Learning Optimal and Sample-Efficient Decision Policies with Guarantees

Daqian Shao
Wolfson College
University of Oxford

A thesis submitted for the degree of
DPhil in Computer Science
Trinity 2025

Acknowledgements

First and foremost, I would like to express my greatest gratitude to my supervisor, Prof. Marta Kwiatkowska. Without her, this thesis would not have been possible. I am truly grateful for her continuous care and advice throughout my DPhil, while being an extremely responsible supervisor and mentor.

Secondly, I would like to thank my colleagues and collaborators who I am fortunate enough to work with during my DPhil journey. To the colleagues in the QAVAS group, thank you for making my DPhil memorable and for sharing this DPhil journey. To my talented collaborators, I have learnt so much from each of you. Thank you for all the insightful meetings and intriguing discussions that shaped my path as a researcher.

I gratefully acknowledge that this work was supported by the EPSRC Prosperity Partnership FAIR (grant number EP/V056883/1). I acknowledge funding from the Turing Institute and Accenture collaboration. The topics studied in this thesis are aligned with my supervisor's projects FUN2MODEL (grant agreement No. 834115, funded by the ERC under the European Union's Horizon 2020 research and innovation programme) and ELSA: European Lighthouse on Secure and Safe AI project (grant agreement No. 101070617 under UK guarantee).

Now, to my parents Dr Yan Shao and Ms. Wenhong Tian. I am forever grateful for all your unwavering support throughout my long studying life. You inspire me to be passionate about the things I do and to be fearless in pursuing my passion. To my other family members, thank you for all the love and joy, encouraging me to carry on.

Finally, to Dr Sichen Liu. I cannot imagine this DPhil journey without you. Thank you for being an important part of my life through the ups and downs, taking care of me and supporting me. It is my privilege to share this journey with you.

Abstract

The paradigm of decision-making has been revolutionised by reinforcement learning and deep learning due to their ability to learn intelligent agents in unknown environments. Although this has led to significant progress in domains such as robotics, healthcare, and finance, the use of reinforcement learning in practice is challenging, particularly when learning decision policies in high-stakes applications that may require guarantees. Traditional reinforcement learning algorithms rely on online interactions with the environment to gather data for policy learning and are often data-inefficient. This is problematic in scenarios where online interactions are costly, dangerous, or infeasible. In this thesis, we work in the setting of offline reinforcement learning, which assumes access to fixed sets of pre-collected data, and develop algorithms for learning effective policies with provable guarantees. However, learning from offline datasets is hindered by the presence of hidden confounders, that is, variables that influence the actions and the outcome but are not observed in the offline dataset. Such confounders can cause spurious correlations in the dataset and can mislead the agent into taking suboptimal or adversarial actions. Firstly, we address the problem of learning from offline datasets in the presence of hidden confounders within the context of reinforcement learning while ensuring sample efficiency and optimality guarantees. To achieve this, we work with instrumental variables (IVs) to identify the causal effect, which we show is an instance of a conditional moment restrictions (CMR) problem. Inspired by double/debiased machine learning, we derive a novel Neumann

orthogonal score function and design a cross-fitting regime to obtain a sample-efficient algorithm for solving CMR problems with convergence and optimality guarantees, which outperforms state-of-the-art on IV regression and proximal causal learning problems. Secondly, we relax the conditions on the hidden confounders in the setting of (offline) imitation learning, and adapt our CMR estimator to derive an algorithm that can learn effective imitator policies with convergence rate guarantees, which outperforms the state of the art. Finally, we consider the problem of learning high-level objectives expressed in linear temporal logic (LTL) and develop a provably optimal learning algorithm that improves sample efficiency over existing methods. The derived methods have been implemented and are made available in open source. Through evaluation on reinforcement learning benchmarks and synthetic and semi-synthetic datasets, we demonstrate the usefulness of the methods developed in this thesis in real-world decision making.

Contents

1 Introduction **1**

1.1 Summary of Contributions 5

1.2 Thesis Outline 7

1.3 Publications and Statement of Authorship 7

2 Background **9**

2.1 Reinforcement Learning 11

2.1.1 Markov Decision Processes 12

2.1.2 Online Reinforcement Learning 13

2.1.3 Offline Reinforcement Learning 20

2.1.4 Imitation Learning 25

2.2 Hidden Confounders 27

2.2.1 Structural Causal Models 27

2.2.2 Conditional Moment Restrictions 29

2.2.3 Instrumental Variable Regression 31

2.2.4 Proximal Causal Learning 33

2.3 High-Level Objectives 35

2.3.1 Linear Temporal Logic 36

2.3.2 Other Temporal Logics 37

2.3.3 Reward Machines 39

3 Literature Review **40**

3.1 Reinforcement Learning 41

3.1.1 RL with Hidden Confounders 41

3.1.2 Imitation Learning 43

3.1.3 Learning High-Level Objectives with RL 45

3.1.4 Robust and Safe RL 47

3.1.5 Causal Reinforcement Learning 48

3.2 Hidden Confounders 51

3.2.1 Instrumental Variable Regression via Conditional Moment Restriction Estimation 51

3.2.2 Proximal Causal Learning 53

3.2.3 Double Machine Learning for Hidden Confounders 54

3.3 Summary 56

4 Learning Policies from Confounded Datasets with Instrumental Variables **59**

4.1 Introduction 60

4.1.1 Double Machine Learning 62

4.2 Problem Formulation 64

4.2.1 Contextual IV Setting 64

4.2.2 Offline IV Bandit 65

4.3 Solving Conditional Moment Restrictions with DML 66

4.3.1 Neyman Orthogonal Score 67

4.3.2 A DML Estimator for Solving CMR Problems — DML-CMR . . . 68

4.3.3 Theoretical Analysis 70

4.3.4 A Computationally Efficient Variant of DML-CMR 77

4.3.5 DML Identifiability Condition and Ill-posedness 78

4.4 Suboptimality for Offline IV Bandit 79

4.5 Experimental Results 80

4.5.1 IV Regression 81
4.5.2 Offline IV Bandit 84
4.5.3 Proximal Causal Learning 85
4.6 Discussion 88

5 Imitating Expert Policies from Confounded Datasets 90

5.1 Introduction 91
5.2 Causal Imitation Learning 93
5.3 A Unifying Framework for Causal Imitation Learning 94
5.3.1 MDPs with Hidden Confounders. 94
5.3.2 Causal Imitation Learning in Our Framework 95
5.3.3 Reducing Our Framework to Prior Work 97
5.4 Causal IL as a CMR Problem 100
5.4.1 Derivation of the CMR Problem 101
5.4.2 Practical Algorithms for Solving the CMR Problem 102
5.4.3 Theoretical Analysis 104
5.5 Experimental Results 108
5.5.1 Ticket Pricing Environment 109
5.5.2 Mujoco Environments 111
5.5.3 Adopting Other CMR Estimators 112
5.6 Discussion 113

6 Learning Policies for High-Level Objectives 115

6.1 Introduction 116
6.2 Problem Formulation 118
6.3 ω-Automata and Limit-Deterministic Büchi Automata 118
6.4 Methodology 121
6.4.1 Product MDP 122

6.4.2 LTL Learning with Q-Learning 128
6.4.3 Counterfactual Imagining 129
6.5 Experimental Results 131
6.5.1 Probabilistic Gate MDP 133
6.5.2 Frozen Lake MDP 133
6.5.3 Office World MDP 133
6.5.4 Runtime Analysis and Sensitivity Analysis 134
6.6 Discussion 135

7 Conclusion **137**
7.1 Discussion 138
7.1.1 Practicality in Real-World Applications 139
7.1.2 Strengths and Weaknesses of the Methods 140
7.2 Open Problems and Future Work 141

Bibliography **143**

A Learning Policies from Confounded Datasets with Instrumental Variables **180**
A.1 The Score Function for Standard Two-Stage CMR Estimators 180
A.2 Proofs 181
A.2.1 DML-CMR $N^{-1/2}$ Convergence Rate Guarantees 181
A.2.2 Ill-posedness and DML Identification 191
A.2.3 Constrained Empirical Risk Minimisation Bounds 193
A.2.4 Suboptimaltiy of the Decision Policy 201
A.3 Datasets Details 203
A.3.1 IV Regression 203
A.3.2 Proximal Causal Learning 206
A.4 Network Architecture and Hyperparameters 209
A.4.1 IV Regression 209

A.4.2 Proximal Causal Learning 211
A.4.3 Validation and Hyperparameter Tuning 214
A.5 Failure of Standard Offline Bandit Algorithms 214
A.6 Additional Experimental Results 216
A.6.1 Effects of Weak Instruments 216
A.6.2 Performance of DML-CMR with Tree-Based Estimators 217
A.6.3 Sensitivity Analysis for Different Hyperparameters 218

B Imitating Expert Policies from Confounded Datasets 220
B.1 Proofs of Main Results 220
B.1.1 Proof of Propositions 220
B.1.2 Guarantees on the Imitation Gap 221
B.1.3 Proofs of Corollaries 225
B.2 Environments and Tasks 228
B.2.1 Dynamic Aeroplane Ticket Pricing 228
B.2.2 Mujoco Environments 229
B.3 Implementation Details 231
B.3.1 Expert Training 231
B.3.2 Imitator Training 231
B.3.3 Imitator Evaluation 232

C Learning Policies for High-Level Objectives 233
C.1 Proofs of Main Results 233
C.1.1 Proof of Theorem 6.6 233
C.1.2 Proof of Lemma 6.9 234
C.1.3 Proof of Theorem 6.10 236
C.1.4 Proof of Theorem 6.11 241
C.2 Additional Experimental Details and Setup 242

C.2.1 Probabilistic Gate 243
C.2.2 Frozen Lake 244
C.2.3 Office World 244
C.2.4 PRISM 244
C.2.5 Rabinizer 249

