditionally independent given the latent variables $\mathbf{Z}_t$ and the observed covariates $\mathbf{X}_t$, we propose using multitask multilayer perceptrons (MLPs) consisting of fully connected (FC) layers:

$$A_{tj} = \text{FC}(\mathbf{X}_t, \mathbf{Z}_t; \theta_j), \tag{4.17}$$

for all $j = 1, \ldots k$ and for all $t = 1, \ldots T$, where θ_j are the parameters in the FC layers used to obtain A_{tj}. We use a single FC hidden layer before the output layer. For binary treatments, the sigmoid activation is used in the output layer. For continuous treatments, MC dropout [162] can instead be applied in the FC layers to obtain $p(A_{tj} \mid \mathbf{X}_j, \mathbf{Z}_j)$. To model the probabilistic nature of factor models we incorporate *variational dropout* [144] in the RNN as illustrated in Figure 4.7. Dropout enables us to obtain samples from $\mathbf{Z}_t$ and treatment assignments A_{tj}. These samples allow us to obtain treatment replicas and to compute predictive checks over time, but also to estimate the *uncertainty* in $\mathbf{Z}_t$ and potential outcomes.

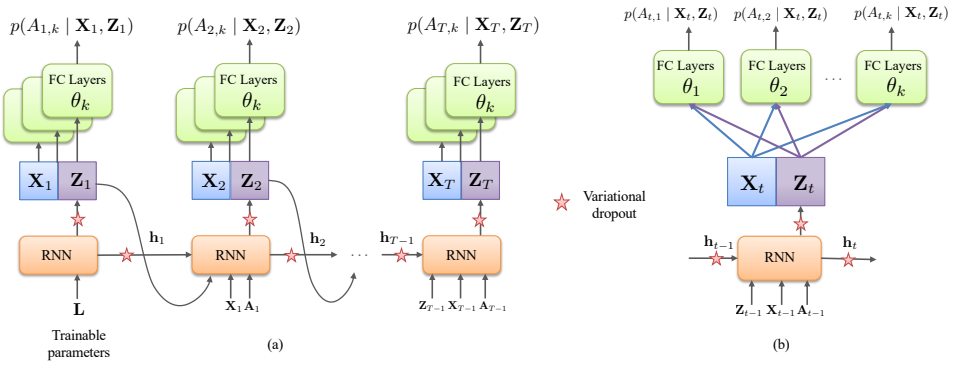


Figure 4.7: (a) Proposed factor model implementation. $\mathbf{Z}_t$ is generated by RNN as a function of the history $\bar{\mathbf{H}}_{t-1}$, given by the hidden state $\mathbf{h}_t$, and current input. Multitask output is used to construct the treatments such that they are independent given $\mathbf{Z}_t$ and $\mathbf{X}_t$. (b) Single timestep.

Using the treatment assignments from the observational dataset, the factor model can be trained using gradient descent based methods. The proposed factor model architecture follows from the theory developed in Section 4.2.4 where at each timestep the latent variable $\mathbf{Z}_t$ is built as a function of the history (parametrized by an RNN). The multitask output is essential for modeling the conditional independence between the assigned treatments given the latent variables generated by the RNN and the observed covariates.

Note that our theory does not put restrictions on the factor model that can be used. Alternative factor models over time are generalized dynamic-factor model [163, 164] or factor-augmented vector autoregressive models [165]. These come from the econometrics literature and explicitly model the dynamics in the data. The use of RNNs in the factor model enables us to learn complex relationships between $\bar{\mathbf{X}}_t$, $\bar{\mathbf{Z}}_t$, and $\bar{\mathbf{A}}_t$ from the data, which is needed in medical applications involving complex diseases. Nevertheless, predictive checks should be used to assess any selected factor model.

4.2.6 Experiments on Synthetic Data

To validate the theory developed in this chapter, we perform experiments on synthetic data where we vary the effect of hidden confounding. It is not possible to validate the method on real datasets since the true extent of hidden confounding is never known [97, 99].

Simulated Dataset

To keep the simulation process general, we propose building a dataset using p-order autoregressive processes. At each timestep t, we simulate k time-varying covariates $X_{t,k}$

representing single cause confounders and a multi-cause hidden confounder Z_t as follows:

$$X_{t,j} = \frac{1}{p}\sum_{i=1}^{p}(\alpha_{i,j}X_{t-i,j} + \omega_{i,j}A_{t-i,j}) + \eta_t \tag{4.18}$$

$$Z_t = \frac{1}{p}\sum_{i=1}^{p}(\beta_i Z_{t-i} + \sum_{j=1}^{k}\lambda_{i,j}A_{t-i,j}) + \epsilon_t, \tag{4.19}$$

for $j = 1, \ldots, k$, $\alpha_{i,k}, \lambda_{i,j} \sim \mathcal{N}(0, 0.5^2)$, $\omega_{i,k}, \beta_i \sim \mathcal{N}(1-(i/p), (1/p)^2)$, and $\eta_t, \epsilon_t \sim \mathcal{N}(0, 0.01^2)$. The value of Z_t changes over time and is affected by treatment assignments. Each treatment assignment $A_{t,j}$ depends on the single-cause confounder $X_{t,j}$ and multi-cause hidden confounder Z_t:

$$\pi_{tj} = \gamma_A \hat{Z}_t + (1-\gamma_A)\hat{X}_{tj} \qquad A_{tj} \mid \pi_{tj} \sim \text{Bernoulli}(\sigma(\lambda\pi_{tj})), \tag{4.20}$$

where $\hat{X}_{tj}$ and $\hat{Z}_t$ are the sum of the covariates and confounders respectively over the last p timesteps, $\lambda = 15$, $\sigma(\cdot)$ is the sigmoid function and γ_A controls the amount of hidden confounding applied to the treatment assignments. The outcome is also obtained as a function of the covariates and the hidden confounder:

$$\mathbf{Y}_{t+1} = \gamma_Y Z_{t+1} + (1-\gamma_Y)\Big(\frac{1}{k}\sum_{j=1}^{k}X_{t+1,j}\Big), \tag{4.21}$$

where γ_Y controls the amount of hidden confounding applied to the outcome. We simulate datasets consisting of 5000 patients, with trajectories between 20 and 30 timesteps, and $k = 3$ covariates and treatments. To induce time dependencies we set $p = 5$. Each dataset undergoes a 80/10/10 split for training, validation and testing respectively. Hyperparameter optimization is performed for each trained factor model as explained in Appendix B.2.2. Using the training observational dataset, we fit the Time Series Deconfounder to perform one-step ahead estimation of treatment responses.

Evaluating Factor Model using Predictive Checks

Our theory for using the inferred latent variables as substitutes for the hidden confounders and obtain unbiased treatment responses relies on the fact that the factor model captures well the distribution of the assigned causes. To assess the suitability of our proposed factor model architecture, we compare it with the following baselines: RNN without multitask output (predicting the k treatment assignments by passing $\mathbf{X}_t$ and $\mathbf{Z}_t$ through a

FC layer and output layer with k neurons) and multilayer perceptron (MLP) used instead of the RNN at each timestep to generate $\mathbf{Z}_t$. The MLP factor model does not use the entire history for generating $\mathbf{Z}_t$. See Appendix B.2.3 for details.

Figure 4.8 shows the p-values over time computed for the test set in 30 simulated datasets with $\gamma_A = \gamma_Y = 0.5$. The p-values for the MLP factor model decrease over time, which means that there is a consistent distribution mismatch between the treatment assignments learned by this model and the ones in the test set. Conversely, the predictive checks for our proposed factor model are closer to the ideal p-value of 0.5. This illustrates that having an architecture capable of capturing time-dependencies and accumulating past information for inferring the latent confounders is crucial. Moreover, the performance for the RNN without multitask is similar to our model, which indicates that the factor model constraint does not affect the performance in capturing the distribution of the causes.

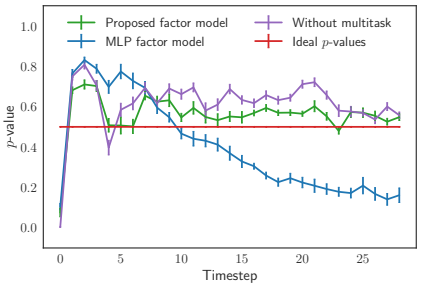


Figure 4.8: Predictive checks over time. We show the mean p-values at each timestep and the std error.

Deconfounding the Estimation of Treatment Responses over Time

We evaluate how well the Time Series Deconfounder [5] can remove hidden confounding bias when used in conjunction with the following outcome models:

Standard Marginal Structural Models (MSMs). MSMs [14, 137] have been widely used in epidemiology to estimate treatment effects over time. MSMs use inverse probability of treatment weighting (IPTW) to adjust for the time-dependent confounding bias present in observational datasets [16, 53]. MSMs compute the propensity weights by using logistic regression; through IPTW, these models construct a pseudo-population from the observational data where the treatment assignment probability no longer depends on the time-varying confounders. The treatment responses over time are computed using linear regression. For full implementation details see Appendix B.2.4.

[5] The implementation of the Time Series Deconfounder can be found at https://github.com/vanderschaarlab/mlforhealthlabpub and at https://github.com/ioanabica/Time-Series-Deconfounder.

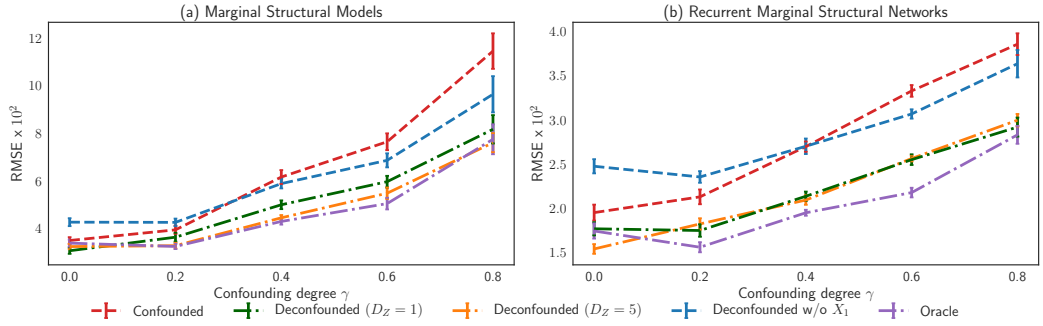


Figure 4.9: Results for deconfounding the one-step ahead estimation of treatment responses in two outcome models: (a) Marginal Structural Models (MSM) and (b) Recurrent Marginal Structural Networks (R-MSN). The average RMSE and the standard error in the results are computed for 30 dataset simulations for each different degree of confounding, as measured by γ.

Recurrent Marginal Structural Networks (R-MSNs). R-MSNs [117] also use IPTW to remove the bias from time-dependent confounders when estimating treatment effects over time. However, R-MSNs estimate the propensity scores using RNNs instead. The use of RNNs is more robust to changes in the treatment assignment policy. To estimate the treatment responses over time R-MSNs also use a model based on RNNs. For implementation details, see Appendix B.2.4. Notice that these outcome models were also chosen because they are capable of handling multiple treatments (that may be assigned simultaneously) at the same timstep. In the simulated dataset, parameters γ_A and γ_Y control the amount of hidden confounding applied to the treatments and outcomes respectively. We vary this amount through $\gamma_A = \gamma_Y = \gamma$.

The Time Series Deconfounder is used to obtain unbiased estimates of one-step ahead treatment responses. For a comparative evaluation, the outcome models are trained to estimate these treatment responses in the following scenarios: without information about $\bar{\mathbf{Z}}_t$ in which case they estimate $\mathbb{E}[\mathbf{Y}_{t+1}(\mathbf{a}_t) \mid \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t]$ (Confounded), with information about the simulated (oracle) $\bar{\mathbf{Z}}_t$ which leads to the following regression model $\mathbb{E}[\mathbf{Y}_{t+1}(\mathbf{a}_t) \mid \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t]$ (Oracle), as well as after applying the Time Series Deconfounder with different model specifications and in this case $\mathbb{E}[\mathbf{Y}_{t+1}(\mathbf{a}_t) \mid \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t, \hat{\bar{\mathbf{Z}}}_t]$ is estimated (Deconfounded). To highlight the importance of Assumption 3, we also apply the Time Series Deconfounder after removing the single-cause confounder X_1, thus violating the assumption.

Figure 4.9 shows the root mean squared error (RMSE) for the one-step ahead estimation of treatment responses for patients in the test set. We notice that the Time Series Deconfounder gives unbiased estimates of treatment responses, i.e. close to the estimates obtained using the simulated (oracle) confounders. The method is robust to model misspecification, performing similarly when $D_Z = 1$ (simulated size of hidden confounders) and when $D_Z = 5$ (misspecified size of inferred confounders). When there are no hidden confounders ($\gamma = 0$), the extra information from $\hat{\hat{\mathbf{Z}}}_t$ does not harm the estimations (although they have higher variance).

When the sequential single strong ignorability assumption (Assumption 3) is invalidated, namely when the single cause confounder X_1 is removed from the observational dataset, we obtain biased estimates of the treatment responses. The performance in this case, however, is comparable to the performance when there is no control for the unobserved confounders.

In Appendix B.2.5, we consider an experimental set-up with a different simulated size for the hidden confounders (true $D_Z = 3$) and show results when the size of the hidden confounders is underestimated in the Time Series Deconfounder. We also include additional results on a simulated setting with static hidden confounders.

Source of gain: To understand the source of gain in the Time Series Deconfounder, consider why the outcome models fail in the scenarios when there are hidden confounders. MSMs and R-MSNs make the implicit assumption that the treatment assignments depend only on the observed history. The existence of any multi-cause confounders not captured by the patient history results in biased estimates of both the propensity weights and of the outcomes. On the other hand, the construction in our factor model rules out the existence of any multi-cause confounders which are not captured by $\mathbf{Z}_t$. By augmenting the data available to the outcome models with the substitute confounders, we eliminate these biases.

4.2.7 Experiments on MIMIC III

Using the Medical Information Mart for Intensive Care (MIMIC III) [148] database consisting of electronic health records from patients in the ICU, we show how the

Outcome model	White blood cell count		Blood pressure		Oxygen saturation	
	MSM	R-MSN	MSM	R-MSN	MSM	R-MSN
Confounded	3.90 ± 0.00	2.91 ± 0.05	12.04 ± 0.00	10.29 ± 0.05	2.92 ± 0.00	1.74 ± 0.03
Deconfounded ($D_Z = 1$)	3.55 ± 0.05	2.62 ± 0.07	11.69 ± 0.14	9.35 ± 0.11	2.42 ± 0.02	1.24 ± 0.05
Deconfounded ($D_Z = 5$)	3.56 ± 0.04	2.41 ± 0.04	11.63 ± 0.10	9.45 ± 0.10	2.43 ± 0.02	1.21 ± 0.07
Deconfounded ($D_Z = 10$)	3.58 ± 0.03	2.48 ± 0.06	11.66 ± 0.14	9.20 ± 0.12	2.42 ± 0.01	1.17 ± 0.06
Deconfounded ($D_Z = 20$)	3.54 ± 0.04	2.55 ± 0.05	11.57 ± 0.12	9.63 ± 0.14	2.40 ± 0.01	1.28 ± 0.08

Table 4.2: Average RMSE $\times 10^2$ and standard error for predicting the effect of antibiotics, vassopressors and mechanical ventilator on three patient covariates. The results are for 10 runs.

Time Series Deconfounder can be applied on a real dataset. From MIMIC III we extracted a dataset with 6256 patients for which there are three treatment options at each timestep: antibiotics, vasopressors, and mechanical ventilator (all of which can be applied simultaneously). These treatments are common in the ICU and are often used to treat patients with sepsis [166, 167]. For each patient, we extracted 25 patient covariates consisting of lab tests and vital signs measured over time that affect the assignment of treatments. We used daily aggregates of the patient covariates and treatments and patient trajectories of up to 50 timesteps. We estimate the effects of antibiotics, vasopressors, and mechanical ventilator on the following patient covariates: white blood cell count, blood pressure, and oxygen saturation.

Hidden confounding is present in the dataset as patient comorbidities and several lab tests were not included. However, since this is a real dataset, it is not possible to evaluate the extent of hidden confounding or to estimate the true (Oracle) treatment responses.

Table 4.2 illustrates the RMSE when estimating one-step ahead treatment responses by using the MSM and R-MSN outcome models directly on the extracted dataset (Confounded) and after applying the Time Series Deconfounder and augmenting the dataset with the substitutes for the hidden confounders of different dimensionality D_Z (Deconfounded). We notice that in all cases, the Time Series Deconfounder enables us to obtain a lower error when estimating the effect of antibiotics, vasopressors, and mechanical ventilator on the patients' white blood cell count, blood pressure, and oxygen saturation. By modeling the dependencies in the assigned treatments for each patient, the factor model part of the Time Series Deconfounder was able to infer latent variables that account for the unobserved information about the patient states. Using these substitutes for the hidden confounders in the outcome models resulted in better estimates of the

treatment responses. While these results on real data require further validation from doctors (which is outside the scope of this chapter), they indicate the potential of the method to be applied in real medical scenarios.

In Appendix B.2.5, we include results for an additional experimental set-up where we remove several patient covariates from the dataset and we show how the Time Series Deconfounder can be used to account for this bias. Moreover, in Appendix B.2.6, we provide further discussion and directions for future work.

4.2.8 Conclusion

The availability of observational data consisting of longitudinal information about patients prompted the development of methods for modeling the effects of treatments on the disease progression in patients. All existing methods for estimating the individualized effects of time-dependent treatment from observational data make the untestable assumption that there are no hidden confounders. In the longitudinal setting, this assumption is even more problematic than in the static setting. As the state of the patient changes over time and the complexity of the treatment assignments and responses increases, it becomes much easier to miss important confounding information.

In this chapter, we proposed the Time Series Deconfounder, a method that takes advantage of the patterns in the multiple treatment assignments over time to infer latent variables that can be used as substitutes for the hidden confounders. Moreover, we developed a deep learning architecture based on an RNN with multitask output and variational dropout for building a factor model over time and computing the latent variables in practice. Through experimental results on both synthetic and real datasets, we show the effectiveness of the Time Series Deconfounder in removing the bias from the estimation of treatment responses over time in the presence of multi-cause hidden confounders.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

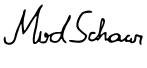
Title of Paper	Time Series Deconfounder: Estimating Treatment Effects over Time in the Presence of Hidden Confounders
Publication Status	☑ Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Ioana Bica, Ahmed M. Alaa, and Mihaela van der Schaar. "Time Series Deconfounder: Estimating Treatment Effects over Time in the Presence of Hidden Confounders". In *International Conference on Machine Learning (ICML)*, 2020.

Student Confirmation

Student Name:	Ioana Bica		
Contribution to the Paper	- I was the lead author responsible for conceptualizing the paper, developing and implementing the method, designing the experimental analysis, obtaining the results, structuring and writing the original draft. - All authors contributed to the development of the paper through multiple discussions. - All authors reviewed the final draft.		
Signature 		Date	30/03/2022

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Mihaela van der Schaar		
Supervisor comments I agree that the candidate has made a substantial contribution to the publication and that the above description is accurate.		
Signature 	Date	30/03/2022

This completed form should be included in the thesis, at the end of the relevant chapter.

5

Using Counterfactuals to Explain Sequential Decision-Making

Contents

5.1 Introduction 95
5.2 Related Works 98
5.3 Problem Formulation 99
5.4 Batch Inverse Reinforcement Learning using Counterfactuals . . 102
5.4.1 Counterfactual $\mu-$learning 103
5.4.2 Finding Optimal Policy for Given Reward Weights 104
5.4.3 Counterfactual Inverse Reinforcement Learning (CIRL) . . . 105
5.5 Experiments 106
5.5.1 Extracting Different Types of Expert Behaviour 107
5.5.2 Case Study on Real-world Dataset: MIMIC III 109
5.6 Discussion 110

Publication Included

- **Ioana Bica**, Daniel Jarrett, Alihan Hüyük and Mihaela van der Schaar. "Learning "What-if" Explanations for Sequential Decision-Making". In *International Conference on Learning Representations (ICLR)*, 2021. Link: https://openreview.net/pdf?id=h0de3QWtGG. [Accepted]

Abstract

Building interpretable parameterizations of real-world decision-making on the basis of demonstrated behavior—i.e. trajectories of observations and actions made by an expert maximizing some unknown reward function—is essential for introspecting and auditing policies in different institutions. In this chapter, we propose learning explanations of expert decisions by modeling their reward function in terms of preferences with respect to "what if" outcomes: Given the current history of observations, *what would happen if we took a particular action*? To learn these cost-benefit tradeoffs associated with the expert's actions, we integrate counterfactual reasoning into batch inverse reinforcement learning. This offers a principled way of defining reward functions and explaining expert behavior, and also satisfies the constraints of real-world decision-making—where active experimentation is often impossible (e.g. in healthcare). Additionally, by estimating the effects of different actions, counterfactuals readily tackle the *off-policy* nature of policy evaluation in the batch setting, and can naturally accommodate settings where the expert policies depend on *histories* of observations rather than just current states. Through illustrative experiments in both real and simulated medical environments, we highlight the effectiveness of our batch, counterfactual inverse reinforcement learning approach in recovering accurate and interpretable descriptions of behavior.

5.1 Introduction

Consider the problem of explaining sequential decision-making on the basis of demonstrated behavior. In healthcare, an important goal lies in being able to obtain an interpretable parameterization of the experts' behavior (e.g in terms of how they assign

treatments) such that we can quantify and inspect policies in different institutions and uncover the trade-offs and preferences associated with expert actions [168–171]. Moreover, modeling the reward function of different clinical practitioners can be revealing as to their tendencies to treat various diseases more/less aggressively [172], which —in combination with patient outcomes—has the potential to inform and update clinical guidelines.

In many settings, such as medicine, decision-makers can be modeled as reasoning about "what-if" patient outcomes: Given the available information about the patient, what would happen if we took a particular action? [52, 173]. As treatments often affect several patient covariates, by having both benefits and side-effects, decision-makers often make choices based on their preferences over these counterfactual outcomes. Thus, in our case, an interpretable explanation of a policy is one where the reward signal for (sequential) actions is parameterized on the basis of preferences over (sequential) counterfactuals (i.e. "what-if" patient outcomes).

Given the observations and actions made by an expert, *inverse reinforcement learning* (IRL) offers a principled way for modeling their behavior by recovering the (unknown) reward function being maximized [174–176]. Standard solutions operate by iterating on candidate reward functions, solving the associated (forward) reinforcement learning problem at each step. In many real-world problems, however, we are specifically interested in the challenge of offline learning—that is, where further experimentation is not possible—such as in medicine. In this *batch* setting, we only have access to trajectories sampled from the expert policy in the form of an observational dataset.

Batch IRL. By their nature, classic IRL algorithms require interactive access to the environment, or full knowledge of the environment's dynamics [174–176]. While batch IRL solutions have been proposed by way of off-policy evaluation [177–179], they suffer from two disadvantages. First, they are limited by the assumption that state dynamics are fully-observable and Markovian. This is hardly true in medicine: Treatment assignments generally depend on how patient covariates have evolved over time. Second, rewards are often parameterized as uninterpretable representations of neural network hidden states and consequently cannot be used to explain sequential decision making.

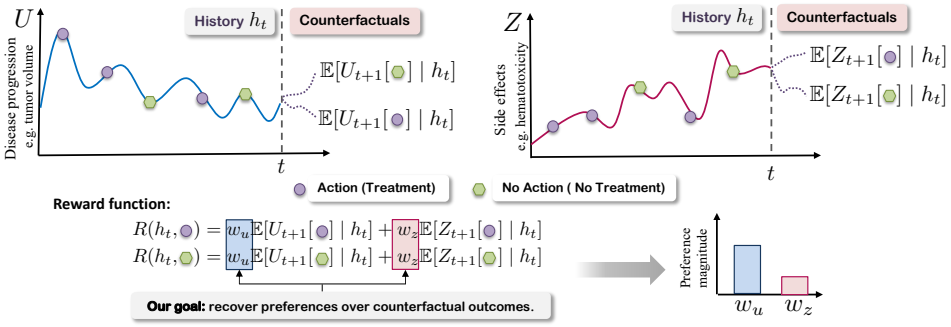


Figure 5.1: Explaining decision-making in terms of preferences over "what if" outcomes. Consider the evolution of tumour volume (U) and side effects (Z) under a binary action. $\mathbb{E}[U_{t+1}[a_t] \mid h_t]$ and $\mathbb{E}[Z_{t+1}[a_t] \mid h_t]$ are the counterfactuals for the patient features under action a_t given history h_t of prior actions and covariates. Parameterizing the reward as the weighted sum of these counterfactuals: $R(h_t, a_t) = w_u \mathbb{E}[U_{t+1}[a_t] \mid h_t] + w_z \mathbb{E}[Z_{t+1}[a_t] \mid h_t]$, allows us to model the preferences of experts: e.g. finding that $|w_u| > |w_z|$ indicates that the expert is treating more aggressively, by placing more weight on reducing tumour volume than on minimizing side effects.

"What-if" Explanations. To address these shortcomings and to obtain a parameterizable interpretation of the expert's behavior, we propose explicitly incorporating counterfactual reasoning into batch IRL. In particular, we focus on "what if" explanations for modeling decision-making, while simultaneously accounting for the partially-observable nature of patient histories. Under the max-margin apprenticeship framework [175, 177, 179], we learn a parameterized reward function $R(h_t, a_t)$ that is defined as a weighted sum over *potential outcomes* [180] for taking action a_t given history h_t.

As highlighted in Figure 5.1, consider the decision making for a binary action given the tumour volume (U) and side effects (Z). Let $\mathbb{E}[U_{t+1}[a_t] \mid h_t]$ and $\mathbb{E}[Z_{t+1}[a_t] \mid h_t]$ be the counterfactual outcomes for the two covariates when action a_t is taken given the history h_t of covariates and previous actions. We define the reward as the weighted sum of these counterfactuals: $R(h_t, a_t) = w_u \mathbb{E}[U_{t+1}[a_t] \mid h_t] + w_z \mathbb{E}[Z_{t+1}[a_t] \mid h_t]$, to take into account the effect of actions and to directly model the preferences of the expert. The ideal scenario is when both the tumour volume and the side effects are zero, so the reward weights of a doctor aiming for this are both negative. However, recovering that $|w_u| > |w_z|$, means that the doctor is treating more aggressively, as they are focusing more on reducing the tumour volume rather than on the side effects of treatments. Alternatively, $|w_u| < |w_z|$ indicates that the side effects are more important and the expert is treating less aggressively. Our motivation for using counterfactuals to define the reward comes from the idea that rational decision making considers the potential effects of actions [52].

Contributions. Exploring the synergy between counterfactual reasoning and batch IRL for understanding sequential decision making confers multiple advantages. First, it offers a principled approach for parameterizing reward functions in terms of preferences over *what-if* patient outcomes, which enables us to explain the cost-benefit tradeoffs associated with an expert's actions. Second, by estimating the effects of different actions, counterfactuals readily tackle the *off-policy* nature of policy evaluation in the batch setting. Furthermore, we demonstrate that not only does this alleviate the *cold-start* problem typical of conventional batch IRL solutions, but also accommodates settings where the usual assumption of full observability fails to hold. Through experiments in both real and simulated medical environments, we illustrate the effectiveness of our batch, counterfactual inverse reinforcement learning approach in recovering accurate and interpretable descriptions of behavior.

5.2 Related Works

In our work, the aim is to explain decision-making by recovering the preferences of experts with respect to the effects of their actions, denoted by the counterfactual outcomes. This goal is fundamentally different from the goal of IRL methods which generally aim to match the performance of experts. We operate under the standard max-margin apprenticeship framework [174, 175], which searches for a reward function that minimizes the margin between feature expectations of the expert and candidate policies. However, our approach to recovering and understanding decision policies is uniquely characterized by incorporating counterfactuals to obtain explainable reward functions. To tackle the challenges posed by real-world decision making, our method also operates in an offline and model-free manner, and accommodates partially-observable environments.

Explainability. By using basis functions [178] or hidden layers of a deep network [179] to define the feature map, the learned rewards of either approach is inherently uninterpretable, and cannot be used to explain differences in expert behavior. An alternative approach for recovering the expert policy (without reward functions) is imitation learning [66, 181–183]. However, these methods do not allow us to fully model the decision-making process of experts and to uncover the trade-offs behind their actions.

Method	Environment	Batch	Feature map for reward	Policy	Feat. expectations
[175]	Model-based	No	$\phi(s_t)$ = basis functions for state s_t	$\pi(a_t \mid s_t)$	Model roll-outs
[176]	Model-based	No	$\sum_s b_t(s)\phi(s, a_t)$ = basis for belief b_t	$\pi(a_t \mid b_t)$	Model roll-outs
[177]	Model-free	Yes	$\phi(x_t)$ = basis functions for state x_t	$\pi(a_t \mid x_t)$	LSTD-Q
[179]	Model-free	Yes	$\phi(x_t, a_t) = \text{concat}(\phi(x_t), a_t)$	$\pi(a_t \mid x_t)$	DSFN
Ours	Model-free	Yes	$\phi(h_t, a_t) = \mathbb{E}[Y_{t+1}[a_t] \mid h_t]$	$\pi(a_t \mid h_t)$	Counterfactual μ-learning

Table 5.1: Comparison of proposed method (batch, counterfactual IRL) with related works.

Batch Learning. [177] propose an off-policy evaluation method based on least squares temporal difference (LSTD-Q) [184] for estimating feature expectations, and [178] use a linear score-based classifier to directly approximate the Q-function offline. However, both methods require the constraining assumptions that rewards are direct, linear functions of fully-observable states—assumptions we cannot afford to make in realistic settings such as medicine. [179] propose a deep successor feature network (DSFN) based on Q-learning to estimate feature expectations. But their approach similarly assumes fully-observable states, and additionally suffers from the "cold-start" problem where off- policy evaluations are heavily biased unless the initial candidate policy is (already) close to the expert.

Partial Observability. No existing batch IRL method accommodates modeling expert policies that depend on patient histories. While [176] and [185] extend the apprenticeship learning paradigm to partially observable environments by considering policies on beliefs over states, both need to interact with the environment.

To the best of our knowledge, we are the first to propose explaining sequential decisions through counterfactual reasoning and to tackle the batch IRL problem in partially-observable environments. Our use of the estimated counterfactuals yields inherently interpretable rewards and simultaneously addresses the cold-start problem in [179]. Table 5.1 highlights the main differences between our method and the relevant related works. See Appendix C.1 for additional related works.

5.3 Problem Formulation

Preliminaries. At timestep t, let random variable $X_t \in \mathcal{X}$ denote the observed patient features and let $A_t \in \mathcal{A}$ denote the action (e.g. treatment) taken, where $\mathcal{A}$ is a finite set of actions. Let x_t and a_t denote realizations of these random variables. Let $h_t =$

$(x_0, a_0, \ldots, x_{t-1}, a_{t-1}, x_t) = (x_{0:t}, a_{0:t-1}) \in \mathcal{H}$ be a realization of the history $H_t \in \mathcal{H}$ of patient observations and actions until timestep t.

A stationary stochastic policy represents a mapping: $\pi : \mathcal{H} \times \mathcal{A} \to [0, 1]$, where $\pi(a \mid h)$ indicates the probability of choosing action $a \in \mathcal{A}$ given history $h \in \mathcal{H}$ and $\sum_{a \in \mathcal{A}} \pi(a \mid h) = 1$. Taking action a_t under history h_t results in observing x_{t+1} and obtaining h_{t+1}. The reward function is $R : \mathcal{H} \times \mathcal{A} \to \mathbb{R}$ where $R(h, a)$ represents the reward for taking action $a \in \mathcal{A}$ given history $h \in \mathcal{H}$. The value function of a policy π, $V : \mathcal{H} \to \mathbb{R}$ is defined as: $V^\pi(h) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(H_t, A_t) \mid \pi, H_0 = h]$, where $\gamma \in [0, 1)$ is the discount factor and $A_t \sim \pi(\cdot \mid H_t)$ for $t \geq 0$. The action-value function $Q : \mathcal{H} \times \mathcal{A} \to \mathbb{R}$ of a policy is defined as $Q^\pi(h, a) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(H_t, A_t) \mid \pi, H_0 = h, A_0 = a]$ where $A_t \sim \pi(\cdot \mid H_t)$ for $t \geq 0$. A higher Q-value indicates that action a will yield better long term returns if taken for history h. We assume we know the discount factor γ which indicates the importance of future rewards for the current history and action pair.

Batch IRL. Let $\mathcal{D} = \{\zeta^i\}_{i=1}^N$ be a batch observational dataset consisting of N patient trajectories: $\zeta^i = (x_0^i, a_0^i, \ldots x_{T^i-1}^i, a_{T^i-1}^i, x_{T^i}^i)$. The trajectory ζ^i for patient i consists of covariates x_t^i and actions a_t^i observed for T^i timesteps. For simplicity, we drop the superscript i unless explicitly needed. The actions $a_t \in \mathcal{D}$ are assigned according to some expert policy π_E such that $a_t \sim \pi_E(\cdot \mid h_t)$.

We work in the apprenticeship learning set-up [175] and we consider a linear reward function $R(h_t, a_t) = w \cdot \phi(h_t, a_t)$, where the weights $w \in \mathbb{R}^d$ satisfy $\|w\|_1 \leq 1$. The feature map $\phi : \mathcal{H} \times A \to \mathbb{R}^d$ also satisfies $\|\phi(\cdot)\|_2 \leq 1$ such that the reward is bounded. We assume that the expert policy π_E is attempting to optimize, without necessarily succeeding, some unknown reward function $R^*(h_t, a_t) = w^* \cdot \phi(h_t, a_t)$, where w^* are the 'true' reward weights. Given $R(h_t, a_t)$, the value of policy π can be re-written as: $\mathbb{E}[V^\pi(H_0)] = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t w \cdot \phi(H_t, A_t) \mid \pi] = w \cdot \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \phi(H_t, A_t) \mid \pi]$, where the expectation is taken with respect to the sequence of histories and action pairs $(H_t, A_t)_{t \geq 0}$ obtained by acting according to π. The feature expectation of policy π, defined as the expected discounted cumulative feature vector obtained when choosing actions according to policy π is $\mu^\pi = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \phi(H_t, A_t) \mid \pi] \in \mathbb{R}^d$ such that: $\mathbb{E}[V^\pi(H_0)] = w \cdot \mu^\pi$.

Our aim is to recover the expert weights w^* as well as find a policy π that is close to the policy of the expert π_E. We take the max-margin IRL approach and we measure the similarity between the feature expectations of the expert's policy and the feature expectations of a candidate policy using $\|\mu^{\pi_E} - \mu^{\pi}\|_2$. In this batch IRL setting, we do not have knowledge of transition dynamics and we cannot sample more trajectories from the environment. Note that in this context, we are the first to model expert policies that depend on patient histories and not just current observations.

Counterfactual reasoning. To explain the expert's behaviour in terms of their trade-off associated with "what if" outcomes, we use counterfactuals to define the feature map $\phi(h_t, a_t)$ part of the reward $R(h_t, a_t) = w \cdot \phi(h_t, a_t)$. We adopt the potential outcomes framework [79–81]. Let $Y[a]$ be the potential outcome, either factual or counterfactual, for treatment $a \in \mathcal{A}$. Using the dataset $\mathcal{D}$ we learn feature map $\phi(h_t, a_t)$ such that:

$$\phi(h_t, a_t) = \mathbb{E}[Y_{t+1}[a_t] \mid h_t], \tag{5.1}$$

where $\mathbb{E}[Y_{t+1}[a_t] \mid h_t]$ is the potential outcome for taking action a_t at time t given the history h_t. For the factual action a_t, assigned under policy $\pi(\cdot \mid h_t)$, the factual outcome is x_{t+1} and this is the same as the potential outcome $\mathbb{E}[Y_{t+1}[a_t] \mid h_t]$. The potential outcomes for the other actions $a_t \in \mathcal{A}$ are the counterfactual ones and they allow us to understand what would happen to the patient if they receive a different treatment a_t. To identify the potential outcomes from the batch data we make the standard assumptions of consistency, positivity and no hidden confounders as described in Appendix C.2. No hidden confounders means that we observe all variables affecting the action assignment and potential outcomes. Overlap means that at each timestep, every action has a non-zero probability and can be satisfied in this setting by having a stochastic expert policy. These assumptions are standard across methods for estimating counterfactual outcome [14, 40, 55]. Note that these assumptions are needed to be able to reliably perform causal inference using observational data. However, they do not constrain the batch IRL set-up.

Estimating the potential outcomes from batch data poses additional challenges that need to be considered. The fact that the expert follows policies that consider the history of patient observations when deciding new actions, gives rise to time-dependent confounding

bias. Standard supervised learning methods for learning $\mathbb{E}[Y_{t+1}[a_t] \mid h_t]$ from $\mathcal{D}$ will be biased by the expert policy used in the observational dataset and will not be able to correctly estimate the counterfactual outcomes under alternative policies [40]. Methods for adjusting for the confounding bias involve using either inverse probability of treatment weighting [14, 117] or building balancing representations [55]. Refer to Appendix C.2 for more details. In the sequel, we consider the model for estimating counterfactuals as a black box such that the feature map $\phi(h_t, a_t)$ represents the effect of taking action a_t for history h_t. The reward is then:

$$R(h_t, a_t) = w \cdot \phi(h_t, a_t) = w \cdot \mathbb{E}[Y_{t+1}[a_t] \mid h_t] \tag{5.2}$$

Defining the reward function using counterfactuals gives an interpretable parameterization of doctor behavior: It allows us to interpret their behavior with respect to the importance weights implicitly assigned to the effects of their actions. This enables describing the relative trade-offs in treatment decisions. Note that we are *not* assuming that the experts themselves actually compute these quantities (nor that they explicitly adopt the same causal inference assumptions); rather, we are simply providing a way to understand how decision-makers are effectively behaving (i.e. in terms of counterfactuals).

5.4 Batch Inverse Reinforcement Learning using Counterfactuals

Max-margin IRL [175] starts with an initial random policy π and iteratively perform the following three steps to recover the expert policy and its reward weighs: (1) estimate feature expectations of candidate policy π, (2) compute new reward weights w and (3) find new candidate policy π that is optimal for reward function $R(h_t, a_t) = w \cdot \phi(h_t, a_t)$. This approach finds a policy $\tilde{\pi}$ that satisfies $\|\mu^{\pi_e} - \mu^{\tilde{\pi}}\|_2 < \epsilon$ such that $\tilde{\pi}$ has an expected value function close the expert policy.

The expert feature expectations can be estimated empirically from the dataset $\mathcal{D}$ using:

$$\mu^{\pi_E} = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T^i} \gamma^t \phi(h_t^i, a_t^i). \tag{5.3}$$

In the batch setting, we cannot estimate the feature expectations of candidate policies by taking the sample mean of on-policy roll-outs: $\mu^{\tilde{\pi}} \neq \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T^i}\gamma^t\phi(h_t^i, \pi(h_t^i))$. To address this off-policy nature of estimating feature expectations, we introduce a new method that leverages the estimated counterfactuals. We also make use of the counterfactuals to learn optimal policies for different reward weights. Figure 5.2 illustrates how we integrate "what if" reasoning into batch IRL.

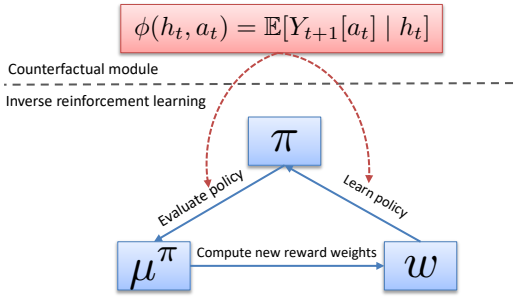


Figure 5.2: Counterfactual inverse reinforcement learning (CIRL). Counterfactuals are used to define $\phi(h, a)$, to estimate feature expectations μ^{π} of candidate policy π in batch setting and to learn optimal policies for reward weights w.

5.4.1 Counterfactual $\mu-$learning

Similar to the approach proposed by [178, 179], we consider a history-action feature expectation defined as follows $\mu^{\pi}(h, a) = \mathbb{E}[\sum_{t=0}^{\infty}\gamma^t\phi(H_t, A_t)|\pi, H_0 = h, A_0 = a]$, where the first action a can be chosen randomly and for $t \geq 1$, $A_t \sim \pi(\cdot \mid H_t)$. This can be re-written as:

$$\mu^{\pi}(h, a) = \phi(h, a) + \mathbb{E}_{h', a' \sim \pi(\cdot|h')}[\sum_{t=1}^{\infty}\gamma^t\phi(H_t, A_t) \mid \pi, H_1 = h', A_1 = a'] \quad (5.4)$$

$$= \phi(h, a) + \gamma\mathbb{E}_{h', a' \sim \pi(\cdot|h')}[\mu^{\pi}(h', a')], \quad (5.5)$$

where h' is the next history. Notice the analogy between $\mu^{\pi}(h, a)$ and the action-value function:

$$Q^{\pi}(h, a) = R(h, a) + \mathbb{E}_{h', a' \sim \pi(\cdot|h')}[\sum_{t=1}^{\infty}\gamma^t R(H_t, A_t) \mid \pi, H_1 = h', A_1 = a'] \quad (5.6)$$

$$= R(h, a) + \gamma\mathbb{E}_{h', a' \sim \pi(\cdot|h')}[Q^{\pi}(h', a')], \quad (5.7)$$

that allows us to use temporal difference learning to estimate feature expectations [186]. Existing methods for estimating feature expectations fall into two extremes: (1) model-based (online) IRL approaches learn a model of the world and then use the model as a simulator to obtain on-policy roll-puts [175] and (2) batch IRL approaches use Q-learning (or alternative methods) for off-policy evaluation [179], and can only be used to evaluate

policies similar to the expert policy and requires warm start. In our case, the counterfactual model allows us to compute $h' = (h, a, \mathbb{E}[Y(a)|h])$ for any $h \in \mathcal{D}$ and any arbitrary action a. Thus, we propose counterfactual μ-learning, a novel method for estimating feature expectations that uses these counterfactuals as part of temporal difference learning with 1-step bootstrapping. This approach falls in-between (1) and (2) and allows us to estimate feature expectations for any candidate policy π in the batch IRL setting.

Counterfactual μ-learning algorithm learns the μ-values for policy π iteratively by updating the current estimates of the μ-values with the feature map plus the μ-values obtained by following policy π in the new counterfactual history $h' = (h, a, \mathbb{E}[Y[a]|h])$:

$$\hat{\mu}^{\pi}(h, a) \leftarrow \hat{\mu}^{\pi}(h, a) + \alpha(\phi(h, a) + \gamma \mathbb{E}_{a' \sim \pi(\cdot|h')}[\hat{\mu}^{\pi}(h', a')] - \hat{\mu}^{\pi}(h, a)), \tag{5.8}$$

where α is the learning rate. We use a recurrent network with parameters θ to approximate $\hat{\mu}^{\pi}(h, a \mid \theta)$ and we train it by minimizing the sequence of loss functions $\mathcal{L}_i$ which changes for every iteration i:

$$\mathcal{L}_i(\theta_i) = \mathbb{E}_{h \sim \mathcal{D}}[||y_i - \hat{\mu}^{\pi}(h, a \mid \theta_i)||_2] \qquad \theta_{i+1} \leftarrow \theta_i + \alpha \nabla(\mathcal{L}_i(\theta_i)) \tag{5.9}$$

where a can be chosen randomly from $\mathcal{A}$ and $y_i = \phi(h, a) + \gamma \mathbb{E}_{a' \sim \pi(\cdot|h')}[\hat{\mu}^{\pi}(h', a' \mid \theta_{i-1})]$ is the target for iteration i. The parameters for the previous iteration θ_{i-1} are held fixed when optimizing $\mathcal{L}_i(\theta_i)$. Refer to Appendix C.3 for full details of the counterfactual μ-learning algorithm. The feature expectations for the policy π are given by $\hat{\mu}^{\pi} = \mathbb{E}_{H_0, A_0 \sim \pi(\cdot|H_0)}[\hat{\mu}^{\pi}(H_0, A_0)]$, which can be estimated empirically from the observational dataset $\mathcal{D}$ using $\hat{\mu}^{\pi} = \frac{1}{N} \sum_{i=1}^{N} \sum_{a \in \mathcal{A}} \hat{\mu}^{\pi}(h_0^i, a) \pi(a \mid h_0^i)$.