List of Figures

2.1 An overview of the types of problems considered in policy learning. 10

2.2 The causal graph of outcome Y, treatment A and hidden confounder U. . . 29

2.3 The causal graph of outcome Y, treatment A, hidden confounder U and an instrumental variable Z. 31

2.4 The causal graph of outcome Y, treatment A, hidden confounder U and proxies V and W. 33

4.1 The causal graph of the contextual IV setting 64

4.2 Results on the ticket demand dataset with low-dimensional context. 82

4.3 Results on the ticket demand dataset with high-dimensional context. 83

4.4 The mean squared error of $\widehat{f}$ and average reward following $\widehat{\pi}$ for the real-world datasets. 84

4.5 The mean squared error of $\widehat{f}$ on the ticket demand dataset for the PCL task. 86

4.6 The mean squared error of $\widehat{f}$ on the dSprites dataset with high dimensional treatment for the PCL task. 87

5.1 A causal graph of MDPs with hidden confounders $u_t = (u_t^o, u_t^\epsilon)$. 95

5.2 Experimental results for DML-IL in the plane ticket environment. 108

5.3 Experimental results for DML-IL in the MuJoCo environment. 110

5.4 Confounded IL using other CMR estimators in the plane ticket environment. 113

5.5 Confounded IL using other CMR estimators in the MuJoCo environment. . 113

6.1 An LDBA for "FG a & G !c" 121
6.2 The experimental results on various MDPs and LTL tasks. 132
6.3 Sensitivity analysis on key hyperparameters. 134

A.1 A graph of the causal function in the ticket demand dataset 205
A.2 An example of dSprites image, which is used as the treatment A in PCL experiments. Its scale, rotation, x position and y position are randomly generated. 208
A.3 Average reward for offline bandit algorithms that do not take IVs into account 215

C.1 The optimal policy for the probabilistic gate MDP task 243
C.2 MDP environments used in the experiments. 243

List of Tables

A.1 Network architecture for DML-CMR and CE-DML-CMR for the ticket demand low-dimensional dataset for IV regression 210
A.2 Network architecture of the feature extractor used for the ticket demand dataset with MNIST for IV regression 211
A.3 Network architecture for DML-CMR and CE-DML-CMR for the ticket demand dataset with MNIST for IV regression 212
A.4 Network architecture for DML-CMR and CE-DML-CMR for the ticket demand dataset for PCL 213
A.5 Network architecture of the feature extractor used for the dSprites image dataset for PCL 214
A.6 Results for the low-dimensional ticket demand dataset when the IV is weakly correlated with the action. 217
A.7 Results for the low-dimensional ticket demand dataset using tree-based estimators compared to DNN estimators. 218
A.8 Results for the low-dimensional ticket demand dataset for a range of hyperparameter values 219
A.9 Results for the high-dimensional ticket demand dataset for a range of hyperparameter values 219

B.1 Network architecture for DML-IL 232

Chapter 1

Introduction

Reinforcement learning (RL) and deep learning (DL) have revolutionised decision-making by enabling intelligent agents to learn optimal policies in unknown environments. This is achieved through either online interactions with the environment or offline pre-collected data, without relying on predefined rules or human supervision. RL algorithms allow agents to discover the sequence of actions that maximise expected cumulative rewards, making them well-suited for complex, sequential decision-making problems. The integration of deep neural networks (DNNs) in deep RL further enhances this capability by allowing agents to learn directly from high-dimensional and unstructured data such as images or sensor readings, without manual feature engineering. This has led to breakthroughs in domains such as robotics, healthcare, and finance, where deep RL agents can make decisions based on raw state inputs and learn intricate strategies that surpass human-level performance. However, many fundamental challenges remain in using RL for real-world decision making, including learning from *offline datasets*, specifying *rewards* that align with the learning objective, and designing *sample-efficient* algorithms.

Traditional RL algorithms learn policies through trial and error, which often require a vast amount of data from online interactions with the environment to learn an optimal policy, making them data-inefficient compared to other machine learning techniques. Al-

though this problem is not severe for (video) games and simulated robotic environments, it is detrimental in many real-world applications where online interactions with the environment are costly, dangerous, or simply infeasible, such as autonomous driving in robotics, credit and loan applications in finance, and medical treatment in healthcare. In these scenarios, *sample-efficient* RL algorithms, which can learn decision policies with a relatively small number of training samples, are desirable. If interactions with the environment are infeasible, previously collected *offline datasets*, such as execution trajectories and logs, can be used to learn optimal policies. This learning paradigm is known as *offline RL*. Most of the offline RL literature focuses on tackling the *distribution shift* problem, where the learnt policy becomes unreliable if it leads the agent outside the state-action distribution of the offline dataset. We instead focus on a less studied but very important problem, prevalent in real-world applications, where the agent may be confounded by *hidden confounders*, which are variables that influence both the actions and the outcome (for example, the next state and reward), but are not observed in the offline dataset. Such hidden confounders cause spurious correlations between the action and the outcome, where the observed effect of an action on the outcome is not its true causal effect. These spurious correlations can mislead the agent to take suboptimal and even adversarial actions. For example, suppose we have aeroplane ticket sales and pricing data (Hartford et al., 2017), and we wish to learn a policy from this offline dataset that maximises revenue. During the holiday season, observational data may contain evidence of a concurrent surge in ticket sales and prices, which may cause the learning algorithm to learn an incorrect policy that higher ticket prices will always drive higher sales.

In the presence of hidden confounders, the ability to learn optimal policies heavily depends on structural assumptions regarding the environment and the offline dataset. To learn the optimal policy by identifying the true causal effects of actions in the environment, additional assumptions are necessary for the environment and the offline dataset (Shpitser & Pearl, 2008). One practical and popular assumption is the existence of *instrumental*

variables (IVs) (Wright, 1928), which are heterogeneous random variables that affect the action, but not the outcome. These IVs have been observed and used extensively in many fields, such as econometrics, drug testing, and social sciences. In the aeroplane ticket example, supply cost shifters (e.g., fuel price) are instrumental variables, as their variations are independent of the demand for aeroplane tickets and affect sales solely via ticket prices. More broadly, a range of causal inference techniques, such as IV regression and proximal causal learning (PCL), can be framed within the general problem of solving *conditional moment restrictions* (CMRs), and we derive algorithms that solve this problem with provable guarantees on sample efficiency and optimality.

However, despite the widespread use of causal inference methods such as IVs to identify the causal effect, these methods often require strong assumptions on the dataset, which are not guaranteed to be satisfied in real-world applications. Therefore, the causal effect of actions in these scenarios may not be identifiable. Fortunately, if the offline dataset is produced by an expert demonstrator, it might still be possible to imitate and match the performance of the expert by learning from their decision trajectories in the offline dataset. This problem is known as *imitation learning* (IL). Like offline RL, most of the IL literature focuses on addressing distribution shifts through interactive expert queries and inverse RL. However, the hidden confounder problem is also very prevalent in IL, which can causally confuse the imitator policy, leading to unreliable performance. We consider the general setting, where the environment has hidden variables that are not recorded in the offline dataset, whereas the expert, on the other hand, can access parts of the hidden variables due to imperfect environment logging and expert knowledge. In this setting, it is desirable to learn a history-dependent policy that captures information regarding the hidden variables from trajectory histories while mitigating the causal confusion caused by the confounding noise. We will show that this complex IL problem can be cast into a CMR problem, where we develop sample-efficient algorithms with suboptimality guarantees.

The RL algorithms introduced above maximise quantitative rewards. However, the de-

sired behaviour and objectives for the learning agent are often qualitative and high level. Therefore, it is desirable for a system, or for an agent within the system, to learn policies that satisfy these high-level specifications. Specifying appropriate reward functions that accurately capture the objectives in these scenarios is non-trivial and can lead to suboptimal and undesirable policies if not carefully designed. *Linear temporal logic* (LTL) is a temporal logic language that can encode formulae regarding properties of an infinite sequence of logic propositions. LTL is widely used for the formal specification of high-level objectives for robotics and multi-agent systems, and it is highly expressive and versatile. For example, infinite-horizon reach-avoidance objectives, sequences of conditional tasks, periodic tasks, and many other safety objectives can all be specified using LTL.

In order to learn an optimal policy that maximises the probability of satisfying LTL objectives using RL, most methods first transform the LTL objective into an automaton and build a product *Markov decision process* (MDP) using the original environment MDP and the automaton. A reward function needs to be constructed for this product MDP to determine when, where, and how much reward to give to the agent. Crucially, such a reward function should guide the agent towards the optimal satisfaction of LTL objectives with theoretical guarantees. However, existing works that provide such optimality guarantees are impractical for two reasons. Firstly, it is unclear how to choose the key hyperparameters that ensure the optimality of the algorithm in practice. Secondly, these algorithms are highly sample-inefficient, rendering them costly or intractable in practice. Therefore, it is important to develop algorithms for LTL learning that are sample-efficient, optimal, and practical.

In this thesis, we tackle the aforementioned problems encountered in RL- and DL-driven decision making in the real world. More specifically, we design practical algorithms that can learn optimal policies in the presence of hidden confounders under different levels of assumptions, and to learn optimal policies with respect to qualitative and high-level LTL objectives, all with optimality and sample efficiency guarantees.

1.1 Summary of Contributions

With the aim of this thesis outlined above, we provide a summary of the contributions made by each main chapter towards fulfilling this aim.

- In Chapter 4, we consider offline RL and study the problem of learning optimal policies in the presence of hidden confounders when IVs are observed in the dataset. The problem of identifying causal effect through IVs is also known as IV regression, which at its core is a *conditional moment restriction* (CMR) problem. We propose DML-CMR, a sample-efficient and novel estimator that solves general CMR problems and converges at $N^{-1/2}$ rate, where N is the sample size. This leads to $O(N^{-1/2})$ suboptimality for the learnt policy when applying DML-CMR to the offline IV bandit problem, which is a single-step RL problem within a confounded environment using IVs. DML-CMR leverages the *double/debiased machine learning* (DML) framework by deriving a novel, *Neyman orthogonal*, score function for CMR problems, and designing a cross-fitting regime for the DML-CMR estimator. We experimentally demonstrate that DML-CMR[1] outperforms other modern IV regression methods (Hartford et al., 2017; Bennett et al., 2019; Singh et al., 2019; Xu et al., 2020) on a range of IV regression and offline IV bandit benchmarks, including two real-world datasets (Hill, 2011; Wyatt et al., 2020). In addition, since DML-CMR can solve general CMR problems, we evaluate DML-CMR on the PCL task, demonstrating superior performance against modern PCL algorithms (Im et al., 2021; Mastouri et al., 2021; Xu et al., 2021; Kompa et al., 2022; Wu et al., 2024).