5.4.2 Finding Optimal Policy for Given Reward Weights

During each iteration of max-margin IRL, we obtain a candidate policy to evaluate by finding the optimal policy for a given vector of reward weights. We use deep recurrent Q-learning [187], a model-free approach for learning Q-values for reward function $R(h, a) = w \cdot \phi(h, a)$. The counterfactuals are used to compute $\phi(h, a)$, and to estimate the next history for the temporal difference updates. See Appendix C.4 for details. After estimating the $Q-$values, $Q(h, a)$ for history h and action a, a new candidate policy is obtained using: $\pi(a \mid h) = \mathbb{1}_{a = \arg\max_{a'} Q(h, a')}$.

Algorithm 1 (Batch, Max-Margin) CIRL

Input: Batch dataset $\mathcal{D}$, max iterations n, convergence
threshold ϵ, feature map $\phi(h_t, a_t) = \mathbb{E}[Y_{t+1}[a_t]|h_t]$
$\mu^{\pi_E} \leftarrow$ compute π_E's feature expectations (Equation 5.3)
$w_0 \leftarrow$ random initial reward weights, $\pi_0 \leftarrow$ compute optimal policy for $R_0 = w_0 \cdot \phi$
$\mu^{\pi_0} \leftarrow$ compute π_0's feature expectations (counterfactual μ-learning)
$\Pi = \{\pi_0\}, \Delta = \{\mu^{\pi_0}\}, \bar{\mu}_0 = \mu^{\pi_0}$
for $k = 1$ to n **do**
    $w_k = \mu_e - \bar{\mu}_{k-1}$, $\pi_k \leftarrow$ compute optimal policy for $R_k = w_k \cdot \phi$
    $\mu^{\pi_k} \leftarrow$ compute π_k's feature expectations (counterfacual μ-learning)
    $\Pi = \Pi \cup \{\pi_k\}, \Delta = \Delta \cup \{\mu^{\pi_k}\}$
    Orthogonally project μ^{π_E} onto line thru. $\bar{\mu}_{k-1}, \mu^{\pi_k}$:
$\bar{\mu}_k = \frac{(\mu^{\pi_k} - \bar{\mu}_{k-1})^T(\mu^{\pi_E} - \bar{\mu}_{k-1})}{(\mu^{\pi_k} - \bar{\mu}_{k-1})^T(\mu^{\pi_k} - \bar{\mu}_{k-1})}(\mu^{\pi_k} - \bar{\mu}_{k-1}) + \bar{\mu}_{k-1}, \qquad t = \|\mu^{\pi_E} - \bar{\mu}_k\|_2$
    if $t < \epsilon$ **then** break
$K = \arg\min_{k:\mu^{\pi_k} \in \Delta} \|\mu^{\pi_E} - \mu^{\pi_k}\|_2$, $\tilde{R}(h, a) = w_K \cdot \phi(h, a)$

Output: $\tilde{R}$, Δ, Π

5.4.3 Counterfactual Inverse Reinforcement Learning (CIRL)

Algorithm 1 describes our proposed counterfactual inverse reinforcement learning (CIRL) method for the batch setting. CIRL is based on the projection algorithm proposed by [175] and iteratively updates the reward weights to minimize the margin between the expert's feature expectations and the feature expectations of intermediate policies. CIRL uses our proposed counterfactual $\mu-$learning algorithm for estimating the feature expectations of intermediate policies in an off-policy manner that is suitable for the batch setting. Compared to the algorithm for batch IRL proposed by [179] that requires the initial policy π_0 to already be similar to the expert policy, CIRL works for any initial policy π that is optimal for the randomly initialized reward weights w_0.

Similarly to [176], the CIRL algorithm returns the reward function $\tilde{R}(h, a)$ that results in a policy with feature expectations closest to the ones of the expert policy. We show experimentally that the reward that yields the closest feature expectations will be similar to the true underlying reward function of the expert. CIRL returns the set of policies tried Π and their feature expectations Δ, which allows us to compute a mixing policy that would yield a similar performance to the expert policy [175]. Let $\tilde{\mu}$ be the closest

point to μ^{π_E} in to convex closure of $\Delta = \{\mu^{\pi_0}, \mu^{\pi_1} \dots \mu^{\pi_k}\}$, which can be computed by solving the quadratic programming problem:

$$\min \|\mu^{\pi_e} - \mu\|_2 \text{ s.t. } \mu = \sum_i \lambda_i \mu^{\pi_i}, \lambda_i \geq 0, \sum_i \lambda_i = 1. \tag{5.10}$$

From the termination criteria of Algorithm 1, μ^{π_E} is separated from the points μ^{π_i} by a margin of at most ϵ. Thus, the solution $\tilde{\mu}$ will satisfy $\|\mu^{\pi_E} - \tilde{\mu}\|_2 \leq \epsilon$. To obtain a policy that is close to the performance of the expert policy, we mix together the policies $\Pi = \{\pi_0, \dots \pi_k\}$ returned by Algorithm 1, where the probability of selecting π_i is λ_i.

5.5 Experiments

We evaluate the ability of CIRL to recover the preferences of experts over the "what if" outcomes of actions. These preferences are denoted by the magnitude of the recovered reward weights. Since we do not have access to the underlying reward weights in real data, we first validate the method in a simulated environment. To show the applicability of CIRL in healthcare, we also perform a case study on an ICU dataset [148].

Benchmarks. For our proposed CIRL method, we use the Counterfactual Recurrent Network, a state-of-the-art model for estimating counterfactual outcomes in a temporal setting [55]. Note that other models for this task are also applicable [117]. Refer to Appendix C.6 for details. We benchmark CIRL against MB-IRL: model-based IRL—i.e. inverse reinforcement learning with model-based policy evaluation (e.g. [188–191]). We consider two versions of this benchmark: MB(h)-IRL which uses the patient history and MB(x)-IRL which only uses the current observations to define the policy. For both MB(x)-IRL and MB(h)-IRL we define the reward as a weighted sum of counterfactual outcomes, but these methods use instead standard supervised methods to estimate the next history h' needed for the counterfactual μ-learning algorithm. These benchmarks are aimed to highlight the need of handling the bias from the time-dependent confounders and using a suitable counterfactual model, but also the importance of handling the patient history. We also compare against the deep successor feature networks (DSFN) proposed by [179], which currently represents the state-of-the-art Batch IRL for the MDP setting. To show that their approach for estimating feature expectations in the batch setting

is suboptimal, we extend their method to also incorporate histories in the DSFN(h) benchmark. Implementation details of the benchmarks can be found in Appendix C.7.

5.5.1 Extracting Different Types of Expert Behaviour

Simulated environment. We propose an environment that uses a general data simulation involving p-order auto-regressive processes. To analyze different types of expert behaviour (e.g. treating more/less aggressively) we simulate data for patient features representing disease progression (x), e.g. tumour volume and side effects (z) and action (a) indicating the binary application of treatment. For time t, we model the evolution of patient covariates according to the treatments as follows:

$$x_t = \frac{1}{p}\sum_{i=1}^{p} x_{t-i} - 2.5\sum_{i=1}^{p} a_{t-i} + 0.5p + \epsilon \qquad z_t = \frac{1}{p}\sum_{i=1}^{p} z_{t-i} + 0.5\sum_{i=1}^{p} a_{t-i} - p + \eta \tag{5.11}$$

where $p = 5$ and $\epsilon \sim \mathcal{N}(0, 0.1^2)$, $\eta \sim \mathcal{N}(0, 0.1^2)$ are noise terms. The initial values are sampled as follows: $x_0 \sim \mathcal{N}(30, 5)$ and $z_0 \sim \mathcal{N}(2, 1)$. We set $x_{max} = 50$ and $z_{max} = 15$. The trajectory terminates when either $x_t = 0$, $x_t \geq x_{max}$, $z_t \geq z_{max}$ or $t \geq 20$. The tumour volume x_t, denoting the disease progression, decreases when we give treatment and increases otherwise. Conversely, the side effects z_t, increase when we give treatment and decrease otherwise. We define a linear reward for taking action a_t given history $h_t = (x_{0:t}, z_{0:t}, a_{0:t-1})$:

$$R(h_t, a_t) = w_1 \frac{x_{t+1}}{x_{max} - x_{min}} + w_2 \frac{z_{t+1}}{z_{max} - z_{min}} \tag{5.12}$$

where $w = [w_1, w_2]$, $||w||_1 \leq 1$ and x_{t+1} and z_{t+1} are simulated according to equations 5.11 to take into account the effect of action a_t for history h_t. The features are normalized to $[0, 1]$. The best scenario for a patient is when both the side effects and tumour volume are zero and a doctor attempting to achieve this will have negative reward weights. However, different settings of the reward weights will result in different expert behaviours. For instance, for $w_1 = -0.7$ and $w_2 = -0.3$, the expert policy will focus more on the disease progression and will consequently treat more aggressively, while this behaviour will be reversed for reward weights set to $w_1 = -0.3$ and $w_2 = -0.7$. We used deep

recurrent $Q-$learning [187] to find a stochastic expert policy that optimizes the reward function for different settings of the weights. The batch dataset $\mathcal{D}$ consists of 10000 trajectories sampled from the expert policy. Refer to Appendix E for more details.

Recovering decision making preferences of experts. We first evaluate the benchmarks on their ability to recover the weights of the reward function optimized by the expert for the experimental setting with $\gamma = 0.99$, $w_1 = -0.3$ and $w_2 = -0.7$. Note that DSFN does not provide interpretable reward weights, and thus cannot be used for understanding the trade-offs in the expert behavior. We train each benchmark 10 times and we plot in Figure 5.3 the reward weights obtained for the different iterations. We show that CIRL performs best at recovering the preferences of the expert, which in this case is to treat less aggressively. While MB(h)-IRL also recovers the correct trade-offs in the expert behavior, the computed weights have a much higher variance. Conversely, MB(x)-IRL, which does not consider the patient history fails to recover the underlying weights of the expert policy.

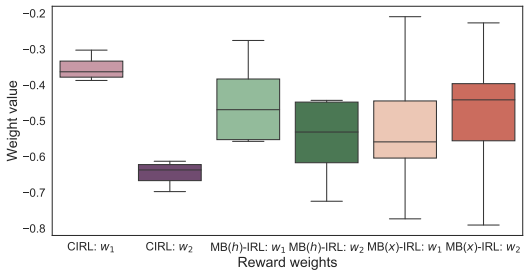


Figure 5.3: Reward weights recovered by benchmarks over 10 runs. The weights of the expert are $w_1 = -0.3$ and $w_2 = -0.7$.

Matching the expert policy. We evaluate the benchmarks' ability to recover policies that match the performance of the expert policy for two settings of the discount factor $\gamma \in \{0.99, 0.5\}$. A lower γ indicates that the expert is optimizing for the immediate effect of actions, while a higher γ means they considered the long term effect of actions. For each γ we learn expert policies for different reward weights and we use the expert policies to generate the batch datasets. We evaluate the policies learned by the benchmarks using two metrics: cumulative reward for running the policy in the simulated environment and accuracy on matching the expert policy. We report in Tables 5.2 and 5.3 the average results and their standard error over 1000 sampled trajectories from the environment. CIRL recovers a policy that has the closest cumulative reward to the expert policy and that can best match the treatments assigned by the expert.

	$\gamma = 0.99$			$\gamma = 0.5$		
Reward weights	$w_1 = -0.3$ $w_2 = -0.7$	$w_1 = -0.7$ $w_2 = -0.3$	$w_1 = -0.5$ $w_2 = -0.5$	$w_1 = -0.3$ $w_2 = -0.7$	$w_1 = -0.7$ $w_2 = -0.3$	$w_1 = -0.5$ $w_2 = -0.5$
MB(x)-IRL	-3.78 ± 0.02	-4.42 ± 0.05	-4.90 ± 0.05	-4.51 ± 0.05	-4.53 ± 0.05	-4.54 ± 0.04
MB(h)-IRL	-3.23 ± 0.02	-4.10 ± 0.02	-4.63 ± 0.04	-4.43 ± 0.04	-3.54 ± 0.05	-4.35 ± 0.03
DSFN	-3.56 ± 0.06	-4.32 ± 0.04	-3.77 ± 0.05	-4.11 ± 0.06	-3.07 ± 0.03	-4.67 ± 0.07
DSFN(h)	-3.31 ± 0.07	-4.33 ± 0.07	-3.60 ± 0.07	-3.95 ± 0.07	-3.05 ± 0.05	-4.61 ± 0.05
CIRL	$\mathbf{-2.89 \pm 0.02}$	$\mathbf{-3.92 \pm 0.03}$	$\mathbf{-3.41 \pm 0.05}$	$\mathbf{-2.79 \pm 0.02}$	$\mathbf{-2.91 \pm 0.02}$	$\mathbf{-4.27 \pm 0.03}$
Expert	-2.72 ± 0.02	-3.61 ± 0.02	-2.81 ± 0.01	-2.65 ± 0.02	-2.36 ± 0.01	-3.97 ± 0.03

Table 5.2: Mean reward and standard deviation for running learnt policy in the environment.

	$\gamma = 0.99$			$\gamma = 0.5$		
Reward weights	$w_1 = -0.3$ $w_2 = -0.7$	$w_1 = -0.7$ $w_2 = -0.3$	$w_1 = -0.5$ $w_2 = -0.5$	$w_1 = -0.3$ $w_2 = -0.7$	$w_1 = -0.7$ $w_2 = -0.3$	$w_1 = -0.5$ $w_2 = -0.5$
MB(x)-IRL	$62.5 \pm 0.41\%$	$61.4 \pm 0.81\%$	$54.6 \pm 0.56\%$	$52.4 \pm 0.63\%$	$60.1 \pm 0.39\%$	$71.8 \pm 0.72\%$
MB(h)-IRL	$77.8 \pm 0.31\%$	$70.2 \pm 0.45\%$	$71.4 \pm 0.69\%$	$66.3 \pm 0.58\%$	$70.2 \pm 0.71\%$	$75.6 \pm 0.52\%$
DSFN	$75.4 \pm 0.32\%$	$68.4 \pm 0.21\%$	$73.4 \pm 0.45\%$	$80.2 \pm 0.37\%$	$70.8 \pm 0.24\%$	$69.8 \pm 0.44\%$
DSFN(h)	$76.3 \pm 0.37\%$	$67.5 \pm 0.32\%$	$80.6 \pm 0.54\%$	$80.4 \pm 0.56\%$	$71.0 \pm 0.35\%$	$70.2 \pm 0.47\%$
CIRL	$\mathbf{81.8 \pm 0.42\%}$	$\mathbf{75.5 \pm 0.51\%}$	$\mathbf{83.7 \pm 0.76\%}$	$\mathbf{89.5 \pm 0.37\%}$	$\mathbf{73.2 \pm 0.43\%}$	$\mathbf{80.4 \pm 0.42\%}$

Table 5.3: Average accuracy and standard deviation for matching the actions in the expert policy.

5.5.2 Case Study on Real-world Dataset: MIMIC III

Suppose we want to explain the decision-making process of doctors assigning antibiotics to patients in the ICU. For this purpose, we consider a dataset with 6631 patients that have received antibiotics during their ICU stay, extracted from the Medical Information Mart for Intensive Care (MIMIC III) database [148]. We used CIRL, with $\gamma = 0.99$, to recover the policy and the reward weights of doctors administering antibiotics to understand their preferences over the effect of antibiotics on the patient features. The relative magnitude of the reward weights for the counterfactual outcomes of the patient features considered are illustrated in Figure 5.4. CIRL found that reducing temperature had the highest weight in the reward function of the expert, followed by WBC. This corresponds to known medical guidelines [192, 193]. Sepsis is a leading cause of morbidity and mortality in the ICU, and several studies have shown that early administration of antibiotics is crucial to decreasing the risk of adverse outcomes [194]. Fever and elevated WBC are among a small subset of variables that indicate a systemic inflammatory response, concerning the development of sepsis [195]. While these findings are not specific to bacterial infection, the risk of failing to treat a potentially serious infection often outweighs the risk of inappropriate antibiotic

administration, thereby driving clinicians to prescribe antibiotics in the setting of these abnormal findings. Similarly, the decision to discontinue antibiotics is complex, but it is often supported by signs of a resolution of infection which included normalization of body temperature and a downtrending of the WBC. As such, our finding that the two highest reward weights for the administration of antibiotics in the ICU are temperature and WBC is consistent with clinical practice. Moreover, note that the model is simply identifying the factors that are driving the decision-making of the clinician represented in this dataset.

In Table 5.4, to verify that explainability does not come at the cost of accuracy we also evaluate the benchmarks on matching the expert actions.

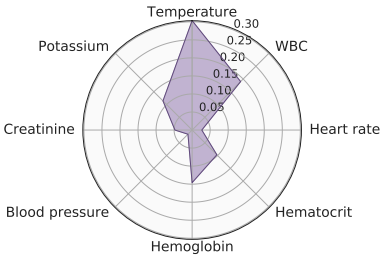


Figure 5.4: Radar plot of reward weights magnitude for assigning antibiotics.

	Accuracy
MB(x)-IRL	$70.1 \pm 0.11\%$
MB(h)-IRL	$77.5 \pm 0.15\%$
DSFN	$73.5 \pm 0.25\%$
DSFN(h)	$75.3 \pm 0.19\%$
CIRL	$\mathbf{83.4 \pm 0.17}\%$

Table 5.4: Accuracy on matching expert actions.

5.6 Discussion

In this chapter, we propose building interpretable parametrizations of sequential decision-making by explaining an expert's behaviour in terms of their preferences over "what-if" outcomes. To achieve this, we introduce CIRL, a new method that incorporates counterfactual reasoning into batch IRL: counterfactuals are used to define the feature map part of the reward function, but also to tackle the off-policy nature of estimating feature expectations in the batch setting. The reward weights recovered by CIRL indicate the relative preferences of the expert over the counterfactual outcomes of their actions. Note that our aim is to provide a description of behavior, i.e. how the expert is effectively behaving under our interpretable parameterization of the reward based on counterfactuals. We are not assuming that the experts actually operate under this specific (linear, in our case) model or that they compute the exact counterfactuals. Instead, our purpose is to show that we can explain an agent's behavior on the basis of counterfactuals, which is useful in that it allows us to audit them, sanity-check their policies and find variation in practice.

There are several limitations of your method and directions for future work. We illustrate in Figure 5.5 how the different components in our method interact and which are their limitations. While our method considers reward functions that are linear in the features, one way of extending it to handle more complex reward functions is to use domain knowledge to define the feature map as a non-linear function over the counterfactual outcomes. Nevertheless, these functions should be defined in a way that still allows us to obtain interpretable explanations of the expert's behaviour. Moreover, although time-invariant reward weights w are standard in the IRL literature [175, 176, 179], time-variant rewards/policies have been considered in the dynamic treatment regimes (reinforcement learning) literature [196, 197]. Thus another direction for future work would be to extend our method to consider non-stationary policies and reward weights that can change over time.

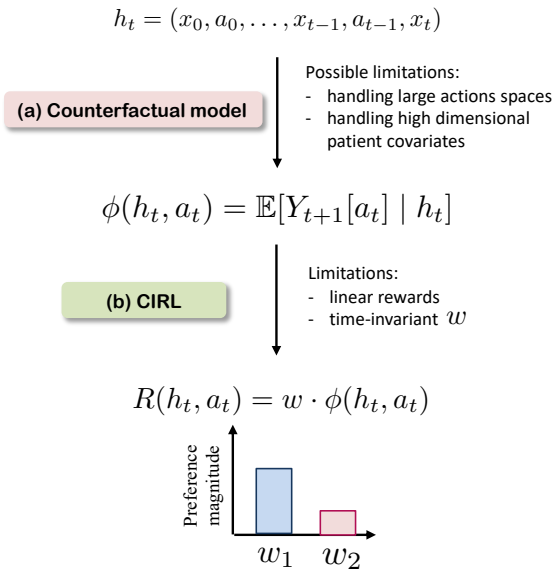


Figure 5.5: The two different components part of our method for obtaining interpretable parametrizations of decision making: (a) Counterfactual model which estimates the potential outcomes for taking action a_t given patient history h_t. Note that we do not propose a new counterfactual estimation algorithm, but instead use already existing methods which could have limitations in terms of how they handle large action spaces and large patient covariates. (b) CIRL algorithm which represents the novelty of this work. We propose incorporating the estimated counterfactuals as part of a method for batch IRL that can recover the preferences of the expert over the "what if" patient outcomes. As part of the limitations of CIRL are the use of linear rewards and time-invariant reward weights w.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Learning "What-if" Explanations for Sequential Decision-Making
Publication Status	☑ Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Ioana Bica, Daniel Jarrett, Alihan Huyuk, and Mihaela van der Schaar. "Learning "What-if" Explanations for Sequential Decision-Making". In *International Conference on Learning Representation (ICLR)*, 2021.

Student Confirmation

Student Name:	Ioana Bica		
Contribution to the Paper	- I was the lead author responsible for conceptualizing the paper, developing and implementing the method, designing the experimental analysis, obtaining the results, structuring and writing the original draft. - All authors contributed to the development of the paper through multiple discussions. - All authors reviewed the final draft.		
Signature 		Date	30/03/2022

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Mihaela van der Schaar		
Supervisor comments I agree that the candidate has made a substantial contribution to the publication and that the above description is accurate.		
Signature 	Date	30/03/2022

This completed form should be included in the thesis, at the end of the relevant chapter.

6

Using Causal Structure to Improve the Robustness of Decision-Making

Contents

6.1 Introduction 114
6.2 Related Works 117
6.3 Problem Formalism 119
6.3.1 Imitation Learning 119
6.3.2 Imitation Learning from Multiple Environments 120
6.4 Invariant Causal Imitation Learning for Generalization across Domains 123
6.4.1 Learning Invariant Causal Representations 124
6.4.2 Matching Expert Behaviour in a Strictly Batch Setting 126
6.5 Experiments 127
6.5.1 Evaluation on OpenAI Gym 128
6.5.2 Evaluation on MIMIC-III 129
6.6 Discussion 131

Publication Included

- **Ioana Bica**⋆, Daniel Jarrett⋆, and Mihaela van der Schaar. "Invariant Causal Imitation Learning for Generalizable Policies". In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. Link: https://openreview.net/forum?id=715E7e6j4gU. [Accepted]

Abstract

Consider learning an imitation policy on the basis of demonstrated behavior from multiple environments, with an eye towards deployment in an unseen environment. Since the observable features from each setting may be different, directly learning individual policies as mappings from features to actions is prone to *spurious correlations*—and may not generalize well. However, the expert's policy is often a function of a shared *latent structure* underlying those observable features that is invariant across settings. By leveraging data from multiple environments, we propose *Invariant Causal Imitation Learning* (ICIL), a novel technique in which we learn a feature representation that is invariant across domains, on the basis of which we learn an imitation policy that matches expert behavior. To cope with transition dynamics mismatch, ICIL learns a *shared* representation of causal features (for all training environments), that is independent from the *specific* representations of noise variables (for each of those environments). Moreover, to ensure that the learned policy matches the observation distribution of the expert's policy, ICIL estimates the energy of the expert's observations and uses a regularization term that minimizes the imitator policy's next state energy. Experimentally, we compare our method against several benchmarks in control and healthcare tasks and show its effectiveness in learning imitation policies capable of generalizing to unseen environments.

6.1 Introduction

Strictly batch imitation learning aims to learn a policy that directly mimics the behaviour of experts, for which we only have access to a set of demonstrations: logged trajectories of observations and actions following the expert's policy [66, 198, 199]. We cannot

interact online with the environment, let alone query the expert any further, nor do we have reward signals for supervision. This setting is relevant in real-world scenarios where live experimentation is risky or costly—such as healthcare and education.

Our aim is to learn an imitation policy in the strictly batch setting that faithfully matches the expert behaviour, while at the same time is able to generalize to unseen environments. In healthcare, learning a generalizable behaviour policy that could achieve expert performance in new environments is an important goal: As a means of providing clinical decision support, it could serve as an "individualized" clinical guideline for actions that can be taken for different patients—especially in a hospital, region, or patient demographic from which we have no access to data during training. In this endeavor, a principal challenge is that the sets of expert demonstrations that we have access to may contain variables that induce selection bias, or are otherwise spuriously correlated with the expert's actions [48, 200–202]. Directly learning an imitation policy from such data may lead to learning those spurious associations, thereby failing to generalize to unseen environments, and perpetuating any biases in the expert's behaviour.

However, in general it is likely that the expert's actions are only causally affected by a subset of the observed variables or by a shared latent structure [203, 204]. For instance, when imitating ideal driving behaviour, the background scenery might change, but the actions should only depend on car and road features. Another example includes the case when the lightning conditions in a room are changing, but physical dynamics of the environment are staying the same [202]. By leveraging expert trajectories from multiple different environments, our aim is to uncover this shared latent structure that causally determines expert actions, which allows us to eliminate the spurious associations and biases. In this way, the learnt policy will better be able to generalize to any unseen environments that share the same latent structure as those used for training.

As illustrated in Figure 6.1, we assume access to observations and actions from the expert's policy in the different environments e. The observations are functions of noise factors η^e (which may differ across environments) and shared latent state representations s (which is invariant across environments)—that encapsulate the causal parents of the expert's actions. Note that the observed features for an environment may simply be the

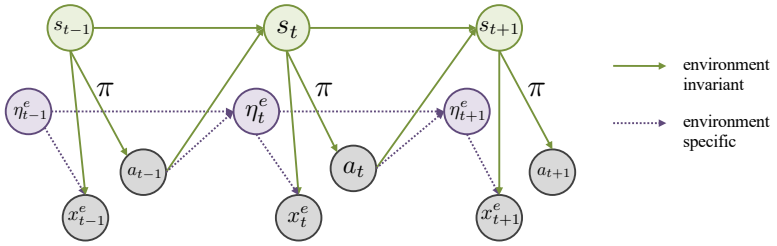


Figure 6.1: Causal diagram for the structure of environments. Expert demonstrations contain information about observations x_t and actions a_t. We assume that observations are decomposable into (1) state representations s_t that consist of the causal parents of the actions, and (2) noise representations η_t that encapsulate any spurious correlations with the actions. To allow for dynamics mismatch, the transitions between the noise representations are specific to each environment. We want to recover the invariant state representation s_t such that the learned policy $\pi(\cdot|s_t)$ generalizes well to new environments.

union of η^e and s, but they may also be any non-linear transformation of them. We shall operate in the setting where there are no hidden confounders, i.e. that we observe all variables that are affecting the expert's actions (and the next states that result from these actions). In addition to spurious correlations, another difficulty stems from learning to imitate sequential behavior in the strictly batch setting itself: While behaviour cloning [205] provides an intrinsically batch solution, it ignores important information contained in the expert's roll-out distribution, and the learned policy may drift from the support of the distribution of states visited by the expert [206, 207].

Contributions: In this chapter, we introduce *Invariant Causal Imitation Learning* (ICIL), a novel method that learns a causal representation of the expert's actions—which is used to build a generalizable imitation policy that matches the expert's behaviour. ICIL operates in the strictly batch setting and does not assume access to data from the target environments. By leveraging expert demonstrations from multiple different training environments, ICIL learns a (shared) invariant causal representation as well as an (environment-specific) noise representation. This accommodates dynamics mismatch across environments, while allowing the imitation policy to be learned by conditioning on the invariant causal representations. First, to satisfy the causal relationships in Figure 6.1, ICIL learns dynamics preserving representations and ensures that the learnt causal and noise representations are marginally independent by minimizing their mutual information. Second, to encourage the learnt imitation policy to stay within the support of the distribution of states visited by the expert's policy, ICIL estimates the energy of the expert's

observations and uses a regularization term that minimizes the imitator policy's next state energy. Third, we evaluate ICIL against benchmarks for batch imitation learning in control and healthcare environments. We also empirically investigate directly using ideas from invariant risk minimization [48] to augment the loss function of existing batch imitation learning methods, and benchmark against their ability to generalize across environments.

6.2 Related Works

We tackle the problem of learning generalizable policies in an offline setting using ideas from causal inference. As such, our work straddles the intersection of research in (1) strictly batch imitation, (2) invariant representation learning, and—more broadly—(3) causality in sequential decision-making.

Strictly Batch Imitation Learning: The simplest approach to imitation learning in the batch setting is behaviour cloning [205] which uses standard supervised learning techniques to learn an imitation policy that minimizes the negative log-likelihood of the observed demonstrator actions. However, behaviour cloning suffers from distributional shift as the learnt imitation policy cannot recover if it reaches a state out-of the distribution of the expert demonstrations [206–209]. To overcome this problem, [198, 209] propose incorporating dynamics-awareness by adding regularization to behaviour cloning by using norm-based penalties on the sparsity of implied rewards. Alternatively, [199] uses a distribution matching approach and proposes an offline objective for estimating the distribution ratio of the imitator policy and the expert policy, while [66] jointly learns a policy function together with an energy-based model of the state distribution. However, none of the existing approaches considers the problem of *generalization* across environments and learning policies robust to spurious correlations.

Invariant Risk Minimization: In the supervised learning setting, Invariant Risk Minimization (IRM) [48] leverages data from multiple domains to learn a data representation that elicits an invariant predictor across the different environments. The training data from each environment corresponds to different interventions on the data generating process. Given data from several training environments, the IRM objective aims to find a representation such that there exists a classifier that is optimal across all training domains,

	Method	Environment	Offline	Dynamics mismatch	Sensory-shift (hidden confounders)	State-distribution matching	No access to target trajectories	Temporal aspect
Imitation Learning	Pomerleau [205]	Model-free	Yes	No	No	No	N/A	No
	Ho & Ermon [210]	Model-free	No	No	No	Model rollouts	N/A	Yes
	Kostrikov et al. [199]	Model-free	Yes	No	No	Adversarial off-policy matching	N/A	Yes
	de Haan *et al.* [203]	Model-free	No	No	No	No	N/A	Yes
Generaliz. in IL	Lu *et al.* [211]	Model-free	No	Yes	No	Model rollouts	No	Yes
	Kim *et al.* [212]	Model-based	No	Yes	No	Model rollouts	No	Yes
	Etsami *et al.* [213]	Model-free	Yes	No	Yes	No	No	Yes
IRM	Arjovsky *et al.*[48]	Model-free	Yes	N/A	N/A	N/A	Yes	No
ICIL (Ours)		Model-based	**Yes**	**Yes**	No	Energy-based	**Yes**	**Yes**

Table 6.1: Comparison of ICIL with related works. ICIL operates in the strictly batch setting, allows for dynamics mismatch, does not require access to target trajectories, and incentivizes the imitation policy to stay in the support of the expert's distribution via energy-based regularization.

i.e. that minimizes the empirical risk in each domain. This represents a challenging, bi-level optimization, and [48] proposes the IRM-v1 objective which is a practical version to optimize. Through this optimization, the IRM objective should learn a predictor that only uses the causal parents of the target variable and that is thus invariant across environments. However, directly using IRM for our *sequential* problem setting is not desirable, since it does not take into account the effect of each action on the subsequent states. Nonetheless, we empirically investigate augmenting existing methods for batch imitation to use the IRM-v1 objective in conjunction with their defined imitation risk, and verify whether they are able to generalize across environments. In our experiments, we observe that, in general, directly applying the IRM objective in this manner is not good enough.

Generalization in Imitation Learning: The problem of domain adaptation and transfer learning for the imitation learning setting has been tackled by several works so far. However, while they consider problems of dynamics-, embodiment-, and/or viewpoint-mismatch between the imitator and expert, existing methods assume access to demonstrations from the target environment [212, 214, 215], assume access to online interaction or simulators in the different environments [211], or focus on the different problem of hidden confounding [46, 213]. Another line of work that is related is learning from demonstrations and meta-learning. While works in meta-learning also aim to generalize learnt policies to new tasks, they require access to one or more expert trajectories from the new task [216–221].

Causality in Imitation Learning and Reinforcement Learning: Several ideas from causality have been used to improve imitation learning and generalization in reinforcement

learning. The idea of conditioning the imitation policy on the causal parents has been employed by [203] to avoid the problem of 'causal confusion' when learning a policy for the single environment setting. However, [203] requires queering the expert or being able to perform interventions in the environment this is not possible in the batch setting. Similarly, [204, 222] also learn causal relationships between the observations, actions and rewards by performing/simulating the effect of interventions in the environment. Alternatively, [223] use ideas from Invariant Risk Minimization [48, 224] to learn optimal reinforcement learning policies that generalize across domains. Perhaps the most similar setting to ours is the one in [202] which studies the problem of generalization in reinforcement learning and also learn a representation that is shared across the domains. However, unlike our imitation setting, they assume access to a *known* reward signal and focus on learning the causal ancestors of that reward to improve reinforcement learning [225].

To the best of our knowledge, we are the first to tackle the problem of learning generalizable imitation policies in the strictly batch setting. Table 6.1 summarizes main differences with relevant related works.

6.3 Problem Formalism

6.3.1 Imitation Learning

We work in the standard Markov decision process (MDP) setting: Let an environment be given by $e = (\mathcal{X}, \mathcal{A}, T, r, \gamma)$, with observations $x \in \mathcal{X}$, actions $a \in \mathcal{A}$, transition function $T \in \Delta(\mathcal{X})^{\mathcal{X}\times\mathcal{A}}$, reward function $r \in \mathbb{R}^{\mathcal{X}\times\mathcal{A}}$, and discount factor γ. Let $\pi \in \Delta(\mathcal{A})^{\mathcal{X}}$ be a policy with the induced occupancy measure $\rho_\pi(x,a) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t p(x_t = x, a_t = a | x_t \sim T(\cdot|x_{t-1}, a_{t-1}), a_t \sim \pi(\cdot|x_t))$ of observations and actions, and let $\rho_\pi(x) = \sum_{a\in\mathcal{A}}\rho_\pi(x,a)$ be the observation occupancy measure.

Unlike in the reinforcement learning setting, where the aim is to learn a policy $\pi(\cdot \mid x)$ that maximizes the cumulative sum of some known reward signal, in imitation learning the reward is neither known nor observed. Instead, we only have access to a dataset of trajectories $\mathcal{D} = \{\tau_i\}_{i=1}^N$ from a demonstrator policy π_D, where each

trajectory $\tau \sim \pi_D = (x_t, a_t, x_{t+1})_{t=0,\dots}$ consists of a sequence of observation, action, next observation tuples that are sampled as $a_t \sim \pi_D(\cdot \mid x_t)$ and $x_{t+1} \sim T(\cdot \mid x_t, a_t)$.

The goal of imitation learning is to seek an imitation policy π that minimizes the following risk:

$$R(\pi) = \mathcal{L}(\pi, \pi_D) \tag{6.1}$$

where $\mathcal{L}$ is a choice of loss function. Now, if we were in the online setting, we would have access to the environment (or a simulator), with which we can interactively perform distribution matching by minimizing the divergence between the expert's state occupancy ρ_D and the imitator's state occupancy ρ_π [210, 226–228]. One example is to use the (forward) KL divergence: $\mathcal{L}(\pi, \pi_D) = D_{KL}(\rho_D || \rho_\pi)$ [228]. However, in the offline setting we have no further access to the environment. As noted above, the simplest solution is behaviour cloning (BC) [205, 229, 230], which minimizes the negative log-likelihood of the demonstrator's actions. However, by disregarding the distribution of the expert's observations, imitation policies learnt by BC often result in compounding error when deployed in practice [206–209].

6.3.2 Imitation Learning from Multiple Environments

Consider a *family* of environments $\mathcal{M} = \{(\mathcal{X}^e, \mathcal{A}, T^e, r^e, \gamma) \mid e \in \mathcal{E}\}$ with observations $x^e \in \mathcal{X}^e$, actions $a \in \mathcal{A}$, transition function $T^e \in \Delta(\mathcal{X})^{\mathcal{X}\times\mathcal{A}}$, reward function $r^e \in \mathbb{R}^{\mathcal{X}\times\mathcal{A}}$, and discount factor γ. This is the primary setting that we shall operate in. Note that the action space and discount factor do not change between environments. For notational simplicity, when considering the union over environments, we shall drop the index e. We assume offline access to a dataset of recorded trajectories from the expert policy π_D in a set of training environments $\mathcal{E}_{train} \subset \mathcal{E}$, $\mathcal{D} = \{\{(\tau_i^e\}_{i=1}^{N_e} \mid e \in \mathcal{E}_{train}\}$. Each trajectory $\tau^e \sim \pi_D = (x_t^e, a_t, x_{t+1}^e)_{t=0,\dots}$ consists of a sequence of environment specific observations, expert actions and next observations sampled as $a_t \sim \pi_D(\cdot \mid x_t^e)$ and $x_{t+1}^e \sim T^e(\cdot \mid x_t^e, a_t)$.

In the presence of multiple environments, our goal is to learn a policy $\pi \in \Delta(\mathcal{A})^{\mathcal{X}}$ that matches the expert behaviour in all possible environments $\mathcal{E}$ that share a certain structure for the observations and the transition dynamics. In particular, this involves

finding a policy that *generalizes* well across these related environments $e \in \mathcal{E}$—that is, the policy should ideally minimize the imitation risk across them:

$$\max_{e \in \mathcal{E}} R^e(\pi) = \mathcal{L}^e(\pi, \pi_D) \tag{6.2}$$

where each $\mathcal{L}^e$ explicitly depends on the characteristics of the environment e. Note that since we know nothing specific about $\mathcal{E}$, it is difficult to optimize for this directly. That said, if we make mild assumptions about the "relatedness" of these environments, we can learn policies that generalize well.

Structure of Observations: First, we assume there is a shared latent structure underlying the observations from different environments—on which the expert policy depends. Finding such a structure would let us discard irrelevant factors as inputs to the learnt policy, improving generalization [48, 202, 231]:

Assumption 6.3.1. *(Shared Latent Structure) Consider decomposing the observations $x^e \in \mathcal{X}^e$ in each environment $e \in \mathcal{E}$ into two components: an invariant representation $s \in \mathcal{S}$ and noise terms $\eta^e \in \mathcal{Z}^e$ (i.e. spurious correlations), such that $x^e = q(s, \eta^e)$ for some invertible transformation $q : \mathcal{S} \times \mathcal{Z}^e \to \mathcal{X}^e$. There exists some q such that π_D only depends on s, and space $\mathcal{S}$ is non-empty.*

In other words, we assume that the demonstrator's policy π_D depends only on information that is shared across the environments, i.e. the state variables s are the causal parents of the expert action $a \sim \pi_D(\cdot \mid s)$. As illustrated in Figure 6.1, the state variables and the noise terms are responsible for generating the patient observations, but the policy depends only on the state variables. Thus, we allow different environments to have different $p(x)$ marginals (as well as different $p(a|x)$). This allows environments to have different structure. The only requirement is that the environments are the same as far as the task is concerned. This means that there exists some $\mathcal{S}$ such that $p(s)$ marginals should be the same (as well as $p(a|s)$). Learning such a representation that is invariant satisfies the standard Environment Invariance Constraint [232]. While this set-up is similar to the one in [202], a crucial difference is that we have no access to any reward functions whatsoever, and that we must learn an imitation policy in a strictly batch setting.

Note that the latent structure induced by the state variables is *shared* across the different environments. This means that the transition dynamics for the state representation $p(s_{t+1} \mid s_t, a_t)$ remain invariant across the environments. On the other hand, as different environments may be characterized by different types of spurious correlations, to allow for flexibility in their structure and evolution, we consider that the transition dynamics of the noise terms $p^e(\eta^e_{t+1} \mid \eta^e_t, a_t)$ are *specific* to each environment. Our goal, then, is to learn a **generalizable policy** π—that is, one that depends only on s.

Structure of Environments: Second, to learn a policy that depends only on s, we must assume that the available training environments are actually different, so that we can learn the invariant state representation using the data from these environments and separate it from the noise representation:

Assumption 6.3.2. *(Environment Interventions) Each available training environment e corresponds to a hard [156] or soft [233] intervention on one or more dimensions of that environment's observation space (where these dimensions do not constitute any causal parents of the demonstrator's actions).*

To ensure that a generalizable policy actually exists, Assumption 6.3.1 requires that $\mathcal{S}$ be non-empty across all environments. Here, to ensure that the space $\mathcal{S}$ can actually be learned, Assumption 6.3.2 requires that $\mathcal{Z}$ be non-empty across the training environments. Note that we require that the interventions inducing the different environments to *not* be on the causal parents of the action, such that Assumption 6.3.1 is not violated.

Overall, in our setting $p(x)$ and $p(a \mid x)$ can differ between the multiple environments. However, Assumption 6.3.1 enforces that the environments and tasks are the same modulo noise, i.e. that there exists some non-empty $\mathcal{S}$ such that $p(s)$ and $p(a \mid s)$ are the same between them. In other words, we have a set of environments that are different (i.e. the dynamics of x_t are different), but the task being performed by the agent is the same (i.e. the dynamics of s_t are the same). This setting applies to the case when lightning conditions in a room are changing, but physical dynamics of the environment are staying the same [202] or when weather conditions are changing, but driving behaviour and dynamics are staying the same. We provide additional explanations and definitions of environment interventions in Appendix D.1.

Figure 6.2 shows a simple example where each observation x_t represents a union of the causal parents of the action (state variables) $s_t = \{x_t^1, x_t^2\}$ and the spurious correlations (noise variables) $\eta_t = \{x_t^3\}$. To satisfy Assumption 6.3.2, the different environments need to correspond to interventions on x_t^3. And to satisfy Assumption 6.3.1, x_t^1 and x_t^2 must not be intervened on. Our aim is to find a representation s of the causal parents $\{x_t^1, x_t^2\}$ of the actions, as well as the mapping between them and the actions a_t, that mimics the expert's policy.

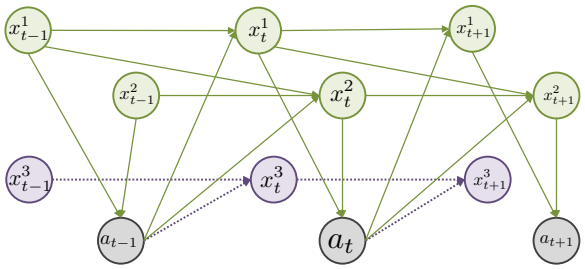


Figure 6.2: Causal diagram illustrating temporal dependencies between causal parents of action $\{x_t^1, x_t^2\}$ and noise variables $\{x_t^3\}$. Different environments are induced by interventions on the noise variables.

Finally, similarly to [202], we also assume that the observations x_t at timestep t can only affect the actions a_t and the observations at the next timestep $t+1$:

Assumption 6.3.3. *(Temporal Causal Mechanism) Let x^i and x^j be any two components of the observation x at timestep t. Then:*

$$x_{t+1}^i \perp\!\!\!\perp x_{t+1}^j \mid x_t, a_t \tag{6.3}$$

Note that Assumption 6.3.3 simply serves to place us within the standard MDP setting: It ensures Markovianity of the temporal transitions, that only the observations x_t at time t will contain the causal parents of the action a_t, and that x_t and a_t are the only factors that determine the next observation x_{t+1}.

6.4 Invariant Causal Imitation Learning for Generalization across Domains

The goal of our Invariant Causal Imitation Learning (ICIL) algorithm is to learn a representation of the state variables s that is invariant across domains, and an imitation policy π that depends on this causal representation and matches the demonstrator's behaviour. We operate in the strictly batch setting, and our aim is for π to generalize to unseen environments $e \in \mathcal{E}$ given the above structural assumptions.

6.4.1 Learning Invariant Causal Representations

To achieve our goal, we decompose the observations x_t^e in each environment e into a representation $s_t = \phi(x_t^e; \theta_s)$ for the causal features of the action a_t, and another representation $\eta_t^e = \mu^e(x_t^e; \theta_\eta^e)$ for the noise variables, where θ_s and θ_η^e are the learnable parameters of ϕ and η^e. Since the causal parents of the action are invariant across the environments, the state representation model $\phi : \mathcal{X} \to \mathcal{S}$ is the same across all environments. On the other hand, $\mu^e : \mathcal{X} \to \mathcal{Z}^e$ is environment-specific in order to allow for dynamics mismatch of the noise variables between the different environments.

In order to satisfy the causal diagram in Figure 6.1 and to learn a minimal causal representation, we need the following conditions to be satisfied: (1) s_t should be *invariant* across the environments, (2) s_t and η_t^e should be *dynamics-preserving*, and (3) s_t and η_t^e should be *independent* from each other.

Firstly, to fulfill condition (1) we train an environment classifier on the shared state representation $c_s : \mathcal{S} \to |\mathcal{E}_{train}|$, parameterized by θ_c using the cross-entropy loss. Similarly to [202], in order to build a state representation that is invariant across domains, we use an adversarial loss [234] that maximizes the entropy of the classifier: $H(c_s(\phi(x_t; \theta_s); \theta_c))$. This gives us the following practical loss function:

$$\mathcal{L}_{inv}(\theta_s) = \sum_{e \in \mathcal{E}_{train}} \mathbb{E}_{x_t^e \sim \rho_D^e} - H(c_s(\phi(x_t^e; \theta_s); \theta_c)) \tag{6.4}$$

Out of all possible representations that are invariant, we specifically seek one that also preserves the transition dynamics, fulfilling condition (2). To ensure that the state and noise representations are dynamics-preserving, we also learn the transition dynamics for the state variables $g_s : \mathcal{S} \times \mathcal{A} :\to \mathcal{S}$, such that $\hat{s}_{t+1} = g_s(s_t, a_t; \theta_{g_s})$ and the environment specific transition dynamics for the noise variables: $g_\eta^e : \mathcal{Z}^e \times \mathcal{A} \to \mathcal{Z}^e$, such that $\hat{\eta}_{t+1}^e = g_\eta^e(\eta_t^e, a_t; \theta_{g_\eta}^e)$. To reconstruct x_{t+1} we also learn $\psi : \mathcal{S} \times \mathcal{Z}^e \to \mathcal{X}$ such that $\hat{x}_{t+1}^e = \psi(s_{t+1}, \eta_{t+1}^e; \theta_\psi)$. This yields the following practical loss function:

$$\mathcal{L}_{dyn}(\theta_s, \theta_{g_s}, \{\theta_\eta^e, \theta_{g_\eta}^e\}_{e \in \mathcal{E}_{train}}, \theta_\psi) = \sum_{e \in \mathcal{E}_{train}} \mathbb{E}_{x_{t+1}^e, a_t, x_{t+1}^e \sim \rho_D^e} \|x_{t+1}^e - \hat{x}_{t+1}^e\|^2 \tag{6.5}$$

Note that while an alternative approach could consider directly building an invertible mapping from x^e to (s, η^e), the motivation for decoding s_{t+1} and η_{t+1}^e into $\hat{x}_{t+1}^e$ is

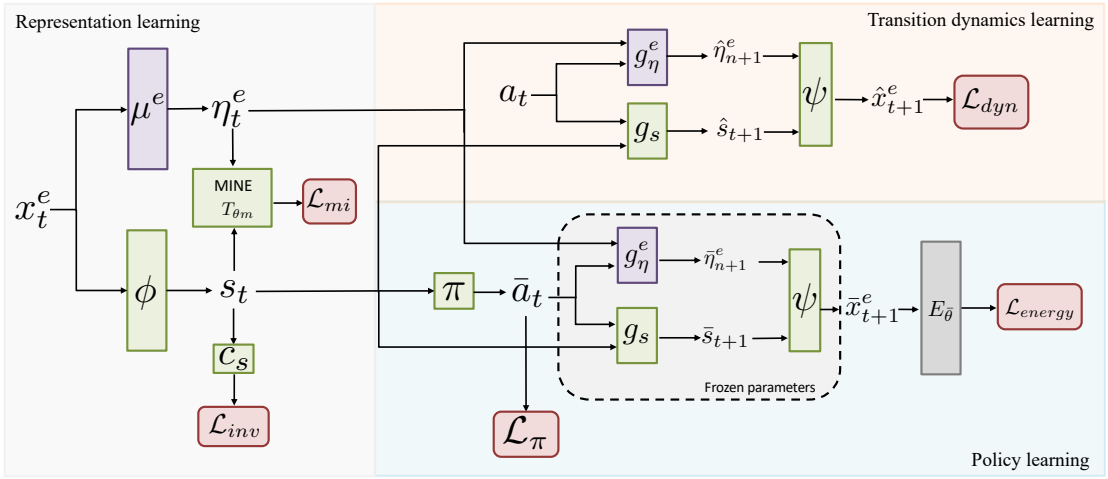


Figure 6.3: Block diagram of our model. ICIL decomposes the observations x_t^e into an invariant causal representation s_t and an environment specific noise representation η^e. To obtain an invariant representation, we maximize entropy of an environment classifier that receives as input s_t ($\mathcal{L}_{inv}$). Moreover, the state and noise representations are learnt to be dynamics preserving by minimizing the prediction error of the next observation ($\mathcal{L}_{dyn}$) and independent by minimizing their mutual information ($\mathcal{L}_{mi}$). We learn a generalizable imitation policy that is conditioned on the invariant causal representation ($\mathcal{L}_{\pi}$) and to ensure that the learnt policy matches the distribution of the expert's observations, we minimize the imitator's policy next state energy ($\mathcal{L}_{energy}$).

twofold. In addition to learning dynamics-preserving representations, as we will see in Section 6.4.2, this also allows us to compute the energy of the next state obtained by following the imitation policy and enforcing this to be similar to the distribution of states visited by the expert's policy.

Finally, to ensure that the state representation and the noise representation are marginally independent per condition (3), we minimize the mutual information between them. We use the Mutual Information Neural Estimation (MINE) framework [235], which provides a way for estimating the mutual information using neural networks. In particular, MINE uses a neural information measure $I(U, V)$ to approximate the mutual information between random variables U and V. Let T_{θ_m} be a statistics network parametrized by θ_m. MINE estimates $I(U, V)$ by ascending the gradient of the following:

$$I(U,V) = \sup_{\theta_m} \mathbb{E}_{\mathbb{P}_{UV}^{(n)}}[T_{\theta_m}] - \log(\mathbb{E}_{\mathbb{P}_U^{(n)} \otimes \mathbb{P}_V^{(n)}}[e^{T_{\theta_m}}]) = \sup_{\theta_m} I(U,V;\theta_m) \tag{6.6}$$

where $\mathbb{P}_{UV}$ is the joint measure of (U, V) and $\mathbb{P}_U = \int_{\mathcal{V}} dP_{UV}$, $\mathbb{P}_V = \int_{\mathcal{U}} dP_{UV}$ are the marginal distributions. $\mathbb{P}^{(n)}$ denotes the empirical distribution associated with n i.i.d samples. As noted in [235], the neural information measure $I(U, V)$ can approximate the

mutual information with arbitrary accuracy. We therefore add the following practical loss function to our optimization objective, which seeks to minimize the mutual information between the state representation and noise representation:

$$\mathcal{L}_{mi}(\theta_s, \{\theta_\eta^e\}_{e\in\mathcal{E}_{train}}) = \sum_{e\in\mathcal{E}_{train}} \mathbb{E}_{x_t^e\sim\rho_D^e} I(\phi(x_t^e;\theta_s), \mu(x_t^e;\theta_\eta^e);\theta_m) \tag{6.7}$$

Note that the parameters θ_m of the statistics network T_{θ_m} used for computing the mutual information are updated through gradient ascent on $I(U, V; \theta_m)$.

6.4.2 Matching Expert Behaviour in a Strictly Batch Setting

On the basis of the causal representation s, we shall learn a generalizable policy π (parameterized by θ_π) in the strictly batch setting, such that it matches the demonstrator's behaviour. To begin, we first condition π on the representation s_t and minimize the negative log-likelihood of the expert's actions:

$$\mathcal{L}_\pi(\theta_\pi, \theta_s) = \sum_{e\in\mathcal{E}_{train}} -\mathbb{E}_{x_t^e, a_t\sim\rho_D^e} \log \pi(a_t \mid \phi(x_t^e;\theta_s);\theta_\pi) \tag{6.8}$$

However, having only this objective corresponds to performing behaviour cloning, which has well-known limitations [206–209]. To mitigate compounding error, we want some form of added regularization to incentivize the imitation policy to stay within the distribution of the expert's observations.

In the online setting, a popular approach is to make sure that the rollout distribution of the imitating policy matches the rollout distribution of the expert's policy—for instance, by minimizing some form of divergence between their induced occupancy measures. However, this requires interactive access to the real environment or simulator to perform rollouts of intermediate policies—which is not possible in our setting. Instead, we propose a method that takes advantage of the learnt transition dynamics. For any current observation $x_t \sim \rho_D$, we shall encourage the next observation $\bar{x}_{t+1}$ obtained by following the imitation policy $\bar{a}_t \sim \pi(\cdot \mid x_t)$ to remain within the occupancy measure of the expert.

Consider approximating the expert's occupancy measure using an Energy Based Model (EBM) such that $\rho_D(x) = \frac{\exp(-E_{\bar{\theta}}(x))}{Z(\bar{\theta})}$ where the function $E_{\bar{\theta}}(x) : \mathcal{X} \to \mathbb{R}$ is the energy function and $Z(\bar{\theta}) = \int_x -E_{\bar{\theta}}(x)dx$ is the partition function. We parameterize

$E_{\bar{\theta}}$ by a neural network. It is not possible to train the EBM directly through maximum likelihood because $Z(\bar{\theta})$ involves integrating over the entire input domain of x which is impractical. Instead, we use contrastive divergence to pre-train the energy function $E_{\bar{\theta}}$ [236, 237]. Contrastive divergence lowers the energy of the observations coming from the expert's occupancy distribution and increases the energy of the observations outside of the expert's occupancy distribution. Refer to Appendix D.2 for details on how we train the EBM. To incentive the imitation policy to stay within the distribution of the expert's observations, we train it to minimize the energy of the next observation obtained by following π given the current observation:

$$\mathcal{L}_{energy}(\theta_\pi; \theta_s, \theta_{g_s}, \{\theta^e_\eta, \theta^e_{g_\eta}\}_{e\in\mathcal{E}_{train}}, \theta_\psi) = \sum_{e\in\mathcal{E}_{train}} \mathbb{E}_{\substack{x^e_t\sim\rho^e_D \\ s_t=\phi(x^e_t),\eta^e_t=\mu^e(x^e_t) \\ \bar{a}_t\sim\pi(\cdot|s_t) \\ \bar{x}_{t+1}=\psi(g_s(s_t,\bar{a}_t),g^e_\eta(\eta^e_t,\bar{a}_t))}} E_{\bar{\theta}}(\bar{x}^e_{t+1}) \tag{6.9}$$

This effectively assigns a high "reward" to the imitation policy for staying within high-density areas of the expert's occupancy measure, and a low "reward" for straying from it. This can be seen as an adaptation of online imitation methods [238, 239] where the expectation would be instead over $x_t \sim \rho_\pi$.

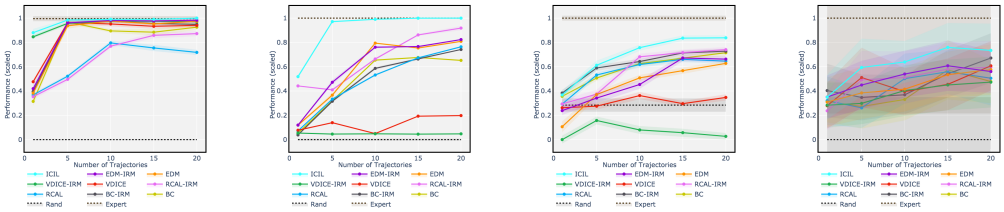
We illustrate in Figure 6.3 all of the components of the our ICIL model. Further details and the full algorithm for optimizing ICIL can be found in Appendix D.4.

6.5 Experiments

We perform experiments on OpenAI gym tasks [240] and on an ICU dataset from the MIMIC III database [148]. In both cases, we generate data from multiple domains by augmenting the feature space with noise variables (spurious correlations).

Benchmarks We compare ICIL[1] against standard methods for strictly batch imitation learning: Behaviour Cloning (BC) [205]; Reward-regularized Classification for Apprenticeship Learning (RCAL), which incorporates dynamics-awareness through a sparsity regularization on the implied rewards [209]; ValueDICE (VDICE) [199], which uses an off-policy objective to estimate distribution ratios needed for distribution matching; as

[1]The code for ICIL can be found at https://github.com/ioanabica/Invariant-Causal-Imitation-Learning.



(a) Acrobot **(b)** Cartpole **(c)** LunarLander **(d)** BeamRider

Figure 6.4: Evaluation on OpenAI gym environments. x-axis indicates the number of trajectories (in $\{1, 5, 10, 15, 20\}$) with expert demonstrations from each training environment given as input to each benchmark and y-axis represents the average return of the learnt imitation policy on the test environments, scaled between 1 (expert performance) and 0 (random policy performance).

well as Energy-based Distribution Matching (EDM) [66], which jointly learns the imitator policy with an energy model of its state distributions. These methods seek to find a policy that approximately matches the expert's behaviour from a single environment, and were not designed with generalization in mind. Hence we augment these benchmarks by using the IRMv1 objective [48] in conjunction with their originally defined imitation risk to obtain the additional benchmarks: BC-IRM, RCAL-IRM, VDICE-IRM, and EDM-IRM. More details about how we used the invariance-based penalty from IRM [48] to augment these existing methods such that they may learn generalizable policies can be found in Appendix D.3. Implementation details about all benchmarks and the hyperparameter settings used can be found in Appendix D.5.

6.5.1 Evaluation on OpenAI Gym

We perform experiments on the following control tasks from OpenAI gym [240]: Acrobot [241], Cartpole [242], LunarLander [240] and BeamRider [243]. For each task, we use pre-trained RL agents from RL Baselines Zoo [244] and Stable OpenAI Baselines [245] to obtain expert policies. We then follow an approach similar to the one in [202] to obtain datasets with demonstrations from the expert in two different environments. In particular, for Acrobot [241], Cartpole [242] and LunarLander [240] we add spurious correlations to the state space of each control task and an environment identifier. The spurious correlations in each environment are different multiplicative factors of a subset of variables in the original state space. The invariant causal state is represented by the

original variables in the state space of each control task. We train the benchmarks on demonstrations from two environments with $1\times$ and $2\times$ multiplicative factors for the spurious correlations and we test on an environments with multiplicative factors sampled from $\mathcal{U}(-1, 1)$. For BeamRider [243], similarly to [202], different camera angles are used for the training and testing environments. In particular, we use two training environments where the game frames are rotated by 10 degrees to the left and to the right respectively, while the test environment has no rotation. The rotation is applied to the entire frame for all trajectories in each environment. However, note that, despite the rotation, the dynamics for the state variables and how they influence the action stay the same. Further details about the train and test environments can be found in Appendix D.5.

We vary the number of demonstrated trajectories from each environment that we give as input to each benchmark and we evaluate them on the average return obtained by deploying the learnt imitation policies on the test environment. Figure 6.4 shows the mean results and standard errors obtained across 10 runs where for each run we train the benchmarks on different trajectories from the train environments and we evaluate on a test environment with newly sampled multiplicative factors for computing the spurious correlations. We notice that our method consistently outperforms the benchmarks and is capable of generalizing better to the unseen target environments. Moreover, we generally found that using the IRMv1 objective [48] together with existing methods for strictly batch imitation learning did not improve performance and resulted in more unstable training. For additional results, see Appendix D.6 where we perform ablation studies to investigate the impact of the different terms in the loss function used to train ICIL on overall performance, compare performance on train vs. test environments and also evaluate robustness to increasing the size of the spurious correlations.

6.5.2 Evaluation on MIMIC-III

We also perform experiments on a healthcare dataset with Intensive Care Unit (ICU) patients extracted from the Medical Information Mart for Intensive Care (MIMIC-III) database [148]. The dataset consists of trajectories of clinical measurements (e.g. heart

	Mechanical ventilator		
Benchmark	ACC	AUC	APR
BC	0.783 ± 0.001	0.762 ± 0.002	0.692 ± 0.001
RCAL	0.790 ± 0.002	0.771 ± 0.002	0.697 ± 0.002
VDICE	0.794 ± 0.001	0.784 ± 0.001	0.716 ± 0.001
EDM	0.786 ± 0.003	0.741 ± 0.011	0.682 ± 0.005
BC-IRM	0.791 ± 0.002	0.767 ± 0.003	0.696 ± 0.002
RCAL-IRM	0.789 ± 0.002	0.766 ± 0.003	0.694 ± 0.003
VDICE-IRM	0.766 ± 0.001	0.730 ± 0.001	0.694 ± 0.001
EDM-IRM	0.781 ± 0.004	0.717 ± 0.015	0.673 ± 0.007
ICIL	$\mathbf{0.855 \pm 0.003}$	$\mathbf{0.856 \pm 0.004}$	$\mathbf{0.789 \pm 0.004}$

Table 6.2: Evaluation on MIMIC-III in terms of action-matching. We compare the actions selected by the benchmark imitation policies with the ones from the clinical expert policy in the test environment and report the accuracy (ACC), the area under the receiving operator characteristic curve (AUC) and the area under the the precision-recall curve (APR).

rate, respiratory rate) recorded every hour. The aim is to learn a generalizable policy for the action of putting patients on the mechanical ventilator.

We define three environments, two for training and one for testing, each consisting of 2000 independent patient trajectories from MIMIC-III. We augment the original feature space by adding spurious correlations (noise variables) that are the same as the expert actions with probabilities $p = 0.1$ and $p = 0.2$ in the training environments and with probability $p = 0.8$ in the testing environment.

In a real setting, such spurious correlations are commonplace. For instance, consider some hospitals (i.e. training environments) where selection bias is present, such that patients with a certain otherwise irrelevant comorbidity happen to receive a treatment more often [246–249]. However, learning an imitation policy that takes into account such a comorbidity when assigning the patient's treatment would fail to generalize to hospitals where fewer patients suffer from this comorbidity but should still receive the treatments. More details about the dataset can be found in Appendix D.5.

Since MIMIC-III is an entirely offline dataset, it is not possible to compute average returns for running the policies in the test environment. Instead, we evaluate the benchmarks in terms of action matching in the test environment. We report in table 6.2 the mean accuracy (ACC), the mean area under the receiving operator characteristic curve (AUC), the mean area under the precision-recall curve (APR) and their standard

deviations over 10 runs. We notice that ICIL learns a policy that best discards the spurious correlations present in the training environment to learn a generalizable policy for putting patients on the mechanical ventilator that best matches the expert's actions in the test environment.

6.6 Discussion

In this chapter, we tackle the problem of learning generalizable imitation policies in the strictly batch setting. Our ICIL model leverages ideas from causality and learns an invariant state representation that minimizes the presence of spurious correlations. By conditioning the imitation policy on this state representation, we obtain a policy that generalizes to environments with the same shared latent structure, but with different noise distribution and dynamics. ICIL also matches expert behaviour by incentivizing the learnt imitation policy to stay within the expert's observations distribution.

In terms of limitations, we believe that future work should consider providing theoretical insights and error bounds on the generalization error. In addition, to be able to learn an invariant state representation, our method requires demonstrated trajectories from at least two training environments with different interventions on the noise variables (spurious correlations), and the method cannot be used if such data is not available in practice. Finally, we bear in mind that—as with any other imitation learning method that aims to match the expert's policy—ICIL can have potential negative societal impacts if the expert's policy is flawed in the first place. Thus, in sensitive applications such as clinical decision support, care must be taken to prevent potentially negative feedback loops.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Invariant Causal Imitation Learning for Generalizable Policies
Publication Status	☑ Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Ioana Bica*, Daniel Jarrett*, and Mihaela van der Schaar. "Invariant Causal Imitation Learning for Generalizable Policies". In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Student Confirmation

Student Name:	Ioana Bica		
Contribution to the Paper	- I was the lead co-author responsible for conceptualizing the paper. - The different components of the proposed Invariant Causal Imitation Learning model were the product of brainstorming discussions between Daniel and me. - I implemented the model and produced all of the results in the paper. - The paper was written jointly by Daniel and me. - Both Daniel and I performed the experimental design and analyzed the results. - All authors contributed to the development of the paper through multiple discussions. - All authors reviewed the final draft.		
Signature 		Date	30/03/2022

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Mihaela van der Schaar		
Supervisor comments: I agree that the candidate has made a substantial contribution to the publication and that the above description is accurate.		
Signature 	Date	30/03/2022

This completed form should be included in the thesis, at the end of the relevant chapter.

7 Discussion and Conclusion

Contents

7.1 Summary of Contributions 134
7.2 Relationship to Other Publications 136
7.3 Limitations and Future Directions 139

In this thesis, we motivated, investigated and proposed new methods to address several challenges in building causal inference tools for supporting decision-making by estimating the causal effects of interventions (Chapters 3 and 4), for obtaining interpretable explanations of sequential decision-making by integrating counterfactual reasoning into batch inverse reinforcement learning (Chapter 5) and for improving the robustness of decision-making guidelines by leveraging causal knowledge as part of imitation learning algorithms (Chapter 6). In this chapter, we provide a summary of our contributions and discuss limitations and directions for future work.

7.1 Summary of Contributions

This section summarizes the main contributions from each chapter of the thesis.

In **Chapter 2**, we provide a thorough literature review of the area of causal inference from observational data in both the static (cross-sectional) setting and the longitudinal setting (where patient history and treatment timing are taken into account). This chapter discusses the main difficulties in learning from observational data, main modelling choices used by machine learning methods and highlights areas that require further improvements and which we consider in the subsequent chapters of the thesis.

Chapter 3 introduces SCIGAN, a causal inference method that builds on the framework of generative adversarial networks to estimate the effects of continuous-valued interventions, i.e. treatments with associated dosages in the static (cross-sectional) setting. We provide theoretical guarantees for the proposed approach and novel neural network architectures that can handle the complexity of having continuous interventions. In addition, we also develop a new benchmarking dataset for evaluating causal inference methods for estimating individualized dose-response curves on which we show the performance gains of our approach.

In **Chapter 4**, we shift our attention to the temporal setting and in Chapter 4.1 we introduce the Counterfactual Recurrent Network (CRN) which is a sequence-to-sequence model capable of estimating, for each patient, potential outcomes for intended sequences of future treatments, conditional on the patient history. To handle the bias from time-dependent confounders, CRN builds at each timestep balancing representations of the

patient history through domain adversarial training. Moreover, in Chapter 4.2 we propose the first method to address the challenge of having hidden confounders in the temporal causal inference setting and develop the Time Series Deconfounder (TSD). The TSD takes advantage of the way multiple treatments are assigned over time and builds a factor model to obtain a sequence of latent variables that can act as substitutes for the hidden confounder. Then, these latent variables can be used to augment existing observational datasets and thus lead to unbiased estimates of the treatment effects.

Chapter 5 explores the intersection of causality and batch inverse reinforcement learning and introduces a method for explaining sequential decision making on the basis of demonstrated behaviour–i.e. trajectories of observations and actions made by an expert (agent) maximizing some unknown reward function. We propose CIRL, the first method for explaining sequential decision-making by integrating counterfactual reasoning into batch inverse reinforcement learning and recovering the preferences of experts over "what-if" outcomes. This allows us to build interpretable parameterisations of experts' behaviour such that we can quantify and inspect policies in different environments and uncover the trade-offs and preferences associated with expert actions. Such methods for understanding and modelling clinical decision making are important for finding the limitations of existing clinical practices, potential biases and variation in practice and overall, for building a more transparent account of clinical practice.

In **Chapter 6**, we propose Invariant Causal Imitation Learning (ICIL), a new method for learning generalizable imitation policies in the strictly batch setting. ICIL leverages ideas from causality and uses data from multiple environments to learn an invariant state representation that minimizes the presence of spurious correlations. By conditioning the imitation policy on this state representation, we obtain a policy robust to spurious correlations and biases that generalizes to environments with the same shared latent structure. ICIL also matches expert behaviour by incentivizing the learnt imitation policy to stay within the expert's observations distribution. By incorporating causal structure into imitation learning, ICIL helps us obtain clinical decision guidelines that are robust across different environments.

The aim of the causal inference methods developed in this thesis is to help support, understand, and subsequently improve decision-making. Throughout this thesis, we make a significant contribution in this direction, by proposing new methodological advances in machine learning capable of reasoning about cause-and-effect relationships that (1) enable us to better estimate the causal effects of treatments, (2) understand expert preferences through counterfactual reasoning, and (3) learn more robust imitation policies that incorporate causal structure. Nevertheless, we acknowledge the fact that the work presented in this thesis is on the theoretical side and significant testing, potentially through clinical trials, will be needed before such methods can be used in clinical practice, particularly due to the life-threatening implications of incorrect estimates.

7.2 Relationship to Other Publications

In this section, we describe the relationship between the work presented in this thesis to the other publications I have contributed to during my doctoral studies, highlighted in Section 1.3.2. The additional publications that I have contributed to, mainly as a second or third author, propose further causal inference tools for improving the estimation of treatment effects or consider new types of interventions (e.g. organs) that have not been well-studied before. Moreover, we also propose frameworks for the evaluation of machine learning methods for time series analysis, which includes the estimation of counterfactual treatment outcomes over time, and of methods for medical decision modelling.

Organ-transplantation allocation. To begin with, in [59, 60] we consider the problem of organ-transplantation allocation where the aim is to decide which patient in the transplantation queue should be matched with an arriving organ, where the optimization goal is to maximising the total life years of the population (i.e. mean life expectancy). By using observational data for organ allocations and transplant outcomes, we propose new methods for reliably estimating the potential outcomes for each patient receiving the different possible organs, i.e transplant outcomes under counterfactual matches not observed in the data. As organs are high-dimensional interventions, our methods propose clustering them either as a pre-processing step [59] or in an end-to-end learning fashion [60] to build loss functions that can handle the selection bias present in the observational

datasets. We then devise new organ allocation policies, that take into account not only the counterfactual outcomes of receiving or not receiving the organ [60], but also the rarity of the organ needed for each patient [59].

Model selection for treatment effects. In addition, we also consider the problem of model selection for treatment effect methods in the unsupervised domain adaptation setting [63]. In this case, we have access to an observational dataset with patient outcomes (labelled samples) in the source domain, but desire selecting a treatment effects model that achieves good performance on a target domain where only unlabeled samples are available. This is common in healthcare where the observational data used to train a treatment effects model may come from a setting where the distribution of patient features is different from the one in the deployment (target) environment, for example, when transferring models across hospitals or countries. Because of this, it is imperative to select treatment effects models that are robust to these covariate shifts across disparate patient populations. To address this problem, we propose to leverage the invariance of causal graphs across domains to develop a model selection metric that evaluates how well the predictions of the treatment effects satisfy known or discovered causal relationships [63].

Estimating treatment effects over time. When considering the setting of treatments over time, we build upon CRN to introduce the Disentangled Counterfactual Recurrent Network (DCRN) [62]. DCRN introduces a novel sequence-to-sequence architecture that learns representations of patient histories which are disentangled into three separate latent factors: a treatment factor that determines the treatment, an outcome factor that affects the outcome and a confounding factor, influencing both. This decomposition allows DCRN to better handle the time-dependent confounding bias and achieves performance improvements over CRN for estimating counterfactual outcomes for sequences of future treatments. In addition, in [61], we introduce SyncTwin, a causal inference method that tackles the setting of longitudinal and irregularly sampled data where the treatment allocation happens at some observed time and then it is fixed for the rest of the study. These are commonly referred to as point treatments and some examples include surgical operations or medication for chronic diseases. SyncTwin makes counterfactual predictions

for a target patient by constructing synthetic twins that match the representation of the pre-treatment observations of the target patient, both for the control and treated cases. Such an approach has the added benefit of interpretability, as medical experts can examine how the synthetic twins used to issue predictions for the target patient have been constructed.

Pipeline toolkits for model development and evaluation. Furthermore, to facilitate the use of machine learning methods for medical time-series by practitioners, we built the Clairvoyance package [67]. Clairvoyance is a unified end-to-end pipeline for medical time series that consists of (1) a software toolkit that facilitates model development and deployment, (2) an empirical standard for model evaluation and (3) an interface for optimization over the pipeline abstraction. As part of Clairvoyance, we also incorporate methods for estimating the causal effects of treatments in the temporal setting, such as CRN. Additionally, we built Medkit [68], which is an environment that provides extensive simulation scenarios for evaluating methods for medical decision modelling such as CIRL. Medkit enables users to obtain batch datasets where the ground-truth policy parameterisations (e.g. reward weights) of experts are known. By being able to simulate various different types of decision-making behaviours, Medkit can enable more efficient progress in the area of understanding human decision-making which can ultimately lead to improved clinical practice.

Finally, while most of the work done during my doctoral studies has been on the machine learning methodology side, I have also contributed to a few papers that aim to bridge the gap with the pharmacology and clinical trials research communities. In particular, [64] highlights the potential of using causal inference tools based on machine learning to improve clinical trials. Moreover, the work on [53] incorporated in Chapter 2 also aims to emphasize the benefits of such methods in the pharmacology community to foster future collaboration across the scientific disciplines. Nevertheless, as previously mentioned, we acknowledge that significant more work is needed to incorporate machine learning methods into clinical practice and we provide a further discussion in the next section.

7.3 Limitations and Future Directions

In this section, we discuss in more depth some of the limitations of the main works presented in this thesis and propose several areas where further work is needed. We note that each thesis chapter discusses the limitations of the proposed methods and highlights relevant technical directions for future work. Therefore, here we focus on providing more general suggestions for improving machine learning methods for causal inference and discussing limitations for incorporating such methods into clinical practice.

We have yet to harness the full potential of observational data for clinical decision support [23]. While the simple case of static, binary treatments has been well-studied in the causal inference literature, Electronic Health Records (EHRs) contain information about significantly more complex treatment scenarios. In order to design more robust and effective machine learning methods for personalized treatment recommendations, it is vital that we gain a deeper theoretical understanding of the challenges and limitations of modelling multiple treatment options, combinations, and treatment dosages from observational data —in both the cross-sectional and time-series settings. In addition, developing machine learning methods that utilize both Randomized Control Trials (RCT) and EHR data represents an important research frontier, as these different data sources may provide more evidence for the effectiveness of treatments [250, 251]. Pharmacokinetic/Pharmacodynamic (PK/PD) approaches may also allow us to introduce priors into machine learning methods and improve their interpretability and performance [22, 252, 253]. Moreover, since counterfactuals cannot be observed, better methods for validating causal inference models that, for instance, make use of PK/PD modelling for simulation or expert knowledge are also crucial for being able to select the best performing method.

In practice, incorporating machine learning models into clinical decision support systems entails challenges of its own. Causal inference methods for estimating treatment effects and making treatment recommendations from observational data require thorough medical validation before real-world deployment. We refer the readers to the work of Eichler et al. [254] for a more thorough discussion of the requirements for making

machine learning methods acceptable for decision-makers and regulators. Domain knowledge is critical for providing assurance that the unconfoundedness assumption actually holds in the observational dataset of interest, and the overlap assumption also requires testing. At the end of the day, since we never observe any counterfactuals, expert knowledge is always needed to validate the model's estimates of the causal effects [22]. Moreover, the risk of incorrectly assigning treatments can also be significantly mitigated by ensuring that such models are used as support systems alongside clinicians, rather than independently.

Furthermore, from a methodological perspective, we must ensure that the patient population used to train the model is representative of the patient population the model is to be deployed on. If this is not the case, additional methods are needed to ensure that causal inference models are robust to distributional shifts in the patient population. Along this direction, developing further approaches for incorporating causal structure as indicative biases into machine learning methods for decision-making will help mitigate this problem and result in safer and more robust models [45].

Equally important for clinical adoption is understanding which patient features are the strongest drivers of treatment response– particularly when patient data are high-dimensional. Developing better tools for this can lead to a better understanding of diseases, or even discoveries about better treatments. In addition, finding relevant patient features for estimating treatment effects can also lead to more transparent, interpretable models. Finally, uncertainty estimates are also beneficial for assessing the model's confidence in its predictions. After all, if the patient outcomes estimated by a model are highly variable, it becomes difficult to trust the model safely and consistently for treatment decisions.

Future work addressing these aforementioned research directions will lead to better and more reliable machine learning methods for causal inference. It is our goal that these methods be safely and effectively integrated into the clinical decision support process.

Appendices

A

Appendix for Chapter 3

Contents

A.1 Expanded Related Works 143
A.2 Proofs of Theoretical Results 145
A.3 Counterfactual Generator Pseudo-code 148
A.4 Inference Network 150
A.5 Architecture 150
A.6 Single Discriminator Model 151
A.7 Ablation Studies Architectures 153
A.8 Dataset Descriptions 153
A.9 Dosage Bias 154
A.10 Benchmarks 155
A.11 Metrics 156
A.12 Additional Results 157

A.1 Expanded Related Works

Most methods for performing causal inference in the static setting focus on the scenario with two or multiple treatment options and no dosage parameter. The approaches taken by such methods to estimate the treatment effects involve either building a separate regression model for each treatment [123, 255, 256] or using the treatment as a feature and adjusting for the imbalance between the different treatment populations. The former does not generalise to the dosage setting due to the now infinite number of possible treatments available. In the latter case, methods for handling the selection bias involve propensity weighting [25, 257, 258], building sub-populations using tree based methods [1, 27, 103, 259] or building balancing representations between patients receiving the different treatments [28, 29, 102, 132]. An additional approach involves modelling the data distribution of the factual and counterfactual outcomes [26, 35].

[131] leverages observational and interventional data to estimate the effects of discrete dosages for a single treatment. In particular, [131] uses observational data to construct a non-stationary covariance function and develop a hierarchical Gaussian process prior to build a distribution over the dose response curve. Then, controlled interventions are employed to learn a non-parametric affine transform to reshape this distribution. The setting in [131] differs significantly from ours as we do not assume access to any interventional data.

Comparison with GANITE

Naive attempts to extend [35] to the continuous setting might involve: (1) discretising the continuous space of interventions; (2) somehow passing entire response curves to the discriminator and asking it to identify the point on the curve that corresponds to the factual outcome. Naturally, discretisation comes with a cost. If the discretisation is too coarse, the response curves will not be well-approximated. On the other hand, we show experimentally that GANITE is incapable of handling a high number of discrete interventions (corresponding to having a finer discretisation). In fact, although SCIGAN was designed for continuous interventions, it can be applied in the discrete setting and we show that it outperforms GANITE when the (discrete) parameter space is not small.

For (2), the problem is in defining a mechanism for generating these response curves in a form that can be passed to the discriminator and ensuring the *continuity* of these curves around the factual outcome so that the discontinuity itself does not make identification trivial for the discriminator. To overcome this we define a discriminator that acts on a finite set of points from each generated response curve (rather than on entire curves), as shown in Fig. 3.1. From *among the chosen points*, the discriminator attempts to identify the factual one. The set of points is sampled randomly *each* time an input would be passed to the discriminator. As our discriminator will be acting on a *set* of random intervention-outcome pairs, we explicitly condition it to behave as a function on a set. In particular, we draw on ideas from [127] to ensure that its output does not depend on the *order* of its input.

In addition, for the setting in which there are multiple possible interventions that *each* have an associated continuous parameter (which is the main setting of the paper), we propose a *hierarchical* discriminator which breaks down the job of the discriminator into determining the factual intervention and determining the factual parameter using separate networks. We show in the experiments section that this approach significantly improves performance and is more stable than using a single network discriminator. In this setting, we also model the generator as a multi-task deep network capable of taking a continuous parameter as an input; this gives us the flexibility to learn heterogeneous response curves for the different interventions.

Off-policy Evaluation and Policy Optimization with Continuous Treatments

A problem related to ours involves evaluating policies and learning optimal policies from logged data [260]. In this context, several methods have been proposed for performing off-policy evaluation and policy optimization with continuous treatments [261–263]. It should be emphasized that there is a significant difference between this line of research, which aims to find optimal policies, and the causal inference setting considered in this paper, where the aim is to learn the counterfactual patient outcomes under all possible treatment options.

For off-policy policy evaluation with continuous treatments, [261] propose a method based on inverse-propensity weighting to estimate the value function, i.e. cumulative

reward of a target policy from observational data. However, their proposed method does not perform any type of ITE estimation and does not have any intermediate per-sample value estimates. Alternatively, [263] does perform intermediate estimation of the value function for each sample. However, their contribution is to construct a doubly-robust estimator for the value function which crucially depends on knowing (or having some prior knowledge of) the parametric form of the response curve's dependency on the treatment. Without an assumed form, it is not possible to construct their estimator. Our proposed model, SCIGAN, does not assume any prior knowledge about the response curve.

The problem of individualised treatment effect estimation (which is our primary goal) is harder than off-policy evaluation as it involves estimating the outcomes (which may differ from the value function) for every sample and every possible action that could have been taken for that sample. After having learned these counterfactuals it is possible to perform policy evaluation, but note that the use cases for treatment effect estimation go beyond policy evaluation. In particular, for a given setting it may be far more beneficial to present a patient with their estimated outcome along with potential side effects and allow them (or the clinician) to come to a decision based on these several factors. In such a setting, each patient may have their own "internal" value function that depends on potential outcomes and side effects differently.

A.2 Proofs of Theoretical Results

In this section we prove Theorem 3.4.1. Note that these results also apply to GANITE (with very minor modifications - the proofs are even simpler in the case of GANITE).

In order to prove Theorem 3.4.1, we analyse the simpler minimax game defined by

$$\min_{\mathbf{G}} \max_{\mathbf{D}} \mathcal{L}(\mathbf{D}, \mathbf{G}) + \lambda \mathcal{L}_S(\mathbf{G}) \,, \tag{A.1}$$

which corresponds to the single discriminator model (instead of the hierarchical model) and then use this to prove our full result.

Lemma A.2.1. *Fix* $\mathbf{G}$ *and* $\tilde{\mathcal{D}} = \bigcup_w \tilde{\mathcal{D}}_w$. *Let* $p_{w,d}(\mathbf{y}|\mathbf{x}) = p_r(y_{w,d}|\mathbf{x}) p_{\mathbf{G}}(\mathbf{y}_{\neg w,d}|\mathbf{x}, y_{w,d})$ *denote the induced joint density of outcomes when restricted to dosages in* $\tilde{\mathcal{D}}$, *where* p_r

denotes the true density that generated the observed outcome and $p_{\mathbf{G}}$ denotes the density induced by $\mathbf{G}$ over the remaining dosages in $\tilde{\mathcal{D}}$. Then the optimal discriminator is

$$\mathbf{D}^*_{w,j}(\mathbf{x},\mathbf{y}) = \frac{\tilde{p}(w,d_j|\mathbf{x})p_{w,d_j}(\mathbf{y}|\mathbf{x})}{\sum_{w'\in\mathcal{W}}\sum_{i=1}^{n_w}\tilde{p}(w',d_i|\mathbf{x})p_{w',d_i}(\mathbf{y}|\mathbf{x})} \tag{A.2}$$

where $\tilde{p}$ is the $\tilde{\mathcal{D}}$-restricted propensity given by

$$\tilde{p}(w,d_j|\mathbf{x}) = p(w|\mathbf{x})(p(d_j|\mathbf{x},w)/\sum_{i=1}^{n_w}p(d_i|\mathbf{x},w)). \tag{A.3}$$

Proof. Fix $\mathbf{G}$ and $\tilde{\mathcal{D}} = \bigcup_w \tilde{\mathcal{D}}_w$. The optimal discriminator is given by $\arg\min_{\mathbf{D}} \mathcal{L}(\mathbf{D},\mathbf{G})$. We have

$$\mathcal{L}(\mathbf{D},\mathbf{G}) = \mathbb{E}\Bigg[\sum_{w\in\mathcal{W}}\sum_{d\in\tilde{\mathcal{D}}_w}\mathbb{I}_{\{T_f=(w,d)\}}\log\mathbf{D}^{w,d}(\mathbf{X},\tilde{\mathbf{Y}}) + \mathbb{I}_{\{T_f\neq(w,d)\}}\log(1-\mathbf{D}^{w,d}(\mathbf{X},\tilde{\mathbf{Y}}))\Bigg] \tag{A.4}$$

$$= \mathbb{E}_{\tilde{\mathcal{D}}}\Bigg[\sum_{w\in\mathcal{W}}\sum_{d\in\tilde{\mathcal{D}}_w}\int_{(\mathbf{x},\mathbf{y})}\tilde{p}(w,d|\mathbf{x})p_{w,d}(\mathbf{y}|\mathbf{x})\log\mathbf{D}^{w,d}(\mathbf{x},\mathbf{y}) + \Bigg(\sum_{w',d'\neq w,d}\tilde{p}(w',d'|\mathbf{x})p_{w',d'}(\mathbf{y}|\mathbf{x})\Bigg)\log(1-\mathbf{D}^{w,d}(\mathbf{x},\mathbf{y}))p(\mathbf{x})d\mathbf{y}d\mathbf{x}\Bigg] \tag{A.5}$$

where we have taken the (conditional on $\tilde{\mathcal{D}}$) expectations inside the sums and replaced indicator functions with densities as appropriate. We now note that $a\log p + b\log(1-p)$ for $p\in(0,1)$ has a unique maximum at $p=\frac{a}{a+b}$, thus implying that the integrand is maximised when

$$\mathbf{D}^{w,d}(\mathbf{x},\mathbf{y}) = \frac{\tilde{p}(w,d|\mathbf{x})p_{w,d}(\mathbf{y}|\mathbf{x})}{\tilde{p}(w,d|\mathbf{x})p_{w,d}(\mathbf{y}|\mathbf{x}) + \sum_{w',d'\neq w,d}\tilde{p}(w',d'|\mathbf{x})p_{w',d'}(\mathbf{y}|\mathbf{x})}\,. \tag{A.6}$$

This gives the required result. □

Using Lemma A.2.1 we can now show that the optimal solution to our single discriminator game is when the marginal distributions of the generated counterfactuals are equal to the true counterfactuals. Importantly, this suffices for estimating μ since the expectation is only concerned with the marginal distribution of $Y(w,d)$.

Lemma A.2.2. *The global minimum of minimax game defined by* $\min_{\mathbf{G}} \max_{\mathbf{D}} \mathcal{L}(\mathbf{D}, \mathbf{G})$ *is achieved if and only if for all* $\tilde{\mathcal{D}}_w$*, for all* $w, w' \in \mathcal{W}$ *and for all* $d \in \tilde{\mathcal{D}}$*,* $d' \in \tilde{\mathcal{D}}_{w'}$ *we have that*

$$p_{w,d}(\mathbf{y}|\mathbf{x}) = p_{w',d'}(\mathbf{y}|\mathbf{x}) \tag{A.7}$$

which in turn implies that for any treatment-dosage pair $(w, d) \in \mathcal{T}$ *we have that the generated counterfactual for outcome* (w, d) *for any sample (that was not assigned* (w, d)*) has the same (marginal) distribution (conditional on the features) as the true marginal distribution for that outcome.*

Proof. For fixed $\tilde{\mathcal{D}}$ and $\mathbf{x}$ we note that by substituting the optimal discriminator into $\mathcal{L}(\mathbf{D}, \mathbf{G})$ and subtracting $\sum_{w \in \mathcal{W}} \sum_{i=1}^{n_w} \log \tilde{p}(w, d_i|\mathbf{x})$ (which is independent of $\mathbf{G}$) we obtain

$$\mathcal{L}(\mathbf{D}^*, \mathbf{G}) - \int_{\mathbf{x}} \left(\sum_{w \in \mathcal{W}} \sum_{i=1}^{n_w} \log \tilde{p}(w, d_i|\mathbf{x}) \right) p(\mathbf{x}) d\mathbf{x} \tag{A.8}$$

$$= \mathbb{E}_{\tilde{\mathcal{D}}} \int_{\mathbf{x}} \mathrm{KL}\Big(p_{w,d}(\mathbf{y}|\mathbf{x}) || \hat{p}(\mathbf{y}|\mathbf{x}) \Big) + \mathrm{KL}\Big(\frac{1}{1 - \tilde{p}(w, d|\mathbf{x})} \sum_{t' \neq (w,d)} \tilde{p}(t'|\mathbf{x}) p_{t'}(\mathbf{y}|\mathbf{x}) || \hat{p}(\mathbf{y}|\mathbf{x}) \Big) d\mathbf{x} \tag{A.9}$$

where KL is the KL divergence and $\hat{p}(\mathbf{y}|\mathbf{x}) = \sum_{t \in \tilde{\mathcal{T}}} \tilde{p}(t|\mathbf{x}) p_t(\mathbf{y}|\mathbf{x})$ where $\tilde{\mathcal{T}}$ is the restriction of $\mathcal{T}$ to the dosages in $\tilde{\mathcal{D}}$. We then note that the KL divergence is minimised if and only if the two densities are equal, and we note by definition of $\hat{p}$ this occurs if and only if $p_{w,d}(\mathbf{y}|\mathbf{x}) = p_{w',d'}(\mathbf{y}|\mathbf{x})$ for all w, d, w', d'. This also directly implies that the marginal distributions for any fixed treatment-dosage pair agree for all factually observed treatments. In particular, if a sample received treatment $t' \neq t$, we have that the counterfactual generated for t for this sample has the same distribution as the true data generating distribution. □

Finally, we prove the following result, from which Theorem 3.4.1 follows immediately.