- In Chapter 5, we study the problem of learning decision policies from expert trajectories, which is known as *imitation learning* (IL), in the presence of hidden confounders. In this chapter, we relax the explicit assumption in Chapter 4 that additional IVs are observed in the dataset. We propose a novel and unifying framework for con-

[1]Link to codebase for Chapter 4: `https://github.com/shaodaqian/DML-CMR`

founded IL, where the expert can have access to parts of the hidden confounders that are not recorded in the dataset. We show that IL under this setting can be transformed into a CMR problem by leveraging trajectory histories as instruments and derive the DML-IL algorithm based on DML-CMR to learn a history-dependent policy. The learnt imitator policy can match the expert policy if the expert trajectory history contains sufficient information regarding the hidden confounder observable to the expert. We prove that the performance gap between our imitator and the expert is of the rate $O(\epsilon T^2)$, where ϵ is some small optimisation error of the algorithm and T is the horizon of the IL task. We experimentally demonstrate that DML-IL[2] can effectively learn good imitator policies in our confounded IL setting on a range of environments including MuJoCo (Todorov et al., 2012), and outperforms other confounded IL methods (Swamy et al., 2022a,b).

- In Chapter 6, we consider the problem of learning LTL objectives optimally and sample-efficiently using online RL. As explained above, existing methods typically define a product MDP with an automaton, where a reward function is constructed to learn the policy. We propose a novel RL algorithm that is guaranteed to converge to the optimal policy for satisfying LTL objectives utilising a novel product MDP design with a generalised reward structure that improves sample efficiency. We provide theoretical optimality guarantees and theoretical analysis for choosing the key hyperparameters to ensure the algorithm returns an optimal policy in practice. To further improve sample efficiency, we adopt counterfactual imagining, a method to exploit the known structure of the LTL objective by creating imagination experiences through counterfactual reasoning. We directly evaluate our algorithms[3] through a novel integration of probabilistic model checkers, which are formal verification tools that analyse stochastic systems to compute the probability of satisfying formal spec-

[2]Link to codebase for Chapter 5: `https://github.com/shaodaqian/Confounded-IL`
[3]Link to codebase for Chapter 6: `https://github.com/shaodaqian/rl-from-ltl`

ifications, within the evaluation pipeline. Our algorithms demonstrate better sample efficiency and training convergence against other RL algorithms that learn LTL objectives (Hahn et al., 2019; Bozkurt et al., 2019; Hasanbeig et al., 2020) on a range of grid-world environments (Brockman et al., 2016; Icarte et al., 2022).

1.2 Thesis Outline

We now outline the structure of this thesis. In Chapter 2, we introduce the preliminary background related to reinforcement learning, paradigms in policy learning, and hidden confounders. In Chapter 3, we provide a literature review, which includes an overview of RL, various aspects of RL relevant to this thesis, and causality frameworks. In Chapter 4, we propose a novel algorithm, DML-CMR, to solve CMR problems, which we use to learn policies from confounded datasets with IVs. In Chapter 5, we present a unifying framework for imitation learning from confounded datasets and transform this problem into a CMR problem, which can be solved by adopting the DML-CMR algorithm. In Chapter 6, we present a novel RL algorithm to learn policies that satisfy high-level LTL objectives with optimality guarantees. Finally, in Chapter 7, we provide a summary of the thesis findings and propose viable avenues for future research. Formal proofs of theorems and other technical details can be found in the Appendix.

1.3 Publications and Statement of Authorship

This thesis is based on several papers that were published (Shao & Kwiatkowska, 2023; Shao et al., 2024, 2025a) or under submission (Shao et al., 2025b) during my DPhil studies. In addition, I contributed to other works that are not included in this thesis (Shao et al., 2023). In this section, my contributions to the aforementioned papers in the context of this thesis are outlined.

Chapter 4 is primarily based on Shao et al. (2024) (published at ICML 2024) and a

journal extension of that paper (Shao et al., 2025b), which is currently under submission. In Shao et al. (2024), the DML-IV algorithm is proposed, the theoretical results are proven, and experiments are conducted for IV regression and policy learning. I formalised the problem, proposed the Neyman orthogonal score and the DML-IV algorithm, derived and proved the theoretical results, implemented the algorithm, and performed all experimental evaluations. In the journal extension, the algorithm and theory are generalised for solving CMR problems, where additional experiments on the PCL task are carried out. I generalised the Neyman orthogonal score and the algorithm to the CMR framework, extended and proved new theoretical results, and performed additional experimental evaluations.

Chapter 5 is primarily based on Shao et al. (2025a) (SCSL workshop at ICLR 2025 and under review at NeurIPS 2025), where a unifying framework for causal imitation learning is proposed, the corresponding CMR problem for causal IL is derived, theoretical results are proved, and experimental evaluations are performed. I proposed the unifying framework, derived the CMR problem, proved the theoretical results, designed and implemented the algorithm, and conducted the experiments.

Chapter 6 is primarily based on Shao & Kwiatkowska (2023) (published at IJCAI 2023), where the novel product MDP and the RL algorithm for LTL learning are proposed with theoretical guarantees and a practical implementation. I proposed the product MDP and the algorithm, implemented it, proved the theoretical results, and conducted experiments.

Chapter 2

Background

Contents

2.1 Reinforcement Learning 11

2.1.1 Markov Decision Processes 12

2.1.2 Online Reinforcement Learning 13

2.1.3 Offline Reinforcement Learning 20

2.1.4 Imitation Learning 25

2.2 Hidden Confounders 27

2.2.1 Structural Causal Models 27

2.2.2 Conditional Moment Restrictions 29

2.2.3 Instrumental Variable Regression 31

2.2.4 Proximal Causal Learning 33

2.3 High-Level Objectives 35

2.3.1 Linear Temporal Logic 36

2.3.2 Other Temporal Logics 37

2.3.3 Reward Machines 39

		Learning Objectives		
Learning Sources		**Expected Reward Maximisation**	**Imitate an Expert**	**High-Level Objectives**
	Online Interactions	(Standard) Online RL	Interactive Imitation Learning*	Online RL with High-Level Objectives **Chapter 6**
	Offline Dataset	(Standard) Offline RL	Behaviour Cloning (Imitation Learning)	Offline RL with High-Level Objectives†
	Confounded Offline Dataset	Confounded Offline RL **Chapter 4**	Confounded Imitation Learning **Chapter 5**	RL with High-Level Objectives from Confounded Datasets†

* Interactive imitation learning assumes access to an online expert that can interact with the environment, in addition to an offline expert dataset.
† Potential fields of research that are not yet established

Figure 2.1: An overview of the types of problems considered in policy learning, categorised by the learning objective and the data source from which the policy learns. Problems considered in this thesis are highlighted with the corresponding chapters.

In this chapter, we cover the essential technical background that underpins the work in this thesis. The chapter is divided into three sections. The first section introduces reinforcement learning, which builds the foundation for the entire thesis and formalises the concept of policy learning. The next two sections introduce concepts that motivate the concrete problems that we solve in the following chapters. Specifically, the second section covers hidden confounders in the context of structural causal models, with a particular focus on instrumental variables, which provide a way to handle hidden confounders when learning policies. The third section concerns high-level objectives as learning targets for RL.

To provide structure to the concepts introduced in this chapter and clarify the relationships between the subsequent chapters, we categorise the concrete policy learning problems in Figure 2.1 based on their learning objectives and data sources for learning. This maps the characteristics of practical problems encountered in policy learning to different clusters of solutions.

Notations

We write uppercase letters such as X to denote random variables and use the corresponding calligraphic letter such as $\mathcal{X}$ to denote the set from which the random variable takes its value. For example, $X \in \mathcal{X} \subseteq \mathbb{R}^d$ is a d-dimensional real-valued random variable in $(\mathcal{X}, \mathcal{B}_{\mathcal{X}})$, where $\mathcal{B}_{\mathcal{X}}$ is the Borel algebra on $\mathcal{X}$. We refer to the probability distribution of X given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ as the probability measure denoted as $F_X : \mathcal{B}_{\mathcal{X}} \to \mathbb{R}$. The support of X is denoted as $supp(X)$, and the set of all distributions supported on $\mathcal{X}$ is denoted as $D(\mathcal{X})$. An observed realisation of X is denoted by a lowercase letter x. We abbreviate $\mathbb{E}[Y|X = x]$, a realisation of the conditional expectation $\mathbb{E}[Y|X]$, as $\mathbb{E}[Y|x]$. $[N]$ denotes the set $\{1, ..., N\}$ for $N \in \mathbb{N}$. We write $\mathbb{E}[X|do(A = a)]$ for the expectation of X under *do* intervention (Pearl, 2000) of setting $A = a$. We use $\|\cdot\|_p$ to denote the functional norm, defined as $\|f\|_p := \mathbb{E}[|f(X)|^p]^{1/p}$, where the measure is implicit from the context. For a function f, we use f_0 to denote the true function and $\hat{f}$ an estimator of the true function. We use O and o to denote big-O and little-o notations (Weisstein, 2023), respectively.

2.1 Reinforcement Learning

In this section, we cover the basics of reinforcement learning (RL) (Sutton & Barto, 2018) and imitation learning (IL). We begin by introducing Markov decision processes, which formulate the potentially stochastic environment in which RL takes place. Next, we elucidate the different paradigms of RL relevant to this thesis, including online RL, offline RL, and IL. We also provide an overview of the key developments in reinforcement learning (RL) algorithms for solving online RL (Section 2.1.2.2) and offline RL (Section 2.1.3.1). While online RL and offline RL represent two problem formulations for RL in different settings, the algorithms designed to address these problems can generally be classified into two categories: model-

free and model-based. Model-free RL learns policies or value functions directly from experience, without an explicit model of the environment, and relies on trial and error. In contrast, model-based RL first learns a model or representation of the environment's dynamics and subsequently uses this model for policy learning. We will introduce and discuss both classes of algorithms for online RL and offline RL, respectively.

2.1.1 Markov Decision Processes

We first define Markov decision processes (MDPs), which will later be extended with atomic proposition labels and reward functions. The definitions presented draw from both formal verification and RL literature.

Definition 2.1 (Markov decision processes (Puterman, 1994))**.** An MDP $\mathcal{M}$ is a tuple $(\mathcal{S}, \mu_0, \mathcal{A}, \mathcal{P})$, where $\mathcal{S}$ is a set of states, $\mu_0 \in D(\mathcal{S})$ is the initial state distribution, $\mathcal{A}$ is a set of actions, and $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the probabilistic transition function. Let $A(s)$ denote the set of available actions in state s. Then, for all $s \in \mathcal{S}$, it holds that $\mathcal{P}(s, a, \cdot)$ is a valid probability distribution if $a \in \mathcal{A}(s)$ and $\mathcal{P}(s, a, \cdot) = 0$ otherwise.

An infinite path is a sequence of states $\sigma = s_0, s_1, s_2...$, where for all $i \geq 0$, there exists $a_{i+1} \in \mathcal{A}(s_i)$ such that $\mathcal{P}(s_i, a_{i+1}, s_{i+1}) > 0$, and a finite path is a finite such sequence. We denote the set of infinite and finite paths of the MDP $\mathcal{M}$ as $Paths^{\mathcal{M}}$ and $FPaths^{\mathcal{M}}$, respectively. We use $\sigma[i]$ to denote s_i, and $\sigma[: i]$ and $\sigma[i :]$ to denote the prefix and suffix of the path, respectively. Furthermore, we assume self-loops: if $A(s) = \varnothing$ for some state s, we let $\mathcal{P}(s, a, s) = 1$ for some $a \in \mathcal{A}$ and $A(s) = a$ such that all finite paths can be extended to an infinite one.

A finite-memory policy π for $\mathcal{M}$ is a function $\pi : FPaths^{\mathcal{M}} \to D(\mathcal{A})$ such that $supp(\pi(\sigma)) \subseteq \mathcal{A}(\sigma[\text{-}1])$, where $\sigma[\text{-}1]$ is the last state of a finite path σ. A policy π is memoryless if it only depends on the current state, that is, $\sigma[\text{-}1] = \sigma'[\text{-}1]$ implies $\pi(\sigma) =$

$\pi(\sigma')$, and a policy is deterministic if $\pi(\sigma)$ is a point distribution for all $\sigma \in FPaths^{\mathcal{M}}$. For a memoryless policy, we let $\pi(s)$ represent $\pi(\sigma)$ where σ[-1] $= s$.

Let $Paths_{\pi}^{\mathcal{M}} \subseteq Paths^{\mathcal{M}}$ denote the subset of infinite paths that follow policy π, and we define the probability space $(Paths_{\pi}^{\mathcal{M}}, \mathcal{F}_{Paths_{\pi}^{\mathcal{M}}}, \mathbb{P}_{\pi})$ on $Paths_{\pi}^{\mathcal{M}}$. Then, for any function $f : Paths_{\pi}^{\mathcal{M}} \to \mathbb{R}$, let $\mathbb{E}_{\pi}[f]$ be the expectation of f over the infinite paths of $\mathcal{M}$ following π.

2.1.1.1 Markov Decision Processes with Atomic Proposition Labels

MDP states can also be labelled with atomic propositions, such that we can specify logic properties and specifications regarding the paths generated by the MDP.

Definition 2.2 (MDP with atomic proposition labels)**.** A Markov decision process (MDP) with atomic proposition labels is a tuple $(\mathcal{S}, \mu_0, \mathcal{A}, \mathcal{P}, \mathcal{AP}, L)$ that extends a standard MDP Definition 2.1 with $\mathcal{AP}$, the set of atomic propositions, and $L : \mathcal{S} \to 2^{\mathcal{AP}}$, the proposition labelling function that maps each state to a subset of atomic propositions.

A Markov chain (MC) induced by MDP $\mathcal{M}$ with atomic proposition labels and deterministic memoryless policy π is a tuple $\mathcal{M}_{\pi} = (\mathcal{S}, \mu_0, \mathcal{P}_{\pi}, \mathcal{AP}, L)$, where $\mathcal{P}_{\pi}(s, s') = \mathcal{P}(s, \pi(s), s')$. A bottom (sink) strongly connected component (BSCC) of a Markov chain is a set of states $C \subseteq \mathcal{S}$ such that, for all pairs $s_1, s_2 \in C$, there exists a path from s_1 to s_2 following the transition function $\mathcal{P}_{\pi}$ (strongly connected), and there is no state $s' \in S \setminus C$ such that $\mathcal{P}_{\pi}(s, s') > 0$ for all $s \in C$ (sink).

2.1.2 Online Reinforcement Learning

Online reinforcement learning (Sutton & Barto, 2018) teaches an agent to select an action from its action space in an unknown environment, in order to maximise rewards over time. The RL agent interacts with the unknown environment and learns from the rewards it receives in the environment. The environment is typically modelled as an MDP

$\mathcal{M} = (\mathcal{S}, \mu_0, \mathcal{A}, \mathcal{P})$, for which a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ and a discount factor function $\gamma : \mathcal{S} \to (0, 1]$ are defined. It is important to note that, in the context of RL, MDPs are sometimes defined with r and γ included in the tuple for simplicity. Given a memoryless policy π, at each time step t, let the agent's current state be s_t. Then, the action a following the distribution $\pi(s_t)$ is chosen, and the next state $s_{t+1} \sim \mathcal{P}(s_t, a, \cdot)$ together with the immediate reward $r(s_t, a, s_{t+1})$ is received from the environment. Then, starting at $s \in \mathcal{S}$, the expected discounted reward following policy π is

$$G^{\pi}(s) = \mathbb{E}_{\pi}[\sum_{i=0}^{\infty}(\prod_{j=0}^{i-1}\gamma(s_j)) \cdot r(s_i, a_i, s_{i+1}) \mid s_0 = s] \tag{2.1}$$

where $\prod_{j=t}^{t-1} := 1$. The agent's goal is to learn the optimal policy π^* that maximises the expected discounted reward G over the initial state distribution. This quantity is sometimes defined as the value of the policy π, defined formally as

$$J(\pi) = \mathbb{E}_{\pi,\mu_0}[\sum_{i=0}^{\infty}(\prod_{j=0}^{i-1}\gamma(s_j)) \cdot r(s_i, a_i, s_{i+1})]. \tag{2.2}$$

In the various reinforcement learning methods, for which we provide a comprehensive survey in Section 3.1, the state-action value function, or the Q function (Watkins & Dayan, 1992), is widely used. The Q function $Q_{\pi}(s, a)$ is defined as the expected discounted reward of taking action a in state s and following policy π thereafter:

$$Q_{\pi}(s, a) = \mathbb{E}_{\pi}[\sum_{i=0}^{\infty}(\prod_{j=0}^{i-1}\gamma(s_j)) \cdot r(s_i, a_i, s_{i+1}) \mid s_0 = s, a_0 = a]. \tag{2.3}$$

For example, in Q-learning (Watkins & Dayan, 1992), which is a classic RL algorithm for MDPs with discrete state and action spaces, the optimal policy π^* is derived from the optimal Q function $Q^* := Q_{\pi^*}$ by selecting the action a with the highest state-action pair value in each state s. The Q-learning algorithm learns the optimal Q function through the agent's experiences within the environment by adopting an exploration policy. Typically,

the ϵ-greedy policy, which selects the action with the highest Q value $\arg\max_{a \in \mathcal{A}} Q(s, a)$ with probability $1 - \epsilon$ and selects a random action with probability ϵ, is used. At each iteration, the agent's experiences, which include the next state s' and immediate reward $r(s, a, s')$, are used to update the Q function:

$$Q(s, a) \overset{\alpha}{\leftarrow} r(s, a, s') + \gamma(s) \max_{a' \in \mathcal{A}} Q(s', a') \tag{2.4}$$

where α is the learning rate and $x \overset{\alpha}{\leftarrow} y$ represents $x \leftarrow x + \alpha(y - x)$. It is proven that Q-learning converges to the optimal Q function Q^* in the limit, provided that each state-action pair is visited infinitely often (Watkins & Dayan, 1992), thereby learning the optimal policy.

2.1.2.1 Finite-horizon Reinforcement Learning

In certain areas, specifically for imitation learning, RL is sometimes defined with a finite time horizon T and without a discount factor γ. In this context, the environment is also modelled as an MDP $\mathcal{M} = (\mathcal{S}, \mu_0, \mathcal{A}, \mathcal{P})$, where a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ and a maximum time horizon T are given. Because of the finite time horizon, the sum of the total rewards of a trajectory is finite, so there is no need for a discount factor. With the maximum time horizon T, the expected total reward that the agent optimises is

$$G^{\pi}(s) = \mathbb{E}_{\pi}[\sum_{i=0}^{T} r(s_i, a_i, s_{i+1}) \mid s_0 = s], \tag{2.5}$$

and the Q-function is defined as

$$Q_{\pi}(s_t, a_t) = \mathbb{E}_{\pi}[\sum_{i=t}^{T} r(s_i, a_i, s_{i+1})]. \tag{2.6}$$

The finite-horizon formulation is sometimes preferred to the infinite-horizon discounted reward formulation if the RL task is an episodic task with a known endpoint or fixed dura-

tion. Moreover, finite-horizon formulation avoids discounting bias, which is the systematic tendency to prioritise short-term rewards over long-term rewards due to the exponential discounting factor in the discounted reward formulation.

2.1.2.2 Online Model-free RL Algorithms

Online reinforcement learning (Sutton & Barto, 2018) teaches an agent in an unknown environment to select an action from its action space to maximise rewards over time. The online RL agent interacts with the unknown environment and learns from the rewards it receives within that environment. With the development of deep neural networks (LeCun et al., 2015), these powerful function approximators have been incorporated into RL, and together with effective training algorithms have enabled RL to achieve exceptional results in numerous applications, including playing Atari games (Mnih et al., 2013; Schrittwieser et al., 2020), the game of Go (Silver et al., 2017), robotic control and manipulation (Levine et al., 2015; Kalashnikov et al., 2018), and resource allocation in business processes (Huang et al., 2011), among others.

We first discuss the various model-free algorithms used for online RL. The first and most direct approach to optimise the RL objective, which is the maximisation of expected rewards, is by employing policy gradient methods (Sutton et al., 1999; Kakade, 2001). These methods directly estimate the gradient of the expected reward with respect to the model parameters and perform gradient descent (Robbins & Monro, 1951). Schulman et al. (2015) then proposed a robust policy gradient variant that guarantees monotonic improvement of the policy, while Thomas (2014) modified the classic policy gradient algorithm to work in the average reward setting. These methods are applicable for general RL problems and can handle continuous state and action spaces, but generally suffer from high variance in gradient estimates, leading to noisy and unstable convergence with poor sample efficiency.