Theorem A.2.1. *An optimal solution to the game defined by Equations 3.8 - 3.9 is also an optimal solution to the game defined by Equation A.1 if the response curves generated by the generator for different treatments are conditionally independent given the features.*

Proof. To prove this result, it suffices to show that for fixed $\mathbf{G}$, $\mathbf{D}_H^* = \mathbf{D}^*$. To show this, we observe that by the same arguments as given for Lemma A.2.1, we have the following:

$$\mathbf{D}_{\mathcal{W}}^{w\,*}(\mathbf{x}, \mathbf{y}) = \frac{p(w|\mathbf{x})\Big(\sum_{i=1}^{n_w} \tilde{p}(d_i|\mathbf{x}, w) p_{w,d_i}(\mathbf{y}|\mathbf{x})\Big)}{\sum_{w' \in \mathcal{W}} \Big(p(w'|\mathbf{x}) \sum_{i=1}^{n_w} \tilde{p}(d_i|\mathbf{x}, w)\Big)} \tag{A.10}$$

$$\mathbf{D}_{w}^{j\,*}(\mathbf{x}, \mathbf{y}_w) = \frac{\tilde{p}(d_j|\mathbf{x}, w) p_{w,d_j}(\mathbf{y}_w|\mathbf{x})}{\sum_{i=1}^{n_w} \tilde{p}(d_i|\mathbf{x}, w) p_{w,d_i}(\mathbf{y}_w|\mathbf{x})} \tag{A.11}$$

where $\mathbf{y}_w$ is the restriction of $\mathbf{y}$ to the outcomes corresponding to treatment w. By multiplying (A.11) by $\frac{p_{w,d_j}(\mathbf{y}_{\neq w}|\mathbf{y},\mathbf{x})}{p_{w,d_j}(\mathbf{y}_{\neq w}|\mathbf{y},\mathbf{x})}$ we obtain

$$\mathbf{D}_{w}^{j\,*}(\mathbf{x}, \mathbf{y}_w) = \frac{\tilde{p}(d_j|\mathbf{x}, w) p_{w,d_j}(\mathbf{y}|\mathbf{x})}{\sum_{i=1}^{n_w} \tilde{p}(d_i|\mathbf{x}, w) p_{w,d_i}(\mathbf{y}|\mathbf{x})} \tag{A.12}$$

since the cond. independence assumption implies that $p_{w,d_j}(\mathbf{y}_{\neq w}|\mathbf{y}, \mathbf{x}) = p_{w,d_i}(\mathbf{y}_{\neq w}|\mathbf{y}, \mathbf{x})$ for all $i, j = 1, ..., n_w$. Multiplying (A.10) and (A.12) together to get $\mathbf{D}_H^{w,j}$, we notice that the denominator in (A.12) cancels with the bracketed term of the numerator in (A.10) to give

$$\mathbf{D}_H^{*\,w,j} = \frac{p(w|\mathbf{x})\tilde{p}(d_j|\mathbf{x}, w) p_{w,d_j}(\mathbf{y}|\mathbf{x})}{\sum_{w' \in \mathcal{W}} \Big(p(w'|\mathbf{x}) \sum_{i=1}^{n_w} \tilde{p}(d_i|\mathbf{x}, w)\Big)} \tag{A.13}$$

$$= \frac{\tilde{p}(w, d_j|\mathbf{x}) p_{w,d_j}(\mathbf{y}|\mathbf{x})}{\sum_{w' \in \mathcal{W}} \sum_{i=1}^{n_w} \tilde{p}(w', d_i|\mathbf{x}) p_{w',d_i}(\mathbf{y}|\mathbf{x})} \tag{A.14}$$

which is equal to the optimal discriminator for the single loss given in Lemma A.2.1. □

A.3 Counterfactual Generator Pseudo-code

Algorithm 2 provides the pseudo-code for training the Counterfactual Generator part of SCIGAN.

Algorithm 2 Training of the generator in SCIGAN

Input: dataset $\mathcal{C} = \{(\mathbf{x}^i, t_f^i, y_f^i) : i = 1, ..., N\}$, batch size n_{mb}, number of dosages per treatment n_d, number of discriminator updates per iteration n_D, number of generator updates per iteration n_G, dimensionality of noise n_z, learning rate α

Initialize: θ_G, $\theta_{\mathcal{W}}$, $\{\theta_w\}_{w \in \mathcal{W}}$

while G has not converged **do**

Discriminator updates

  for $i = 1, ..., n_D$ **do**

    Sample $(\mathbf{x}_1, (w_1, d_1), y_1), ..., (\mathbf{x}_{n_{mb}}, (w_{n_{mb}}, d_{n_{mb}}), y_{n_{mb}})$ from $\mathcal{C}$

    Sample noise $\mathbf{z}_j = (z_1^j, ..., z_{n_z}^j)$ from $\mathcal{U}([0,1]^{n_z})$ for $j = 1, ..., n_{mb}$

    for $w \in \mathcal{W}$ **do**

      for $j = 1, ..., n_{mb}$ **do**

        Sample $\tilde{D}_w^j = (d_1^{w,j}, ..., d_{n_d}^{w,j})$ from $(\mathcal{D}_w)^{n_d}$

        Set $\tilde{\mathbf{y}}_w^j$ according to Eq. 3.4

      Calculate gradient of dosage discriminator loss

$$g_w \leftarrow \nabla_{\theta_w} - \left[\sum_{\{j:w_j=w\}} \sum_{k=1}^{n_d} \mathbb{I}_{\{d_j = d_k^{w,j}\}} \log \mathbf{D}_w(\mathbf{x}_j, \tilde{\mathbf{y}}_w^j) + \mathbb{I}_{\{d_j \neq d_k^{w,j}\}} \log(1 - \mathbf{D}_w(\mathbf{x}_j, \tilde{\mathbf{y}}_w^j)) \right]$$

      Update dosage discriminator parameters $\theta_w \leftarrow \theta_w + \alpha g_w$

    Set $\tilde{\mathbf{y}}_j = (\tilde{\mathbf{y}}_w^j)_{w \in \mathcal{W}}$

    Calculate gradient of treatment discriminator loss

$$g_{\mathcal{W}} \leftarrow \nabla_{\theta_{\mathcal{W}}} - \left[\sum_{j=1}^{n_{mb}} \sum_{w \in \mathcal{W}} \mathbb{I}_{\{w_j = w\}} \log \mathbf{D}_{\mathcal{W}}(\mathbf{x}_j, \tilde{\mathbf{y}}_j) + \mathbb{I}_{\{w_j \neq w\}} \log(1 - \mathbf{D}_{\mathcal{W}}(\mathbf{x}_j, \tilde{\mathbf{y}}_j)) \right]$$

    Update treatment discriminator parameters $\theta_{\mathcal{W}} \leftarrow \theta_{\mathcal{W}} + \alpha g_{\mathcal{W}}$

  Generator updates

  for $i = 1, ..., n_G$ **do**

    Sample $(\mathbf{x}_1, (w_1, d_1), y_1), ..., (\mathbf{x}_{n_{mb}}, (w_{n_{mb}}, d_{n_{mb}}), y_{n_{mb}})$ from $\mathcal{C}$

    Sample noise $\mathbf{z}_j = (z_1^j, ..., z_{n_z}^j)$ from $\mathcal{U}([0,1]^{n_z})$ for $j = 1, ..., n_{mb}$

    Sample $(\tilde{D}_w^j)_{w \in \mathcal{W}}$ from $\Pi_{w \in \mathcal{W}}(\mathcal{D}_w)^{n_d}$ for $j = 1, ..., n_{mb}$

    Set $\tilde{\mathbf{y}}$ according to Eq. 3.4

    Calculate gradient of generator loss

$$g_G \leftarrow \nabla_{\theta_G} \left[\sum_{j=1}^{n_{mb}} \sum_{w \in \mathcal{W}} \sum_{l=1}^{n_d} \mathbb{I}_{\{w_j = w, d_j = d_l^{w,j}\}} \log(\mathbf{D}_{\mathcal{W}}^w(\mathbf{x}_j, \tilde{\mathbf{y}}_j)_w \times \mathbf{D}_w^l(\mathbf{x}_j, \tilde{\mathbf{y}}_w^j)_l) \right.$$
$$\left. + \mathbb{I}_{\{w_j \neq w, d_j \neq d_l^{w,j}\}} \log(1 - (\mathbf{D}_{\mathcal{W}}^w(\mathbf{x}_j, \tilde{\mathbf{y}}_j) \times D_w^l(\mathbf{x}_j, \tilde{\mathbf{y}}_w^j))) \right]$$

    Update generator parameters $\theta_G \leftarrow \theta_G + \alpha g_G$

Output: G

A.4 Inference Network

To generate dose-response curves for new samples, we learn an inference network, $\mathbf{I} : \mathbf{X} \times \mathcal{T} \to \mathcal{Y}$. This inference network is trained using the original dataset and the learned counterfactual generator. As with the training of the generator and discriminator, we train using a random set of dosages, $\tilde{\mathcal{D}}_w$. The loss is given by

$$\mathcal{L}_I(\mathbf{I}) = \mathbb{E}\left[\sum_{w \in \mathcal{W}} \sum_{d \in \tilde{\mathcal{D}}_w} (\tilde{Y}(w,d) - \mathbf{I}(\mathbf{X},(w,d)))^2\right], \tag{A.15}$$

where $\tilde{Y}(w,d)$ is Y_f if $T_f = (w,d)$ or given by the generator if $T_f \neq (w,d)$. The expectation is taken over $\mathbf{X}, T_f, Y_f, \mathbf{Z}$ and $\tilde{\mathcal{D}}_w$.

Algorithm 3 provides the pseudo-code for training the Inference Network.

Algorithm 3 Training of the inference network in SCIGAN

Input: dataset $\mathcal{C} = \{(\mathbf{x}^i, t_f^i, y_f^i) : i = 1, ..., N\}$, trained generator $\mathbf{G}$, batch size n_{mb}, number of dosages per treatment n_d, dimensionality of noise n_z, learning rate α
Initialize: θ_I
while $\mathbf{I}$ has not converged **do**
  Sample $(\mathbf{x}_1, (w_1, d_1), y_1), ..., (\mathbf{x}_{n_{mb}}, (w_{n_{mb}}, d_{n_{mb}}), y_{n_{mb}})$ from $\mathcal{C}$
  Sample generator noise $\mathbf{z}_j = (z_1^j, ..., z_{n_z}^j)$ from Unif$([0,1]^{n_z})$ for $j = 1, ..., n_{mb}$
  for $j = 1, ..., n_{mb}$ **do**
    for $w \in \mathcal{W}$ **do**
      Sample $\tilde{D}_w^j = (d_1^{w,j}, ..., d_{n_d}^{w,j})$ independently and uniformly from $(\mathcal{D}_w)^{n_d}$
      Set $\tilde{\mathbf{y}}_w^j$ according to Eq. 3.4 3.4
  Calculate gradient of inference network loss

$$g_I \leftarrow -\nabla_{\theta_I}\left[\sum_{j=1}^{n_{mb}} \sum_{w \in \mathcal{W}} \sum_{l=1}^{n_d} (\tilde{\mathbf{y}}_w^j)_l - \mathbf{I}(\mathbf{x}_j, (w, d_l^{w,j}))^2\right]$$

  Update inference network parameters $\theta_I \leftarrow \theta_I + \alpha g_I$
Output: $\mathbf{I}$

A.5 Architecture

Definitions of Permutation Invariance and Permutation Equivariance

The notions of what it means for a function to be *permutation invariant* and *permutation equivariant* with respect to (a subset of) its inputs are given below in definitions A.5.1

and A.5.2, respectively. Let $\mathcal{U}$, $\mathcal{V}$, $\mathcal{C}$ be some spaces. Let $m \in \mathbb{Z}^+$.

Definition A.5.1. *A function $f : \mathcal{U}^m \times \mathcal{V} \to \mathcal{C}$ is permutation* invariant *with respect to the space $\mathcal{U}^m$ if for every $\mathbf{u} = (u_1, ..., u_m) \in \mathcal{U}^m$, every $v \in \mathcal{V}$ and every permutation, σ, of $\{1, ..., m\}$ we have $f(u_1, ..., u_m, v) = f(u_{\sigma(1)}, ..., u_{\sigma(m)}, v)$.*

Definition A.5.2. *A function $f : \mathcal{U}^m \times \mathcal{V} \to \mathcal{C}^m$ is permutation* equivariant *with respect to the space $\mathcal{U}^m$ if for every $\mathbf{u} \in \mathcal{U}^m$, every $v \in \mathcal{V}$ and every permutation, σ, of $\{1, ..., m\}$ we have $f(u_{\sigma(1)}, ..., u_{\sigma(m)}, v) = (f_{\sigma(1)}(\mathbf{u}, v), ..., f_{\sigma(m)}(\mathbf{u}, v))$, where $f_j(\mathbf{u}, v)$ is the jth element of $f(\mathbf{u}, v)$.*

To build up functions that are permutation invariant and permutation equivariant we make the following observations: (1) the composition of any function with a permutation invariant function is permutation invariant, (2) the composition of two permutation equivariant functions is permutation equivariant.

As noted in Section 3.5.2, the basic building block we use for equivariant functions is defined in terms of equivariance input, $\mathbf{u}$, and auxiliary input, $\mathbf{v}$ by

$$f_{equi}(\mathbf{u}, \mathbf{v}) = \sigma(\lambda \mathbf{I}_m \mathbf{u} + \gamma(\mathbf{1}_m \mathbf{1}_m^T)\mathbf{u} + (\mathbf{1}_m \Theta^T)\mathbf{v}) \,. \tag{A.16}$$

A.6 Single Discriminator Model

In the paper we developed a hierarchical discriminator and demonstrated that it performs significantly better than the single discriminator setup that we now describe in this section.

Single Discriminator

In the single model, we learn a single discriminator, $\mathbf{D}$, that outputs $\mathbb{P}((W_f, D_f) = (w, d) | \mathbf{X}, \tilde{\mathcal{D}}_w, \tilde{\mathbf{Y}})$ for each $w \in \mathcal{W}$ and $d \in \tilde{\mathcal{D}}_w$. We will write $\mathbf{D}^{w,d}(\cdot)$ to denote the output of $\mathbf{D}$ that corresponds to the treatment-dosage pair (w, d). We define the loss, $\mathcal{L}_D$, to be

$$\mathcal{L}_D(\mathbf{D}; \mathbf{G}) = -\mathbb{E}\Bigg[\sum_{w \in \mathcal{W}} \sum_{d \in \tilde{\mathcal{D}}_w} \mathbb{I}_{\{T_f = (w,d)\}} \log \mathbf{D}^{w,d}(\mathbf{X}, \tilde{\mathbf{Y}}) + \mathbb{I}_{\{T_f \neq (w,d)\}} \log(1 - \mathbf{D}^{w,d}(\mathbf{X}, \tilde{\mathbf{Y}})) \Bigg] \tag{A.17}$$

where the expectation is taken over $\mathbf{X}, \{\tilde{\mathcal{D}}_w\}_{w\in\mathcal{W}}, \tilde{\mathbf{Y}}, W_f$ and D_f and we note that the dependence on $\mathbf{G}$ is through $\tilde{\mathbf{Y}}$. Our single discriminator will be trained to minimise this loss directly. The generator GAN-loss, $\mathcal{L}_G$, is then defined by

$$\mathcal{L}_G(\mathbf{G}) = -\mathcal{L}_D(\mathbf{D}^*; \mathbf{G}) \tag{A.18}$$

where $\mathbf{D}^*$ is the optimal discriminator given by minimising $\mathcal{L}_D$. The generator will be trained to minimise $\mathcal{L}_G + \lambda\mathcal{L}_S$.

Single Discriminator Architecture

In the case of the single discriminator, we want the output of $\mathbf{D}$ corresponding to each treatment $w \in \mathcal{W}$, i.e. $(\mathbf{D}^{w,1}, ..., \mathbf{D}^{w,n_w})$, to be permutation equivariant with respect to $\tilde{\mathbf{y}}_w$ and permutation invariant with respect to each $\tilde{\mathbf{y}}_v$ for $v \in \mathcal{W} \setminus \{w\}$. To achieve this, we first define a function $f : \prod_{w\in\mathcal{W}}(\mathcal{D}_w \times \mathcal{Y})^{n_w} \to \mathcal{H}_S$ and require that this function be permutation invariant with respect to each of the spaces $(\mathcal{D}_w \times \mathcal{Y})^{n_w}$. For each treatment, $w \in \mathcal{W}$, we introduce a multitask head, $f_w : \mathcal{X} \times \mathcal{H}_S \times (\mathcal{D}_w \times \mathcal{Y})^{n_w} \to [0,1]^{n_w}$, and require that each of these functions be permutation equivariant with respect to their corresponding input space $(\mathcal{D}_w \times \mathcal{Y})^{n_w}$ but they can depend on the features, $\mathbf{x} \in \mathcal{X}$, and invariant latent representation coming from f arbitrarily. Writing f_w^j to denote the jth output of f_w, the output of the discriminator given input features, $\mathbf{x}$, and generated outcomes, $\tilde{\mathbf{y}}$, is defined by

$$\mathbf{D}^{w,j}(\mathbf{x}, \tilde{\mathbf{y}}) = f_w^i(\mathbf{x}, f(\tilde{\mathbf{y}}), \tilde{\mathbf{y}}_w). \tag{A.19}$$

To construct the function f, we concatenate the outputs of several invariant layers of the form given in Eq. A.16 that each individually act on the spaces $(\mathcal{D}_w \times \mathcal{Y})^{n_w}$. That is, for each treatment, $w \in \mathcal{W}$ we define a map $f_{inv}^w : (\mathcal{D}_w \times \mathcal{Y})^{n_w} \to \mathcal{H}_S^w$ by substituting $\tilde{\mathbf{y}}_w$ for $\mathbf{u}$ in Eq. A.16. We then define $\mathcal{H}_S = \prod_{w\in\mathcal{W}} \mathcal{H}_S^w$ and $f(\tilde{\mathbf{y}}) = (f_{inv}^{w_1}(\tilde{\mathbf{y}}_{w_1}), ..., f_{inv}^{w_k}(\tilde{\mathbf{y}}_{w_k}))$.

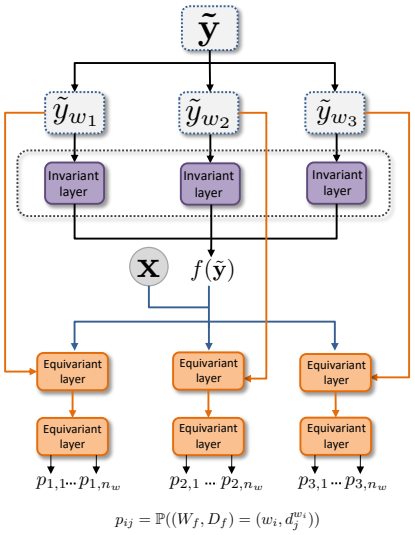
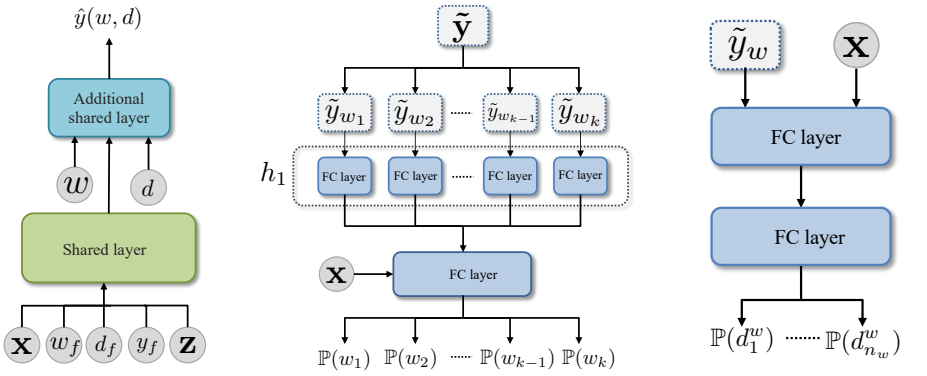


Figure A.1: Overview of the single discriminator architecture.

Each f_w will consist of two layers of the form given in Eq. (A.16) with the equivariance input, $\mathbf{u}$, to first layer being $\tilde{\mathbf{y}}_w$ and to the second layer being the output of the first layer and the auxiliary input, $\mathbf{v}$, to the first layer being the concatenation of the features and invariant representation, i.e. $(\mathbf{x}, f(\tilde{\mathbf{y}}))$ and then no auxiliary input to the second layer. A diagram depicting the architecture of the single discriminator model can be found in Fig. A.1.

A.7 Ablation Studies Architectures



(a) Generator without multitask heads

(b) Treatment discriminator with fully connected (FC) layers instead of invariant layers

(c) Dosage discriminator with FC layers instead of equivariant layers

Figure A.2: Architecture of the generator without multitask and discriminator without the invariant/equivariant layers used in the ablation studies.

A.8 Dataset Descriptions

TCGA: The TCGA dataset consists of gene expression measurements for cancer patients [129]. There are 9659 samples for which we used the measurements from the 4000 most variable genes. The gene expression data was log-normalized and each feature was scaled in the $[0, 1]$ interval. For each patient, the features were scaled to have norm 1. We give meaning to our treatments and dosages by considering the treatment as being chemotherapy/radiotherapy/immunotherapy and their corresponding dosages. The outcome can be thought of as the risk of cancer recurrence [36]. We used the same version of the TCGA dataset as used by DRNet https://github.com/d909b/drnet.

News: The News dataset consists of word counts for news items. We extracted 10000 news items (randomly sampled) each with 2858 features. As in [29, 36], we give meaning to our treatments and dosages by considering the treatment as being the viewing device (e.g. phone, tablet etc.) used to read the article and the dosage as being the amount of time spent reading it. The outcome can be thought of as user satisfaction. We used the same version of the News dataset as used by DRNet https://github.com/d909b/drnet.

MIMIC III: The Medical Information Mart for Intensive Care (MIMIC III) [130] database consists of observational data from patients in the ICU. We extracted 3000 patients (randomly sampled) that receive antibiotics treatment and we used as features 9 clinical covariates, namely age, temperature, heart rate, systolic and diastolic blood pressure, SpO2, FiO2, glucose, and white blood cell count, measured at start of ICU stay. Again, the features were scaled in the $[0, 1]$ interval. In this setting, we can considered as treatments the different antibiotics and their corresponding dosages.

For a summary description of the datasets, see table A.1. The datasets are split into 64/16/20% for training, validation and testing respectively. The validation dataset is used for hyperparameter optimization.

	TCGA	**News**	**MIMIC**
Number of samples	9659	10000	3000
Number of features	4000	2858	9
Number of treatments	3*	3	2

Table A.1: Summary description of datasets. *: for our final experiment in Appendix A.12 we increase the number of treatments in TCGA to 6 and 9.

A.9 Dosage Bias

In order to create dosage-assignment bias in our datasets, we assign dosages according to $d_w|\mathbf{x} \sim \text{Beta}(\alpha, \beta_w)$. The selection bias is controlled by the parameter $\alpha \geq 1$. When we set $\beta_w = \frac{\alpha-1}{d_w^*} + 2 - \alpha$ (which ensures that the mode of our distribution is d_w^*), we

can write the variance of d_w in terms of α and d_w^* as follows

$$\text{Var}(d_w) = \frac{\frac{\alpha^2-\alpha}{d_w^*} + 2\alpha - \alpha^2}{(\frac{\alpha-1}{d_w^*} + 2)^2(\frac{\alpha-1}{d_w^*} + 3)} \approx \frac{c\alpha^2}{d\alpha^3}\,. \tag{A.20}$$

We see that the variance of our Beta distribution therefore decreases with α, resulting in the sampled dosages being closer to the optimal dosage, thus resulting in higher dosage-selection bias. In addition we note that the Beta$(1, 1)$ distribution is in fact the uniform distribution, corresponding to the dosages being sampled independently of the patient features, resulting in no selection bias when $\alpha = 1$.

A.10 Benchmarks

We use the publicly available GitHub implementation of DRNet provided by [36]: `https://github.com/d909b/drnet`. Moreover, we also used a GPS implementation similar to the one from `https://github.com/d909b/drnet` which uses the `causaldrf` R package [264]. More spcifically, the GPS implementation uses a normal treatment model, a linear treatment formula and a 2-nd degree polynomial for the outcome. Moreover, for the TCGA and News datasets, we performed PCA and only used the 50 principal components as input to the GPS model to reduce computational complexity.

Hyperparameter Optimization: The validation split of the dataset is used for hyperparameter optimization. For the DRNet benchmarks we use the same hyperparameter optimization proposed by [36]. For SCIGAN, we use the hyperparameter optimization method proposed in GANITE [35], where we use the complete dataset from the counterfactual generator to evaluate the MISE on the inference network. We perform a random search [265] for hyperparameter optimization over the search ranges in Table A.2. The number of hidden layers per multitask head is set to 2, the number of dosage samples is set to 5, λ to 1 and we use Adam Moment Optimization. For all experiments with SCIGAN, we used 5000 training iterations for the GAN network and 10000 training iterations for the inference network. This number of training iterations was chosen to ensure convergence of the generator loss, discriminator loss, as well as of the supervised

loss. For a fair comparison, for the MLP-M model we used the same architecture used in the inference network of SCIGAN. Similarly, for the MLP model we use the same architecture as for the MLP-M, but without the multitask heads. The experiments were run on a system with 6CPUs, an Nvidia K80 Tesla GPU and 56GB of RAM.

Hyperparameter	Search range
Batch size	64, 128, 256
Number of units per hidden layer	32, 64, 128
Size of invariant and equivariant representations	16, 32, 64, 128

Table A.2: Hyperparameters search range for SCIGAN.

The hyperparameters used to generate the results for SCIGAN are given in Table A.3.

Hyperparameter	TCGA	News	MIMIC
Batch size	128	256	128
Number of units per hidden layer	64	128	32
Size of invariant and equivariant representations	16	32	16

Table A.3: Hyperparameters used for obtaining results.

A.11 Metrics

The Mean Integrated Squared Error (MISE) measures how well the models estimates the patient outcome across the entire dosage space:

$$\text{MISE} = \frac{1}{N}\frac{1}{k}\sum_{w\in\mathcal{W}}\sum_{i=1}^{N}\int_{\mathcal{D}_w}\left(y^i(w,u)-\hat{y}^i(w,u)\right)^2 \mathrm{d}u\,. \tag{A.21}$$

In addition to this, we also compute the mean dosage policy error (DPE) [36] to assess the ability of the model to estimate the optimal dosage point for every treatment for each individual:

$$\text{DPE} = \frac{1}{N}\frac{1}{k}\sum_{w\in\mathcal{W}}\sum_{i=1}^{N}\left(y^i(w,d_w^*)-y^i(w,\hat{d}_w^*)\right)^2, \tag{A.22}$$

where d_w^* is the true optimal dosage and $\hat{d}_w^*$ is the optimal dosage identified by the model. The optimal dosage points for a model are computed using SciPy's implementation of Sequential Least SQuares Programming.

Finally, we compute the mean policy error (PE) [36] which compares the outcome of the true optimal treatment-dosage pair to the outcome of the optimal treatment-dosage pair as selected by the model:

$$\text{PE} = \frac{1}{N}\sum_{i=1}^{N}\left(y^i(w^*, d_w^*) - y^i(\hat{w}^*, \hat{d}_w^*)\right)^2, \tag{A.23}$$

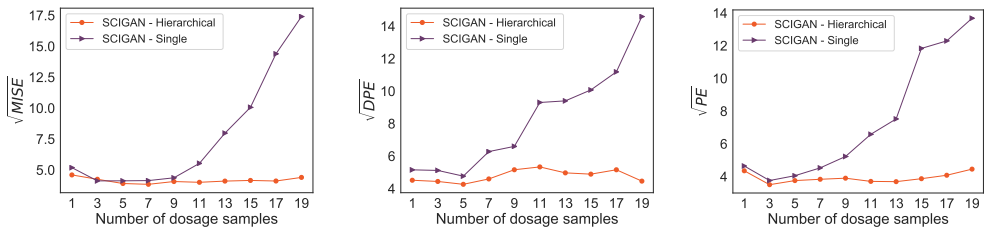
where w^* is the true optimal treatment and $\hat{w}^*$ is the optimal treatment identified by the model. The optimal treatment-dosage pair for a model is selected by first computing the optimal dosage for each treatment and then selecting the treatment with the best outcome for its optimal dosage. Each of these metrics are computed on a held out test-set.

A.12 Additional Results

Investigating Hyperparameter Sensitivity (n_w)

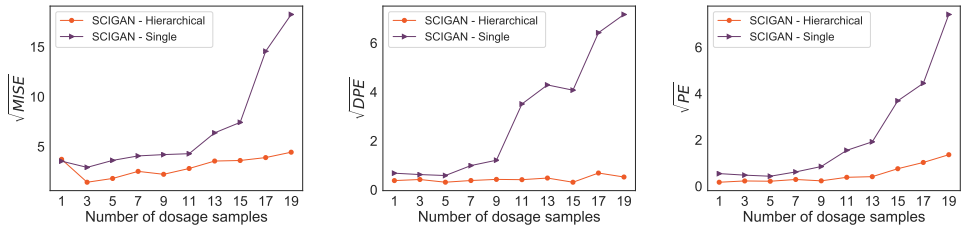
The performance of the single discriminator causes significant performance drops around $n_w = 9$ across all metrics. As previously noted, this is due to the dimension of the output space (which for $n_w = 9$ is 27) being too large. Conversely, we see that our hierarchical discriminator shows much more stable performance even when $n_w = 19$.

Here we present additional results for our investigation of the hyperparameters n_w. Fig. A.3 reports each of the 3 performance metrics as we increase the number of dosage samples, n_w, used to train the discriminators on the News dataset. As with the TCGA results in the main paper we see that the single discriminator suffers a significant performance decrease when n_w is set too high.



(a) $\sqrt{\text{MISE}}$ **(b)** $\sqrt{\text{DPE}}$ **(c)** $\sqrt{\text{PE}}$

Figure A.3: Performance of single vs. hierarchical discriminator when increasing the number of dosage samples (n_w) on News dataset.



(a) $\sqrt{\text{MISE}}$ **(b)** $\sqrt{\text{DPE}}$ **(c)** $\sqrt{\text{PE}}$

Figure A.4: Performance of single vs. hierarchical discriminator when increasing the number of dosage samples (n_w) on TCGA dataset.

Dosage Policy Error for Benchmark Comparison

In Table A.4 we report the Dosage Policy Error (DPE) corresponding to Section 3.6.3.

Methods	TCGA $\sqrt{\text{DPE}}$	News $\sqrt{\text{DPE}}$	MIMIC $\sqrt{\text{DPE}}$
SCIGAN	$\mathbf{0.31} \pm 0.05$	$\mathbf{4.14} \pm 0.11$	$\mathbf{0.51} \pm 0.05$
DRNet	$0.51 \pm 0.05^*$	$4.39 \pm 0.11^*$	0.52 ± 0.05
DRN-W	$0.50 \pm 0.05^*$	4.21 ± 0.11	0.53 ± 0.05
GPS	$1.38 \pm 0.01^*$	$6.40 \pm 0.01^*$	$1.41 \pm 0.12^*$
MLP-M	$0.92 \pm 0.05^*$	$4.94 \pm 0.16^*$	$0.77 \pm 0.05^*$
MLP	$1.04 \pm 0.05^*$	$5.18 \pm 0.12^*$	$0.80 \pm 0.05^*$

Table A.4: Performance of individualized treatment-dose response estimation on three datasets. Bold indicates the method with the best performance for each dataset. *: performance improvement is statistically significant.

Varying the Number of Treatments

In this experiment, we increase the number of treatments by defining 3 or 6 additional treatments. The parameters $\mathbf{v}_1^w, \mathbf{v}_2^w, \mathbf{v}_3^w$ are defined in exactly the same way as for 3 treatments. The outcome shapes for treatments 4 and 7 are the same as for treatment 1, similarly for 5, 8 and 2 and for 6, 9 and 3. In Table A.5 we report MISE, DPE and PE on the TCGA dataset with 6 treatments (TCGA-6) and with 9 treatments (TCGA-9). Note that we use 3 dosage samples for training SCIGAN in this experiment.

Sample Efficiency

We have also performed a further experiment to evaluate model performance in terms of sample efficency. For the MIMIC dataset, in Table 1 we report evaluation metrics

Method	TCGA - 6			TCGA - 9		
	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$
SCIGAN	2.37 ± 0.12	0.43 ± 0.05	0.32 ± 0.05	2.79 ± 0.05	0.51 ± 0.05	0.54 ± 0.05
DRNET	4.09 ± 0.16	0.52 ± 0.05	0.71 ± 0.05	4.31 ± 0.12	0.59 ± 0.05	0.74 ± 0.05
GPS	6.62 ± 0.01	2.04 ± 0.01	2.61 ± 0.00	7.58 ± 0.01	3.14 ± 0.01	2.91 ± 0.01

Table A.5: Performance of SCIGAN and the benchmarks when we increase the number of treatments to 6 and 9. Bold indicates the method with the best performance for each dataset.

for training SCIGAN with different number of training samples N and evaluating on the same test set.

	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$
$N = 100$	31.12 ± 63.39	7.72 ± 2.57	18.94 ± 29.07
$N = 500$	13.36 ± 10.46	4.07 ± 1.92	2.63 ± 0.94
$N = 1000$	3.80 ± 1.04	2.46 ± 1.75	1.03 ± 1.13
$N = 1500$	2.95 ± 0.37	0.70 ± 0.17	0.63 ± 0.12
$N = 1920$	2.09 ± 0.12	0.51 ± 0.05	0.32 ± 0.05

Table A.6: Sample efficiency analysis for MIMIC. Metrics are reported as Mean $\pm$ Std.

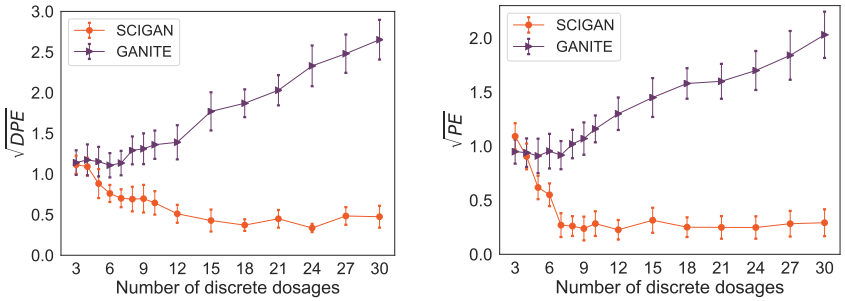
Discrete Dosage Set-up and Comparison with GANITE

In this set-up, we use the TCGA dataset and treatments 2 and 3 from Table 3.1 with dose-response curves $f_2(\mathbf{x}, d)$ and $f_3(\mathbf{x}, d)$ respectively. Let β be the number of discrete dosages for which we want to generate data. We chose β equally spaced points in the interval $[0, 1]$ as our set of discrete dosages: $\Delta = \{\frac{k}{\beta-1}\}_{k=0}^{\beta-1}$. To create factual dosages for our dataset, we sample the dosages as before $d_w \mid x \sim \text{Beta}(\alpha, \beta_w)$ (as described in Appendix A.9), and choose the closest discrete dosage from the set Δ.

To evaluate SCIGAN in this setting, we maintain the same architecture for the multi-task generator and hierarchical discriminator. The only difference is that we now randomly sample dosages for the SCIGAN discriminator from Δ.

We adopt the GANITE implementation proposed by [35]. To be able to have a fair comparison with SCIGAN we also use a multi-task architecture for the GANITE generator and we give as input to each multitask head the dosage parameter. The GANITE generator will generate outcomes for all possible discrete dosages in Δ and these will be passed to the GANITE discriminator to distinguish the factual one. For the GANITE

generator we use a similar architecture to the SCIGAN generator with 2 hidden layers for each multitask head and 64 neurons in each layer. The GANITE discriminator consists of 2 fully connected layers with 64 neurons in each. We also set $\lambda = 1$. In addition, to maintain a similar set-up to SCIGAN, we train an inference network to learn the counterfactual outcomes with data from the GANITE generator. The inference network has the same architecture as the GANITE generator.



(a) $\sqrt{\text{DPE}}$ **(b)** $\sqrt{\text{PE}}$

Figure A.5: Comparison between SCIGAN and GANITE in the discrete dosage set-up.

We clearly see from Fig. A.5 that SCIGAN achieves a similar performance to GANITE for a small number of dosages (< 6) but then significantly outperforms GANITE for more dosages than 6. In fact, we see that while GANITE's performance degrades with an increasing number of dosages, SCIGAN's improves and then stabilises at around 12 dosages. This is due to the fact that the single discriminator in GANITE simply cannot handle a large number of dosages. Our hierarchical model, however, can. The worse performance of SCIGAN for the lower dosages can be attributed to the fact that for such few dosages (e.g. 3 dosages corresponds to only 6 total different interventions), the SCIGAN architecture is overly complex, and the sub-sampling of dosages for the discriminator is not actually necessary.

Mixing Dosage and No-dosage Treatment Options

We also evaluate the case when one of the treatments does not have a dosage parameter. For this experiment we generate data for the treatment that has a dosage parameter d using $f_3(\mathbf{x}, d)$ and for the treatment without an associated dosage using $2C(\mathbf{v}_0^T\mathbf{x})$, where $\mathbf{v}_0$ are parameters, $\mathbf{x}$ are patient features and $C =$ is the scaling parameter. This set-up

also corresponds to the scenario where we want to compare giving a treatment with a dosage and not giving any treatment.

SCIGAN can be easily extended to incorporate an additional treatment that does not come with a dosage parameter. Such treatments will not need a dosage discriminator but will be passed to the treatment discriminator. A head can be added to the generator for each such non-dosage treatment but will not need to take dosage as an input. As the DRNet public implementation does not allow for this set-up, we compared SCIGAN with the multilayer perceptron model with multitask heads (MLP-M). This model is trained using supervised learning to minimize error on the factual outcomes and consists of two multitask heads: one head for the treatment option which receives as input the dosage and estimates the dose-response curve and one head for the no-treatment option.

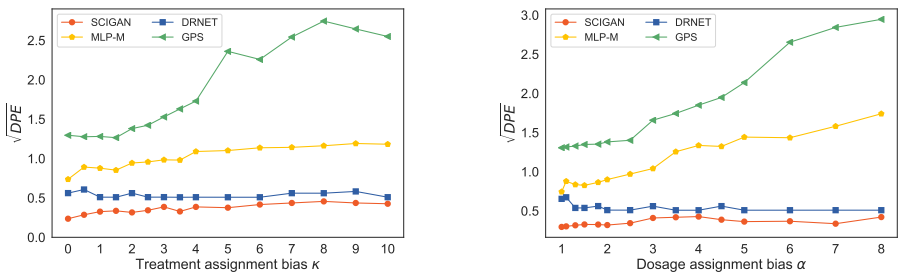
As can be seen in Table A.7, SCIGAN is capable of handling this setting and lends itself naturally to potentially mixed dosage and no-dosage treatment options.

Method	**TCGA**			**News**			**MIMIC**		
	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$
SCIGAN	**1.28** $\pm$ 0.09	**1.37** $\pm$ 0.07	**1.56** $\pm$ 0.06	**3.18** $\pm$ 0.15	**2.04** $\pm$ 0.09	**2.49** $\pm$ 0.12	**0.61** $\pm$ 0.08	**1.82** $\pm$ 0.02	**1.89** $\pm$ 0.03
MLP-M	2.08 $\pm$ 0.12	1.85 $\pm$ 0.16	2.02 $\pm$ 0.07	4.68 $\pm$ 0.11	2.45 $\pm$ 0.08	2.64 $\pm$ 0.08	1.56 $\pm$ 0.08	2.04 $\pm$ 0.03	2.14 $\pm$ 0.5

Table A.7: Performance of individualized treatment-dose response when mixing treatment with no-treatment options. Bold indicates the method with the best performance for each dataset.

Additional Results on Selection Bias

In Fig. A.6 we report the DPE for our treatment and dosage bias experiment from Section 3.6.5 of the main paper.



(a) Treatment selection bias **(b)** Dosage selection bias

Figure A.6: Additional performance metrics of the 4 methods on datasets with varying bias levels on TCGA dataset

B

Appendix for Chapter 4

Contents

B.1 Counterfactual Recurrent Network 163
B.1.1 Extended Related Work 163
B.1.2 Assumptions 164
B.1.3 Time-dependent Confounding 165
B.1.4 Proof of Theorem 4.1.1 167
B.1.5 Training Procedure for CRN 168
B.1.6 Pharmacokinetic-Pharmacodynamic Model of Tumour Growth 170
B.1.7 Marginal Structural Models 171
B.1.8 Recurrent Marginal Structural Networks 172
B.1.9 Baseline RNN and Linear Model 173
B.1.10 Hyperparameter Optimization for CRN 174
B.1.11 Full Results for Counterfactual Prediction 176
B.1.12 Test Set Generation for Evaluating Timing of Treatment . . . 177
B.1.13 Results on Factual Prediction on MIMIC III 178
B.2 Time Series Deconfounder 180
B.2.1 Proof for Theorem 4.2.1 180
B.2.2 Implementation Details for the Factor Model 183
B.2.3 Baselines for Evaluating Factor Model 184
B.2.4 Outcome Models 186
B.2.5 Additional Results 188
B.2.6 Discussion 190

B.1 Estimating Counterfactual Treatment Outcomes over Time through Adversarially Balanced Representations

B.1.1 Extended Related Work

Causal inference in the static setting: A large number of methods have been proposed to learn treatment effects from observational data in the static setting. In this case, it is needed to adjust for the selection bias; bias caused by the fact that, in the observational dataset, the treatment assignments depend on the patient features. Several ways of handling the selection bias involve using propensity matching [266–268], building representations where treated and un-treated populations had similar distributions [28, 29, 102, 132] or performing propensity-aware hyperparameter tuning [26, 31]. However, these methods for the static setting cannot be extended directly to time-varying treatments [141, 269].

Learning optimal policies: A related problem to ours involves learning the optimal treatment policies from logged data [260, 270, 271]. That is, learning the treatment option that would give the best reward. Note the difference to the causal inference setting considered in this paper, where the aim is to learn the counterfactual patient outcomes under all possible treatment options. Learning all of the counterfactual outcomes is a harder problem and can also be used for finding the optimal treatment.

A method for learning optimal policies, proposed by [271] uses domain adversarial training to build a representation that is invariant to the following two domains: observational data and simulated randomized clinical trial data, where the treatments have equal probabilities. [271] only considers the static setting and aims to choose the optimal treatment instead of estimating all of the counterfactual outcomes. In our paper the aim is to eliminate the bias from the time-dependent confounders and reliably estimate *all of the potential outcomes*; thus, at each timestep t we build a representation that is invariant to the treatment.

Off-policy evaluation in reinforcement learning: In reinforcement learning, a similar problem to ours is off-policy evaluation, which uses retrospective observational data, also known as logged bandit feedback [272–274]. In this case, the retrospective

observational data consists of sequences of states, actions and rewards which were generated by an agent operating under an unknown policy. The off-policy evaluation methods aim to use this data to estimate the expected reward of a target policy. These methods use algorithms based on importance sampling [275–277], action-value function approximation (model based) [278] or doubly robust combination of both approaches [279]. Nevertheless, these methods focus on obtaining average rewards of policies, while in our case the aim is to estimate individualized patient outcomes for future treatments.

B.1.2 Assumptions

The standard assumptions needed for identifying the treatment effects are [40, 81, 117]:

Assumption 1: Consistency. If $\mathbf{A}_t = \mathbf{a}_t$ for a given patient, then the potential outcome for treatment $\mathbf{a}_t$ is the same as the observed (factual) outcome: $\mathbf{Y}_{t+1}[\mathbf{a}_t] = \mathbf{Y}_{t+1}$.

Assumption 2: Positivity (Overlap) [110]: If $P(\bar{\mathbf{A}}_{t-1} = \bar{\mathbf{a}}_{t-1}, \bar{\mathbf{X}}_t = \bar{\mathbf{x}}_t) \neq 0$ then $P(\mathbf{A}_t = \mathbf{a}_t \mid \bar{\mathbf{A}}_{t-1} = \bar{\mathbf{a}}_{t-1}, \bar{\mathbf{X}}_t = \bar{\mathbf{x}}_t) > 0$ for all $\bar{\mathbf{a}}_t$.