The second class of algorithms involves approximate dynamic programming (Powell, 2007), which estimates the expected reward from a state or a state-action pair following a policy. These expected rewards are also referred to as value or Q functions, and with accurate value functions one can recover a near-optimal policy. In fact, the value functions must satisfy the Bellman equation (Bellman, 1957) for dynamic programming, and Watkins & Dayan (1992) proposed Q-learning to find the Q value functions by minimising the error between both sides of the Bellman equation. Lagoudakis & Parr (2003) used temporal-difference learning to improve the performance of the Q function approximation and proved convergence of the algorithm to the optimal policy. A group of algorithms known as fitted Q-iteration (Ernst et al., 2005; Riedmiller, 2005) fully minimises the difference between both sides of the Bellman equation at each iteration. Lin (1992), instead of performing a single update of the Q function at each iteration, employed a replay buffer to store past agent experiences. This is also known as off-policy RL, as the replay buffer includes data collected from all past policies, which differs from the current policy being updated. To handle high-dimensional and continuous state spaces, Mnih et al. (2013) proposed Deep Q-Network (DQN), which uses deep neural networks (DNNs) and gradient descent to estimate the Q function by minimising the Bellman error from a replay buffer. For improved training stability, many approaches used target networks for the Q function (Hausknecht & Stone, 2015; Hasselt et al., 2018; Wang et al., 2015), which are less frequently updated, to calculate the Bellman error.

The third class of algorithms, actor-critic algorithms (Konda & Tsitsiklis, 2000), combines the basic ideas of the above two approaches by approximating the Q function to provide a better estimate of the policy gradient for policy optimisation. The algorithm Deterministic Policy Gradient (DPG) (Silver et al., 2014) is an off-policy actor-critic algorithm that learns a deterministic target policy from Q function estimations, which are learnt by minimising the Bellman error. It was later extended to use DNNs (Lillicrap et al., 2016) and to a distributed framework (Barth-Maron et al., 2018). Advantage actor-critic (A2C) (Mnih

et al., 2016) proposed to use advantage values, which measure the improvement by taking a particular action compared to the expected state value, for more stable policy gradient updates. Soft actor-critic (Haarnoja et al., 2018) extended this approach for continuous action tasks by also maximising the entropy of the policy. Recent advances include a better theoretical sample complexity for global convergence of actor-critic algorithms (Kumar et al., 2024), and a value-improved actor-critic algorithm (Oren et al., 2024).

Lastly, we discuss methods that constrain the policy update process, which improves the stability and efficiency of RL training. Trust Region Policy Optimisation (TRPO) (Schulman et al., 2015) is a policy optimisation method that ensures each policy update does not change the policy too drastically, thereby preventing performance collapse. To overcome the computational challenges of TRPO, Proximal Policy Optimization (PPO) (Schulman et al., 2017) proposes to use a clipped surrogate objective, which directly restricts how much the new policy can deviate from the old one. More recent improvements of PPO and TRPO include extensions to multi-agent RL (Yu et al., 2021a; Liu & Liu, 2024), extensions to model-based RL (Janner et al., 2019), and safe RL (Yao et al., 2023; Milosevic et al., 2024).

2.1.2.3 Online Model-based RL Algorithms

The above methods find the optimal policy without learning the explicit transition function or dynamic model of the environment, where the agents interact with the environment and select actions without knowing the next state after the transition. We refer to this group of algorithms as model-free reinforcement learning. Naturally, there exists another class of RL algorithms that uses explicit estimates of the dynamic model for policy learning, namely model-based reinforcement learning. Tassa et al. (2012) proposed a method to learn only the dynamic model of the environment for planning at test time, using trajectory optimisation. Improvements on this method have also been developed by adopting neural networks as dynamic models (Nagabandi et al., 2017). Chua et al. (2018) further improved

the performance of model-based RL using a method called probabilistic ensembles with trajectory sampling, while an end-to-end extension (Amos et al., 2018) to continuous state and action spaces via a differentiable policy has been proposed. Hafner et al. (2018) developed the PlaNet algorithm that accurately learns a latent dynamic model for image tasks and uses fast online planning algorithms in the latent space.

Another group of online model-based RL methods employs the dynamic model to generate synthetic data, also called model rollouts, which augment the procedure of policy optimisation. The classic Dyna algorithm (Sutton, 1991) combines model learning and data generation with Q-learning for policy learning. Many model-free RL algorithms incorporate this data generation functionality to improve sample complexity and accelerate learning (Gu et al., 2016; Kurutach et al., 2018; Clavera et al., 2018). Janner et al. (2019) later formalised a model-based RL approach with a monotonic improvement guarantee that uses short model-generated rollouts to avoid the model generalisation error for long-term rollouts. Hafner et al. (2020) proposed DreamerV2, that learns the policy purely from synthetic data generated from a learnt world model with a discrete latent space. Schiewer et al. (2024) studied hierarchical world models that are abstractions of the environment model across hierarchical levels. Theoretical works also exist for model-based RL, with the most recent progress by Li et al. (2023) who proposed a conservative model-based algorithm that achieved minimax-optimal sample complexity for infinite-horizon MDPs.

Last but not least, the learnt dynamic model can also improve the value estimates used as targets (Buckman et al., 2018) in temporal difference learning and as the critic (Feinberg et al., 2018) in actor-critic algorithms. This is achieved by estimating the uncertainty through model rollouts while measuring the impact of an imperfect dynamics model, ensuring that the performance of the learning algorithm is not degraded.

2.1.3 Offline Reinforcement Learning

Although online RL has demonstrated outstanding performance across various fields (Silver et al., 2017; Levine et al., 2015; Kalashnikov et al., 2018), the intrinsic online nature of RL algorithms remains a significant barrier to their wider adoption. Online RL involves repeatedly interacting with the environment using the current policy to collect new experiences, which are then used to improve the policy. However, in many cases, online interaction is impractical due to the high cost of data collection (e.g., robotics, educational agents, healthcare) or safety risks (e.g., autonomous driving, healthcare). Even when online interaction is feasible, relying on previously collected data can be preferable, particularly in complex domains where effective generalisation necessitates large datasets. These challenges motivate the development of offline RL.

The offline RL problem can be considered as a *data-driven* formulation of the RL problem. The goal remains to learn the optimal policy π^* that maximises the expected discounted reward G, but the agent no longer has the ability to interact with the environment and collect additional transitions. An offline RL algorithm, instead, is given a static, or offline, dataset $\mathcal{D} = \{(s_t^i, a_t^i, s_{t+1}^i, r_t^i)\}$ of transitions from different trajectories, where i is the trajectory index and t represents the time-step. All trajectories are generated following the transition function $\mathcal{P}$ and a data collection policy, which we denote as the behaviour policy π_β. An offline RL algorithm is required to learn a policy π solely from this fixed offline dataset $\mathcal{D}$ that achieves the highest expected reward when evaluated online in the environment.

The terms off-policy RL, batch RL, and offline RL can all describe algorithms that learn from a pre-collected dataset rather than interacting with the environment in real time. Off-policy RL allows learning from data collected by any policy, but it often involves additional online data collection. Batch RL is sometimes used interchangeably with offline RL, but this can lead to confusion, as it may refer to iterative batch learning that still gathers new data. To avoid ambiguity, offline RL is used in this thesis to describe learning exclusively

from a fixed offline dataset, without any further data collection.

2.1.3.1 Offline Model-free RL Algorithms

The classic paradigm of online RL fundamentally involves online learning, where the agent iteratively interacts with the environment to collect experience and uses that experience to improve its policy. However, such interactions may be costly and dangerous in many settings, and online RL algorithms can suffer from exponential sample complexity (Kakade, 2003). This problem has driven research in various directions, including building high-fidelity simulators to learn policies for tasks like autonomous driving and robotics control, and then transferring these policies to the real world (Sadeghi & Levine, 2016; Chebotar et al., 2018). On the other hand, offline RL is a direct approach that provides a paradigm to use previously collected offline data (Fu et al., 2020; Kumar et al., 2019), without any online interaction with the environment. This enables the exploitation of modern scalable machine learning algorithms that can potentially achieve better performance in the RL setting if more data is used for training. Examples of successful offline RL applications include robotic manipulation (Kalashnikov et al., 2018), robotic navigation skills (Kahn et al., 2020), and human dialogue generation (Jaques et al., 2019), among others.

However, as discussed in Chapter 1, offline RL presents major challenges including *distribution shift* (Levine et al., 2020) and *hidden confounders*. Since the hidden confounder problem is fundamentally a causal inference problem, it is often not considered in conjunction with offline RL in the literature. In fact, most literature considers the offline RL problem exclusively as a distribution shift problem. Therefore, in this section, we mainly introduce offline RL algorithms that address the distribution shift problem. Later, we dedicate a separate section in Section 3.1.1 to the literature on the intersection between RL and hidden confounders.

Before introducing algorithms specifically designed for offline reinforcement learning

(RL), it is important to note that any off-policy RL algorithm, such as Q-learning (Watkins & Dayan, 1992), allows the replay buffer to differ from the distribution used for evaluation and, in principle, can be applied directly to the offline setting. However, off-policy RL (Harutyunyan et al., 2016) typically assumes that new in-distribution experience can be added to the dataset through online interactions with the environment. This naturally limits (Munos et al., 2016) the degree of distribution shift in the dataset. However, most of these methods are not nearly as effective in the offline setting.