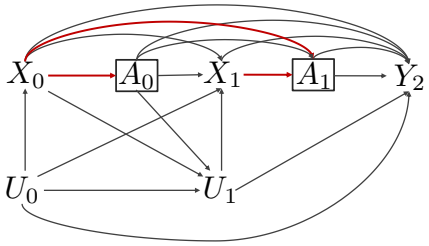
Assumption 3: Sequential strong ignorability. $\mathbf{Y}_{t+1}[\mathbf{a}_t] \perp\!\!\!\perp \mathbf{A}_t \mid \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t, \forall \mathbf{a}_t \in \mathcal{A}, \forall t$.

Assumption 2 means that, for each timestep, each treatment has non-zero probability of being assigned. Assumption 3 means that there are no hidden confounders, that is, all of covariates affecting both the treatment assignment and the outcomes are present in the the observational dataset. Note that while assumption 3 is standard across all methods for estimating treatment effects, it is not testable in practice [14, 280].

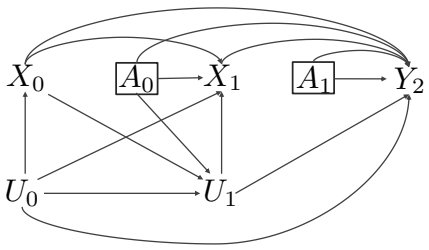
B.1.3 Time-dependent Confounding

Figure B.1 illustrates the causal graphs for a time-varying exposures with 2-steps [14]. In Figure B.1 (a), the covariate X is a time-dependent confounder because it affects the treatment assignments and at the same time, its value is changed by past treatments [16], as illustrated by the red arrows. Thus, the treatment probabilities at each time t depend on the history of covariate X and past treatments. Note that U_0 and U_1 are hidden variables which only affect the covariates, i.e. they do not have arrows into the treatments. Thus, the no hidden confounders assumption (Assumption 4.2.3) is satisfied.

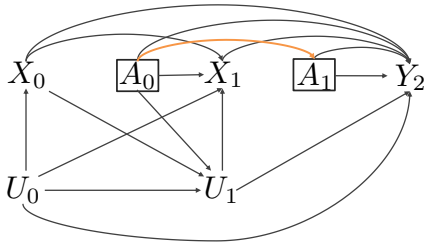
Figure B.1 (a) and (b) illustrate the two cases when there is no bias from time-dependent confounding. In Figure B.1 (a) the treatment probabilities are independent, while in Figure B.1 (b) they depend on past treatments.



(a) Bias from time-dependent confounders.



(a) Option 1 for removing bias from time-dependent confounders.



(a) Option 2 for removing bias from time-dependent confounders.

Figure B.1: Causal graphs for 2-step time-varying exposures [14]. X_0, X_1 are patient covariates, A_0, A_1 are treatments, U_0, U_1 are unobserved variable and Y_2 is the outcome.

Marginal Structural Models: To remove the association between time-dependent confounders and time-varying treatments, Marginal Structural Models propose using inverse probability of treatment weighting (IPTW) [14]. Without loss of generality, consider the use of MSMs with univariate treatments, baseline variables and outcomes. The outcome after t timesteps is parametrized as follows: $\mathbf{E}[Y_{t+1} \mid \mathbf{a}_1, \dots \mathbf{a}_t, V] = g(\mathbf{a}_1, \dots \mathbf{a}_n, V; \theta)$, where $g(\cdot)$ is usually a linear function with parameters θ. To remove

the bias from the time-dependent confounders present in the observational dataset, in the regression model $g(\cdot)$ MSMs weights each patients using either stabilized weights:

$$SW(t) = \prod_{l=1}^{t} \frac{f(\mathbf{A}_l \mid \bar{\mathbf{A}}_{l-1})}{f(\mathbf{A}_l \mid \bar{\mathbf{X}}_l, \bar{\mathbf{A}}_{l-1}, \mathbf{V})} \tag{B.1}$$

or unstabilized weights:

$$W(t) = \prod_{l=1}^{t} \frac{1}{f(\mathbf{A}_l \mid \bar{\mathbf{X}}_l, \bar{\mathbf{A}}_{l-1}, \mathbf{V})}, \tag{B.2}$$

where $f(\cdot)$ represents the conditional probability mass function for discrete treatments.

Inverse probability of treatment weighting (IPTW) creates a pseudo-population where each member consists of themselves and $W - 1$ (or $SW - 1$) copies added though weighting. In this pseudo-population, Robins [142] shows that $\bar{\mathbf{X}}_t$ does not predict treatment $\mathbf{A}_t$, thus removing the bias from time-dependent confounders.

When using unstabilized weights W, the causal graph in the pseudo-population is the one in Figure B.1 (a) where $P(\mathbf{A}_t \mid \bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}, V) = P(\mathbf{A}_t)$. On the other hand, when using stabilized weights SW, causal graph in the pseudo-population is the one in Figure B.1 (b) where $P(\mathbf{A}_t \mid \bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}, V) = P(\mathbf{A}_t \mid \bar{\mathbf{A}}_{t-1})$.

Counterfactual Recurrent Networks. Instead of using IPTW, we proposed building a representation of $\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}, V$ that is not predictive of treatment $\mathbf{A}_t$. At timestep t, we have k different possible treatments $\mathbf{A}_t \in \{A_1, \dots A_K\}$. We build a representation of the history and covariates and treatments that has the same distribution across the different possible treatments: $P(\Phi(\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}, \mathbf{V}) \mid \mathbf{A}_t = A_1) = \cdots = P(\Phi(\bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}, \mathbf{V}) \mid \mathbf{A}_t = A_K)$. By breaking the association between past exposure and current treatments $\mathbf{A}_t$, we satisfy the causal graph in Figure B.1 (a) and thus we remove the bias from time-dependent confounders.

B.1.4 Proof of Theorem 4.1.1

We first prove the following proposition.

Proposition B.1.1. *For fixed Φ, let $x' = \Phi(\bar{\mathbf{h}}_t)$. Then the optimal prediction probabilities of G_a are given by*

$$G_a^{j\,*}(x') = \frac{P_j^\Phi(x')}{\sum_{i=1}^K P_i^\Phi(x')}\,. \tag{B.3}$$

Proof. For fixed Φ, the optimal prediction probabilities are given by

$$G_a^* = \arg\max_{G_a} \sum_{j=1}^K \int_{x'} \log(G_a^j(x'))P_j^\Phi(x')dx' \qquad \text{subject to } \sum_{j=1}^K G_a^j(x') = 1\,. \tag{B.4}$$

Maximising the value function pointwise and applying Lagrange multiplies, we get

$$G_a^* = \arg\max_{G_a} \sum_{j=1}^K \log(G_a^j(x'))P_j^\Phi(x') + \lambda\left(\sum_{j=1}^K G_a^j(x') - 1\right). \tag{B.5}$$

Setting the derivative (w.r.t. $G_a^{j\,*}(x')$) to 0 and solving for $G_a^{j\,*}(x')$ we get

$$G_a^{j\,*}(x') = -\frac{P_j^\Phi(x')}{\lambda} \tag{B.6}$$

where λ can now be solved for using the constraint to be $\lambda = -\sum_{i=1}^K P_i^\Phi(x')$. This gives the result. □

Proof. (of **Theorem 4.1.1**) By substituting the expression from Proposition 1 into the minimax game defined in Eq. 4.6, the objective for Φ becomes

$$\min_\Phi \sum_{j=1}^K \mathbb{E}_{x'\sim P_j^\Phi}\left[\log\left(\frac{P_j^\Phi(x')}{\sum_{i=1}^K P_i^\Phi(x')}\right)\right]. \tag{B.7}$$

We then note that

$$\begin{aligned}
&\sum_{j=1}^K \mathbb{E}_{x'\sim P_j^\Phi}\left[\log\left(\frac{P_j^\Phi(x')}{\sum_{i=1}^K P_i^\Phi(x')}\right)\right] + K\log K = \\
&\qquad = \sum_{j=1}^K \left(\mathbb{E}_{x'\sim P_j^\Phi}\left[\log\left(\frac{P_j^\Phi(x')}{\sum_{i=1}^K P_i^\Phi(x')}\right)\right] + \log K\right) && \text{(B.8)}\\
&\qquad = \sum_{j=1}^K \mathbb{E}_{x'\sim P_j^\Phi}\left[\log\left(\frac{P_j^\Phi(x')}{\frac{1}{K}\sum_{i=1}^K P_i^\Phi(x')}\right)\right] && \text{(B.9)}\\
&\qquad = \sum_{j=1}^K KL\left(P_j^\Phi(x')\middle\|\frac{1}{K}\sum_{i=1}^K P_i^\Phi(x')\right) && \text{(B.10)}\\
&\qquad = K\cdot JSD(P_1^\Phi, ..., P_K^\Phi), && \text{(B.11)}
\end{aligned}$$

where $KL(\cdot||\cdot)$ is Kullback-Leibler divergence and $JSD(\cdot, ..., \cdot)$ is multi-distribution Jensen-Shannon Divergence [135]. Since $K \log K$ is a constant and the multi-distribution JSD is non-negative and 0 if and only if all distributions are equal, we have that $P_1^\Phi = ... = P_K^\Phi$. □

B.1.5 Training Procedure for CRN

Let $\mathcal{D} = \left\{\{\mathbf{x}_t^{(i)}, \mathbf{a}_t^{(i)}, \mathbf{y}_{t+1}^{(i)}\}_{t=1}^{T^{(i)}} \cup \{\mathbf{v}^{(i)}\}\right\}_{i=1}^{N}$ be an observational dataset consisting of information about N independent patients that we use to train CRN. The encoder and decoder networks part of CRN are trained into two separate steps.

To begin with, the encoder is trained to built treatment invariant representations of the patient history and to perform one-step ahead prediction. After the encoder is optimized, we use it to compute the balancing representation $\mathbf{br}_t^{(i)}$ for each timestep in the trajectory of patient (i). To train the decoder, we modify the training dataset as follows. For each patient (i), we split their trajectory into shorter sequences of the $\tau_{\max}$ timesteps of the form:

$$\left\{\mathbf{br}_l^{(i)} \cup \{\mathbf{y}_{l+t}^{(i)}, \mathbf{a}_{l+t}^{(i)}, \mathbf{y}_{l+t+1}^{(i)}\}_{t=1}^{\tau_{max}} \cup \mathbf{v}^{(i)}\right\}, \tag{B.12}$$

for $l = 1, \dots T^{(i)} - \tau_{\max}$. Thus, each patients contributes with $T^{(i)} - \tau_{\max}$ examples in the dataset for training the decoder. The different sequences obtained for all patents are randomly grouped into minibatches and used for training.

The pseudocode in Algorithm 4 shows the training procedure used for the encoder and decoder networks part of CRN. The model was implemented in TensorFlow and trained on an NVIDIA Tesla K80 GPU. The Adam optimizer [281] was used for training and both the encoder and the decoder are trained for 100 epochs.

Algorithm 4 Pseudo-code for training CRN

Input: Training data: $\mathcal{D} = \left\{ \{\mathbf{x}_t^{(i)}, \mathbf{a}_t^{(i)}, \mathbf{y}_{t+1}^{(i)}\}_{t=1}^{T^{(i)}} \cup \mathbf{v}^{(i)} \right\}_{i=1}^{N}$

(1) Encoder optimization: parameters $\theta_{E,r}, \theta_{E,a}, \theta_{E,y}$.
Learning rate: μ
for $p = 1, \ldots,$ max epochs **do**
$\lambda_p = \dfrac{2}{1 + \exp(-10 \cdot p)} - 1$
for Batch $\mathcal{B} = \left\{ \{\mathbf{x}_t^{(i)}, \mathbf{a}_t^{(i)}, \mathbf{y}_{t+1}^{(i)}\}_{t=0}^{T^{(i)}} \cup \mathbf{v}^{(i)} \right\}_{i=1}^{|\mathcal{B}|}$ in epoch **do**
Compute $\mathcal{L}_{E,a}^{\mathcal{B}}(\theta_{E,r}, \theta_{E,a}) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{t=1}^{T^{(i)}} \mathcal{L}_{t,a}^{(i)}(\theta_{E,r}, \theta_{E,a})$
Compute $\mathcal{L}_{E,y}^{\mathcal{B}}(\theta_{E,r}, \theta_{E,y}) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{t=1}^{T^{(i)}} \mathcal{L}_{t,y}^{(i)}(\theta_{E,r}, \theta_{E,y})$
$\theta_{E,r} \leftarrow \theta_{E,r} - \mu \left(\frac{\partial \mathcal{L}_{E,y}^{\mathcal{B}}(\theta_{E,r}, \theta_{E,y})}{\partial \theta_{E,r}} - \lambda_p \frac{\partial \mathcal{L}_{E,a}^{\mathcal{B}}(\theta_{E,r}, \theta_{E,a})}{\partial \theta_{E,r}} \right)$
$\theta_{E,y} \leftarrow \theta_{E,y} - \mu \frac{\partial \mathcal{L}_{E,y}^{\mathcal{B}}(\theta_{E,r}, \theta_{E,y})}{\partial \theta_{E,y}}$
$\theta_{E,a} \leftarrow \theta_{E,a} - \mu \frac{\partial \mathcal{L}_{E,a}^{\mathcal{B}}(\theta_{E,r}, \theta_{E,a})}{\partial \theta_{E,a}}$

(2) Compute the encoder balanced representation and use it to initialize the decoder hidden state.
for $i = 1, ..., N$ **do**
for $t = 1, \ldots, T^{(i)}$ **do**
$\mathbf{br}_t^{(i)} = \text{encoder}(\bar{\mathbf{x}}_t^{(i)}, \bar{\mathbf{a}}_{t-1}^{(i)}, \mathbf{v}^{(i)}; \theta_{E,r})$

(3) Split dataset in sequences of $\tau_{\max}$ timesteps:

$$\left\{ \left\{ \mathbf{br}_l^{(i)} \cup \{\mathbf{y}_{l+t}^{(i)}, \mathbf{a}_{l+t}^{(i)}, \mathbf{y}_{l+t+1}^{(i)}\}_{t=1}^{\tau_{max}} \cup \mathbf{v}^{(i)} \right\}_{l=1}^{T^{(i)} - \tau_{\max}} \right\}_{i=1}^{N}$$

(4) Optimize decoder: parameters $\theta_{D,r}, \theta_{D,a}, \theta_{D,y}$
Learning rate: μ
for p = 1, ..., max epochs **do**
$\lambda_p = \dfrac{2}{1 + \exp(-10 \cdot p)} - 1$
for Batch $\mathcal{B} = \left\{ \mathbf{br}_l^{(i)} \cup \{\mathbf{y}_{l+t}^{(i)}, \mathbf{a}_{l+t}^{(i)}, \mathbf{y}_{l+t+1}^{(i)}\}_{t=0}^{\tau_{max}} \cup \{\mathbf{v}^{(i)}\} \right\}_{i=1}^{|\mathcal{B}|}$ in epoch **do**
Compute $\mathcal{L}_{D,a}^{\mathcal{B}}(\theta_{D,r}, \theta_{D,a}) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{t=1}^{\tau_{max}} \mathcal{L}_{t,a}^{(i)}(\theta_{D,r}, \theta_{D,a})$
Compute $\mathcal{L}_{D,y}^{\mathcal{B}}(\theta_{D,r}, \theta_{D,y}) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{t=1}^{\tau_{max}} \mathcal{L}_{t,y}^{(i)}(\theta_{D,r}, \theta_{D,y})$
$\theta_{D,r} \leftarrow \theta_{D,r} - \mu \left(\frac{\partial \mathcal{L}_{D,y}^{\mathcal{B}}(\theta_{D,r}, \theta_{D,y})}{\partial \theta_{D,r}} - \lambda_p \frac{\partial \mathcal{L}_{D,a}^{\mathcal{B}}(\theta_{D,r}, \theta_{D,a})}{\partial \theta_{D,r}} \right)$
$\theta_{D,y} \leftarrow \theta_{D,y} - \mu \frac{\partial \mathcal{L}_{D,y}^{\mathcal{B}}(\theta_{D,r}, \theta_{D,y})}{\partial \theta_{D,y}}$
$\theta_{D,a} \leftarrow \theta_{D,a} - \mu \frac{\partial \mathcal{L}_{D,a}^{\mathcal{B}}(\theta_{D,r}, \theta_{D,a})}{\partial \theta_{D,a}}$

Output: Trained CRN encoder (parameters $\theta_{E,r}, \theta_{E,a}, \theta_{E,y}$) and trained CRN decoder (parameters $\theta_{D,r}, \theta_{D,a}, \theta_{D,y}$.)

B.1.6 Pharmacokinetic-Pharmacodynamic Model of Tumour Growth

To evaluate CRN on counterfactual estimation, we need access to the data generation mechanism to build a test set that consists of patient outcomes under all possible treatment options. For this purpose, we use the state-of-the-art pharmacokinetic-pharmacodynamic (PK-PD) model of tumour growth proposed by [118] and also used by [117] for evaluating RMSMs. The PK-PD model characterizes patients suffering from non-small cell lung cancer and models the evolution of their tumour under the combined effects of chemotherapy and radiotherapy. In addition, the model includes different distributions of tumour sizes based on the cancer stage at diagnosis.

The volume of tumour t days after diagnosis is modelled as follows:

$$V(t+1) = \Big(1 + \underbrace{\rho \log(\frac{K}{V(t)})}_{\text{Tumor growth}} - \underbrace{\beta_c C(t)}_{\text{Chemotherapy}} - \underbrace{\Big(\alpha_r d(t) + \beta_r d(t)^2\Big)}_{\text{Radiotherapy}} + \underbrace{e_t}_{\text{Noise}}\Big) V(t) \tag{B.13}$$

where the parameters $K, \rho, \beta_c, \alpha_r, \beta_r$ are sampled from the distributions described in [118] and $e_t \sim \mathcal{N}(0, 0.01^2)$ is a noise term that accounts for randomness. To incorporate heterogeneity among patient responses, due to, for instance, gender or genetic factors [282], the prior means for β_c and α_r are adjusted to create three patient subgroups $S^{(i)} \in \{1, 2, 3\}$ as described in [117]. This way, we incorporate in the model of tumour growth specific characteristics that affect the patient's individualized response to treatments. Thus, the prior mean μ_{β_c} of β_c and the prior mean μ_{α_r} of α_r are augmented as follows.

$$\mu'_{\beta_c}(i) = \begin{cases} 1.1\mu_{\beta_c}, \text{if} S^{(i)} = 3 \\ \mu_{\beta_c}, \text{otherwise} \end{cases} \qquad \mu'_{\alpha_r}(i) = \begin{cases} 1.1\mu_{\alpha_r}, \text{if} S^{(i)} = 1 \\ \mu_{\alpha_r}, \text{otherwise} \end{cases} \tag{B.14}$$

where μ_{β_c} and μ_{α_r} are from [118] and $\mu'_{\beta_c}(i)$ and $\mu'_{\alpha_r}(i)$ are the parameters used in the data simulation. The patient subgroup $S^{(i)} \in \{1, 2, 3\}$ is used as baseline features.

Chemotherapy drug concentration follows an exponential decay with half life of 1 day $C(t) = \tilde{C}(t) + C(t-1)/2$, where $\tilde{C}(t) = 5.0mg/m^3$ of Vinblastine if chemotherapy is given at time t. $d(t) = 2.0Gy$ fractions of radiotherapy if the radiotherapy treatment is applied at timestep t.

Time-varying confounding is introduced by modelling chemotherapy and radiotherapy assignment as Bernoulli random variables, with probabilities p_c and p_r depending on

the tumour diameter:

$$p_c(t) = \sigma\left(\frac{\gamma_c}{D_{\max}}(\bar{D}(t) - \delta_c)\right) \qquad p_r(t) = \sigma\left(\frac{\gamma_r}{D_{\max}}(\bar{D}(t) - \delta_r)\right), \tag{B.15}$$

where $\bar{D}(t)$ is the average tumour diameter over the last 15 days, $D_{\max} = 13$cm is the maximum tumour diameter and $\sigma(\cdot)$ is the sigmoid activation function. The parameters δ_c and δ_r are set to $\delta_c = \delta_r = D_{\max}/2$ such that there is 0.5 probability of receiving treatment when tumour is half of its maximum size. γ_c, γ_r control the amount of time-dependent confounding; the higher $\gamma_\star$ is, the more important the history of tumour diameter is in assigning treatments. Thus, at each timestep, there are four treatment options options: no treatment (A_1), chemotherapy (A_2), radiotherapy (A_3), combined chemotherapy and radiotherapy (A_4).

Since the work most relevant to ours is the one of [117] we used the same data simulation and same settings for $\gamma = \gamma_c = \gamma_r$ as in their case. When $\gamma = 0$, there is no time-dependent confounding and the treatments are randomly assigned. By increasing γ we increase the influence of the volume size history (encoded in $\bar{D}(t)$) on the treatment probability. For example, assume $\bar{D}(t) = \frac{3D_{max}}{4}$. From equation (7), the probability of chemotherapy in this case is $p_c(t) = \sigma(\frac{\gamma_c}{D_{max}}(\bar{D}(t) - \frac{D_{max}}{2})) = \sigma(0.25\gamma_c)$, where $\sigma(\cdot)$ is the sigmoid function. When $\gamma = 1$, $p_c(t) = 0.56$, when $\gamma = 5$, $p_c(t) = 0.77$ and when $\gamma = 10$, $p_c(t) = 0.92$ in this example. γ can be increased further to increase the bias. However, the values used in the experiments evaluate the model on a wide range of settings for the time-dependent confounding bias.

B.1.7 Marginal Structural Models

Marginal Structural Models [14, 137] have been widely used in epidemiology and as part of follow up studies. In our case, we would like to estimate the effects of a sequence of treatments in the future given the current patient history:

$$\mathbb{E}(\mathbf{Y}_{t+\tau} \mid \bar{\mathbf{A}}(t, t+\tau-1) = \bar{\mathbf{a}}(t, t+\tau-1), \bar{\mathbf{H}}_t) = g(\tau, a(t, t+\tau-1), \bar{\mathbf{H}}_t), \tag{B.16}$$

where g is a generic function and $\bar{\mathbf{a}}(t, t+\tau-1) = [\mathbf{a}_t, \dots \mathbf{a}_{t+\tau-1}]$ represents a possible sequence of treatments from t just until before the potential outcome $\mathbf{Y}_{t+\tau}$ is observed.

To implement MSMs, we encode the treatments at timestep t in the model of tumour growth as $\mathbf{A}_t = [A_{t,c}, A_{t,d}]$ to indicate the binary application of chemotherapy and radiotherapy. To remove the time-dependent confounding bias and estimate future outcomes, we use the stabilized weights of MSMs to weight each patient in the dataset:

$$SW(t,\tau) = \prod_{n=t}^{t+\tau} \frac{f(\mathbf{A}_n \mid \bar{\mathbf{A}}_{n-1})}{f(\mathbf{A}_n \mid \bar{\mathbf{A}}_{n-1}, \bar{\mathbf{X}}_n, \mathbf{V})} = \prod_{n=t}^{t+\tau} \frac{\prod_{k\in\{c,d\}} f(A_{n,k} \mid \bar{\mathbf{A}}_{n-1})}{\prod_{k\in\{c,d\}} f(A_{n,k} \mid \bar{\mathbf{A}}_{n-1}, \bar{\mathbf{X}}_n, \mathbf{V})}, \quad \text{(B.17)}$$

where $f(\cdot)$ represents the conditional probability mass function for discrete treatments.

We adopt the implementation in [117, 137, 283] for MSMs and use logistic regression for estimating the propensity weights as follows:

$$f(A_{t,k} \mid \bar{\mathbf{A}}_{t-1}) = \sigma\Big(\sum_{j=1}^{k} \omega_k (\sum_{i=1}^{t-1} A_{t,j})\Big) \quad \text{(B.18)}$$

$$f(A_{t,k} \mid \bar{\mathbf{H}}_t) = \sigma\Big(\sum_{k\in\{c,d\}} \phi_k (\sum_{i=1}^{t-1} A_{t,k}) + \mathbf{w}_1 \mathbf{X}_t + \mathbf{w}_2 \mathbf{X}_{t-1} + \mathbf{w}_3 \mathbf{V}\Big) \quad \text{(B.19)}$$

where $\omega_\star, \phi_\star$ and $\mathbf{w}_\star$ are regression coefficients, $k \in \{c,d\}$ indicates chemotherapy or radiotherapy and $\sigma(\cdot)$ is the sigmoid function. For predicting the outcome, the following regression model is used, where each patient is weighted by its propensity score:

$$g(\tau, a(t, t+\tau-1), \bar{\mathbf{H}}_t) = \sum_{k\in\{c,d\}} \beta_k (\sum_{n=t}^{t+\tau-1} A_{n,k}) + \mathbf{l}_1 \mathbf{X}_t + \mathbf{l}_2 \mathbf{X}_{t-1} + \mathbf{l}_3 \mathbf{V} \quad \text{(B.20)}$$

where $\beta_\star$ and $\mathbf{l}_\star$ are regression coefficients. MSMs do not require hyperparameter tuning so we use the patients from both the train and validation sets for training.

B.1.8 Recurrent Marginal Structural Networks

MSMs are very sensitive to model mis-specification in computing the propensity weights and estimating the outcomes. Recurrent Marginal Structural Models (RMSNs) [117] overcome this problem by using recurrent neural networks to estimate the propensity scores and to build the outcome model. RNNs are more robust to changes in the treatment assignment policy. RMSNs were implemented as descried in [117][1].

For implementing RMSNs, we also encode the treatments at timestep t in the model of tumour growth as $\mathbf{A}_t = [A_{t,c}, A_{t,d}]$ to indicate the binary application of

[1] We used the implementation from https://github.com/sjblim/rmsn_nips_2018.

chemotherapy and radiotherapy. The propensity weights are estimated using recurrent neural networks as follows:

$$f(A_{t,k} \mid \bar{\mathbf{A}}_{t-1}) = \text{RNN}_{SW_n}(\bar{\mathbf{A}}_{t-1}) \quad f(A_{t,k} \mid \bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}) = \text{RNN}_{SW_d}(\bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t, \mathbf{V}) \tag{B.21}$$

For predicting one-step-ahed outcome, RMSNs use an encoder network:

$$g(1, a(t,t), \bar{\mathbf{H}}_t) = \text{RNN}_E(\mathbf{a}_t, \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t, \mathbf{V}), \tag{B.22}$$

where in the loss function, each patient is weighted by their stabilized IPTW.

For estimating the treatment responses for a sequence of treatments in the future, RMSNs use a decoder network:

$$g(\tau, a(t, t+\tau-1), \bar{\mathbf{H}}_t) = \text{RNN}_D(\mathbf{a}_t, \ldots, \mathbf{a}_{t+\tau-1}, \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t, \mathbf{V}). \tag{B.23}$$

See [117] for more details about the RMSN model architecture and training procedure of the propensity weights, encoder and decoder networks. Tables B.1 and B.2 show the hyperparameter search ranges used to optimize this model for evaluation in our paper. The hyperparameters were selected in the same way as proposed by [117], based on the error on the factual outcomes in the validation dataset. All of the models are trained using Adam optimizer for 100 epochs.

Hyperparameter	Search range
Iterations of Hyperparameter Search	50
Learning rate	0.01, 0.005, 0.001
Minibatch size	64, 128, 256
RNN state size	0.5C, 1C, 2C, 3C, 4C
Dropout rate	0.1, 0.2, 0.3, 0.4, 0.5
Max Gradient Norm	0.5, 1.0, 2.0

Table B.1: Hyperparameter search range for RMSNs propensity networks and encoder (same as in [117]). C is the size of the input.

B.1.9 Baseline RNN and Linear Model

For the baseline linear model, we fit the same regression model used for Marginal Structural Networks, but without using the IPTW. The baseline RNN uses an LSTM

Hyperparameter	Search range
Iterations of Hyperparameter Search	20
Learning rate	0.01, 0.001, 0.0001
Minibatch size	256, 512, 1024
RNN state size	1C, 2C, 4C, 8C, 16C
Dropout Rate	0.1, 0.2, 0.3, 0.4, 0.5
Max Gradient Norm	0.5, 1.0, 2.0, 4.0

Table B.2: Hyperparameter search range for RMSNs decoder [117]. C is the input size.

unit and, at each timestep, receives as input the current treatment, the patient covariates and the patient static features to perform one-step-ahead prediction. To have a model of similar capacity to the CRN (similar number of parameters), we add a fully connected layer on top of the output of the LSTM unit in order to obtain the outcomes. Table B.3 shows the hyperparameter search range used to optimize this model. The hyperparameters were selecting according to the error on the factual outcomes in the validation set. We train the baseline RNN using the Adam optimizer for 100 epochs.

Hyperparameter	Search range
Iterations of Hyperparameter Search	50
Learning rate	0.01, 0.001, 0.0001
Minibatch size	64, 128, 256
RNN hidden units	0.5C, 1C, 2C, 3C, 4C
FC hidden units	0.5C, 1C, 2C, 3C, 4C
RNN dropout probability	0.1, 0.2, 0.3, 0.4, 0.5

Table B.3: Hyperparameter search range for baseline RNN model. C is the size of the input.

B.1.10 Hyperparameter Optimization for CRN

As described in Appendix C, the dataset for training the decoder are used by splitting the sequences of the patients in the training set. This creates a larger dataset for training (where each patient (i) contributes $T^{(i)} - \tau_{\max}$ times to the dataset) which requires a different hyperparameter search range. Moreover, the balancing representations computed by the encoder are used to initialize the state of the RNN for the decoder. Thus, the decoder RNN size is equal to the size of the balancing representation size of the encoder. Table B.4 shows the hyperparameter search ranges for the encoder and decoder networks

in CRN. We selected hyperparameters based on the error of the model on the factual outcomes in the validation dataset. All models are trained for 100 epochs.

Hyperparameter	Search range encoder	Search range decoder
Iterations of Hyperparameter Search	50	30
Learning rate	0.01, 0.001, 0.0001	0.01, 0.001, 0.0001
Minibatch size	64, 128, 256	256, 512, 1024
RNN hidden units	0.5C, 1C, 2C, 3C, 4C	Balancing representation size of encoder
Balancing representation size	0.5C, 1C, 2C, 3C, 4C	0.5C, 1C, 2C, 3C, 4C
FC hidden units	0.5R, 1R, 2R, 3R, 4R	0.5R, 1R, 2R, 3R, 4R
RNN dropout probability	0.1, 0.2, 0.3, 0.4, 0.5	0.1, 0.2, 0.3, 0.4, 0.5

Table B.4: Hyperparameter search range for CRN encoder. C is the size of the input and R is the size of the balancing representation.

In addition, Tables B.5 and B.6 illustrate the optimal hyperparameters chosen.

	$\gamma_\star = 0$	$\gamma_\star = 1$	$\gamma_\star = 2$	$\gamma_\star = 3$	$\gamma_\star = 4$	$\gamma_\star = 5$
Learning rate	0.001	0.1	0.001	0.01	0.01	0.001
Minibatch size	64	64	64	128	64	128
RNN hidden units	12	18	24	18	24	24
Balancing representation size	18	18	12	18	6	12
FC hidden units	18	18	36	54	24	48
RNN dropout probability	0.1	0.1	0.1	0.2	0.2	0.1
	$\gamma_\star = 6$	$\gamma_\star = 7$	$\gamma_\star = 8$	$\gamma_\star = 9$	$\gamma_\star = 10$	
Learning rate	0.001	0.001	0.01	0.001	0.01	
Minibatch size	64	64	128	128	128	
RNN hidden units	24	18	12	24	24	
Balancing representation size	12	18	24	18	12	
FC hidden units	48	72	12	36	12	
RNN dropout probability	0.1	0.2	0.1	0.1	0.1	
	$\gamma_c = 0, \gamma_r = 5$		$\gamma_c = 5, \gamma_r = 0$			
Learning rate	0.01		0.001			
Minibatch size	128		64			
RNN hidden units	12		12			
Balancing representation size	18		24			
FC hidden units	36		96			
RNN dropout probability	0.1		0.1			

Table B.5: Optimal hyperparameters for the CRN encoder when different degrees of time-dependent confounding are applied in the model of tumour growth. The parameters γ_c and γ_r measure the degree of time-dependent confounding applied. When γ_c and γ_r are set to the same value, we denote this with $\gamma_\star$.

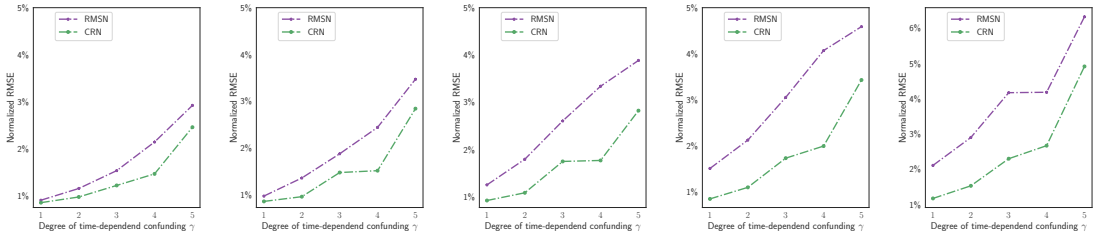
	$\gamma_\star = 1$	$\gamma_\star = 2$	$\gamma_\star = 3$	$\gamma_\star = 4$	$\gamma_\star = 5$
Learning rate	0.001	0.001	0.001	0.001	0.001
Minibatch size	1024	1024	512	1024	1024
RNN hidden units	18	12	18	6	12
Balancing representation size	18	18	6	18	3
FC hidden units	18	36	18	72	6
RNN dropout probability	0.1	0.2	0.3	0.1	0.1
	$\gamma_c = 0, \gamma_r = 5$	$\gamma_c = 5, \gamma_r = 0$			
Learning rate	0.01	0.001			
Minibatch size	512	1024			
RNN hidden units	18	24			
Balancing representation size	18	12			
FC hidden units	36	24			
RNN dropout probability	0.1	0.03			

Table B.6: Optimal hyperparameters for the CRN decoder when different degrees of time-dependent confounding are applied in the model of tumour growth. The parameters γ_c and γ_r measure the degree of time-dependent confounding applied. When γ_c and γ_r are set to the same value, we denote this with $\gamma_\star$

B.1.11 Full Results for Counterfactual Prediction

Multi-step Ahead Prediction of Counterfactuals

Figure B.2 shows the normalized RMSE for multiple step-ahead prediction of counterfactuals. The RMSE is normalized by the maximum tumour volume: $V_{max} = 1150\text{cm}^3$. The counterfactuals in this case are generated as described in Section 6.3 and Appendix I. We notice that performance gains of CRN compared to RMSN increase with the number of future timesteps for which the counterfactuals are estimated.



(a) Two-step ahead **(b) Three**-step ahead **(c) Four**-step ahead **(d) Five**-step ahead **(e) Six**-step ahead

Figure B.2: Results for prediction of patient counterfactuals for multiple steps ahead.

Detailed Results for the Counterfactual Predictions

Tables B.7 and B.8 show detailed results for the counterfactual predictions.

	$\gamma = 0$	$\gamma = 1$	$\gamma = 2$	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$
Linear (no IPTW)	0.99%	1.08%	1.36%	1.68%	2.11%	2.77%
MSM	0.99%	1.08%	1.34%	1.63%	2.02%	2.61%
RNN	0.70%	0.70%	0.84%	1.05%	1.24%	1.69%
CRN (λ = 0)	0.66%	0.77%	0.92%	0.95%	1.24%	1.54%
RMSN	0.60%	0.61%	0.72%	0.81%	0.94%	1.23%
CRN	**0.56%**	**0.57%**	**0.62%**	**0.67%**	**0.87%**	**1.20%**

	$\gamma = 6$	$\gamma = 7$	$\gamma = 8$	$\gamma = 9$	$\gamma = 10$
Linear (no IPTW)	3.55%	4.15%	4.80%	5.09%	5.22%
MSM	3.30%	3.79%	4.30%	4.47%	4.47%
RNN	2.03%	2.52%	2.88%	3.79%	4.01%
CRN (λ = 0)	1.98%	2.42%	2.73%	3.17%	3.57%
RMSN	1.70%	2.18%	2.37%	2.77%	2.83%
CRN	**1.48%**	**1.56%**	**2.05%**	**2.36%**	**2.41%**

Table B.7: Normalized RMSE for one-step-ahead prediction of counterfactuals. The parameter γ measures the degree of time-dependent confounding applied.

		$\gamma = 1$	$\gamma = 2$	$\gamma = 3$	$\gamma = 4$	$\gamma = 5$
$\tau = 2$	RMSN	0.90%	1.15%	1.53%	2.14%	2.91%
	CRN	**0.84%**	**0.96%**	**1.21%**	**1.46%**	**2.45%**
$\tau = 3$	RMSN	0.97%	1.36%	1.87%	2.44%	3.47%
	CRN	**0.86%**	**0.96%**	**1.47%**	**1.51%**	**2.84%**
$\tau = 4$	RMSN	1.24%	1.79%	2.60%	3.33%	3.88%
	CRN	**0.91%**	**1.08%**	**1.74%**	**1.76%**	**2.82%**
$\tau = 5$	RMSN	1.51%	2.13%	3.06%	4.07%	4.58%
	CRN	**0.85%**	**1.10%**	**1.73%**	**2.00%**	**3.43%**
$\tau = 6$	RMSN	2.10%	2.89%	3.06%	4.16%	6.32%
	CRN	**1.16%**	**1.52%**	**2.29%**	**2.66%**	**4.91%**

Table B.8: Normalized RMSE for τ-step-ahead prediction of counterfactuals. The parameter γ measures the degree of time-dependent confounding applied.

B.1.12 Test Set Generation for Evaluating Timing of Treatment

In order to evaluate how well the models select the correct treatment and timing of treatment we simulate counterfactual outcomes as follows. We generate 1000 test samples using the model of tumour growth described in Section 6. Let $\bar{\mathbf{H}}_t$ be the current history of the patient and let τ be a future time horizon. For each timestep in the future, we have 4 treatment options at: no treatment (A_0), chemotherapy (A_1), radiotherapy (A_2), chemotherapy and radiotherapy. (A_3). Using the model of tumour growth where the

outcome $\mathbf{Y}_{t+\tau}$ is the volume of the tumour, we generate the following 2τ counterfactuals:

Chemotherapy application

$$\mathbf{Y}_{t+\tau} \mid \mathbf{a}_t = A_1, \mathbf{a}_{t+1} = A_0, \dots \mathbf{a}_{t+\tau-1} = A_0, \bar{\mathbf{H}}_t \quad \text{(B.24)}$$

$$\mathbf{Y}_{t+\tau} \mid \mathbf{a}_t = A_0, \mathbf{a}_{t+1} = A_1, \dots \mathbf{a}_{t+\tau-1} = A_0, \bar{\mathbf{H}}_t \quad \text{(B.25)}$$

$$\dots$$

$$\mathbf{Y}_{t+\tau} \mid \mathbf{a}_t = A_0, \mathbf{a}_{t+1} = A_0, \dots \mathbf{a}_{t+\tau-1} = A_1, \bar{\mathbf{H}}_t \quad \text{(B.26)}$$

Radiotherapy application

$$\mathbf{Y}_{t+\tau} \mid \mathbf{a}_t = A_2, \mathbf{a}_{t+1} = A_0, \dots \mathbf{a}_{t+\tau-1} = A_0, \bar{\mathbf{H}}_t \quad \text{(B.27)}$$

$$\mathbf{Y}_{t+\tau} \mid \mathbf{a}_t = A_0, \mathbf{a}_{t+1} = A_2, \dots \mathbf{a}_{t+\tau-1} = A_0, \bar{\mathbf{H}}_t \quad \text{(B.28)}$$

$$\dots$$

$$\mathbf{Y}_{t+\tau} \mid \mathbf{a}_t = A_0, \mathbf{a}_{t+1} = A_0, \dots \mathbf{a}_{t+\tau-1} = A_2, \bar{\mathbf{H}}_t \quad \text{(B.29)}$$

We perform this for each patient in the test set and at each time t in the history. For instance, for a patient with 50 timesteps in the model of tumour growth and for time horizon $\tau = 3$, we generate $2 \cdot 3 \cdot 50 = 300$ counterfactuals.

Using the true generated couterfactual data, we select the treatment that has the lowest $\mathbf{Y}_{t+\tau}$ among the τ options generated for each treatment. Then, we select the time of applying treatment (among $t, t+1, \dots t+\tau-1$) that resulted in the lowest $\mathbf{Y}_{t+\tau}$. For each model, we generate the counterfactuals under the same treatment plans and patient histories. Then, we perform the selection of treatment and timing of treatment in the same way and we compare these with the true ones. Note that in order to account for numerical instability (two outcomes $Y_{t+\tau}$ having very similar values), we consider two outcomes the same if they are within $\epsilon = 0.001$ of each other.

B.1.13 Results on Factual Prediction on MIMIC III

Using the Medical Information Mart for Intensive Care (MIMIC III) [148] database consisting of electronic health records from patients in the ICU, we also show how the CRN can be used on a real medical dataset. From MIMIC III we extracted the patients on antibiotics, with trajectories up to 30 timesteps, thus obtaining a dataset with 3487

patients. For each patient, we extracted 25 patient covariates including lab tests and vital signs measured over time, as well as static patient features such as age and gender.

We used a binary treatment at each timestep indicating whether the patient was administered antibiotics or not. Note that for the longitudinal covariates we used aggregate value for each day since the ICU admission. The reason for this is because antibiotic treatment is decided daily for the patient. We split the dataset into $2826/313/348$ patients for training, validation and testing respectively. We performed hyperparameter optimization on the validation patient set, using the search ranges in Table B.4 and we again selected hyperparameters based on the error on the factual outcomes.

We estimate the individualized effect of antibiotics assigned over time on the patient's white blood cell count. A high white blood cell count is associated with severe illness and poor outcome for ICU patients [284]. Antibiotic administration in the ICU aims to reduce the white blood cell count. However, the effectiveness of the antibiotics treatment in reducing the white blood cell count is highly dependent on the time they are administered with respect to the history of the patient covariates. In this context we again have time-dependent confounders: the patient features change over time and are affected by the previous administration of antibiotics. Moreover, the history of the patient features also determines antibiotics administration and affects future patient outcomes [285, 286].

In Table B.9 we report the root mean squared error for factual prediction of the patients' white blood cell count for multiple prediction horizons τ. Note that for this dataset we do not have access to counterfactual data, which is why we report error on factual predictions.

	$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$
RMSN	2.84	3.87	4.46	4.79
CRN	2.68	3.54	4.07	4.67

Table B.9: RMSE for τ-step-ahead prediction of factual outcomes on MIMIC III.

We notice that CRN also achieves better performance than RMSN in estimating factual outcomes in a real-world dataset containing electronic health records. In this context, where couterfactual data is not available, domain expert knowledge is required to validate the model's counterfactual predictions under other antibiotic treatment alternatives. This further medical validation is outside the scope of this chapter.

B.2 Time Series Deconfounder: Estimating Treatment Effects over Time in the Presence of Hidden Confounders

B.2.1 Proof for Theorem 4.2.1

Before proving Theorem 4.2.1, we introduce several definitions and lemmas that will aid with the proof. Note that the these are extended from the static setting in [99]. Remember that at each timestep t, the random variable $\mathbf{Z}_t \in \mathcal{Z}_t$ is constructed as a function of the history until timestep t: $\mathbf{Z}_t = g(\bar{\mathbf{H}}_{t-1})$, where $\bar{\mathbf{H}}_{t-1} = (\bar{\mathbf{Z}}_{t-1}, \bar{\mathbf{X}}_{t-1}, \bar{\mathbf{A}}_{t-1})$ takes values in $\bar{\mathcal{H}}_{t-1} = \bar{\mathcal{Z}}_{t-1} \times \bar{\mathcal{X}}_{t-1} \times \bar{\mathcal{A}}_{t-1}$ and $g : \bar{\mathcal{H}}_{t-1} \to \mathcal{Z}$. In order to obtain **sequential ignorable treatment assignment** using the substitutes for the hidden confounders $\mathbf{Z}_t$, the following property needs to hold:

$$\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \perp\!\!\!\perp (A_{t1}, \ldots, A_{tk}) \mid \bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{Z}}_t, \tag{B.30}$$

$\forall \bar{\mathbf{a}}_{\geq t}$ and $\forall t \in \{0, \ldots, T\}$.