Most offline RL algorithms involve approximate dynamic methods with various techniques to limit the distribution shift. The first group of works focuses on imposing constraints on the policy to limit the degree of distribution shift from the behaviour policy. Jaques et al. (2019) proposed to explicitly add the Kullback–Leibler (KL) divergence between the behaviour policy and the current policy as a penalty to the Q values, and Wu et al. (2019) proposed Behaviour Regularised actor-critic (BRAC), where both the policy improvement step and the policy evaluation step can be penalised. However, these explicit f-divergence methods (Nowozin et al., 2016) (a superset of KL-divergence) require explicit estimation of the behaviour policy, which is difficult to estimate accurately, especially for multimodal distributions. Another line of implicit f-divergence constraint methods, such as AWR (Peng et al., 2019), AWAC (Nair et al., 2020) and ABM (Siegel et al., 2020), finds a nonparametric representation of the optimal policy updates for f-divergence constraints. Integral probability metrics (Sriperumbudur et al., 2009) are also used for policy constraints, with BEAR (Kumar et al., 2019) adopting the maximum mean discrepancy distance to represent the policy constraint and Wu et al. (2019) proposing a first-order Wasserstein distance policy constraint.

The second group of works focuses on finding a conservative estimate of the actual Q function to mitigate the effect of out-of-distribution actions. The early work of Jaksch et al. (2010) maintains an explicit confidence set of upper and lower confidence bounds to capture uncertainty. Osband et al. (2016) used bootstrap ensembles of Q functions to

represent samples from the distribution of Q functions to measure the uncertainty, while O'Donoghue et al. (2017) parameterised the distribution of Q functions using a Gaussian distribution with learnt parameters. However, high precision in uncertainty estimation is required, and the uncertainties must faithfully capture the reliability of the Q function. To avoid the difficulty in computing accurate uncertainties of the Q function, Kumar et al. (2020) proposed regularising the value function directly by penalising high Q values, particularly for out-of-distribution actions, to achieve conservative Q functions.

The third group of offline RL algorithms uses importance sampling (Kloek & van Dijk, 1978) to perform offline policy evaluation (OPE) and policy optimisation. Precup et al. (2000) first proposed the use of per-action importance sampling to estimate the expected reward. However, the variance of this estimator is too high to be effective, leading to the proposal of a doubly robust estimator (Thomas & Brunskill, 2016) that uses Q-value approximations alongside methods that optimise the trade-off between bias and variance (Wang et al., 2016) to reduce variance. Thomas et al. (2015) presented a method that guarantees the performance of the learnt policy with high probability for safety-critical applications. Alternative approaches (Precup et al., 2000, 2001; Peshkin & Shelton, 2002) have been proposed to directly estimate the policy gradient using similar per-action importance sampling techniques. To address the high variance of the policy gradient estimates, various approaches have been proposed including doubly robust estimators for the policy gradient (Gu et al., 2017; Huang & Jiang, 2019; Cheng et al., 2019), regularisation of the shift (Levine & Koltun, 2013; Schulman et al., 2017), estimating the state-marginal importance ratio (Sutton et al., 2015; Hallak & Mannor, 2017; Gelada & Bellemare, 2019; Liu et al., 2019), and using the backward Bellman equation (Lee & He, 2019; Nachum et al., 2019; Nachum & Dai, 2020).

Other works in offline RL include the use of one-step methods (Brandfonbrener et al., 2021; Kostrikov et al., 2022) that perform a single policy update step from the behaviour policy for out-of-distribution actions, and imitation learning (see Section 3.1.2 for more

details) that mimics the behaviour policy by finding and exploiting the well-performing parts of the behaviour policy.

2.1.3.2 Offline Model-based RL Algorithms

As discussed above, there exists an active field of research on model-based RL that uses learnt predictive transition models. These models can also be powerful in the offline RL setting, as they can leverage supervised learning to learn the dynamic model from large datasets.

In fact, online model-based RL already suffers from the distribution shift induced by model errors, and previous work in the online setting addresses the distribution shift problem with uncertainty estimation. In the offline setting, some online model-based RL methods are applied directly to robotics problems (Kahn et al., 2018, 2020), yet they provide few theoretical guarantees of performance. Berkenkamp et al. (2017) proposed a method to identify safe regions in the state and action space and imposed constraints to ensure Lyapunov stability. Based on model-based policy optimisation methods that use uncertainty estimates (Janner et al., 2019), two concurrent approaches, MOReL (Kidambi et al., 2020) and MOPO (Yu et al., 2020), have been proposed to produce conservative value estimates by modifying the learnt MDP model with reward penalties based on epistemic uncertainty estimates. In both cases, it is shown that the policy value estimates under the modified MDP provide a lower bound on the policy's true value. Yu et al. (2021b) proposed COMBO that extends the work on offline RL conservative Q functions (Kumar et al., 2020) to the model-based setting that provides a guaranteed bound on the true policy value with model-based policy value and model errors.

A different line of work, called sequential trajectory optimisation, is proposed to learn a model of the trajectory distribution induced by the behaviour policy. The model can then produce trajectories that we can use to plan an optimal set of actions from the initial state. Janner et al. (2021) proposed a method called the Trajectory Transformer that used Transformer models (Vaswani et al., 2017) to learn the trajectory distribution, and finally

adopted beam search for planning. Concurrently, Chen et al. (2021a) proposed Decision Transformer, which used a similar formulation of trajectory optimisation as Janner et al. (2021), but during training, minimised trajectory error only for actions. These approaches leverage large supervised learning models to exploit large offline datasets, but they are costly to train.

2.1.4 Imitation Learning

Imitation learning (IL) offers a powerful alternative to RL that simplifies the challenge of maximising rewards and iteratively optimising policies. Instead of designing complex reward functions and relying on extensive trial-and-error interactions with the environment, IL learns directly from expert demonstrations. This is particularly useful in scenarios where human experts can easily demonstrate the desired behaviour but cannot encode it programmatically. IL is well suited for tasks that require human-like behaviour, making it an attractive approach in robotics, autonomous driving, and other domains involving complex decision-making.

The main purpose of imitation learning (IL) is to enable agents to learn a policy that performs a specific task or behaviour by imitating an expert through the provision of expert demonstrations in a static dataset. The dataset $\mathcal{D} = \{(s_t^i, a_t^i)\}$ typically consists of state and action pairs, but does not include the reward. These demonstrations are used to train agents to perform a task by learning a mapping between states and expert actions. We will introduce three broad categories of IL approaches: behavioural cloning (BC), interactive IL, and inverse reinforcement learning (IRL).

2.1.4.1 Behavioural Cloning

Behavioural cloning (BC) is an IL technique that treats the problem of learning expert behaviour as a supervised learning task. BC algorithms use the expert demonstration dataset $\mathcal{D}$ of state and action pairs to train a policy $\pi(s)$ that mimics an expert's behaviour by

learning the mapping between the state and action. The supervised learning target of BC is $\mathbb{E}[s_t|a_t]$, which is the expectation of the states conditional on the actions. This objective is typically minimised using log-likelihood or least squares methods.

The main advantage of BC is its simplicity. BC does not require knowledge of the environment dynamics or the reward function, as it relies solely on expert demonstrations to learn the required behaviour. Moreover, BC is computationally efficient, as it involves supervised learning, a problem that is extensively researched in machine learning.

2.1.4.2 Other Imitation Learning Paradigms

Existing BC approaches, however, suffer from drawbacks. The main drawback is the covariate shift problem (Pomerleau, 1988), which occurs when the learnt policy encounters states that are not well represented in the training dataset at test time, leading to high test-time errors. Furthermore, unlike RL, BC does not explicitly correct errors once the learnt policy deviates from the expert, and this lack of feedback correction results in compounding errors.

A popular area of research that mitigates the covariate shift problem is interactive IL. This class of algorithms operates on the premise that the agent can access an online expert throughout the training process. The first interactive imitation learning method, Dataset aggregation (DAgger) (Ross et al., 2011), addresses the discrepancy between training and testing environments by instructing the agent on its self-generated state distribution. Subsequent interactive IL algorithms extend DAgger in various ways (Kelly et al., 2018; Mandlekar et al., 2020; Hoque et al., 2021), but the main concept of using an interactive agent remains unchanged.

Another related research domain is inverse reinforcement learning (IRL). IRL infers the reward function that underlies the observed demonstrations, which are assumed to come from an expert who acts optimally (Piot et al., 2017). The IRL agent then uses the inferred reward function to optimise an imitator policy iteratively through standard RL. It is

noteworthy that IRL, unlike BC, assumes that the agent can interact with the environment online for the final reinforcement learning step after a reward function has been learned.

In Chapter 5, we consider the problem of IL from offline confounded expert demonstrations, without online access to the environment or an interactive expert. Among the IL paradigms introduced, our setting is most similar to BC.

2.2 Hidden Confounders

In this section, we introduce the problem of hidden confounders within the context of *structural causal models* (SCMs) and discuss instrumental variables, which we will use to address the hidden confounder issue in the remainder of this thesis.

Hidden confounders (Pearl, 2000) are unobserved variables that influence multiple observable variables simultaneously, confounding them and introducing spurious correlations between them. In causal inference literature, it is often assumed that the hidden confounder affects two specific variables: the *action* (or *intervention*) and the *outcome*, where the goal is to understand the causal relationships between them. To properly account for the hidden confounder and understand the true causal effect of actions, we need to model the causal (or structural) relationship between the action and the outcome, which is expressed through a *causal function*. However, learning the causal function in the presence of hidden confounders is known to be challenging and sometimes infeasible (Shpitser & Pearl, 2008). To formalise the concept of hidden confounders and provide a framework for specifying the underlying causal mechanisms in a data-generating process, we introduce structural causal models.

2.2.1 Structural Causal Models

Definition 2.3 (Structural Causal Model)**.** An SCM M is a tuple $(U, V, F, P(U))$, where U is a set of exogenous (i.e., outside the model) random variables, which are typically

unobserved; V is a set of observed endogenous (i.e., inside the model) variables; $F = \{f_i\}$ is a set of deterministic functions such that, for each $V_i \in V$, $f_i(PA_i, U_i) = V_i$, where PA_i denotes the endogenous parents of V_i; and $P(U)$ is the joint distribution of exogenous variables.

In this definition, endogenous variables are variables for which we would like to study the causal effects (e.g., price and revenue). Exogenous variables serve as external sources of noise (e.g., seasonality) that can confound the causal relationships among endogenous variables. To better illustrate the relationships between variables in an SCM, a graphical abstraction, known as a causal graph, can be used to represent the causal relationships between variables within an SCM. A causal graph is a directed acyclic graph, where the nodes represent variables in the SCM. A solid directed edge indicates a direct causation between two endogenous variables: $V_i \rightarrow V_j$; and a dashed directed edge implies direct causation between an exogenous variable and an endogenous variable: $U_j \rightarrow V_j$.