Definition B.2.1. ***Sequential Kallenberg construction***

At timestep t, we say that the distribution of assigned causes $(A_{t1}, \ldots A_{tk})$ admits a sequential Kallenberg construction from the random variables $\mathbf{Z}_t = g(\bar{\mathbf{H}}_{t-1})$ and $\mathbf{X}_t$ if there exist measurable functions $f_{tj} : \mathcal{Z}_t \times \mathcal{X}_t \times [0,1] \to \mathcal{A}_j$ and random variables $U_{jt} \in [0,1]$, with $j = 1, \ldots, k$ such that:

$$A_{tj} = f_{tj}(\mathbf{Z}_t, \mathbf{X}_t, U_{tj}), \tag{B.31}$$

where U_{tj} marginally follow Uniform$[0,1]$ and jointly satisfy:

$$(U_{t1}, \ldots U_{tk}) \perp\!\!\!\perp \mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \mathbf{Z}_t, \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}, \tag{B.32}$$

for all $\bar{\mathbf{a}}_{\geq t}$.

Lemma B.2.1. ***Sequential Kallenberg construction at each timestep $t \Rightarrow$ Sequential strong ignorability****. If at every timestep t, the distribution of assigned causes $(A_{t1}, \ldots A_{tk})$ admits a Kallenberg construction from $\mathbf{Z}_t = g(\bar{\mathbf{H}}_{t-1})$ and $\mathbf{X}_t$ then we obtain sequential strong ignorability.*

Proof. Assume that $\mathcal{A}_j$ for $j = 1, \ldots, k$ are Borel spaces. For any $t \in \{1, \ldots, T\}$ assume $\mathcal{Z}_t$ and $\mathcal{X}_t$ are measurable spaces and that $A_{tj} = f_{tj}(\mathbf{Z}_t, \mathbf{X}_t, U_{tj})$, where f_{tj} are measurable and

$$(U_{t1}, \ldots U_{tk}) \perp\!\!\!\perp \mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \mathbf{Z}_t, \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}, \tag{B.33}$$

for all $\bar{\mathbf{a}}_{\geq t}$. This implies that:

$$(\mathbf{Z}_t, \mathbf{X}_t, U_{t1}, \ldots U_{tk}) \perp\!\!\!\perp \mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \mathbf{Z}_t, \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}. \tag{B.34}$$

A_{tj}'s are measurable functions of $(\mathbf{Z}_t, \mathbf{X}_t, U_{t1}, \ldots U_{tk})$ and $\bar{\mathbf{H}}_{t-1} = (\bar{\mathbf{Z}}_{t-1}, \bar{\mathbf{X}}_{t-1}, \bar{\mathbf{A}}_{t-1})$, which means that that sequential strong ignorability holds:

$$(A_{t1}, \ldots A_{tk}) \perp\!\!\!\perp \mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \bar{\mathbf{X}}_t, \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{Z}}_t, \tag{B.35}$$

$\forall \bar{\mathbf{a}}_{\geq t}$ and $\forall t \in \{0, \ldots, T\}$. □

Lemma B.2.2. ***Factor models for the assigned causes*** $\Rightarrow$ ***Sequential Kallenberg construction at each timestep*** t. *Under weak regularity conditions, if the distribution of assigned causes* $p(\bar{\mathbf{a}}_T)$ *can be written as the factor model* $p(\theta_{1:k}, \bar{\mathbf{x}}_T, \bar{\mathbf{z}}_T, \bar{\mathbf{a}}_T)$ *then we obtain a sequential Kallenberg construction for each timestep.*

Regularity condition: The domains of the causes $\mathcal{A}_j$ *for* $j = 1, \ldots, k$ *are Borel subsets of compact intervals. Without loss of generality, assume* $\mathcal{A}_j = [0, 1]$ *for* $j = 1, \ldots, k$.

The proof for Lemma B.2.2 uses Lemma 2.22 in [287] (kernels and randomization): Let μ be a probability kernel from a measurable space S to a Borel space T. Then there exists some measurable function $f : S \times [0, 1] \to T$ such that if ϑ is $U(0, 1)$, then $f(s, \vartheta)$ has distribution $\mu(s, \dot{})$ for every $s \in S$.

Proof. For timestep t, consider the random variables $A_{t1} \in \mathcal{A}_1, \ldots A_{tk} \in \mathcal{A}_k, \mathbf{X}_t \in \mathcal{X}_t, \mathbf{Z}_t = g(\bar{\mathbf{H}}_{t-1}) \in \mathcal{Z}_t$ and $\theta_j \in \Theta$. Assume sequential single strong ignorability holds. Without loss of generality, assume $\mathcal{A}_j = [0, 1]$ for $j = 1, \ldots, k$.

From Lemma 2.22 in Kallenberg (1997), there exists some measurable function $f_{tj} : \mathcal{Z}_t \times \mathcal{X}_t \times [0, 1] \to [0, 1]$ such that $U_{tj} \sim$ Uniform$[0, 1]$ and:

$$A_{tj} = f_{tj}(\mathbf{Z}_t, \mathbf{X}_t, U_{tj}) \tag{B.36}$$

and there exists some measurable function $h_{tj} : \Theta \times [0,1] \to [0,1]$ such that:

$$U_{tj} = h_{tj}(\theta_j, \omega_{tj}), \tag{B.37}$$

where $\omega_{tj} \sim \text{Uniform}[0,1]$ and $j = 1, \ldots, k$.

From our definition of the factor model we have that ω_{tj} for $j = 1, \ldots, k$ are jointly independent. Otherwise, $A_{tj} = f_{tj}(\mathbf{Z}_t, \mathbf{X}_t, h_{tj}(\theta_j, \omega_{tj}))$ would not have been conditionally independent given $\mathbf{Z}_t, \mathbf{X}_t$.

As sequential single strong ignorability holds at each timestep t, we have that $A_{tj} \perp\!\!\!\perp \mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \mathbf{X}_t, \bar{\mathbf{H}}_{t-1} \;\; \forall \bar{\mathbf{a}} \in \bar{\mathcal{A}}, \forall t \in \{0, \ldots, T\}$ and $j = 1, \ldots, k$ which implies:

$$\omega_{tj} \perp\!\!\!\perp \mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}, \tag{B.38}$$

$\forall \bar{\mathbf{a}}_{\geq t}$ and $\forall j \in \{1, \ldots, k\}$. Using this, we can write:

$$\begin{aligned}
p(\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \;, \omega_{t1}, \ldots, \omega_{tk} \mid \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}) &= \\
&= p(\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}) \cdot p(\omega_{t1}, \ldots, \omega_{tk} \mid \mathbf{Y}(\bar{\mathbf{a}}_{\geq t}), \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}) \\
&= p(\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}) \cdot \prod_{j=1}^{k} p(\omega_{tj} \mid \omega_{t1}, \ldots, \omega_{t,j-1}, \mathbf{Y}(\bar{\mathbf{a}}_{\geq t}), \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}) \\
&= p(\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}) \cdot \prod_{j=1}^{k} p(\omega_{tj} \mid \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}) \\
&= p(\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}) \cdot p(\omega_{t1}, \ldots, \omega_{tk} \mid \mathbf{X}_t, \bar{\mathbf{H}}_{t-1})
\end{aligned}$$

where the second and third steps follow form equation (B.38) and the fact that $\omega_{t1}, \ldots, \omega_{tk}$ are jointly independent. This gives us:

$$(\omega_{t1}, \ldots, \omega_{tk}) \perp\!\!\!\perp \mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \mathbf{X}_t, \bar{\mathbf{H}}_{t-1} \tag{B.39}$$

Moreover, since the latent random variable $\mathbf{Z}_t$ is constructed without knowledge of $\mathbf{Y}(\bar{\mathbf{a}}_{\geq t})$, but rather as a function of the history $\bar{\mathbf{H}}_{t-1}$ we have:

$$(\omega_{t1}, \ldots, \omega_{tk}) \perp\!\!\!\perp \mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \mathbf{Z}_t, \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}. \tag{B.40}$$

$\theta_{1:k}$ are parameters in the factor model and can be considered point masses, so we also have that:

$$(\theta_1, \ldots, \theta_k) \perp\!\!\!\perp \mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \mid \mathbf{Z}_t, \mathbf{X}_t, \bar{\mathbf{H}}_{t-1}, \tag{B.41}$$

Since $U_{tj} = (h_{ij}(\theta_j, \omega_{tj}))$ are measurable functions of θ_j and ω_{tj} we have that:

$$(U_{t1}, \ldots, U_{tk}) \perp\!\!\!\perp \mathbf{Y}(\mathbf{a}_{\geq t}) \mid \mathbf{Z}_t, \mathbf{X}_t, \bar{\mathbf{H}}_{t-1} \tag{B.42}$$

We have thus obtained a sequential Kallenberg construction at timestep t. □

Theorem B.2.1. *If the distribution of the assigned causes $p(\bar{\mathbf{a}}_T)$ can be written as the factor model $p(\theta_{1:k}, \bar{\mathbf{x}}_T, \bar{\mathbf{z}}_T, \bar{\mathbf{a}}_T)$ then we obtain sequential ignorable treatment assignment:*

$$\mathbf{Y}(\bar{\mathbf{a}}_{\geq t}) \perp\!\!\!\perp (A_{t1}, \ldots, A_{tk}) \mid \bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t, \bar{\mathbf{A}}_{t-1}, \tag{B.43}$$

for all $\bar{\mathbf{a}}_{\geq t}$ and for all $t \in \{0, \ldots, T\}$.

Proof. Theorem 4.2.1 follows from Lemmas B.2.1 and B.2.2. In particular, using the proposed factor graph, we can obtain a sequential Kallenberg construction at each timestep and then obtain sequential strong ignorability. □

B.2.2 Implementation Details for the Factor Model

The factor model described in Section 4.2.5 was implemented in Tensorflow [288] and trained on an NVIDIA Tesla K80 GPU. For each synthetic dataset (simulated as described in Section 4.2.6), we obtained 5000 patients, out of which 4000 were used for training, 500 for validation, and 500 for testing. Using the validation set, we perform hyperparameter optimization using 30 iterations of random search to find the optimal values for the learning rate, minibatch size (M), RNN hidden units, multitask FC hidden units and RNN dropout probability. LSTM [143] units are used for the RNN implementation. The search range for each hyperparameter is described in Table B.10.

The trajectories for the patients do not necessarily have to be equal. However, to be able to train the factor model, we zero-padded them such that they all had the same length. The patient trajectories were then grouped into minibatches of size M and the factor model was trained using the Adam optimizer [281] for 100 epochs.

Table B.11 illustrates the optimal hyperparameters obtained for the factor model under the different amounts of hidden confounding applied (as described by the experiments in Section 4.2.6). Since the results for assessing the Time Series Deconfounder are averaged

Hyperparameter	Search range
Learning rate	0.01, 0.001, 0.0001
Minibatch size	64, 128, 256
RNN hidden units	32, 64, 128, 256
Multitask FC hidden units	32, 64, 128
RNN dropout probability	0.1, 0.2, 0.3, 0.4, 0.5

Table B.10: Hyperparameter search range for the proposed factor model implemented using a recurrent neural network with multitask output and variational dropout.

Hyperparameter	$\gamma = 0$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$
Learning rate	0.01	0.01	0.01	0.01	0.001
Minibatch size	64	64	64	64	128
RNN hidden units	32	64	64	128	128
Multitask FC hidden units	64	128	64	128	128
RNN dropout probability	0.2	0.2	0.1	0.3	0.3

Table B.11: Optimal hyperparameters for the factor model when different amounts of hidden confounding (measured by γ) are used in the synthetic dataset.

across 30 different simulated datasets, we report here the optimal hyperparameters identified through majority voting. We note that when the effect of the hidden confounders on the treatment assignments and the outcome is large, more capacity is needed in the factor model to be able to infer them.

B.2.3 Baselines for Evaluating Factor Model

Figure B.3 illustrates the architecture at each timestep for our proposed factor model and the baselines used for comparison. Figure B.3(a) represents our proposed architecture for the factor model consisting of a recurrent neural network with multitask output and variational dropout. We want to ensure that the multitask constraint does not cause a decrease in the capability of the network to capture the distribution of the assigned causes. To do so, we compare our proposed factor model with the network in Figure B.3(b) where we predict the k treatment assignments by passing $\mathbf{X}_t$ and $\mathbf{Z}_t$ through a hidden layer and having an output layer with k neurons. Moreover, to highlight the importance of learning time-dependencies to estimate the substitutes for the hidden confounders, we also use as a baseline the factor model in Figure B.3(c). In this case, a multilayer perceptron (MLP) is shared across the timesteps and it infers the latent

variable Z_t using only the previous covariates and treatments. Note that in this case there is no dependency on the entire history.

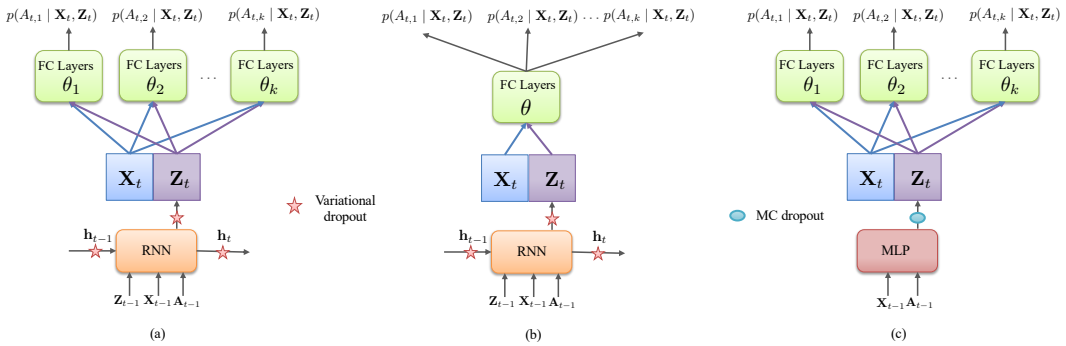


Figure B.3: (a) Proposed factor model using a recurrent neural network with multitask output and variational dropout. (b) Alternative design without multitask output. (c) Factor model using an MLP (shared across timestep) and multitask output. This baseline does not capture time-dependencies. MC dropout [162] is applied in the MLP to be able to sample from the substitutes for the hidden confounders.

The baselines were optimised under the same set-up described for our proposed factor model in Appendix B.2.2. Tables B.12 and B.13 describe the search ranges used for the hyperparameters in each of the baselines.

Hyperparameter	Search range
Learning rate	0.01, 0.001, 0.0001
Minibatch size	64, 128, 256
Max gradient norm	1.0, 2.0, 4.0
RNN hidden units	32, 64, 128, 256
Multitask FC hidden units	32, 64, 128
RNN dropout probability	0.1, 0.2, 0.3, 0.4, 0.5

Table B.12: Hyperparameter search range for factor model without multitask (Figure B.3(b)).

Hyperparameter	Search range
Learning rate	0.01, 0.001, 0.0001
Minibatch size	64, 128, 256
MLP hidden layer size	32, 64, 128, 256
Multitask FC hidden units	32, 64, 128
MLP dropout probability	0.1, 0.2, 0.3, 0.4, 0.5

Table B.13: Hyperparameter search range for MLP factor model. Figure B.3(c))

B.2.4 Outcome Models

After inferring the substitutes for the hidden confounders using the factor model, we implement outcome models to estimate the individualised treatment responses:

$$\mathbb{E}[\mathbf{Y}_{t+1}(\mathbf{a}_t) \mid \bar{\mathbf{A}}_{t-1}, \bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t] = h(\bar{\mathbf{A}}_t, \bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t) \tag{B.44}$$

We train the outcome models and evaluate them on predicting the treatment responses for each timestep, i.e. one-step-ahead predictions, for the patients in the test set. For training and tuning the outcome models, we use the same train/validation/test splits that we have used for the factor model. This means that the substitutes for the hidden confounders estimated using the fitted factor model on the test set are also used for testing purposes in the outcome models.

Marginal Structural Models

MSMs [14, 137] have been widely used in epidemiology to perform causal inference in longitudinal data. MSMs use inverse probability of treatment weighting during training to construct a pseudo-population from the observational data that resembles the one in a clinical trial and thus remove the bias introduced by time-dependent confounders [133]. The propensity scores for each timestep are computed as follows:

$$\text{SW}_t = \frac{f(\mathbf{A}_t \mid \bar{\mathbf{A}}_{t-1})}{f(\mathbf{A}_t \mid \bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t, \bar{\mathbf{A}}_{t-1})} = \frac{\prod_{j=1}^{k} f(A_{t,j} \mid \bar{\mathbf{A}}_{t-1})}{\prod_{j=1}^{k} f(A_{t,j} \mid \bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t, \bar{\mathbf{A}}_{t-1})} \tag{B.45}$$

where $f(\cdot)$ is the conditional probability mass function for discrete treatments and the conditional probability density function for continuous treatments. We adopt the implementation in [137, 283] for MSMs and estimate the propensity weights using logistic regression as follows:

$$f(A_{t,k} \mid \bar{\mathbf{A}}_{t-1}) = \sigma\Big(\sum_{j=1}^{k} \omega_k (\sum_{i=1}^{t-1} A_{t,j})\Big) \tag{B.46}$$

$$f(A_{t,k} \mid \bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t, \bar{\mathbf{A}}_{t-1}) = \sigma\Big(\sum_{j=1}^{k} \phi_k (\sum_{i=1}^{t-1} A_{t,j}) + \mathbf{w}_1 \mathbf{X}_t + \mathbf{w}_2 \mathbf{X}_{t-1} + \mathbf{w}_3 \mathbf{Z}_t + \mathbf{w}_4 \mathbf{Z}_{t-1}\Big) \tag{B.47}$$

where $\omega_\star, \phi_\star$ and $\mathbf{w}_\star$ are regression coefficients and $\sigma(\cdot)$ is the sigmoid function.

For predicting the outcome, the following regression model is used, where each individual patient is weighted by its propensity score:

$$h(\bar{\mathbf{A}}_t, \bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t) = \sum_{j=1}^{k} \beta_k (\sum_{i=1}^{t} A_{t,j}) + \mathbf{l}_1 \mathbf{X}_t + \mathbf{l}_2 \mathbf{X}_{t-1} + \mathbf{l}_3 \mathbf{Z}_t + \mathbf{l}_4 \mathbf{Z}_{t-1} \tag{B.48}$$

where $\beta_\star$ and $\mathbf{l}_\star$ are regression coefficients. Since MSMs do not require hyperparameter tuning, we train them on the patients from both the train and validation sets.

Recurrent Marginal Structural Networks

R-MSNs, implemented as descried in [117][2], use recurrent neural networks to estimate the propensity scores and to build the outcome model. The use of RNNs is more robust to changes in the treatment assignment policy. Moreover, R-MSNs represent the first application of deep learning in predicting time-dependent treatment effects. The propensity weights are estimated using recurrent neural networks as follows:

$$f(A_{t,k} \mid \bar{\mathbf{A}}_{t-1}) = \text{RNN}_1(\bar{\mathbf{A}}_{t-1}) \qquad f(A_{t,k} \mid \bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t, \bar{\mathbf{A}}_{t-1}) = \text{RNN}_2(\bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t, \bar{\mathbf{A}}_{t-1}) \tag{B.49}$$

For predicting the outcome, the following prediction network is used:

$$h(\bar{\mathbf{A}}_t, \bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t) = \text{RNN}_3(\bar{\mathbf{X}}_t, \bar{\mathbf{Z}}_t, \bar{\mathbf{A}}_t), \tag{B.50}$$

where in the loss function, each patient is weighted by its propensity score. Since the purpose of our method is not to improve predictions, but rather to assess how well the R-MSNs can be deconfounded using our method, we use the optimal hyperparameters for this model, as identified by [117]. R-MSNs are then trained on the combined set of patients from the training and validation sets.

R-MSNs [117], can also be used to forecast treatment responses for an arbitrary number of steps in the future. In our paper we focus on one-step ahead predictions of the treatment responses. However, the Time Series Deconfounder can also be applied to estimate the effects of a sequence of future treatments.

[2]We used the publicly available immlementation from https://github.com/sjblim/rmsn_nips_2018.

Hyperparameter	Propensity networks		Prediction
	$f(\mathbf{A}_t \mid \bar{\mathbf{A}}_{t-1})$	$f(\mathbf{A}_t \mid \bar{\mathbf{H}}_t)$	network
Dropout rate	0.1	0.1	0.1
State size	6	16	16
Minibatch size	128	64	128
Learning rate	0.01	0.01	0.01
Max norm	2.0	1.0	0.5

Table B.14: Hyperparameters used for R-MSN.

B.2.5 Additional Results

Experiments on Synthetic Data

We considered an additional experimental set-up where we have simulated hidden confounders of dimension $D_Z = 3$. In Figure B.4 we illustrate the root mean squared error (RMSE) for one-step-ahead estimation of treatment responses for patients in the test set without adjusting for the bias from the hidden confounders (Confounded), when using the simulated hidden confounders (Oracle) and after applying the Time Series Deconfounder with different model specifications (Deconfounded). We notice that the Time Series Deconfounder can still account for the bias from hidden confounders when the true size for the hidden confounders is underestimated in the factor model and set to ($D_Z = 1$). The performance is improved when setting D_Z to the true number of hidden confounders or when overestimating the number of hidden confounders.

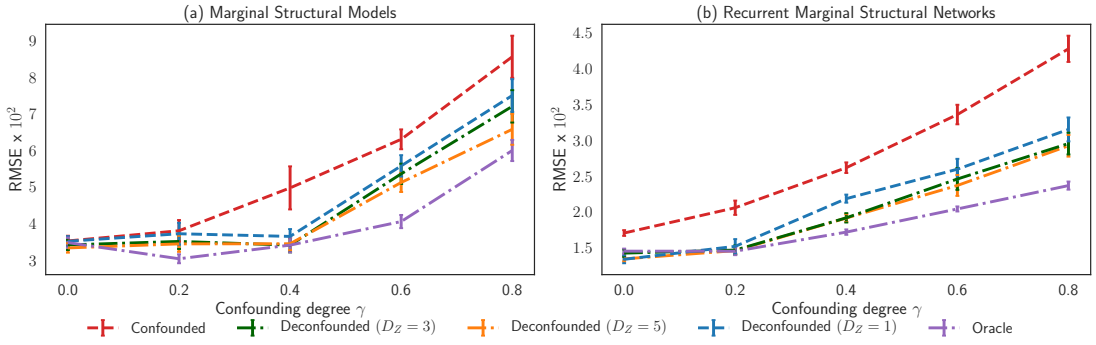


Figure B.4: Results for deconfounding one-step ahead estimation of treatment responses in two outcome models: (a) Marginal Structural Models (MSM) and (b) Recurrent Marginal Structural Networks (R-MSN). The simulated (true) size of the hidden confounders is $D_Z = 3$. The average RMSE and the standard error in the results are computed for 30 dataset simulations for each different degree of confounding, as measured by γ.

Model of Tumour Growth

To show the applicability of our method in a more realistic simulation, we use the pharmacokinetic-pharmacodynamic (PK-PD) model of tumor growth under the effects of chemotherapy and radiotherapy proposed by [118]. The tumor volume after t days since diagnosis is modeled as follows:

$$V(t) = \Big(1 + \underbrace{\rho \log(\frac{K}{V(t-1)})}_{\text{Tumor growth}} - \underbrace{\beta_c C(t)}_{\text{Chemotherapy}} - \underbrace{\big(\alpha_r d(t) + \beta_r d(t)^2\big)}_{\text{Radiotherapy}} + \underbrace{e_t}_{\text{Noise}}\Big) V(t-1) \tag{B.51}$$

where $K, \rho, \beta_c, \alpha_r, \beta_r, e_t$ are sampled as described in [118]. $C(t)$ is the chemotherapy drug concentration and $d(t)$ is the dose of radiation. Chemotherapy and radiotherapy prescriptions are modeled as Bernoulli random variables that depend on the tumor size. Full details about treatments are in [117].

To account for patient heterogeneity due to genetic features [282], the prior means for β_c and α_r are adjusted according to three patient subgroups as described in [117]. The patient subgroup $S^{(i)} \in \{1, 2, 3\}$ represents a confounder because it affects the tumor growth and subsequently the treatment assignments. We reproduced the experimental set-up in [117] and simulated datasets with 10000 patients for training, 1000 for validation, and 1000 for testing. We simulated 30 datasets and averaged the results for testing the MSM and R-MSN outcome models without the information about patient types (confounded), with the true simulated patient types, as well as after applying the Time Series Deconfounder with $D_Z \in \{1, 5, 10\}$.

Outcome model	MSM	R-MSN
Confounded	7.29 ± 0.14	5.31 ± 0.16
Deconfounded ($D_Z = 1$)	6.47 ± 0.16	4.76 ± 0.17
Deconfounded ($D_Z = 5$)	6.25 ± 0.14	4.79 ± 0.19
Deconfounded ($D_Z = 10$)	6.31 ± 0.11	4.54 ± 0.17
Oracle	6.92 ± 0.19	5.00 ± 0.15

Table B.15: Average RMSE $\times 10^2$ (normalised by max. tumour volume) and the standard error in the results for predicting the effect of chemotherapy and radiotherapy on the tumour volume.

The results in Table B.15 indicate that our method can infer substitutes for static hidden confounders such as patient subgroups which affect the treatment responses over

time. By construction, $\bar{\mathbf{Z}}_t$ also captures time dependencies which help with the prediction of outcomes. This is why the performance of the deconfounded models is slightly better than of the oracle model which uses static patient groups.

MIMIC III

We performed an additional experiment using the dataset extracted from the MIMIC III database where we have removed 3 patient covariates from the dataset (temperature, glucose, hemoglobin). In Table B.16 we report the results for estimating the effects of antibiotics, vasopressors, and mechanical ventilator on the patient's white blood cell count when including all variables, after removing these 3 patient covariates (which we notice that further confound the results) and after applying the Time Series Deconfounder with different settings for D_Z.

	White blood cell	
Outcome model	MSM	R-MSN
All patient covariates	3.90 ± 0.00	2.91 ± 0.05
Removed 3 covariates	4.12 ± 0.00	3.11 ± 0.03
Deconfounded ($D_Z = 1$)	3.98 ± 0.02	3.05 ± 0.05
Deconfounded ($D_Z = 3$)	3.91 ± 0.03	2.87 ± 0.08
Deconfounded ($D_Z = 5$)	3.85 ± 0.04	2.81 ± 0.03

Table B.16: Average RMSE $\times 10^2$ and the standard error in the results for predicting the effect of antibiotics, vasopressors, and mechanical ventilator on white blood cell count. The results are for 10 runs.

B.2.6 Discussion

The Time Series Deconfounder firstly builds a factor model to infer substitutes for the multi-cause hidden confounders. If Assumption 4.2.3 holds and the fitted factor model captures well the distribution of the assigned causes, which can be assessed through predictive checks, the substitutes for the hidden confounders help us obtain sequential strong ignorability (Theorem 4.2.1). Then, the Time Series Deconfounder uses the inferred substitutes for the hidden confounders in an outcome model that estimates individualized treatment responses. The experimental results show the applicability of the Time Series Deconfounder both in a controlled simulated setting and in a real dataset

consisting of electronic health records from patients in the ICU. In these settings, the Time Series Deconfounder was able to remove the bias from hidden confounders when estimating treatment responses conditional on patient history.

In the static causal inference setting, several methods have been proposed to extend the deconfounder algorithm in [99]. For instance, [289] augment the theory in the deconfounder algorithm in [99] by extending it to causal graphs and show that by using some of the causes as proxies of the shared confounder in the outcome model one can identify the effects of the other causes. [161] also suggests using proxy variables to obtain non-parametric identification of the mean potential outcomes [290]. Additionally, [291] proves that identification of causal effects is possible in the multi-cause setting when the treatments are normally distributed and the outcome is binary and follows a logistic structural equation model.

For the Time Series Deconfounder, similarly to [99], identifiability can be assessed by computing the uncertainty in the outcome model estimates, as described in Section 4.2.4. When the treatment effects are non-identifiable, the Time Series Deconfounder estimates will have high variance. Thus, future work could explore building upon the results in [289] and [161] and using proxy variables in the outcome model to prove identifiability of causal estimates in the multi-cause time-series setting.

C

Appendix for Chapter 5

Contents

C.1 Additional Related Works 193
C.2 Counterfactual Estimation and Time-dependent Confounding . . 193
C.3 Counterfactual μ-learning 195
C.4 Finding Optimal Policy for Given Reward Weights 196
C.5 Data Simulation 198
C.6 Implementation Details CIRL 200
C.7 Implementation Details Benchmarks 200

C.1 Additional Related Works

Alternative approaches for IRL require a known MDP or POMDP model. In this context, several Bayesian approaches to IRL have also been proposed which make use of the know transition dynamics provided by the model of the environment [292, 293]. We do not assume access to a MDP or POMDP model. [185] propose a method for estimating the unknown parameters of a POMDP model of the environment; however their method is only applicable to discrete observation spaces. We do not make any assumptions about the observation space. While several model-free IRL methods have also been developed, these are also not applicable in the batch setting where we only have access to a fixed set of expert trajectories. Relative Entropy IRL [294] requires non-expert trajectories following arbitrary policy; such experimentation to obtain non-expert trajectories is not possible in the healthcare setting. Distance Minimization IRL (DM-IRL) [295] learns reward functions from scores assigned by experts to sub-optimal demonstrations. Moreover, guided cost learning proposed by [296] which performs model-free maximum entropy optimization also requires the ability to sample trajectories from the environment. Similarly, adversarial IRL, proposed by [297] builds upon the maximum entropy framework in [292], but also requires the ability to execute trajectories from candidate policies.

C.2 Counterfactual Estimation and Time-dependent Confounding

Observational patient data (e.g. electronic health records) contain information about how actions, such as treatments are performed by doctors and how they affect the patient's trajectory. Such data can be used by causal inference methods to estimate counterfactual outcomes–what would happen to the patient if the expert takes a particular action given a history of observations? To identify the counterfactual outcomes from observational data we make the standard assumptions of consistency, positivity and no hidden confounders as described in Assumption C.2.1 [10, 14]. Under Assumption C.2.1,

$\mathbb{E}[Y_{t+1}[a_t] \mid h_t] = \mathbb{E}[X_{t+1} \mid a_t, h_t]$ and this can be estimated by training a regression model on the batch observational data.

Assumption C.2.1 (Consistency, Ignorability and Overlap)**.** *For any individual i, if action a_t is taken at time t, we observe $X_{t+1} = Y_{t+1}[a_t]$. Moreover, we have sequential strong ignorability $\{Y_{t+1}[a]\}_{a \in \mathcal{A}} \perp\!\!\!\perp a_t \mid h_t$, $\forall t$ and sequential overlap $\Pr(A_t = a \mid h_t) > 0$, $\forall t, \forall a \in \mathcal{A}$.*

The sequential strong ignorability assumption, also known as no hidden confounders, means that we observe all variables affecting the action assignment and potential outcomes. Sequential overlap means that at each timestep, every action has a non-zero probability, which can be satisfied by having a stochastic expert policy. These assumptions are standard across causal inference methods [14, 40, 55, 117]. We emphasize that these assumptions are needed to be able to reliably perform causal inference using observational data. Nevertheless, they do not constrain the batch IRL set-up. Using observational data to estimate the counterfactual outcomes poses additional challenges in this set-up that need to be considered. In particular, direct estimation of the treatment effects is biased by the presence of time-varying confounders [134], which are patient covariates that influence the assignment of treatments and which, in turn, are affected by past actions.

For instance, consider that the doctor's policy of assigning treatments takes into account whether the patients' covariates denoting disease progression, such as tumor volume, has been increasing above normal thresholds for several timesteps. If these patients are also more likely to have severe side effects, without adjusting for the bias introduced by time-varying confounders, we may incorrectly conclude that the treatment is harmful to the patients [133]. Using standard supervised learning methods to estimate the patients' response to treatments in this setting will be biased by the doctor's policy and will not be able to estimate the effect of treatments under different policies. This represents an issue for our batch IRL setting where we need to be able to evaluate candidate policies that are different from the expert policy. Thus, being able to correctly estimate counterfactual treatment outcomes is crucial for our method. Methods for adjusting for the confounding bias use either inverse probability of treatment weighting

(IPTW) or balancing representations. The first approach, IPTW, involves creating a pseudo-population where the probability of taking action a_t does not depend on $x_{0:t}$ [14, 117], while the second approach builds a balancing representation of the history that is invariant to the treatment [55]. Using either approach will result in an unbiased estimation of the counterfactual outcomes. See [14, 53, 117] for more details about time-dependent confounding as well as for a more in-depth review of alternative methods for removing the bias from time-dependent confounders and estimating counterfactual outcomes.

C.3 Counterfactual μ-learning

Our counterfactual μ-learning algorithm involves learning the μ-values for policy π iteratively by updating the current estimates of the μ-values with the feature map plus the μ-values obtained by following policy π in the new counterfactual history $h' = (h, a, \mathbb{E}[Y[a]|h])$:

$$\hat{\mu}^{\pi}(h,a) \leftarrow \hat{\mu}^{\pi}(h,a) + \alpha(\phi(h,a) + \gamma \mathbb{E}_{a' \sim \pi(\cdot|h')}[\hat{\mu}^{\pi}(h',a')] - \hat{\mu}^{\pi}(h,a)), \tag{C.1}$$

where α is the learning rate and γ is the discount factor. We approximate $\hat{\mu}^{\pi}(h, a \mid \theta_f)$ using an RNN with parameters θ_f. The RNN consists of an LSTM unit with linear output. To stabilize training we use a target network $\bar{\mu}^{\pi}$ with parameters θ_f^- that provide updates for the main network. Note that using a target network is standard in $Q-$learning [187, 298]. The target network $\bar{\mu}^{\pi}$ has the same architecture as the main network $\hat{\mu}^{\pi}$ and its parameters θ_f^- are updated to match θ_f every M^- iterations. The loss function for updating the μ network for estimating feature expectations at iteration i is:

$$\mathcal{L}_{f,i}(\theta_i) = \mathbb{E}_{h \sim \mathcal{D}}[||y_{f,i} - \hat{\mu}^{\pi}(h, a \mid \theta_{f,i})||_2] \tag{C.2}$$

$$\theta_{f,i+1} \leftarrow \theta_{f,i} + \alpha \nabla(\mathcal{L}_i(\theta_{f,i})) \tag{C.3}$$

where α is the learning rate, $\mathcal{D}$ is the batch observational dataset and

$$y_{f,i} = \begin{cases} \phi(h,a), & \text{if trajectory terminates at history } h \\ \phi(h,a) + \gamma \mathbb{E}_{a' \sim \pi(\cdot|h')}[\bar{\mu}^{\pi}(h', a' \mid \theta_f^-)], & \text{otherwise} \end{cases} \tag{C.4}$$

is the target for iteration i obtained using the target network $\bar{\mu}^\pi$. The action a is chosen by following an ε-greedy policy by selecting action $a \sim \pi(\cdot \mid h)$ with probability $1-\varepsilon$ and a random action with probability ϵ. We perform M training iterations. Algorithm 5 describes the full training procedure used.

Algorithm 5 Counterfactual $\mu-$learning

Input: Batch dataset $\mathcal{D} = \{(x_0^i, a_0^i, \dots x_{T^i-1}^i, a_{T^i-1}^i, x_{T^i}^i)\}_{i=1}^N$, policy to evaluate π, counterfactual model $\phi(h,a)$
Initialize feature expectations network $\hat{\mu}^\pi$ with random weights θ_f
Initialize target feature expectations network $\bar{\mu}^\pi$ with random weights $\theta_f^- = \theta_f$
for $i = 1$ to M **do**
    Sample random minibatch $\mathcal{B} = \{h^b\}_{b=1}^B$ of histories h from $\mathcal{D}$
    for $b = 1$ to B **do**
        With probability ϵ select random action a^b, otherwise select $a^b \sim \pi(\cdot \mid h^b)$
        Compute counterfactual histories $h^{b\prime} = (h^b, a^b, \mathbb{E}[Y[a^b]|h^b])$ using the counterfactual model: $\mathbb{E}[Y[a^b]|h^b] = \phi(h^b, a^b)$.
        Set targets:

$$y_{f,i}^b = \begin{cases} \phi(h^b, a^b), & \text{if trajectory terminates at } h^b \\ \phi(h^b, a^b) + \gamma \mathbb{E}_{a' \sim \pi(\cdot|h^{b\prime})}[\bar{\mu}^\pi(h^{b\prime}, a' \mid \theta_f^-)], & \text{otherwise} \end{cases} \tag{C.5}$$

    Perform gradient descent step on $\sum_{b=1}^B ||y_{f,i}^b - \hat{\mu}^\pi(h^b, a^b \mid \theta_{f,i})||_2$ with respect to parameters θ_f
    if i mod $M^- = 0$ **then** $\theta_f^- = \theta_{f,i}$
$\hat{\mu}^\pi = \frac{1}{N} \sum_{i=1}^N \sum_{a \in \mathcal{A}} \hat{\mu}^\pi(h_0^i, a) \pi(a \mid h_0^i)$

Output: $\hat{\mu}^\pi$

Table C.1 indicates the search ranges used for the various hyperparameters involved in the counterfactual μ-learning algorithm. The hyperparameters were selected to ensure stability of training and convergence of the algorithm.

C.4 Finding Optimal Policy for Given Reward Weights

We use deep recurrent Q-learning [187] to find the optimal policy for each setting of the reward weights $R(h,a) = w \cdot \phi(h,a)$.

For history h, let the next history for taking action a be $h' = (h, a, \mathbb{E}[Y(a) \mid h])$, where $\mathbb{E}[Y(a) \mid h]$ is estimated by the counterfactual model. Using the batch dataset $\mathcal{D}$, we

Hyperparameter	Hyperparameter search range	Hyperparameter value
LSTM size	64, 128, 256	128
Batch size	128, 256, 512	256
Learning rate	0.00001, 0.0001, 0.001	0.001
Target network update M^-	100, 200, 500	100
Min ε	-	0.0
Max ε	-	0.9
ε decay	-	0.00001
Number of training iterations M	-	20000
Optimizer	-	Adam

Table C.1: Hyperparameters for training μ-network for estimating feature expectations.

learn the $Q-$values iteratively by updating the current estimate of the $Q-$values towards the reward plus the maximum Q-value over all possible actions in the new history h'.

$$Q(h,a) \leftarrow Q(h,a) + \alpha(R(h,a) + \gamma \max_{a'} Q(h',a') - Q(h,a)) \tag{C.6}$$

where α is the learning rate and γ is the discount factor.

We approximate $Q(h,a|\theta_q)$ using an RNN parameterized by θ_q. The RNN consists of an LSTM unit with linear output. We use the standard practices for training $Q-$networks [187, 298] and we employ a target network $\bar{Q}$ with parameters θ_q^- to provide the updates for the main network and to stabilize training. The target network $\bar{Q}$ is the same architecture as the main network Q and its parameters θ_q^- are updated to match θ_q every M^- iterations. We use the following loss for the $Q-$learning update at iteration i:

$$\mathcal{L}_{q,i}(\theta_i) = \mathbb{E}_{h\sim\mathcal{D}}[(y_{q,i} - Q(h,a|\theta_{q,i}))^2] \tag{C.7}$$

$$\theta_{q,i+1} \leftarrow \theta_{q,i} + \alpha\nabla(\mathcal{L}_{q,i}(\theta_{q,i})) \tag{C.8}$$

where α is the learning rate and

$$y_{q,i} = \begin{cases} R(h,a), & \text{if patient trajectory terminates at history } h \\ R(h,a) + \gamma \max_{a'} \bar{Q}(h',a'|\theta_q^-), & \text{otherwise} \end{cases} \tag{C.9}$$

is the stale update target obtained from the target network $\bar{Q}$. The action a is chosen by following an ε-greedy policy by selecting action $a = \arg\max_{a'} Q(h,a'|\theta_q)$ with probability $1-\varepsilon$ and a random action with probability ϵ. We perform M training iterations. Table C.2 indicates the search ranges used for the various hyperparameters

involved in the deep recurrent $Q-$learnign algorithm. The hyperparameters were selected to ensure stability of training and convergence of the algorithm. We set min ε to 0.0, max ϵ to 0.9, ε decay to 0.00001. We train for 20000 iterations using the Adam optimizer.

Hyperparameter	Hyperparameter search range	Hyperparameter value
LSTM size	64, 128, 256	128
Batch size	128, 256, 512	256
Learning rate	0.00001, 0.0001, 0.001	0.001
Target network update M^-	100, 200, 500	100

Table C.2: Hyperparameters used for training Q-network to find optimal policy for a setting of the reward weights.

C.5 Data Simulation

Using the dynamics of the simulated environment described in Equation 5.11, we trained a deep recurrent Q-learning agent [187] to find a stochastic optimal policy that optimizes the reward function for different settings of the reward weights w_1 and w_2 in Equation 5.12.

Let $h_t = (x_{0:t}, v_{0:t}, a_{0:t-1})$ contain the history of the simulated patient covariates representing disease progression x and side effects v. Let $h_{t+1} = (x_{0:t+1}, v_{0:t+1}, a_{0:t})$ be the new history after taking action a_t given h_t and let $r_t = R(h_t, a_t)$ be the corresponding reward. We again learn the $Q-$values iteratively by updating the current estimate of the $Q-$values for history h_t towards the reward plus the maximum Q-value over all possible actions in the next history h_{t+1}.

$$Q(h_t, a_t) \leftarrow Q(h_t, a_t) + \alpha(R(h_t, a_t) + \gamma \max_{a'} Q(h_{t+1}, a') - Q(h_t, a_t)) \tag{C.10}$$

where α is the learning rate and γ is the given discount factor.

We approximate the action-value function $Q(h_t, a_t|\theta_e)$ using an RNN parameterized by θ_e. We use an LSTM [143] unit as part of the RNN with a liner output activation for estimating the Q-values. We use the standard practices for training $Q-$networks [187, 298].

Let $\mathcal{E} = \{(h_t, a_t, r_t, h_{t+1})\}_{t=0}^{E}$ be the experience replay memory of maximum capacity E obtained by simulating patient trajectories from the environment. During training, the $Q-$learning agent selects and executes actions in the environment using an $\varepsilon-$greedy policy that follows the greedy policy $a = \arg\max_{a'} Q(h, a'|\theta_e)$ with probability $1 - \varepsilon$

and selects random action with probability ϵ. The trajectories obtained from the agent's behaviour are added to the experience replay memory which is then used to train the $Q-$network. At each training iteration, we sample a batch B of examples from $\mathcal{E}$.

Moreover, we use a target network $\bar{Q}$ with parameters θ_e^- to provide the updates to the main network. The target network $\bar{Q}$ is the same architecture as the main network Q and its parameters θ_e^- are updated to match θ_e every M^- iterations. The purpose of the target network is to stabilize training.

We use the following loss function for the $Q-$learning update at iteration i:

$$\mathcal{L}_{e,i}(\theta_{e,i}) = \mathbb{E}_{(h_t,a_t,r_t,h_{t+1})\sim\mathcal{E}}[(y_{e,i} - Q(h_t, a_t|\theta_{e,i}))^2] \tag{C.11}$$

$$\theta_{e,i+1} \leftarrow \theta_{e,i} + \alpha\nabla(\mathcal{L}_{e,i}(\theta_{e,i})) \tag{C.12}$$

where α is the learning rate and

$$y_{e,i} = \begin{cases} r_t, & \text{if patient trajectory terminates at timestep } t \\ r_t + \gamma \max_{a'} \bar{Q}(h_{t+1}, a'|\theta_e^-), & \text{otherwise} \end{cases} \tag{C.13}$$

is the stale update target obtained from the target network $\bar{Q}$. We perform M training iterations. Table C.3 indicates the search ranges used for the various hyperparameters involved in the deep recurrent $Q-$learnign algorithm. We selected hyperparameters based on the cummulative reward obtained from the learnt optimal greedy policy in the simulated environment. We set min ε to 0.0, max ϵ to 0.9, ε decay to 0.00005. We train for 40000 iterations using the Adam optimizer.

Hyperparameter	Hyperparameter search range	Hyperparameter value
LSTM size	64, 128, 256	128
Experience replay memory capacity E	10000, 50000	10000
Batch size	128, 256, 512	256
Learning rate	0.00001, 0.0001, 0.001	0.001
Target network update M^-	100, 200, 500	200

Table C.3: Hyperparameters used for training Q-network to solve the simulated environment.

After training the $Q-$network we create a batch dataset $\mathcal{D}$ containing 10000 trajectories from the following stochastic expert policy:

$$\pi(a_t \mid h_t) = \text{Bernoulli}(\text{sigmoid}(\kappa Q(h_t, a_t)). \tag{C.14}$$

where κ introduces time-dependent confounding bias in the expert policy.

C.6 Implementation Details CIRL

The CIRL algorithm relies on using a counterfactual model to estimate the potential outcomes $\mathbb{E}[Y[a] \mid h]$ which are used to define the reward and to estimate the feature expectations. For this, we use the Counterfactual Recurrent Network (CRN) [55] which removes the bias from the time-dependend counfounders by building a balancing representation that is invariant to the treatment received by the patient at each timestep. Refer to [55] for details about the model architecture. Note that alternative counterfactual models can be used for this purpose [117].

For each simulated batch observational dataset, we use 9000 samples for training CRN and 1000 for validation (hyperparameter optimization). We perform hyperparameter optimization for the counterfactual model using the hyperparameter search ranges described in Table C.4. We select the model that has the lowest error in estimating the factual outcomes on the validation dataset. In the CIRL algorithm, we set the maximum number of iterations to 50 and the convergence threshold ϵ to 0.001. The experiments were run on a system with 2 NVIDIA K80 Tesla GPUs, 12CPUs, and 112GB of RAM.

Hyperparameter	Hyperparameter search range
Iterations of Hyperparameter Search	20
Learning rate	0.01, 0.001, 0.0001
Minibatch size	64, 128, 256
RNN hidden units	0.5C, 1C, 2C, 3C, 4C
Balancing representation size	0.5C, 1C, 2C, 3C, 4C
FC hidden units	0.5R, 1R, 2R, 3R, 4R
RNN dropout probability	0.1, 0.2, 0.3, 0.4, 0.5

Table C.4: Hyperparameter search range for CRN. C is the size of the input and R is the size of the balancing representation built by CRN.

C.7 Implementation Details Benchmarks

For benchmarks, we integrate as part of the proposed batch inverse reinforcement learning algorithm model-based policy evaluation reinforcement learning methods that use standard supervised learning approaches to estimate the next history [189]. These methods use standard supervised learning to estimate the next history h' needed for the counterfactual

μ-learning algorithm and for finding the optimal policy given a vector of reward weights. Our MB(h) benchmark uses a standard RNN to estimate the next history and define the policy, while MB(x) uses a multi-layer perceptron (MLP) that only considers the current observations for this purpose. The aim of these benchmarks is to highlight the importance of handling the the bias from the time-depdendent confounders and using a suitable counterfactual model, but also the importance of handling the patient history.

For each simulated batch observational datasets, we use 9000 of the samples for training and 1000 for validation. We chose hyperparameters according to the error of the models in estimating the next patient history on the validation set.

The MB(h) benchmark receives as input the patient history h_t and current action a_t and estimates the next observations x_{t+1} which are used to form the next history h_{t+1}. For this purpose, the MB(h) benchmark uses an LSTM unit, with a fully connected layer (FC) on top with ELU activation. The hyperparameter search ranges used for this model are described in Table C.5.

Hyperparameter	Hyperparameter search range
Iterations of Hyperparameter Search	20
Learning rate	0.01, 0.001, 0.0001
Minibatch size	64, 128, 256
RNN hidden units	1C, 2C, 3C, 4C, 5C
FC hidden units	1C, 2C, 3C, 4C, 5C

Table C.5: Hyperparameter search range for the RNN. C is the size of the input.

Conversely, the MB(x) benchmark receives as input the current patient observation x_t and current action a_t and is trained to estimate the next observation x_{t+1}. The MB(x) uses a multi-layer perceptron with two fully connected layers and ELU activation. The hyperparameter search ranges used for this model are described in Table C.6.

Hyperparameter	Hyperparameter search range
Iterations of Hyperparameter Search	20
Learning rate	0.01, 0.001, 0.0001
Minibatch size	64, 128, 256
FC hidden units	2C, 3C, 4C, 5C

Table C.6: Hyperparameter search range for MLP. C is the size of the input.

D

Appendix for Chapter 6

Contents

D.1 Structure of Environment and Interventions 203
D.2 Train Energy Based Model 204
D.3 Causal Features for Imitation using Invariant Risk Minimization . 205
D.4 Algorithm 206
D.5 Experimental Details 206
D.6 Additional Experiments 211

D.1 Structure of Environment and Interventions

Similarly to [48], we consider that all environments have the same underlying Structural Causal Model (SCM) and that the different environments correspond to different interventions on the SCM. We provide here the formal definition for SCMs and interventions.

Definition D.1.1. *(Structural Causal Model) [48]: A structural causal model (SCM)* $\mathcal{C} = (S, N)$ *governing the random vector* $X = (X_1, \dots X_m)$ *is a collection* S *of* m *assignments:*

$$S_j : X_j \leftarrow f_j(Pa(X_j), N_j), \text{ for } j = 1, \dots m \tag{D.1}$$

where $Pa(X_j) \subseteq \{X_1, X_2, \dots X_m\}/\{X_j\}$ *are the parents of* X_j *and the* N_j *are the independent noise variables. We say that* X_i *causes* X_j *if* $X_i \in Pa(X_j)$.

Definition D.1.2. *(Intervention) [48]: Consider a SCM* $\mathcal{C} = (S, N)$. *An intervention* e *on* $\mathcal{C}$ *consists of replacing one or several of its structural equations to obtain an intervened SCM* $\mathcal{C}^e = (S^e, N^e)$ *with structural equations:*

$$S_j^e : X_j^e \leftarrow f_j(Pa(X_j^e), N_j^e), \text{ for } j = 1, \dots m \tag{D.2}$$

The variable X^e *is intervened on if* $S_i \neq S_i^e$ *or* $N_i \neq N_i^e$.

In our setting, the variables forming the SCM are the different observations and actions at each timestep. Moreover, Assumption 6.3.3 that ensures the Markovianity of the temporal transitions restrict the relationships that can be present in the SCM. In addition, Assumption 6.3.2 requires that the interventions to not be on the causal parents of the action.

Soft interventions [233] do not remove any edges in the causal graph induced by the SCM, but instead modify the conditional probability distributions of the variables intervened on. On the other hand, hard interventions [156] on a variable remove all incoming edges from the parents of that variables.

D.2 Train Energy Based Model

We train the energy-based model for the expert demonstrations using Persistent Contrastive Divergence [237]. To sample from the energy-based model, we use Markov Chain Monte Carlo using Langevin Dynamics [299]. Algorithm 6 outlines the method used to learn the energy-based model $E_{\bar{\theta}}$ of the expert observations. See [237] for details on training EBMs in this manner.

Algorithm 6 Learning energy-based model $E_{\bar{\theta}}$ of expert demonstrations

Input: Dataset with expert demonstrations $\mathcal{D}$, number of steps K, step size α, noise variance σ, Mini-batch size N
Initialize: energy-based model parameters $\bar{\theta}$, buffer $\mathcal{B} \leftarrow \varnothing$
while not converged **do**
  Sample N positive samples from expert demonstrations $x_i^+ \sim \mathcal{D}$
  Sample initial negative samples: $x_i^0 \sim \mathcal{B}$ with 95% probability and $x_i^0 \sim \mathcal{U}(-1, 1)$ otherwise
  for sample step $k = 1$ to K **do** ▷ Generate sample via Langevin dynamics
    $\tilde{x}_i^k \leftarrow \tilde{x}_i^{k-1} - \alpha \cdot \nabla_x E_{\bar{\theta}}(\tilde{x}_i^{k-1}) + \omega$, where $\omega \sim \mathcal{N}(0, \sigma), \forall i \in \{1, \dots N\}$
  $x_i^- = \Omega(x_i^K), \forall i \in \{1, \dots N\}$ ▷ Ω: stop gradient operator
  Contrastive divergence loss $\mathcal{L}_{CD} = \frac{1}{N} \sum_i E_{\bar{\theta}}(x_i^+) - E_{\bar{\theta}}(x_i^-)$
  Regularization loss: $\mathcal{L}_{RG} = \frac{1}{N} \sum_i E_{\bar{\theta}}(x_i^+)^2 + E_{\bar{\theta}}(x_i^-)^2$
  Update parameters $\bar{\theta}$ by backpropagating $\nabla_{\bar{\theta}}(\mathcal{L}_{CD} + \mathcal{L}_{RG})$
  Add samples to buffer: $\mathcal{B} \leftarrow \mathcal{B} \cup \{x_i^-\}_{i=1}^N$

For the Acrobot [241], CartPole [242], LunarLander [240] control tasks and the experiments on the healthcare dataset extracted from MIMIC III [148], we use a neural network with 2 fully-connected hidden layers of size 64 and with ReLU activation to define $E_{\bar{\theta}}$. For Acrobot [241], CartPole [242], LunarLander [240], we set the hyperparameters to number of steps $K = 100$, step size $\alpha = 0.01$, noise variance $\sigma = 0.01$ and mini-batch size $N = 64$. We optimize $\bar{\theta}$ by using the Adam optimizer for 1000 training iterations with the learning rate set to 0.001. For the experiments on the healthcare dataset extracted from MIMIC III [148], we use the following hyperparameters for the number of steps $K = 50$, step size $\alpha = 0.01$, noise variance $\sigma = 0.01$ and mini-batch size $N = 128$. We optimize $\bar{\theta}$ by using the Adam optimizer for 1000 training iterations with the learning rate set to 0.0005.

To define $E_{\bar{\theta}}$ for the BeamRider Atari environment [243] we use a convolutional neural network with 3 convolutional layers with 32-64-64 filters, followed by a fully connected layer of size 64, with all layers followed by ReLU activations. The hyperparameters are set as follows: number of steps $K = 100$, step size $\alpha = 0.01$, noise variance $\sigma = 0.01$ and mini-batch size $N = 64$. $\bar{\theta}$ is optimized by using the Adam optimizer for 1000 training iterations with the learning rate set to 0.001.

D.3 Causal Features for Imitation using Invariant Risk Minimization

In the supervised learning setting, Invariant Risk Minimization (IRM) [48] leverage data from multiple domains to learn a data representation $\Phi : \mathcal{X} \to \mathcal{H}$ that elicits an invariant predictor $w : \mathcal{H} \to \mathcal{Y}$ across the different environments, where $\mathcal{X}$ and $\mathcal{Y}$ are the input and output spaces respectively. The training data from each environment $e \in \mathcal{E}$ corresponds to different interventions on the data generating process and R^e corresponds to the empirical risk of the classifier in each domain. We use the following formal definition from [48] to describe the characteristics we want from the learnt representation.

Definition D.3.1. *[48]: We say that a data representation $\Phi : \mathcal{X} \to \mathcal{H}$ elicits an invariant predictor across environments $\mathcal{E}$ if there is a classifier $w : \mathcal{H} \to \mathcal{A}$ simultaneously optimal for all environments, that is, $w = \arg\min_{\bar{w}:\mathcal{H}\to\mathcal{A}} R^e(\bar{w} \circ \Phi)$, for all $e \in \mathcal{E}$.*

Given data from training environments $\mathcal{E}_{train}$, the IRM objective aims to find a representation Φ such that there exists a classifier w that is optimal across all training domains:

$$\min_{\Phi:\mathcal{X}\to\mathcal{H}, w:\mathcal{H}\to\mathcal{Y}} \sum_{e\in\mathcal{E}_{train}} R^e(\bar{w} \circ \Phi) \text{ subject to } w \in \arg\min_{\bar{w}:\mathcal{H}\to\mathcal{A}} R^e(\bar{w} \circ \Phi), \forall e \in \mathcal{E}_{train} \tag{D.3}$$

This represents a challenging, bi-level optimization and [48] propose the IRM-v1 objective which is a practical version to optimize:

$$\min_{\phi:\mathcal{X}\to\mathcal{H}} \sum_{e\in\mathcal{E}_{train}} R^e(\Phi) + \lambda_{penalty} \cdot \|\nabla_{w|w=1.0} R^e(w \cdot \Phi)\| \tag{D.4}$$

Through this optimization, the IRM objective should learn a predictor that only uses the causal parents of the target variable and that is thus invariant across environments.

In the supervised setting considered by IRM [48], R^e is the risk of the classifier in environment e. For classification and regression problems, this can represent for instance the cross-entropy loss.

Imitation learning risk: We propose extending IRM to the imitation learning setting by using instead an imitation risk R^e as described in equation 6.2. Moreover, in this setting, our aim is to find an invariant policy π across the different environments (instead of the invariant classifier w). The risk R^e will therefore be specific to the imitation learning algorithm used. For instance, in behaviour cloning (BC) [205], for categorical actions $R^e_{\text{BC}} = \text{Cross entropy}(\pi(\cdot \mid x_t), a_t)$. Alternatively, ValueDice(VDICE) [199] minimizes the disparity between the occupancy measure of the expert policy vs the imitator policy $R^e_{\text{VDICE}} = D_{KL}(\rho^e_D || \rho_\pi)$. We use the IRMv1 objective in conjunction with the following imitation learning algorithms Behaviour Cloning (BC) [205], Reward-regularized Classification for Apprenticeship Learning (RCAL) [209], ValueDice(VDICE) [199] and Energy-based Distribution Matching (EDM) [66].

D.4 Algorithm

Algorithm 7 has the pseudo-code for training Invariant Causal Imitation Learning (ICIL).

D.5 Experimental Details

Environments Details

We use the following control tasks from OpenAI gym for experiments [240]: Acrobot [241], Cartpole [242], Lunar Lander [240] and BeamRider [243]. For each task, we use pre-trained RL agents from RL Baselines Zoo [244] and Stable OpenAI Baselines [245] to obtain expert policies. We provide in Table D.1 details about the different environments used including the size of the observation and action space, the agent used as demonstrator, the demonstrator's performance (average return) as well as the performance of an agent randomly selecting actions. The performance on the OpenAI gym tasks is measured in terms of average return of running the agent in the environment.

Algorithm 7 Invariant Causal Imitation Learning

Input: Dataset with expert demonstrations $\mathcal{D}$, learning rate λ, mini-batch size N
Initialize: $\theta_s, \theta_{g_s}, \{\theta^e_\eta, \theta^e_{g_\eta}\}_{e\in\mathcal{E}_{train}}, \theta_\psi, \theta_c, \theta_\pi, \theta_m$
while not converged **do**
    Sample mini-batch of N demonstrations $(x_i^{e_i}, a_i, x_{i+1}^{e_i}) \sim \mathcal{D}$ and
    Sample permutation of $[N] = \{1, \dots N\}$ from uniform distribution over the set of all permutations S_N: $\kappa \sim \mathcal{U}(S_N)$
    ICIL update:
    Invariance loss: $\mathcal{L}_{inv} = \frac{1}{N}\sum_{i=1}^{N} -H(c_s(\phi(x_i^{e_i})))$
    Transition dynamics loss:

$$\mathcal{L}_{dyn} = \frac{1}{N}\sum_{i=1}^{N} \|x_{i+1}^{e_i} - \psi(g_s(\phi(x_i^{e_i}), a_i), g_\eta^{e_i}(\mu^{e_i}(x_i^{e_i}), a_i))\|^2$$

    Mutual information (MI) loss:

$$\mathcal{L}_{mi} = \sum_{i=1}^{N} T_{\theta_m}(\phi(x_i^{e_i}), \mu(x_i^{e_i})) - \log(\sum_{i=1}^{N} \exp T_{\theta_m}(\phi(x_i^{e_i}), \mu(x_{\kappa(i)}^{e_{\kappa(i)}})))$$

    Policy loss: $\mathcal{L}_\pi = \frac{1}{N}\sum_{i=1}^{N}$ Cross entropy$(\pi(\cdot \mid \phi(x_i^{e_i})), a_i)$
    for $i = 1 \dots N$ **do**
        $\bar{a}_i \sim$ Gumbel Softmax$(\pi(\cdot \mid \phi(x_i^{e_i})))$
    Next state energy loss: $\mathcal{L}_{energy} = \frac{1}{N}\sum_{i=1}^{N} E_{\bar{\theta}}(\psi(g_s(\phi(x_i^{e_i}), \bar{a}_i), g_\eta^{e_i}(\mu^{e_i}(x_i^{e_i}), \bar{a}_i)))$

    Parameters update:
    $\theta_s \leftarrow \theta_s - \lambda\nabla_{\theta_s}(\mathcal{L}_{inv} + \mathcal{L}_{dyn} + \mathcal{L}_{mi} + \mathcal{L}_\pi)$
    $\theta_{g_s} \leftarrow \theta_{g_s} - \lambda\nabla_{\theta_s}\mathcal{L}_{dyn}$
    $\theta_\psi \leftarrow \theta_\psi - \lambda\nabla_{\theta_\psi}\mathcal{L}_{dyn}$
    for $e \in \mathcal{E}_{train}$ **do**
        $\theta^e_\eta \leftarrow \theta^e_\eta - \lambda\nabla_{\theta^e_\eta}(\mathcal{L}_{dyn} + \mathcal{L}_{mi})$
        $\theta^e_{g_\eta} \leftarrow \theta^e_{g_\eta} - \lambda\nabla_{\theta^e_{g_\eta}}\mathcal{L}_{dyn}$
    $\theta_\pi \leftarrow \theta_\pi - \lambda\nabla_{\theta_\pi}(\mathcal{L}_\pi + \mathcal{L}_{energy})$

    Environment classifier update: ▷ Used to define the invariance loss.
    $\mathcal{L}_c = \frac{1}{N}\sum_{i=1}^{N}$ Cross entropy$(c_s(\phi(x_i^{e_i}), e_i)$
    $\theta_c \leftarrow \theta_c - \lambda\nabla_{\theta_c}\mathcal{L}_c$

    Mutual information (MINE) update: ▷ Used to define the MI loss.
    Update T_{θ_m} by ascending the gradient of the MI loss: $\theta_m \leftarrow \theta_m + \nabla_{\theta_m}\mathcal{L}_{mi}$
Output: Learnt parameters $\theta_s, \theta_{g_s}, \{\theta^e_\eta, \theta^e_{g_\eta}\}_{e\in\mathcal{E}_{train}}, \theta_\psi, \theta_c, \theta_\pi, \theta_m$

Note that the expert uses the original observation space for each task. To create the different environments with spurious correlations used to train and test the imitation

learning benchmarks on Acrobot [241], CartPole [242] and Lunar Lander [240], we augment the observation space as follows. We add 3 noise variables that are different multiplicative factors of the last 3 variables in the original state space. For Acrobot, these are cosine of the second rotational angle and the two joint angular velocities, for CartPole these are Cart Velocity, Pole Angle, Pole Angular Velocity and for LunarLander these are Lander angular velocity, leg 1 ground contact and leg 2 ground contact. The invariant causal state is represented by the original variables in the state space of each control task. We train the benchmarks on demonstrations from two environments with $1\times$ and $2\times$ multiplicative factors for the spurious correlations and we test on an environments with multiplicative factors sampled from $\mathcal{U}(-1, 1)$.

Alternatively, for BeamRider [243], we create the different environments for training and testing by using different camera angles, similarly to [202]. More specifically, we use 2 training environments where the game frames are rotated by 10 degrees to the left and to the right respectively and a test environment that does not have any rotation.

For each OpenAI Gym task, and for each benchmark, we obtain datasets with $N_{traj} \in \{1, 5, 10, 15, 20\}$ demonstrated trajectories from each of the two training environments. We evaluate each benchmark in the test environment by computing the average return over 300 episodes roll-outs. We repeat each experiments 10 times, each time sampling different expert demonstrations for training.

In addition, we also use a dataset from the Medical Information Mart for Intensive Care (MIMIC-III) database [148]. For each patient, we extract 52 clinical covariates including vital signs (e.g. respiratory rate, heart rate, temperature, O2 saturation) and lab test (e.g. glucose, hemoglobin, magnesium, potassium, platelet count, white blood cell count) that are aggregated every hour during their ICU stay. We consider patient trajectories that are up to 24 hours. Moreover, we concatenate the last 4 hours to build the observations received by each imitation learning algorithm. The expert in this case is the doctor and we consider as action the ventilator support. We consider three environment each with 2000 independent patient trajectories from MIMIC III. Two of the environments are used for training and one for testing. We augment the original feature space by adding 20 spurious correlations (noise variables) that are the same as the expert actions with

Environments	Original Obs. Space	Action Space	Demonstrator	Random Perf.	Demonstrator Perf.
Acrobot-v1	Continuous (6)	Discrete (3)	PPO2 Agent	-439.92 ± 13.14	-87.32 ± 12.02
CartPole-v1	Continuous (4)	Discrete (2)	DQN Agent	19.12 ± 1.76	500.00 ± 0.00
LunarLander-v2	Continuous (8)	Discrete (4)	PPO2 Agent	-452.22 ± 61.24	271.71 ± 17.88
BeamRider-v4	Continuous ($210 \times 160 \times 3$)	Discrete (9)	PPO2 Agent	754.84 ± 214.85	1623.80 ± 482.27
MIMIC-III	Continuous (208)	Discrete (2)	Clinician	-	-

Table D.1: Environment details. The random and demonstrator performances are averaged over 1,000 episodes roll-outs.

probabilities $p = 0.1$ and $p = 0.2$ in the training environments and with probability $p = 0.8$ in the testing environment.

Implementation Details

Similarly to [66] and for a fair comparison, whenever possible, we use the same policy network architecture for all imitation learning benchmarks. For Acrobot [241], Cartpole [242], Lunar Lander [240] and MIMIC-III [148] we use a policy network consisting of two fully-connected hidden layers with 64 units each and with ELU activation. Alternatively, for BeamRider [243], we use as the policy network a convolutional neural network with 3 convolutional layers consisting of 32-64-64 filters, followed by a fully connected layer of size 64, with all layers followed by ReLU activations.

We consider discrete actions in all environments; thus, the output layer of the policy network has the same number of dimensions as the action space. For all the different environments used for evaluation we optimize the parameters using the Adam Optimizer for 10k iterations with learning rate $\lambda = 0.001$ and batch size 64 [66]. Moreover, we use the publicly available code for the different benchmarks used and other than the standardized policy network, we keep the optimal hyperparameters in the original implementations. The experiments were run on a system with 6CPUs, an Nvidia K80 Tesla GPU and 56GB of RAM.

Invariant Causal Imitation Learning: uses a policy network as described above and neural network architectures with two fully connected hidden layers with 64 units

and with ELU activation for each of ϕ, μ^e, g_s, g_η, ψ, c_s and T_{θ_m} in the Acrobot [241], Cartpole [242], Lunar Lander [240] and MIMIC-III [148] experiments. On the other hand, for the BeamRider environment [243], we use a convolutional neural network with 3 convolutional layers consisting of 32-64-64 filters, followed by a fully connected layer of size 64, with all layers followed by ReLU activations for ϕ and μ^e. Moreover, we use neural network architectures with two fully connected hidden layers with 64 units and with ELU activation for g_s, g_η, c_s, T_{θ_m} and for the policy network π. Finally, for ψ we use a neural network with 2 fully connected layers of 64 and $64 \times 7 \times 7$ hidden units followed by 3 transposed convolution layers consisting of 64-64-32 filters.

Behaviour cloning (BC): We implement behaviour cloning by using a policy network architecture as described above. We train the model using cross entropy loss and we optimize it as described above. We use the same hyperparameters for BC-IRM but instead use IRM-v1 objective.

Reward-regularized Classification for Apprenticeship Learning (RCAL): We implement RCAL by adding a sparsity-based loss on the implied rewards [209] and we set the sparsity-based regularization coefficient to 0.01. We use the same policy network architecture for optimization procedure as described above. Moreover, we use the same hyperparameters for RCAL-IRM but instead use IRM-v1 objective.

Energy-based Distribution Matching (EDM): We use the publicly available implementation for EDM [171] from: https://github.com/vanderschaarlab/mlforhealthlabpub. We use the same policy network and optimization procedure as above, which corresponds to the ones used in [171]. Moreover, following the implementation details provided in [171] we set the joint EBM training hyperparameters to noise coefficient $\sigma = 0.01$, buffer size $\kappa = 10000$, length $l = 20$, re-initialization $\delta = 0.05$ and SGLD step size $\alpha = 0.01$. For EDM-IRM we use the same hyperparameters, but instead optimize the IRM-v1 objective.

ValueDice (VDICE): We use the publicly available implementation for VDICE [199] from https://github.com/google-research/google-research/tree/master/value_dice. However, to adapt the model to discrete actions we modify the last layer of the actor network to use Gumbel-softmax. For Acrobot [241],

Cartpole [242], Lunar Lander [240] and MIMIC-III [148], the actor and discriminator network architecture used have two fully connected hidden layers with 64 units and ReLU activation. Conversely, for BeamRider [243] the actor and discriminator networks consist of 3 convolutional layers consisting of 32-64-64 filters, followed by a fully connected layer of size 64, with all layers followed by ReLU activations. As described in [199], orthogonal regularization is used for the actor and a learning rate of 0.00001. The discriminator uses a learning rate of 0.001. For VDICE-IRM we use the same hyperparameters, but instead optimize the IRM-v1 objective.

D.6 Additional Experiments

Ablations

To understand the impact of the different components in the overall loss function used to train ICIL, we performed an ablation experiment on the CartPole [242] control task from OpenAI gym [240]. We follow the same training and testing set-up described in Section 6.5.1.

Let $L = L_\pi + L_{\text{dyn}} + L_{\text{inv}} + L_{\text{mi}} + L_{\text{energy}}$ be the full loss function used for training ICIL. Refer to Section 6.4.1 for details of how each component in L is defined. The results in Figure D.1 illustrate the impact of removing different terms from this loss function on overall performance (average return of the learnt imitation policy in the test environment). The average return is scaled between 1 (expert performance) and 0 (random policy performance). The setting of only using L_π corresponds to the Behaviour Cloning (BC) benchmark.

We notice that while each term in the loss L used to train ICIL is important for the overall performance, the loss term L_{inv} which ensures that the state representation is invariant across environments plays the most significant role on the performance in the test environment. This is due to the fact that L_{inv} is crucial for learning the shared latent structure across the different environments that consists of the causal parents of the actions.

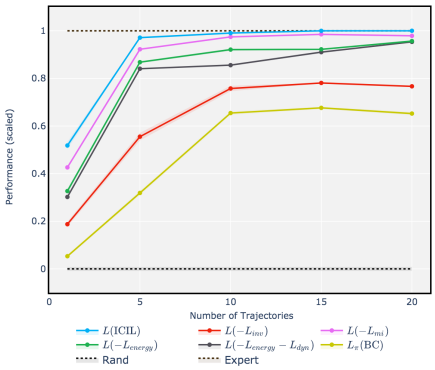


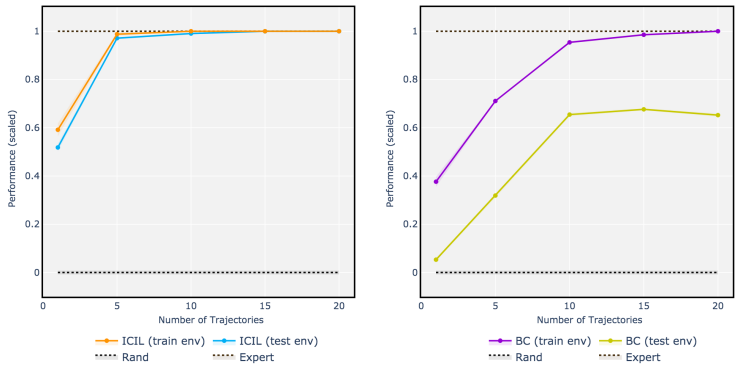
Figure D.1: Evaluating the impact of the different loss components on overall performance for the CartPole control task. x-axis indicates the number of trajectories (in $\{1, 5, 10, 15, 20\}$) with expert demonstrations from each training environment given as input to each ablated version of ICIL and y-axis represents average return of the learnt imitation policy on the test environments, scaled between 1 (expert performance) and 0 (random policy performance).

Train vs. Test Performance

We report here the evaluation metrics for both training and testing environments on the CartPole [242] control task. We follow the same experimental set-up described in Section 6.5.1. In Figure D.2 we report the performance of ICIL and BC when evaluated both on 300 new episode roll-outs from one of the environments used for training as well as when evaluated on a test environment with different multiplicative factors for the noise variables. We notice that the imitation policy learnt by BC relies on the spurious correlations and thus fails to generalize beyond the environments it has been exposed to. On the other hand, ICIL learns an imitation policy that correctly depends on the state variables, which are the ones that are being shared across the different environments.

Robustness to Increasing the Size of Spurious Correlations

Finally, we investigate the robustness of the different imitation learning methods to increasing the size of the spurious correlations (i.e. the number of noise variables in each environment). In this case, we consider again the CartPole [242] control task and the setting where 5 trajectories from each training environment are given as input to each benchmark during training. The spurious correlations in each environment are different multiplicative factors of the last 3 variables in the original state space of the control task. We follow the same set-up described in Appendix D.5 for setting the multiplicative



(a) ICIL **(b)** BC

Figure D.2: Train vs. test performance on CartPole. x-axis indicates the number of trajectories (in $\{1, 5, 10, 15, 20\}$) with expert demonstrations from each training environment given as input to each benchmark and y-axis represents average return of the learnt imitation policy when evaluated on both the train and test environments, scaled between 1 (expert performance) and 0 (random policy performance).

factors for the train and test environments. Figure D.3 illustrates the performance of ICIL, BC and EDM when increasing the number of noise variables used for the different environments. We notice that ICIL is robust to having more spurious correlations, while the performance of BC and EDM degrades more significantly in the case where the observations from the expert demonstrations have a large number of noise variables.

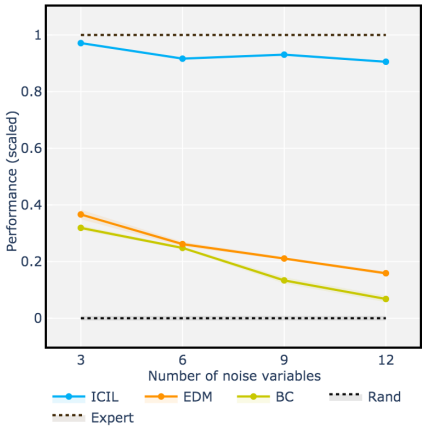


Figure D.3: Robustness to increasing the number of noise variables. x-axis indicates the number of noise variables (in $\{3, 6, 9, 12\}$) that are part of the observations in each environment and y-axis represents average return of the learnt imitation policy on the test environments, scaled between 1 (expert performance) and 0 (random policy performance).

References

[1] Susan Athey and Guido Imbens. “Recursive Partitioning for Heterogeneous Causal Effects”. In: *Proceedings of the National Academy of Sciences* 113.27 (2016), pp. 7353–7360.

[2] Jan-Eric Gustafsson. “Causal inference in educational effectiveness research: A comparison of three methods to investigate effects of homework on student achievement”. In: *School Effectiveness and School Improvement* 24.3 (2013), pp. 275–295.

[3] Paolo Frumento, Fabrizia Mealli, Barbara Pacini, and Donald B Rubin. “Evaluating the effect of training on wages in the presence of noncompliance, nonemployment, and missing outcome data”. In: *Journal of the American Statistical Association* 107.498 (2012), pp. 450–466.

[4] Miguel A Hernán, Sonia Hernández-Dıaz, and James M Robins. “A structural approach to selection bias”. In: *Epidemiology* (2004), pp. 615–625.

[5] Elias Bareinboim and Judea Pearl. “Controlling selection bias in causal inference”. In: *Artificial Intelligence and Statistics*. PMLR. 2012, pp. 100–108.

[6] Mette Nørgaard, Vera Ehrenstein, and Jan P Vandenbroucke. “Confounding in observational studies based on large health care databases: problems and potential solutions–a primer for the clinician”. In: *Clinical epidemiology* 9 (2017), p. 185.

[7] Christopher Paul Adams and Van Vu Brantner. “Spending on new drug development 1”. In: *Health economics* 19.2 (2010), pp. 130–141.

[8] Jared C Foster, Jeremy MG Taylor, and Stephen J Ruberg. “Subgroup identification from randomized clinical trial data”. In: *Statistics in medicine* 30.24 (2011), pp. 2867–2880.

[9] CM Booth and IF Tannock. “Randomised controlled trials and population-based observational research: partners in the evolution of medical evidence”. In: *British journal of cancer* 110.3 (2014), p. 551.

[10] Paul R Rosenbaum and Donald B Rubin. “The central role of the propensity score in observational studies for causal effects”. In: *Biometrika* 70.1 (1983), pp. 41–55.

[11] Alexandre Belloni, Victor Chernozhukov, and Christian Hansen. “Inference on treatment effects after selection among high-dimensional controls”. In: *The Review of Economic Studies* 81.2 (2014), pp. 608–650.

[12] Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

[13] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. “Double/debiased machine learning for treatment and causal parameters”. In: *arXiv preprint arXiv:1608.00060* (2016).

[14] James M Robins, Miguel Angel Hernan, and Babette Brumback. "Marginal structural models and causal inference in epidemiology". In: *Epidemiology* 11.5 (2000), pp. 550–560.

[15] Zoe Fewell, Miguel A Hernán, Frederick Wolfe, Kate Tilling, Hyon Choi, Jonathan AC Sterne, et al. "Controlling for time-dependent confounding using marginal structural models". In: *Stata J* 4.4 (2004), pp. 402–420.

[16] Mohammad Ali Mansournia, Mahyar Etminan, Goodarz Danaei, Jay S Kaufman, and Gary Collins. "Handling time varying confounding in observational research". In: *bmj* 359 (2017), j4587.

[17] Rhian M Daniel, SN Cousens, BL De Stavola, Michael G Kenward, and JAC Sterne. "Methods for dealing with time-dependent confounding". In: *Statistics in medicine* 32.9 (2013), pp. 1584–1618.

[18] Bryan Lim and Mihaela van der Schaar. "Disease-atlas: Navigating disease trajectories with deep learning". In: *Machine Learning for Healthcare Conference* (2018).

[19] Nenad Tomašev, Xavier Glorot, Jack W Rae, Michal Zielinski, Harry Askham, Andre Saraiva, Anne Mottram, Clemens Meyer, Suman Ravuri, Ivan Protsyuk, et al. "A clinically applicable approach to continuous prediction of future acute kidney injury". In: *Nature* 572.7767 (2019), pp. 116–119.

[20] MD McCauley and Dawood Darbar. "A new paradigm for predicting risk of Torsades de Pointes during drug development: Commentary on:"Improved prediction of drug-induced Torsades de Pointes through simulations of dynamics and machine learning algorithms"". In: *Clinical Pharmacology & Therapeutics* 100.4 (2016), pp. 324–326.

[21] Ahmed M Alaa and Mihaela van der Schaar. "Prognostication and risk factors for cystic fibrosis via automated machine learning". In: *Scientific reports* 8.1 (2018), pp. 1–19.

[22] Brett K Beaulieu-Jones, Samuel G Finlayson, William Yuan, Russ B Altman, Isaac S Kohane, Vinay Prasad, and Kun-Hsing Yu. "Examining the Use of Real-World Evidence in the Regulatory Process". In: *Clinical Pharmacology & Therapeutics* (2019).

[23] Alison Cave, Nikolai C Brun, Fergus Sweeney, Guido Rasi, Thomas Senderovitz, and HMA-EMA Joint Big Data Taskforce. "Big Data–How to Realize the Promise". In: *Clinical Pharmacology & Therapeutics* 107.4 (2020), pp. 753–761.

[24] Jennifer L Hill. "Bayesian nonparametric modeling for causal inference". In: *Journal of Computational and Graphical Statistics* 20.1 (2011), pp. 217–240.

[25] Ahmed M Alaa, Michael Weisz, and Mihaela Van Der Schaar. "Deep Counterfactual Networks with Propensity-Dropout". In: *ICML 2017 Workshop on Principled Approaches to Deep Learning,* (2017).

[26] Ahmed M Alaa and Mihaela van der Schaar. "Bayesian Inference of Individualized Treatment Effects using Multi-Task Gaussian Processes". In: *Advances in Neural Information Processing Systems*. 2017, pp. 3424–3432.

[27] Stefan Wager and Susan Athey. "Estimation and Inference of Heterogeneous Treatment Effects using Random Forests". In: *Journal of the American Statistical Association* 113.523 (2018), pp. 1228–1242.

[28] Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. "Representation Learning for Treatment Effect Estimation from Observational Data". In: *Advances in Neural Information Processing Systems*. 2018, pp. 2633–2643.

[29] Fredrik Johansson, Uri Shalit, and David Sontag. "Learning Representations for Counterfactual Inference". In: *International Conference on Machine Learning*. 2016, pp. 3020–3029.

[30] Fredrik D Johansson, Nathan Kallus, Uri Shalit, and David Sontag. "Learning Weighted Representations for Generalization across Designs". In: *arXiv preprint arXiv:1802.08598* (2018).

[31] Ahmed Alaa and Mihaela van der Schaar. "Limits of estimating heterogeneous treatment effects: Guidelines for practical algorithm design". In: *International Conference on Machine Learning*. 2018, pp. 129–138.

[32] Yao Zhang, Alexis Bellot, and Mihaela van der Schaar. "Learning overlapping representations for the estimation of individualized treatment effects". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)* (2020).

[33] Alicia Curth and Mihaela van der Schaar. "Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 1810–1818.

[34] Alicia Curth and Mihaela van der Schaar. "On inductive biases for heterogeneous treatment effect estimation". In: *Advances in Neural Information Processing Systems* 34 (2021).

[35] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. "GANITE: Estimation of Individualized Treatment Effects using Generative Adversarial Nets". In: *International Conference on Learning Representations (ICLR)* (2018).

[36] Patrick Schwab, Lorenz Linhardt, Stefan Bauer, Joachim M Buhmann, and Walter Karlen. "Learning Counterfactual Representations for Estimating Individual Dose-Response Curves". In: *arXiv preprint arXiv:1902.00981* (2019).

[37] Jean Kaddour, Yuchen Zhu, Qi Liu, Matt J Kusner, and Ricardo Silva. "Causal Effect Inference for Structured Treatments". In: *Advances in Neural Information Processing Systems* 34 (2021).

[38] Zhaozhi Qian, Alicia Curth, and Mihaela van der Schaar. "Estimating Multi-cause Treatment Effects via Single-cause Perturbation". In: *Advances in Neural Information Processing Systems* 34 (2021).

[39] Sonali Parbhoo, Stefan Bauer, and Patrick Schwab. "Ncore: Neural counterfactual representation learning for combinations of treatments". In: *arXiv preprint arXiv:2103.11175* (2021).

[40] Peter Schulam and Suchi Saria. "Reliable decision support using counterfactual models". In: *Advances in Neural Information Processing Systems*. 2017, pp. 1697–1708.

[41] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. "Shortcut learning in deep neural networks". In: *Nature Machine Intelligence* 2.11 (2020), pp. 665–673.

[42] Junzhe Zhang. "Designing optimal dynamic treatment regimes: A causal reinforcement learning approach". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 11012–11022.

[43] Trent Kyono, Yao Zhang, and Mihaela van der Schaar. "Castle: Regularization via auxiliary causal graph discovery". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1501–1512.

[44] Trent Kyono, Yao Zhang, Alexis Bellot, and Mihaela van der Schaar. “MIRACLE: Causally-Aware Imputation via Learning Missing Data Mechanisms”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[45] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. “Towards causal representation learning”. In: *arXiv preprint arXiv:2102.11107* (2021).

[46] Cheng Zhang, Kun Zhang, and Yingzhen Li. “A causal view on robustness of neural networks”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 289–301.

[47] Bernhard Schölkopf. “Causality for machine learning”. In: *Probabilistic and Causal Inference: The Works of Judea Pearl*. 2022, pp. 765–804.

[48] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. “Invariant risk minimization”. In: *arXiv preprint arXiv:1907.02893* (2019).

[49] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[50] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation Learning: A Review and New Perspectives.” In: *arXiv preprint arXiv:1206.5538* (2012).

[51] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. “Domain-adversarial training of neural networks”. In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 2096–2030.

[52] Benjamin Djulbegovic, Shira Elqayam, and William Dale. “Rational decision making in medicine: implications for overuse and underuse”. In: *Journal of evaluation in clinical practice* 24.3 (2018), pp. 655–665.

[53] Ioana Bica, Ahmed M Alaa, Craig Lambert, and Mihaela van der Schaar. “From real-world patient data to individualized treatment effects using machine learning: Current and future methods to address underlying challenges”. In: *Clinical Pharmacology & Therapeutics* (2020).

[54] Ioana Bica, James Jordon, and Mihaela van der Schaar. “Estimating the effects of continuous-valued interventions using generative adversarial networks”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 16434–16445.

[55] Ioana Bica, Ahmed M Alaa, James Jordon, and Mihaela van der Schaar. “Estimating counterfactual treatment outcomes over time through adversarially balanced representations”. In: *International Conference on Learning Representations* (2020).

[56] Ioana Bica, Ahmed Alaa, and Mihaela Van Der Schaar. “Time series deconfounder: Estimating treatment effects over time in the presence of hidden confounders”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 884–895.

[57] Ioana Bica, Daniel Jarrett, Alihan Hüyük, and Mihaela van der Schaar. “Learning" What-if" Explanations for Sequential Decision-Making”. In: *International Conference on Learning Representations* (2021).

[58] Ioana Bica, Daniel Jarrett, and Mihaela van der Schaar. “Invariant Causal Imitation Learning for Generalizable Policies”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[59] Jeroen Berrevoets, James Jordon, Ioana Bica, Mihaela van der Schaar, et al. "OrganITE: Optimal transplant donor organ offering using an individual treatment effect". In: *Advances in neural information processing systems* 33 (2020), pp. 20037–20050.

[60] Can Xu, Ahmed Alaa, Ioana Bica, Brent Ershoff, Maxime Cannesson, and Mihaela van der Schaar. "Learning matching representations for individualized organ transplantation allocation". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 2134–2142.

[61] Zhaozhi Qian, Yao Zhang, Ioana Bica, Angela Wood, and Mihaela van der Schaar. "SyncTwin: Treatment Effect Estimation with Longitudinal Outcomes". In: *Advances in Neural Information Processing Systems* 34 (2021).

[62] Jeroen Berrevoets, Alicia Curth, Ioana Bica, Eoin McKinney, and Mihaela van der Schaar. "Disentangled Counterfactual Recurrent Networks for Treatment Effect Inference over Time". In: *arXiv preprint arXiv:2112.03811* (2021).

[63] Trent Kyono, Ioana Bica, Zhaozhi Qian, and Mihaela van der Schaar. "Selecting Treatment Effects Models for Domain Adaptation Using Causal Knowledge". In: *arXiv preprint arXiv:2102.06271* (2021).

[64] William R Zame, Ioana Bica, Cong Shen, Alicia Curth, Hyun-Suk Lee, Stuart Bailey, James Weatherall, David Wright, Frank Bretz, and Mihaela van der Schaar. "Machine learning for clinical trials in the era of COVID-19". In: *Statistics in biopharmaceutical research* 12.4 (2020), pp. 506–517.

[65] Pedro Baqui, Ioana Bica, Valerio Marra, Ari Ercole, and Mihaela van Der Schaar. "Ethnic and regional variations in hospital mortality from COVID-19 in Brazil: a cross-sectional observational study". In: *The Lancet Global Health* 8.8 (2020), e1018–e1026.

[66] Daniel Jarrett, Ioana Bica, and Mihaela van der Schaar. "Strictly batch imitation learning by energy-based distribution matching". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7354–7365.

[67] Daniel Jarrett, Jinsung Yoon, Ioana Bica, Zhaozhi Qian, Ari Ercole, and Mihaela van der Schaar. "Clairvoyance: A pipeline toolkit for medical time series". In: *International Conference on Learning Representations*. 2020.

[68] Alex J Chan, Ioana Bica, Alihan Huyuk, Daniel Jarrett, and Mihaela van der Schaar. "The Medkit-Learn (ing) Environment: Medical Decision Modelling through Simulation". In: *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track* (2021).

[69] Daniel Jarrett, Ioana Bica, and Mihaela van der Schaar. "Time-series Generation by Contrastive Imitation". In: *Advances in Neural Information Processing Systems* 34 (2021).

[70] Larry C Clark, Gerald F Combs, Bruce W Turnbull, Elizabeth H Slate, Dan K Chalker, James Chow, Loretta S Davis, Renee A Glover, Gloria F Graham, Earl G Gross, et al. "Effects of selenium supplementation for cancer prevention in patients with carcinoma of the skin: a randomized controlled trial". In: *Jama* 276.24 (1996), pp. 1957–1963.

[71] Christine Léauté-Labrèze, Peter Hoeger, Juliette Mazereeuw-Hautier, Laurent Guibaud, Eulalia Baselga, Gintas Posiunas, Roderic J Phillips, Hector Caceres, Juan Carlos Lopez Gutierrez, Rosalia Ballona, et al. "A randomized, controlled trial of oral propranolol in infantile hemangioma". In: *New England Journal of Medicine* 372.8 (2015), pp. 735–746.