We now introduce the concept of causal interventions, which are tools that allow us to study causal effects between variables. Interventions are defined through a mathematical operator called $do(x)$ (Pearl, 2000). An intervention, denoted by $do(X = x)$, simulates a physical intervention by removing the natural dependencies of X on its parent variables in the SCM and forcing it to take a specific value x, while keeping the rest of the model unchanged. The resulting causal model after the intervention is denoted M_x. The post-intervention distribution resulting from $do(X = x)$ is given by the equation:

$$\mathbb{P}_M(y|do(x)) = \mathbb{P}_{M_x}(y), \tag{2.7}$$

where the post-intervention distribution of a variable Y is defined as the distribution of Y in the intervened model M_x. Causal interventions allow researchers to analyse the causal effect of X on other variables while eliminating confounding factors.

For example, in a causal model where A (treatment) affects Y (outcome), an obser-

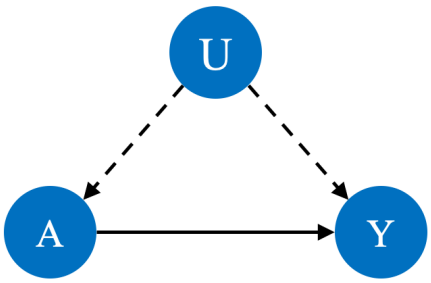


Figure 2.2: The causal graph of outcome Y, treatment A and hidden confounder U.

vational study may show correlation, but an intervention $do(A = a)$ would simulate a randomised experiment, ensuring that changes in Y are caused by A and not by other confounding factors.

In the SCM formulation, an exogenous random variable U is considered to be a *hidden confounder* if it affects two or more endogenous variables, such as V_i and V_j, and is unobserved. Consider an SCM that specifies two endogenous variables: the outcome $Y \in \mathcal{Y}$ and the treatment $A \in \mathcal{A}$,

$$Y = f(A, U), \tag{2.8}$$

where $U \in \mathcal{U}$ is a hidden confounder that affects both A and Y, as illustrated in the causal graph depicted in Figure 2.2. Due to the presence of this hidden confounder, with only observational data, standard regression methods (e.g., ordinary least squares) generally fail to produce consistent estimates of the causal relationship between A and Y (Pearl, 2000), that is, $\mathbb{E}[Y \mid do(A)]$. Therefore, the ability to identify the causal relationship between A and Y requires additional assumptions. Next, we introduce techniques that allow for the identification of $\mathbb{E}[Y \mid do(A)]$ under the assumption of observing additional variables.

2.2.2 Conditional Moment Restrictions

To begin with, we introduce the problem of *conditional moment restrictions* (CMRs). CMRs describe a family of problems in causal inference, including *instrumental variable*

(IV) regression and *proximal causal learning* (PCL), that aim to identify the causal effect $\mathbb{E}[Y \mid do(A)]$ in the presence of hidden confounders.

Definition 2.4 (Conditional Moment Restriction)**.** Let $X \in \mathcal{X} \subseteq \mathbb{R}^d$, $C \in \mathcal{C} \subseteq \mathbb{R}^p$ and $Y \in \mathcal{Y} \subseteq \mathbb{R}$ be random variables with their corresponding distributions. A CMR problem aims to estimate a function of interest $f \in \mathcal{F} \subseteq (\mathcal{X} \to \mathbb{R})$ in a hypothesis function space $\mathcal{F}$, where the true function f_0 satisfies the following condition:

$$\mathbb{E}[Y - f_0(X)|C = c] = 0, \text{ for all } c \in \mathcal{C}.$$

For simplicity, we denote the CMR problem as the following equation:

$$\mathbb{E}[Y - f(X)|C] = 0. \tag{2.9}$$

CMR problems are central to many problems in various fields, including statistical learning (Vapnik, 1998), integral equations (Honerkamp & Weese, 1990), deconvolution (Carrasco et al., 2007) and econometrics (Carrasco et al., 2007) (see Carrasco et al. 2007, Section 1.3 for more examples).

However, solving CMRs analytically is ill-posed (Nashed & Wahba, 1974) because it is an inverse problem for definite integrals that requires the derivation of a function inside conditional expectation. Many CMR estimators (e.g., Angrist et al. 1996; Newey & Powell 2003; Hartford et al. 2017; Singh et al. 2019) estimate f_0 with $\widehat{f}$ in some function space $\mathcal{F}$ by solving the following objective function using a two-stage approach:

$$\hat{f} \in \arg\min_{f \in \mathcal{F}} \mathbb{E}[(Y - \mathbb{E}[f(X)|C])^2]. \tag{2.10}$$

Specifically, the method of Newey & Powell (2003) employs a minimax approach that indirectly optimises this objective by solving a minimax unconditional moment problem in two stages. Angrist et al. (1996); Singh et al. (2019); Hartford et al. (2017), on the other

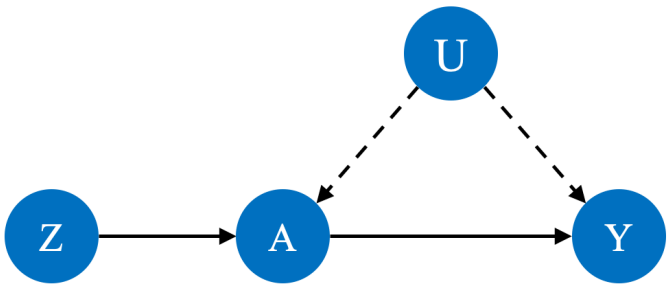


Figure 2.3: The causal graph of outcome Y, treatment A, hidden confounder U and an instrumental variable Z.

hand, directly optimise the above objective in two stages. The first stage involves learning the conditional expectation $\mathbb{E}[f(X)|c]$ through density estimation or kernel methods from observations. In the second stage, the objective in Equation (2.10) is minimised using the estimates obtained in the first stage. For both stages, linear regression, sieve methods, and deep neural networks (DNNs) are used for estimation, respectively, for each work. In Chapter 4, we point out that two-stage estimators often suffer from slow convergence with a lack of guarantees, and we propose a novel two-stage CMR estimator that mitigates this problem with fast convergence rate guarantees. Next, we introduce two specific instances of CMR problems in causal inference that enable causal identification under hidden confounders.

2.2.3 Instrumental Variable Regression

Instrumental variable (IV) regression (Newey & Powell, 2003) is a CMR problem that relies on the observation of additional IVs to identify the causal effect $\mathbb{E}[Y \mid do(A)]$. To begin with, we introduce the concept of IVs.

Definition 2.5 (Instrumental Variable)**.** Under the SCM of outcome $Y \in \mathcal{Y}$, treatment $A \in \mathcal{A}$ and hidden confounder $U \in \mathcal{U}$ defined in Equation (2.8), an instrumental variable $Z \in \mathcal{Z}$ is an observable variable that satisfies the following conditions (Newey & Powell, 2003):

- *Unconfounded Instrument*: $Z \perp\!\!\!\perp U$, that is, Z is independent of U;

- *Relevance*: $\mathbb{P}(A|Z)$ is not constant in Z;
- *Exclusion*: Z does not directly affect Y: $Z \perp\!\!\!\perp Y \mid (A, U)$,

where a causal graph with an instrumental variable Z is depicted in Figure 2.3.

Intuitively, IVs are heterogeneous random variables that only directly affect the action, but not the outcome. IVs have been used extensively to identify the causal effect of actions in many applications, including econometrics (Reiersöl, 1945; Angrist & Pischke, 2009), drug testing (Angrist et al., 1996), and social sciences (Angrist, 1990).

However, in order to identify the causal effect $\mathbb{E}[Y \mid do(A)]$, an additional assumption of *additive noise* is required, where we assume that

$$Y = f(A) + \epsilon(U) \text{ with } \mathbb{E}[\epsilon(U)] = 0. \tag{2.11}$$

The additive noise assumption, in conjunction with the IV conditions, is standard for the IV settings (Newey & Powell, 2003; Xu et al., 2020; Shao et al., 2024) and ensures the minimal condition to identify the causal effect. Another observation we can make here is that, following Equation (2.11),

$$\mathbb{E}[Y \mid do(A)] = \mathbb{E}[f(A) \mid do(A) + \mathbb{E}[\epsilon(U) \mid do(A)] \tag{2.12}$$

$$= f(A) + \mathbb{E}[\epsilon(U)] = f(A), \tag{2.13}$$

which means the task of identifying the *causal effect* $\mathbb{E}[Y \mid do(A)]$ is equivalent to learning the *causal function* $f(A)$.

In the IV setting, since the hidden confounder U affects both A and Y, it is generally the case that $\mathbb{E}[\varepsilon(U) \mid A] \neq 0$. This makes standard regression methods, such as ordinary least squares, fail to estimate the correct causal effect. However, there exist many IV regression methods that use IV to estimate the causal function $f(A)$. In order to identify $f(A)$, a key observation (Newey & Powell, 2003) is that, by taking the expectation on both sides

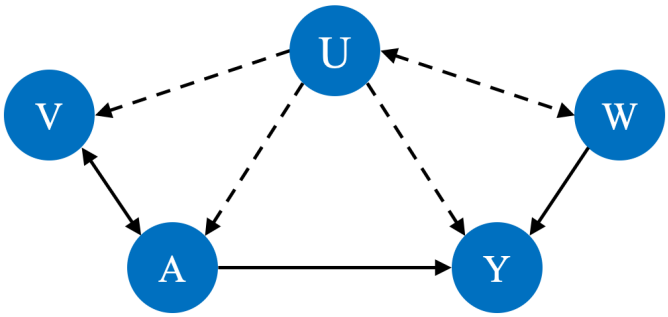


Figure 2.4: The causal graph of outcome Y, treatment A, hidden confounder U and proxies V and W.

of Equation (2.11) conditional on Z, we have

$$\begin{aligned}\mathbb{E}[Y|Z] &= \mathbb{E}[f(A) + \epsilon(U)|Z] \\ &= \mathbb{E}[f(A)|Z] + \mathbb{E}[\epsilon(U)] \\ &= \mathbb{E}[f(A)|Z] = \int f(A)\mathbb{P}(A|Z)dA, \end{aligned} \tag{2.14}$$

where the expectation $\mathbb{E}[Y|Z]$ and the distribution $\mathbb{P}(A|Z)$ are both observable. Therefore, it becomes possible to solve for f that satisfies the above equality. From Equation (2.14), we can rearrange and conclude that IV regression can be reduced to the following CMRs:

$$\mathbb{E}[Y - f(A)|Z] = 0, \tag{2.15}$$

where the variables X and C in the general CMR formulation (2.9) are renamed to A and Z in IV regression.