[72] Peter M Rothwell. "External validity of randomised controlled trials:"to whom do the results of this trial apply?"" In: *The Lancet* 365.9453 (2005), pp. 82–93.

[73] Laura E Bothwell, Jeremy A Greene, Scott H Podolsky, and David S Jones. *Assessing the gold standard—lessons from the history of RCTs*. 2016.

[74] Thomas R Frieden. "Evidence for health decision making—beyond randomized, controlled trials". In: *New England Journal of Medicine* 377.5 (2017), pp. 465–475.

[75] Laura F Hutchins, Joseph M Unger, John J Crowley, Charles A Coltman Jr, and Kathy S Albain. "Underrepresentation of patients 65 years of age or older in cancer-treatment trials". In: *New England Journal of Medicine* 341.27 (1999), pp. 2061–2067.

[76] A Ramamoorthy, MA Pacanowski, J Bull, and L Zhang. "Racial/ethnic differences in drug disposition and response: review of recently approved drugs". In: *Clinical Pharmacology & Therapeutics* 97.3 (2015), pp. 263–273.

[77] Julia Adler-Milstein, A Jay Holmgren, Peter Kralovec, Chantal Worzala, Talisha Searcy, and Vaishali Patel. "Electronic health record adoption in US hospitals: the emergence of a digital "advanced use" divide". In: *Journal of the American Medical Informatics Association* 24.6 (2017), pp. 1142–1148.

[78] Peipei Chen, Wei Dong, Xudong Lu, Uzay Kaymak, Kunlun He, and Zhengxing Huang. "Deep representation learning for individualized treatment effect estimation using electronic health records". In: *Journal of biomedical informatics* 100 (2019), p. 103303.

[79] Jersey Neyman. "Sur les applications de la théorie des probabilités aux experiences agricoles: Essai des principes". In: *Roczniki Nauk Rolniczych* 10 (1923), pp. 1–51.

[80] Donald B Rubin. "Bayesian inference for causal effects: The role of randomization". In: *The Annals of statistics* (1978), pp. 34–58.

[81] James M Robins and Miguel A Hernán. "Estimation of the causal effects of time-varying exposures". In: *Longitudinal data analysis*. Chapman and Hall/CRC, 2008, pp. 547–593.

[82] Michael Wallington, Emma B Saxon, Martine Bomb, Rebecca Smittenaar, Matthew Wickenden, Sean McPhail, Jem Rashbass, David Chao, John Dewar, Denis Talbot, et al. "30-day mortality after systemic anticancer treatment for breast and lung cancer in England: a population-based, observational study". In: *The Lancet Oncology* 17.9 (2016), pp. 1203–1216.

[83] Judea Pearl. "Understanding propensity scores". In: *Causality: models, reasoning, and inference* 2 (2009), pp. 348–352.

[84] Colin B Fogarty, Mark E Mikkelsen, David F Gaieski, and Dylan S Small. "Discrete optimization for interpretable study populations and randomization inference in an observational study of severe sepsis mortality". In: *Journal of the American Statistical Association* 111.514 (2016), pp. 447–458.

[85] Alexander D'Amour, Peng Ding, Avi Feller, Lihua Lei, and Jasjeet Sekhon. "Overlap in observational studies with high-dimensional covariates". In: *arXiv preprint arXiv:1711.02582* (2017).

[86] Fredrik D Johansson, Dennis Wei, Michael Oberst, Tian Gao, Gabriel Brat, David Sontag, and Kush R Varshney. "Characterization of Overlap in Observational Studies". In: *AISTATS* (2020).

[87] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. "Estimating the support of a high-dimensional distribution". In: *Neural computation* 13.7 (2001), pp. 1443–1471.

[88] Gabriel A Brat, Denis Agniel, Andrew Beam, Brian Yorkgitis, Mark Bicket, Mark Homer, Kathe P Fox, Daniel B Knecht, Cheryl N McMahill-Walraven, Nathan Palmer, et al. "Postsurgical prescriptions for opioid naive patients and association with overdose and misuse: retrospective cohort study". In: *Bmj* 360 (2018), j5790.

[89] Andrew Mente, Lawrence de Koning, Harry S Shannon, and Sonia S Anand. "A systematic review of the evidence supporting a causal link between dietary factors and coronary heart disease". In: *Archives of internal medicine* 169.7 (2009), pp. 659–669.

[90] Marcel Młyńczak and Hubert Krysztofiak. "Discovery of causal paths in cardiorespiratory parameters: a time-independent approach in elite athletes". In: *Frontiers in Physiology* 9 (2018), p. 1455.

[91] James M Robins, Andrea Rotnitzky, and Daniel O Scharfstein. "Sensitivity analysis for selection bias and unmeasured confounding in missing data and causal inference models". In: *Statistical models in epidemiology, the environment, and clinical trials*. Springer, 2000, pp. 1–94.

[92] Daniel Scharfstein, Aidan McDermott, Iván Dıaz, Marco Carone, Nicola Lunardon, and Ibrahim Turkoz. "Global sensitivity analysis for repeated measures studies with informative drop-out: A semi-parametric approach". In: *Biometrics* 74.1 (2018), pp. 207–219.

[93] Alex Franks, Alex D'Amour, and Avi Feller. "Flexible sensitivity analysis for observational studies without observable implications". In: *Journal of the American Statistical Association* (2019), pp. 1–38.

[94] Timothy L Lash, Matthew P Fox, Richard F MacLehose, George Maldonado, Lawrence C McCandless, and Sander Greenland. "Good practices for quantitative bias analysis". In: *International journal of epidemiology* 43.6 (2014), pp. 1969–1985.

[95] Manabu Kuroki and Judea Pearl. "Measurement bias and effect restoration in causal inference". In: *Biometrika* 101.2 (2014), pp. 423–437.

[96] Nathan Kallus, Xiaojie Mao, and Madeleine Udell. "Causal inference with noisy and missing covariates via matrix factorization". In: *Advances in Neural Information Processing Systems*. 2018, pp. 6921–6932.

[97] Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. "Causal effect inference with deep latent-variable models". In: *Advances in Neural Information Processing Systems*. 2017, pp. 6446–6456.

[98] Changhee Lee, Nicholas Mastronarde, and Mihaela van der Schaar. "Estimation of Individual Treatment Effect in Latent Confounder Models via Adversarial Learning". In: *NIPS Machine Learning for Health Workshop 2018* (2018).

[99] Yixin Wang and David M Blei. "The blessings of multiple causes". In: *Journal of the American Statistical Association* just-accepted (2019), pp. 1–71.

[100] Rajesh Ranganath and Adler Perotte. "Multiple causal inference with latent confounding". In: *arXiv preprint arXiv:1805.08273* (2018).

[101] Linying Zhang, Yixin Wang, Anna Ostropolets, Jami J Mulgrave, David M Blei, and George Hripcsak. "The medical deconfounder: assessing treatment effects with electronic health records". In: *Machine Learning for Healthcare Conference*. PMLR. 2019, pp. 490–512.

[102] Uri Shalit, Fredrik D Johansson, and David Sontag. "Estimating Individual Treatment Effect: Generalization Bounds and Algorithms". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 3076–3085.

[103] Hugh A Chipman, Edward I George, Robert E McCulloch, et al. "BART: Bayesian additive regression trees". In: *The Annals of Applied Statistics* 4.1 (2010), pp. 266–298.

[104] P Richard Hahn, Jared S Murray, and Carlos Carvalho. "Bayesian regression tree models for causal inference: regularization, confounding, and heterogeneous effects". In: *arXiv preprint arXiv:1706.09523* (2017).

[105] Ahmed M Alaa and Mihaela van der Schaar. "Bayesian nonparametric causal inference: Information rates and learning algorithms". In: *IEEE Journal of Selected Topics in Signal Processing* 12.5 (2018), pp. 1031–1046.

[106] T Wendling, K Jung, A Callahan, A Schuler, NH Shah, and B Gallego. "Comparing methods for estimation of heterogeneous treatment effects using observational data from health care databases". In: *Statistics in medicine* 37.23 (2018), pp. 3309–3324.

[107] Sahil Khera, Dhaval Kolte, Tanush Gupta, Andrew Goldsweig, Poonam Velagapudi, Ankur Kalra, Gilbert HL Tang, Wilbert S Aronow, Gregg C Fonarow, Deepak L Bhatt, et al. "Association between hospital volume and 30-day readmissions following transcatheter aortic valve replacement". In: *JAMA cardiology* 2.7 (2017), pp. 732–741.

[108] Dhaval Kolte, Kevin F Kennedy, Mehdi H Shishehbor, J Dawn Abbott, Sahil Khera, Peter Soukas, Shafiq T Mamdani, Omar N Hyder, Douglas E Drachman, and Herbert D Aronow. "Thirty-day readmissions after endovascular or surgical therapy for critical limb ischemia: analysis of the 2013 to 2014 Nationwide Readmissions Databases". In: *Circulation* 136.2 (2017), pp. 167–176.

[109] Guido W Imbens. "The Role of the Propensity Score in Estimating Dose-Response Functions". In: *Biometrika* 87.3 (2000), pp. 706–710.

[110] Kosuke Imai and David A Van Dyk. "Causal Inference with General Treatment Regimes: Generalizing the Propensity Score". In: *Journal of the American Statistical Association* 99.467 (2004), pp. 854–866.

[111] Keisuke Hirano and Guido W Imbens. "The Propensity Score with Continuous Treatments". In: *Applied Bayesian Modeling and Causal Inference from Incomplete-Data Perspectives* 226164 (2004), pp. 73–84.

[112] James Robins. "A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect". In: *Mathematical modelling* 7.9-12 (1986), pp. 1393–1512.

[113] James M Robins. "Correcting for non-compliance in randomized trials using structural nested mean models". In: *Communications in Statistics-Theory and methods* 23.8 (1994), pp. 2379–2412.

[114] Yanbo Xu, Yanxun Xu, and Suchi Saria. "A Bayesian nonparametric approach for estimating individualized treatment-response curves". In: *Machine Learning for Healthcare Conference*. 2016, pp. 282–300.

[115] Jason Roy, Kirsten J Lum, and Michael J Daniels. "A Bayesian nonparametric approach to marginal structural models for point treatments and a continuous or survival outcome". In: *Biostatistics* 18.1 (2016), pp. 32–47.

[116] Hossein Soleimani, Adarsh Subbaswamy, and Suchi Saria. "Treatment-response models for counterfactual reasoning with continuous-time, continuous-valued interventions". In: *arXiv preprint arXiv:1704.02038* (2017).

[117] Bryan Lim, Ahmed Alaa, and Mihaela van der Schaar. "Forecasting Treatment Responses Over Time Using Recurrent Marginal Structural Networks". In: *Advances in Neural Information Processing Systems*. 2018, pp. 7493–7503.

[118] Changran Geng, Harald Paganetti, and Clemens Grassberger. "Prediction of Treatment Response for Combined Chemo-and Radiation Therapy for Non-Small Cell Lung Cancer Patients Using a Bio-Mathematical Model". In: *Scientific reports* 7.1 (2017), p. 13542.

[119] Ahmed Alaa and Mihaela Van Der Schaar. "Validating Causal Inference Models via Influence Functions". In: *International Conference on Machine Learning*. 2019, pp. 191–201.

[120] Solveig Badillo, Balazs Banfai, Fabian Birzele, Iakov I Davydov, Lucy Hutchinson, Tony Kam-Thong, Juliane Siebourg-Polster, Bernhard Steiert, and Jitao David Zhang. "An Introduction to Machine Learning". In: *Clinical Pharmacology & Therapeutics* 107.4 (2020), pp. 871–885.

[121] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: the approach based on influence functions*. Vol. 196. John Wiley & Sons, 2011.

[122] James Robins, Lingling Li, Eric Tchetgen, Aad van der Vaart, et al. "Higher order influence functions and minimax estimation of nonlinear functionals". In: *Probability and statistics: essays in honor of David A. Freedman*. Institute of Mathematical Statistics, 2008, pp. 335–421.

[123] Dimitris Bertsimas, Nathan Kallus, Alexander M Weinstein, and Ying Daisy Zhuo. "Personalized Diabetes Management using Electronic Medical Records". In: *Diabetes Care* 40.2 (2017), pp. 210–217.

[124] Donna Döpp-Zemel and AB Johan Groeneveld. "High-dose norepinephrine treatment: determinants of mortality and futility in critically ill patients". In: *American Journal of Critical Care* 22.1 (2013), pp. 22–32.

[125] Kyle Wang, Michael J Eblan, Allison M Deal, Matthew Lipner, Timothy M Zagar, Yue Wang, Panayiotis Mavroidis, Carrie B Lee, Brian C Jensen, Julian G Rosenman, et al. "Cardiac Toxicity after Radiotherapy for Stage III Non–Small-Cell Lung Cancer: Pooled Analysis of Dose-Escalation Trials Delivering 70 to 90 Gy". In: *Journal of Clinical Oncology* 35.13 (2017), p. 1387.

[126] Peter Spirtes. "A Tutorial On Causal Inference". In: (2009).

[127] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. "Deep Sets". In: *Advances in Neural Information Processing Systems*. 2017, pp. 3391–3401.

[128] Donald B Rubin. "Bayesianly justifiable and relevant frequency calculations for the applies statistician". In: *The Annals of Statistics* (1984), pp. 1151–1172.

[129] John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, Joshua M Stuart, Cancer Genome Atlas Research Network, et al. "The Cancer Genome Atlas Pan-Cancer Analysis Project". In: *Nature Genetics* 45.10 (2013), p. 1113.

[130] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. "MIMIC-III, A Freely Accessible Critical Care Database". In: *Scientific Data* 3 (2016), p. 160035.

[131] Ricardo Silva. "Observational-Interventional Priors for Dose-Response Learning". In: *Advances in Neural Information Processing Systems*. 2016, pp. 1561–1569.

[132] Sheng Li and Yun Fu. "Matching on Balanced Nonlinear Representations for Treatment Effects Estimation". In: *Advances in Neural Information Processing Systems*. 2017, pp. 929–939.

[133] Robert W Platt, Enrique F Schisterman, and Stephen R Cole. "Time-modified confounding". In: *American journal of epidemiology* 170.6 (2009), pp. 687–694.

[134] Mohammad Ali Mansournia, Goodarz Danaei, Mohammad Hossein Forouzanfar, Mahmood Mahmoodi, Mohsen Jamali, Nasrin Mansournia, and Kazem Mohammad. "Effect of physical activity on functional performance and knee pain in patients with osteoarthritis: analysis with marginal structural models". In: *Epidemiology* (2012), pp. 631–640.

[135] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. "Deep domain generalization via conditional invariant adversarial networks". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 624–639.

[136] Alice Schoenauer Sebag, Louise Heinrich, Marc Schoenauer, Michèle Sebag, Lani Wu, and Steven Altschuler. "Multi-Domain Adversarial Learning". In: *ICLR'19-Seventh annual International Conference on Learning Representations*. 2019.

[137] Miguel A Hernán, Babette Brumback, and James M Robins. "Marginal structural models to estimate the joint causal effect of nonrandomized treatments". In: *Journal of the American Statistical Association* 96.454 (2001), pp. 440–448.

[138] Kathleen M Mortimer, Romain Neugebauer, Mark Van Der Laan, and Ira B Tager. "An application of model-fitting procedures for marginal structural models". In: *American Journal of Epidemiology* 162.4 (2005), pp. 382–388.

[139] Judith J Lok et al. "Statistical modeling of causal effects in continuous time". In: *The Annals of Statistics* 36.3 (2008), pp. 1464–1507.

[140] Elja Arjas and Jan Parner. "Causal reasoning from longitudinal data". In: *Scandinavian Journal of Statistics* 31.2 (2004), pp. 171–187.

[141] Miguel Ángel Hernán, Babette Brumback, and James M Robins. "Marginal structural models to estimate the causal effect of zidovudine on the survival of HIV-positive men". In: *Epidemiology* (2000), pp. 561–570.

[142] James M Robins. "Association, causation, and marginal structural models". In: *Synthese* 121.1 (1999), pp. 151–179.

[143] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[144] Yarin Gal and Zoubin Ghahramani. “A theoretically grounded application of dropout in recurrent neural networks”. In: *Advances in neural information processing systems*. 2016, pp. 1019–1027.

[145] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. “Phased lstm: Accelerating recurrent network training for long or event-based sequences”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 3882–3890.

[146] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. “Detecting change in data streams”. In: *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment. 2004, pp. 180–191.

[147] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. “Analysis of representations for domain adaptation”. In: *Advances in Neural Information Processing Systems*. 2007, pp. 137–144.

[148] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. “MIMIC-III, a freely accessible critical care database”. In: *Scientific data* 3.1 (2016), p. 160035.

[149] Panagiotis J Vlachostergios and Bishoy M Faltas. “Treatment resistance in urothelial carcinoma: an evolutionary perspective”. In: *Nature Reviews Clinical Oncology* (2018).

[150] Frank Kroschinsky, Friedrich Stölzel, Simone von Bonin, Gernot Beutel, Matthias Kochanek, Michael Kiehl, and Peter Schellongowski. “New drugs, new toxicities: severe side effects of modern targeted and immunotherapy of cancer and their management”. In: *Critical Care* 21.1 (2017), p. 89.

[151] Stefan Wager and Susan Athey. “Estimation and inference of heterogeneous treatment effects using random forests”. In: *Journal of the American Statistical Association* just-accepted (2017).

[152] Michael E Tipping and Christopher M Bishop. “Probabilistic principal component analysis”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622.

[153] Rajesh Ranganath, Linpeng Tang, Laurent Charlin, and David Blei. “Deep exponential families”. In: *Artificial Intelligence and Statistics*. 2015, pp. 762–771.

[154] Dustin Tran and David M Blei. “Implicit Causal Models for Genome-wide Association Studies”. In: *arXiv preprint arXiv:1710.10742* (2017).

[155] David Heckerman. “Accounting for hidden common causes when infering cause and effect from observational data”. In: *arXiv preprint arXiv:1801.00727* (2018).

[156] J. Pearl. *Causality*. Causality: Models, Reasoning, and Inference. Cambridge University Press, 2009. URL: https://books.google.com/books?id=f4nuexsNVZIC.

[157] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000.

[158] Philippe Leray, Stijn Meganek, Sam Maes, and Bernard Manderick. “Causal graphical models with latent variables: learning and inference”. In: *Innovations in Bayesian Networks*. Springer, 2008, pp. 219–249.

[159] Fattaneh Jabbari, Joseph Ramsey, Peter Spirtes, and Gregory Cooper. "Discovery of causal models that contain latent variables through Bayesian scoring of independence constraints". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2017, pp. 142–157.

[160] Vineet K Raghu, Joseph D Ramsey, Alison Morris, Dimitrios V Manatakis, Peter Sprites, Panos K Chrysanthis, Clark Glymour, and Panayiotis V Benos. "Comparison of strategies for scalable causal discovery of latent variable models from mixed data". In: *International journal of data science and analytics* 6.1 (2018), pp. 33–45.

[161] Alexander D'Amour. "On Multi-Cause Causal Inference with Unobserved Confounding: Counterexamples, Impossibility, and Alternatives". In: *arXiv preprint arXiv:1902.10286* (2019), pp. 3478–3486.

[162] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning". In: *International Conference on Machine Learning*. 2016, pp. 1050–1059.

[163] Mario Forni, Marc Hallin, Marco Lippi, and Lucrezia Reichlin. "The generalized dynamic-factor model: Identification and estimation". In: *Review of Economics and statistics* 82.4 (2000), pp. 540–554.

[164] Mario Forni, Marc Hallin, Marco Lippi, and Lucrezia Reichlin. "The generalized dynamic factor model: one-sided estimation and forecasting". In: *Journal of the American Statistical Association* 100.471 (2005), pp. 830–840.

[165] Ben S Bernanke, Jean Boivin, and Piotr Eliasz. "Measuring the effects of monetary policy: a factor-augmented vector autoregressive (FAVAR) approach". In: *The Quarterly journal of economics* 120.1 (2005), pp. 387–422.

[166] Gregory A Schmidt, Jess Mandel, Daniel J Sexton, and Robert S Hockberger. "Evaluation and management of suspected sepsis and septic shock in adults". In: *UpToDate. Available online: https://www. uptodate. com/contents/evaluation-and-management-of-suspected-sepsisand-septic-shock-in-adults (accessed on 29 September 2017)* (2016).

[167] Thomas WL Scheeren, Jan Bakker, Daniel De Backer, Djillali Annane, Pierre Asfar, E Christiaan Boerma, Maurizio Cecconi, Arnaldo Dubin, Martin W Dünser, Jacques Duranteau, et al. "Current use of vasopressors in septic shock". In: *Annals of intensive care* 9.1 (2019), p. 20.

[168] Brent C James and M Elizabeth H Hammond. "The challenge of variation in medical practice". In: *Archives of pathology & laboratory medicine* 124.7 (2000), pp. 1001–1003.

[169] Gert P Westert, Stef Groenewoud, John E Wennberg, Catherine Gerard, Phil DaSilva, Femke Atsma, and David C Goodman. "Medical practice variation: public reporting a first necessary step to spark change". In: *International Journal for Quality in Health Care* 30.9 (2018), pp. 731–735.

[170] Jessica Van Parys and Jonathan Skinner. "Physician Practice Style Variation—Implications for Policy". In: *JAMA internal medicine* 176.10 (2016), pp. 1549–1550.

[171] Daniel Jarrett and Mihaela van der Schaar. "Inverse Active Sensing: Modeling and Understanding Timely Decision-Making". In: *International Conference on Machine Learning (ICML)* (2020).

[172] Matthew A Rysavy, Lei Li, Edward F Bell, Abhik Das, Susan R Hintz, Barbara J Stoll, Betty R Vohr, Waldemar A Carlo, Seetha Shankaran, Michele C Walsh, et al. "Between-hospital variation in treatment and outcomes in extremely preterm infants". In: *N Engl J Med* 372 (2015), pp. 1801–1811.

[173] Brent M McGrath. *How doctors think*. 2009.

[174] Andrew Y Ng and Stuart J Russell. "Algorithms for inverse reinforcement learning." In: *Icml*. Vol. 1. 2000, pp. 663–670.

[175] Pieter Abbeel and Andrew Y Ng. "Apprenticeship learning via inverse reinforcement learning". In: *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 1.

[176] Jaedeug Choi and Kee-Eung Kim. "Inverse reinforcement learning in partially observable environments". In: *Journal of Machine Learning Research* 12.Mar (2011), pp. 691–730.

[177] Edouard Klein, Matthieu Geist, and Olivier Pietquin. "Batch, off-policy and model-free apprenticeship learning". In: *European Workshop on Reinforcement Learning*. Springer. 2011, pp. 285–296.

[178] Edouard Klein, Matthieu Geist, Bilal Piot, and Olivier Pietquin. "Inverse reinforcement learning through structured classification". In: *Advances in neural information processing systems*. 2012, pp. 1007–1015.

[179] Donghun Lee, Srivatsan Srinivasan, and Finale Doshi-Velez. "Truly Batch Apprenticeship Learning with Deep Successor Features". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 5909–5915. URL: https://doi.org/10.24963/ijcai.2019/819.

[180] Donald B Rubin. "Causal inference using potential outcomes: Design, modeling, decisions". In: *Journal of the American Statistical Association* 100.469 (2005), pp. 322–331.

[181] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. "Imitation learning: A survey of learning methods". In: *ACM Computing Surveys (CSUR)* 50.2 (2017), pp. 1–35.

[182] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. "An algorithmic perspective on imitation learning". In: *Foundations and Trends® in Robotics* 7.1-2 (2018), pp. 1–179.

[183] Faraz Torabi, Garrett Warnell, and Peter Stone. "Recent advances in imitation learning from observation". In: *arXiv preprint arXiv:1905.13566* (2019).

[184] Michail G Lagoudakis and Ronald Parr. "Least-squares policy iteration". In: *Journal of machine learning research* 4.Dec (2003), pp. 1107–1149.

[185] Takaki Makino and Johane Takeuchi. "Apprenticeship learning for model parameters of partially observable environments". In: *arXiv preprint arXiv:1206.6484* (2012).

[186] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*. Vol. 135. MIT press Cambridge, 1998.

[187] Matthew Hausknecht and Peter Stone. "Deep recurrent q-learning for partially observable mdps". In: *2015 AAAI Fall Symposium Series*. 2015.

[188] Hang Yin, Patrıcia Alves-Oliveira, Francisco S Melo, Aude Billard, and Ana Paiva. "Synthesizing robotic handwriting motion by learning from human demonstrations". In: *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. CONF. 2016.

[189] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 7559–7566.

[190] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. "Model-based reinforcement learning for atari". In: *arXiv preprint arXiv:1903.00374* (2019).

[191] Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. "Woulda, coulda, shoulda: Counterfactually-guided policy search". In: *International Conference on Representation Learning (ICLR)* (2018).

[192] Paul E Marik. "Fever in the ICU". In: *Chest* 117.3 (2000), pp. 855–869.

[193] Lucy B Palmer, Gerald C Smaldone, John J Chen, Daniel Baram, Tao Duan, Melinda Monteforte, Marie Varela, Ann K Tempone, Thomas O'Riordan, Feroza Daroowalla, et al. "Aerosolized antibiotics and ventilator-associated tracheobronchitis in the intensive care unit". In: *Critical care medicine* 36.7 (2008).

[194] Jean-Ralph Zahar, Jean-Francois Timsit, Maıte Garrouste-Orgeas, Adrien Francais, Aurélien Vesim, Adrien Descorps-Declere, Yohann Dubois, Bertrand Souweine, Hakim Haouache, Dany Goldgran-Toledano, et al. "Outcomes in severe sepsis and patients with septic shock: pathogen species and infection sites are not associated with mortality". In: *Critical care medicine* 39.8 (2011), pp. 1886–1895.

[195] Remi Neviere, Polly E Parsons, and Geraldine Finlay. "Sepsis syndromes in adults: Epidemiology, definitions, clinical presentation, diagnosis, and prognosis". In: *Monografia en Internet]. Wolters Kluwer: UpToDate* (2017).

[196] Bibhas Chakraborty. *Statistical methods for dynamic treatment regimes*. Springer, 2013.

[197] Junzhe Zhang and Elias Bareinboim. "Near-optimal reinforcement learning in dynamic treatment regimes". In: *Advances in Neural Information Processing Systems*. 2019, pp. 13401–13411.

[198] Bilal Piot, Matthieu Geist, and Olivier Pietquin. "Bridging the gap between imitation learning and inverse reinforcement learning". In: *IEEE transactions on neural networks and learning systems* 28.8 (2016), pp. 1814–1826.

[199] Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. "Imitation learning via off-policy distribution matching". In: *arXiv preprint arXiv:1912.05032* (2019).

[200] Amy Zhang, Yuxin Wu, and Joelle Pineau. "Natural environment benchmarks for reinforcement learning". In: *arXiv preprint arXiv:1811.06032* (2018).

[201] Antonio Torralba and Alexei A Efros. "Unbiased look at dataset bias". In: *CVPR 2011*. IEEE. 2011, pp. 1521–1528.

[202] Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. "Invariant causal prediction for block mdps". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 11214–11224.

[203] Pim de Haan, Dinesh Jayaraman, and Sergey Levine. "Causal confusion in imitation learning". In: *arXiv preprint arXiv:1905.11979* (2019).

[204] Timothy E Lee, Jialiang Zhao, Amrita S Sawhney, Siddharth Girdhar, and Oliver Kroemer. "Causal Reasoning in Simulation for Structure and Transfer Learning of Robot Manipulation Policies". In: *arXiv preprint arXiv:2103.16772* (2021).

[205] Dean A Pomerleau. "Efficient training of artificial neural networks for autonomous navigation". In: *Neural computation* 3.1 (1991), pp. 88–97.

[206] Stéphane Ross and Drew Bagnell. "Efficient reductions for imitation learning". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 661–668.

[207] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. "A reduction of imitation learning and structured prediction to no-regret online learning". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 627–635.

[208] Francisco S Melo and Manuel Lopes. "Learning from demonstration using mdp induced metrics". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2010, pp. 385–401.

[209] Bilal Piot, Matthieu Geist, and Olivier Pietquin. "Boosted and reward-regularized classification for apprenticeship learning". In: *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. 2014, pp. 1249–1256.

[210] Jonathan Ho and Stefano Ermon. "Generative adversarial imitation learning". In: *arXiv preprint arXiv:1606.03476* (2016).

[211] Yiren Lu and Jonathan Tompson. "ADAIL: Adaptive Adversarial Imitation Learning". In: *arXiv preprint arXiv:2008.12647* (2020).

[212] Kuno Kim, Yihong Gu, Jiaming Song, Shengjia Zhao, and Stefano Ermon. "Domain adaptive imitation learning". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5286–5295.

[213] Jalal Etesami and Philipp Geiger. "Causal transfer for imitation learning and decision making under sensor-shift". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 2020, pp. 10118–10125.

[214] Pierre Sermanet, Corey Lynch, Jasmine Hsu, and Sergey Levine. "Time-contrastive networks: Self-supervised learning from multi-view observation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2017, pp. 486–487.

[215] YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. "Imitation from observation: Learning to imitate behaviors from raw video via context translation". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1118–1125.

[216] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1126–1135.

[217] Yan Duan, Marcin Andrychowicz, Bradly C Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. "One-shot imitation learning". In: *arXiv preprint arXiv:1703.07326* (2017).

[218] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. "One-shot visual imitation learning via meta-learning". In: *Conference on Robot Learning*. PMLR. 2017, pp. 357–368.

[219] Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. "One-shot imitation from observing humans via domain-adaptive meta-learning". In: *arXiv preprint arXiv:1802.01557* (2018).

[220] Stephen James, Michael Bloesch, and Andrew J Davison. "Task-embedded control networks for few-shot imitation learning". In: *Conference on Robot Learning*. PMLR. 2018, pp. 783–795.

[221] Pratyusha Sharma, Deepak Pathak, and Abhinav Gupta. "Third-person visual imitation learning via decoupled hierarchical controller". In: *arXiv preprint arXiv:1911.09676* (2019).

[222] Sergei Volodin, Nevan Wichers, and Jeremy Nixon. "Resolving spurious correlations in causal models of environments via interventions". In: *arXiv preprint arXiv:2002.05217* (2020).

[223] Anoopkumar Sonar, Vincent Pacelli, and Anirudha Majumdar. "Invariant policy optimization: Towards stronger generalization in reinforcement learning". In: *arXiv preprint arXiv:2006.01096* (2020).

[224] Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. "Invariant risk minimization games". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 145–155.

[225] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1861–1870.

[226] Nir Baram, Oron Anschel, and Shie Mannor. "Model-based adversarial imitation learning". In: *arXiv preprint arXiv:1612.02179* (2016).

[227] Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. "A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models". In: *arXiv preprint arXiv:1611.03852* (2016).

[228] Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. "A divergence minimization perspective on imitation learning methods". In: *Conference on Robot Learning*. PMLR. 2020, pp. 1259–1277.

[229] Michael Bain and Claude Sammut. "A Framework for Behavioural Cloning." In: *Machine Intelligence 15*. 1995, pp. 103–129.

[230] Umar Syed and Robert E Schapire. "A reduction from apprenticeship learning to classification". In: *Advances in neural information processing systems* 23 (2010), pp. 2253–2261.

[231] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. "Causal inference by using invariant prediction: identification and confidence intervals". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 78.5 (2016), pp. 947–1012.

[232] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. "Environment inference for invariant learning". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 2189–2200.

[233] Frederick Eberhardt and Richard Scheines. "Interventions and causal inference". In: *Philosophy of science* 74.5 (2007), pp. 981–995.

[234] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. "Adversarial discriminative domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7167–7176.

[235] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. "Mutual information neural estimation". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 531–540.

[236] Geoffrey E Hinton. "Training products of experts by minimizing contrastive divergence". In: *Neural computation* 14.8 (2002), pp. 1771–1800.

[237] Yilun Du and Igor Mordatch. "Implicit generation and modeling with energy based models". In: *Advances in Neural Information Processing Systems* 32 (2019).

[238] Siddharth Reddy, Anca D Dragan, and Sergey Levine. "Sqil: Imitation learning via reinforcement learning with sparse rewards". In: *arXiv preprint arXiv:1905.11108* (2019).

[239] Minghuan Liu, Tairan He, Minkai Xu, and Weinan Zhang. "Energy-based imitation learning". In: *arXiv preprint arXiv:2004.09395* (2020).

[240] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. "Openai gym". In: *arXiv preprint arXiv:1606.01540* (2016).

[241] Alborz Geramifard, Christoph Dann, Robert H Klein, William Dabney, and Jonathan P How. "RLPy: a value-function-based reinforcement learning framework for education and research." In: *J. Mach. Learn. Res.* 16.1 (2015), pp. 1573–1578.

[242] Andrew G Barto, Richard S Sutton, and Charles W Anderson. "Neuronlike adaptive elements that can solve difficult learning control problems". In: *IEEE transactions on systems, man, and cybernetics* 5 (1983), pp. 834–846.

[243] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. "The arcade learning environment: An evaluation platform for general agents". In: *Journal of Artificial Intelligence Research* 47 (2013), pp. 253–279.

[244] Antonin Raffin. *RL Baselines Zoo*. https://github.com/araffin/rl-baselines-zoo. 2018.

[245] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. *Stable Baselines*. https://github.com/hill-a/stable-baselines. 2018.

[246] Chloe E Taylor, Helen Jones, Mohammad Zaregarizi, Nigel T Cable, Keith P George, and Greg Atkinson. "Blood pressure status and post-exercise hypotension: an example of a spurious correlation in hypertension research?" In: *Journal of human hypertension* 24.9 (2010), pp. 585–592.

[247] Thomas Desautels, Ritankar Das, Jacob Calvert, Monica Trivedi, Charlotte Summers, David J Wales, and Ari Ercole. "Prediction of early unplanned intensive care unit readmission in a UK tertiary care hospital: a cross-sectional machine learning approach". In: *BMJ open* 7.9 (2017).

[248] Andrea Soo, Danny J Zuege, Gordon H Fick, Daniel J Niven, Luc R Berthiaume, Henry T Stelfox, and Christopher J Doig. "Describing organ dysfunction in the intensive care unit: a cohort study of 20,000 patients". In: *Critical Care* 23.1 (2019), pp. 1–15.

[249] Adarsh Subbaswamy, Roy Adams, and Suchi Saria. "Evaluating Model Robustness and Stability to Dataset Shift". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 2611–2619.

[250] Sebastiaan C Goulooze, Laura B Zwep, Julia E Vogt, Elke HJ Krekels, Thomas Hankemeier, John N van den Anker, and Catherijne AJ Knibbe. "Beyond the randomized clinical trial: innovative data science to close the pediatric evidence gap". In: *Clinical Pharmacology & Therapeutics* (2019).

[251] Hua Xu, Jianfu Li, Xiaoqian Jiang, and Qingxia Chen. "Electronic Health Records for Drug Repurposing: Current Status, Challenges, and Future Directions." In: *Clinical Pharmacology and Therapeutics* 107.4 (2020), pp. 712–714.

[252] Benjamin Ribba, Sherri Dudal, Thierry Lavé, and Richard W Peck. "Model-Informed Artificial Intelligence: Reinforcement Learning for Precision Dosing". In: *Clinical Pharmacology & Therapeutics* 107.4 (2020), pp. 853–857.

[253] Gilbert Koch, Marc Pfister, Imant Daunhawer, Melanie Wilbaux, Sven Wellmann, and Julia E Vogt. "Pharmacometrics and machine learning partner to advance clinical data analysis". In: *Clinical Pharmacology & Therapeutics* (2020).

[254] Hans-Georg Eichler, Franz Koenig, Peter Arlett, Harald Enzmann, Anthony Humphreys, Frank Pétavy, Brigitte Schwarzer-Daum, Bruno Sepodes, Spiros Vamvakas, and Guido Rasi. "Are novel, nonrandomized analytic methods fit for decision making? The need for prospective, controlled, and transparent validation". In: *Clinical Pharmacology & Therapeutics* 107.4 (2020), pp. 773–779.

[255] J Stoehlmacher, DJ Park, W Zhang, D Yang, S Groshen, S Zahedy, and HJ Lenz. "A Multivariate Analysis of Genomic Polymorphisms: Prediction of Clinical Outcome to 5-FU/Oxaliplatin Combination Chemotherapy in Refractory Colorectal Cancer". In: *British Journal of Cancer* 91.2 (2004), p. 344.

[256] Min Qian and Susan A Murphy. "Performance Guarantees for Individualized Treatment Rules". In: *Annals of Statistics* 39.2 (2011), p. 1180.

[257] Richard K Crump, V Joseph Hotz, Guido W Imbens, and Oscar A Mitnik. "Nonparametric Tests for Treatment Effect Heterogeneity". In: *The Review of Economics and Statistics* 90.3 (2008), pp. 389–405.

[258] Claudia Shi, David M Blei, and Victor Veitch. "Adapting Neural Networks for the Estimation of Treatment Effects". In: *arXiv preprint arXiv:1906.02120* (2019).

[259] Nathan Kallus. "Recursive Partitioning for Personalization using Observational Data". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1789–1798.

[260] Adith Swaminathan and Thorsten Joachims. "Batch learning from logged bandit feedback through counterfactual risk minimization." In: *Journal of Machine Learning Research* 16.1 (2015), pp. 1731–1755.

[261] Nathan Kallus and Angela Zhou. "Policy evaluation and optimization with continuous treatments". In: *arXiv preprint arXiv:1802.06037* (2018).

[262] Dimitris Bertsimas and Christopher McCord. "Optimization over continuous and multi-dimensional decisions with observational data". In: *Advances in Neural Information Processing Systems*. 2018, pp. 2962–2970.

[263] Victor Chernozhukov, Mert Demirer, Greg Lewis, and Vasilis Syrgkanis. "Semi-Parametric Efficient Policy Learning with Continuous Actions". In: *Advances in Neural Information Processing Systems*. 2019, pp. 15039–15049.

[264] Douglas Galagate. "Causal Inference with a Continuous Treatment and Outcome: Alternative Estimators for Parametric Dose-Response function with Applications." PhD thesis. 2016.

[265] James Bergstra and Yoshua Bengio. "Random search for hyper-parameter optimization". In: *Journal of machine learning research* 13.Feb (2012), pp. 281–305.

[266] Peter C Austin. "An introduction to propensity score methods for reducing the effects of confounding in observational studies". In: *Multivariate behavioral research* 46.3 (2011), pp. 399–424.

[267] Kosuke Imai and Marc Ratkovic. "Covariate balancing propensity score". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 76.1 (2014), pp. 243–263.

[268] Alberto Abadie and Guido W Imbens. "Matching on the estimated propensity score". In: *Econometrica* 84.2 (2016), pp. 781–807.

[269] Enrique F Schisterman, Stephen R Cole, and Robert W Platt. "Overadjustment bias and unnecessary adjustment in epidemiologic studies". In: *Epidemiology (Cambridge, Mass.)* 20.4 (2009), p. 488.

[270] Adith Swaminathan and Thorsten Joachims. "The self-normalized estimator for counterfactual learning". In: *Advances in Neural Information Processing Systems*. 2015, pp. 3231–3239.

[271] Onur Atan, William R Zame, and Mihaela van der Schaar. "Learning optimal policies from observational data". In: *International Conference on Machine Learning CausalML workshop* (2018).

[272] William Hoiles and Mihaela van der Schaar. "A non-parametric learning method for confidently estimating patient's clinical state and dynamics". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2020–2028.

[273] Cosmin Păduraru, Doina Precup, Joelle Pineau, and Gheorghe Comănici. "An empirical analysis of off-policy learning in discrete mdps". In: *European Workshop on Reinforcement Learning*. 2013, pp. 89–102.

[274] Shayan Doroudi, Philip S Thomas, and Emma Brunskill. "Importance Sampling for Fair Policy Selection." In: *Grantee Submission* (2017).

[275] Doina Precup. "Eligibility traces for off-policy policy evaluation". In: *Computer Science Department Faculty Publication Series* (2000), p. 80.

[276] Philip S Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. "High-confidence off-policy evaluation". In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.

[277] Zhaohan Guo, Philip S Thomas, and Emma Brunskill. "Using options and covariance testing for long horizon off-policy policy evaluation". In: *Advances in Neural Information Processing Systems*. 2017, pp. 2492–2501.

[278] Assaf Hallak, François Schnitzler, Timothy Mann, and Shie Mannor. "Off-policy model-based learning under unknown factored dynamics". In: *International Conference on Machine Learning*. 2015, pp. 711–719.

[279] Nan Jiang and Lihong Li. "Doubly robust off-policy value evaluation for reinforcement learning". In: *arXiv preprint arXiv:1511.03722* (2015).

[280] Judea Pearl et al. "Causal inference in statistics: An overview". In: *Statistics surveys* 3 (2009), pp. 96–146.

[281] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[282] Helmut Bartsch, Heike Dally, Odilia Popanda, Angela Risch, and Peter Schmezer. "Genetic risk profiles for cancer susceptibility and therapy response". In: *Cancer Prevention*. Springer, 2007, pp. 19–36.

[283] Chanelle J Howe, Stephen R Cole, Shruti H Mehta, and Gregory D Kirk. "Estimating the effects of multiple time-varying exposures using joint marginal structural models: alcohol consumption, injection drug use, and HIV acquisition". In: *Epidemiology (Cambridge, Mass.)* 23.4 (2012), p. 574.

[284] U Waheed, P Williams, S Brett, G Baldock, and N Soni. "White cell count and intensive care unit outcome". In: *Anaesthesia* 58.2 (2003), pp. 180–182.

[285] Liesbet De Bus, Bram Gadeyne, Johan Steen, Jerina Boelens, Geert Claeys, Dominique Benoit, Jan De Waele, Johan Decruyenaere, and Pieter Depuydt. "A complete and multifaceted overview of antibiotic use and infection diagnosis in the intensive care unit: results from a prospective four-year registration". In: *Critical Care* 22.1 (2018), p. 241.

[286] Muhammad Ali, Humaira Naureen, Muhammad Haseeb Tariq, Muhammad Junaid Farrukh, Abubakar Usman, Shahana Khattak, and Hina Ahsan. "Rational use of antibiotics in an intensive care unit: a retrospective study of the impact on clinical outcomes and mortality rate". In: *Infection and Drug Resistance* 12 (2019), p. 493.

[287] Olav Kallenberg. *Foundations of modern probability*. Springer Science & Business Media, 2006.

[288] Martın Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[289] Yixin Wang and David M Blei. "Multiple Causes: A Causal Graphical View". In: *arXiv preprint arXiv:1905.12793* (2019).

[290] Wang Miao, Zhi Geng, and Eric J Tchetgen Tchetgen. "Identifying causal effects with proxy variables of an unmeasured confounder". In: *Biometrika* 105.4 (2018), pp. 987–993.

[291] Dehan Kong, Shu Yang, and Linbo Wang. “Multi-cause causal inference with unmeasured confounding and binary outcome”. In: *arXiv preprint arXiv:1907.13323* (2019).

[292] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. “Maximum entropy inverse reinforcement learning.” In: *Aaai*. Vol. 8. Chicago, IL, USA. 2008, pp. 1433–1438.

[293] Deepak Ramachandran and Eyal Amir. “Bayesian Inverse Reinforcement Learning.” In: *IJCAI*. Vol. 7. 2007, pp. 2586–2591.

[294] Abdeslam Boularias, Jens Kober, and Jan Peters. “Relative entropy inverse reinforcement learning”. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 2011, pp. 182–189.

[295] Benjamin Burchfiel, Carlo Tomasi, and Ronald Parr. “Distance minimization for reward learning from scored trajectories”. In: *Thirtieth AAAI Conference on Artificial Intelligence*. 2016.

[296] Chelsea Finn, Sergey Levine, and Pieter Abbeel. “Guided cost learning: Deep inverse optimal control via policy optimization”. In: *International Conference on Machine Learning*. 2016, pp. 49–58.

[297] Justin Fu, Katie Luo, and Sergey Levine. “Learning robust rewards with adversarial inverse reinforcement learning”. In: *arXiv preprint arXiv:1710.11248* (2017).

[298] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. “Playing atari with deep reinforcement learning”. In: *arXiv preprint arXiv:1312.5602* (2013).

[299] Max Welling and Yee W Teh. “Bayesian learning via stochastic gradient Langevin dynamics”. In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. Citeseer. 2011, pp. 681–688.