2.2.4 Proximal Causal Learning

Proximal causal learning (PCL) is also a CMR problem that allows for the identification of the causal effect between variables under hidden confounders. Following the SCM of outcome $Y \in \mathcal{Y}$, treatment $A \in \mathcal{A}$ and hidden confounder $U \in \mathcal{U}$ defined in Equation (2.8), PCL uses two proxy variables to identify the causal effect of the treatment A on the outcome

Y, i.e., $\mathbb{E}[Y \mid do(A)]$. The first proxy $V \in \mathcal{V}$ is a treatment-inducing proxy, and the second proxy $W \in \mathcal{W}$ is an outcome-inducing proxy. For V and W to be valid proxies, they must satisfy the following *conditional independence conditions*:

- $Y \perp\!\!\!\perp V \mid (A, U)$;
- $W \perp\!\!\!\perp (A, V) \mid U$,

where a causal graph with proxies V and W is depicted in Figure 2.4.

In addition, for the identifiability of the causal effect, proxies should satisfy *completeness assumptions*. Let $l : \mathcal{U} \to \mathbb{R}$ be any square-integrable function, that is, $\|l\|_2 \leq \infty$. The following conditions hold for any $a \in \mathcal{A}$:

$$\mathbb{E}[l(U)|A = a, W = w] = 0 \quad \forall w \in \mathcal{W} \iff l(u) = 0 \textit{ almost everywhere} \text{ on } \mathbb{P}(U);$$
$$\mathbb{E}[l(U)|A = a, V = v] = 0 \quad \forall v \in \mathcal{V} \iff l(u) = 0 \textit{ almost everywhere} \text{ on } \mathbb{P}(U).$$

It has been shown that, when the conditional independence conditions and the completeness assumptions are satisfied, it is possible to identify $\mathbb{E}[Y|do(A)]$.

Proposition 2.6 (Miao et al. (2018))**.** *Let the conditional independence and completeness assumptions hold, then there exists at least one solution to the following equality,*

$$\mathbb{E}[Y|A, V] = \mathbb{E}[h(A, W)|A, V] \tag{2.16}$$
$$= \int h(A, W)\mathbb{P}(A, W|A, V)dW, \tag{2.17}$$

for all $(A, V) \in \mathcal{A} \times \mathcal{V}$*. Let* h^* *be a solution of Equation* (2.16)*, then the causal effect* $\mathbb{E}[Y \mid do(A)]$ *can be estimated by* $\mathbb{E}_W[h(A, W)]$.

From this proposition, we can see that the problem of estimating the causal effect $\mathbb{E}[Y \mid do(A)]$ can be reduced to estimating h^*, which we denote as the bridge function following Miao et al. (2018). Therefore, estimating the causal effect in the PCL setting can

be reduced to the following CMR problem:

$$\mathbb{E}[Y - h(A, W)|A, V] = 0. \tag{2.18}$$

In Chapter 4, we additionally evaluate our proposed CMR estimator on the PCL task, where state-of-the-art performance is demonstrated.

Remark 2.7. For both IV and PCL, it is possible to include additional observed confounders $\mathbf{C}$ that affect both the treatment and the outcome as additional information or context. The resulting CMRs would be $\mathbb{E}[Y - f(A, \mathbf{C})|Z, \mathbf{C}] = 0$ for IV regression and $\mathbb{E}[Y - h(A, W, \mathbf{C})|A, V, \mathbf{C}] = 0$ for PCL. Note that, for IV regression, we can estimate $f(A, \mathbf{C}) = \mathbb{E}[Y|do(A), \mathbf{C}]$, which is the conditional causal effect. This can be used to derive a decision policy that depends on the context $\mathbf{C}$ to maximise the expected outcome, as we do in Chapter 4. However, for PCL, we can only estimate the expected causal effect $\mathbb{E}[Y|do(A)] = \mathbb{E}[h(A, W, X)]$, which implies that a context-dependent policy cannot be derived.

2.3 High-Level Objectives

In this section, we introduce the various ways to specify high-level objectives as targets for RL. While the target for standard RL is to maximise the expected reward with respect to some reward function r, specifying the RL target in terms of reward functions for objectives such as patrolling between points while avoiding obstacles can be challenging, both in terms of human interpretability and the risk of the reward mismatching the true objective. Therefore, we consider specifying RL objectives directly using high-level objectives, specifically in terms of logic languages, since they are concise and intuitive to humans. In addition, we discuss automata-based objectives, such as reward machines (Icarte et al., 2018), which are widely used in the literature to specify high-level objectives.

2.3.1 Linear Temporal Logic

A very commonly used and expressive logic is linear temporal logic (LTL). LTL provides a high-level description for the specifications of a system. In LTL, "temporal" means that the logic includes temporal operators expressing the order of events and can encode formulae regarding the future behaviour of a system. "Linear" means that it considers a linear, or a single, sequence of time points. LTL is very expressive and can describe specifications with an infinite horizon.

Definition 2.8 ((Baier & Katoen, 2008))**.** An LTL formula over atomic propositions $\mathcal{AP}$ is defined iteratively by the grammar:

$$\varphi ::= true \mid p \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \mathsf{X}\,\varphi \mid \varphi_1 \mathsf{U}\,\varphi_2, \quad p \in \mathcal{AP},$$

where X represents *next* and U represents *until*. Other Boolean and temporal operators are derived as follows: or: $\varphi_1 \vee \varphi_2 = \neg(\neg\varphi_1 \wedge \neg\varphi_2)$; implies: $\varphi_1 \rightarrow \varphi_2 = \neg\varphi_1 \vee \varphi_2$; *eventually*: $\mathsf{F}\,\varphi = true\mathsf{U}\,\varphi$; and *always*: $\mathsf{G}\,\varphi = \neg(\mathsf{F}\,\neg\varphi)$.

From the definition of an LTL formula, a language of infinite sequences of atomic propositions can be generated iteratively. In the context of RL with MDP and atomic proposition labels $(\mathcal{S}, \mu_0, \mathcal{A}, \mathcal{P}, \mathcal{AP}, L)$, the satisfaction of an LTL formula φ by an infinite MDP path $\sigma \in Paths^{\mathcal{M}}$, denoted by $\sigma \models \varphi$, is defined by induction on the structure of φ:

$$\begin{aligned} \sigma \text{ satisfies } \varphi \text{ if} \quad p \in L(\sigma[0]) &\quad \text{for } p \in \mathcal{AP}; \\ \sigma[1:] \models \varphi &\quad \text{for } \mathsf{X}\,\varphi; \\ \exists i, \sigma[i] \models \varphi_2 \text{ and } \forall j < i, \sigma[j] \models \varphi_1 &\quad \text{for } \varphi_1 \mathsf{U}\,\varphi_2, \end{aligned}$$

with the satisfaction of Boolean operators $\wedge$ and $\neg$ defined by their default meaning. Note that we will use $\neg$ and $\wedge$ as well as ! and & interchangeably in this thesis. To better explain the inductive satisfaction of LTL formulae, we provide some examples of LTL formulae

and the infinite paths that satisfy them.

Example 2.9. Examples of LTL formulae and the infinite paths satisfying them in increasing complexity. For $a, b, c \in \mathcal{AP}$:

1. LTL formula "a": any infinite path where the first element is labelled a by L.

2. LTL formula "X a": any infinite path where the second element is labelled a.

3. LTL formula "!a U b": any infinite path where there exist an element labelled b with all previous elements not labelled a.

4. LTL formula "G a": any infinite path where no element is not labelled a, meaning all elements in the path must be labelled a.

5. LTL formula "FG a": any infinite path where, after some finite index i, all elements are labelled a.

6. LTL formula "FG a & G !c": any infinite path where no element is labelled c, and, after some finite index i, all elements are labelled a.

2.3.2 Other Temporal Logics

Beyond LTL, there also exist other temporal logics that can specify conditions regarding the future of a system. We are specifically considering temporal logics because we would like to focus on objectives that reason about systems in terms of time. Before we justify the choice of LTL, we briefly overview other temporal logics.

Computation tree logic (CTL) is a branching-time temporal logic, meaning that time is viewed as a tree of possible futures. It allows for reasoning about different possible execution paths of a system and includes two path quantifiers: A means all possible future paths, and E means there exists at least one possible future path. By extending CTL with temporal operators such as X and U as defined in LTL, we have the full computation tree

logic (CTL*), which is a more expressive logic that generalises both LTL and CTL. This extension allows for the combination of path quantifiers with LTL-style operators in a more flexible way.

Interval temporal logic (ITL) reasons about entire intervals of time and includes a duration operator D that specifies time constraints on intervals. Signal temporal logic (STL) incorporates time-constrained temporal operators, which allow STL to specify inequalities over real-valued functions of time that can evolve continuously or discretely (these functions are called signals). These temporal logics are suitable for specifying properties for systems where time duration matters.

Finally, we mention additional variants of the previously introduced temporal logics. While LTL assumes infinite traces, which means that temporal operators such as *always* G extend indefinitely, there are also finite variants of LTL. LTLf is a variant of LTL that only considers finite traces, while truncated LTL considers LTL with a finite horizon. There are also probabilistic versions of temporal logic, such as PLTL and PCTL, which extend temporal logic by incorporating probability. This allows for specifications such as "The system eventually reaches a failure state with probability at most 0.1".

In this thesis, we use LTL to specify high-level objectives for RL for the following reasons. Firstly, under the discounted reward setting in RL, each environment run produces a single infinite-length trajectory, which is well suited for linear-time logics with an infinite horizon. Secondly, the goal of RL is to maximise an objective, and with LTL objectives, the natural goal would be to maximise the probability of satisfying LTL objectives. Therefore, probabilistic LTL's ability to specify objectives with a specific probability is less useful and introduces additional complexity when providing optimality guarantees. Last but not least, LTL can be translated into ω-automata (finite state automata on infinite sequences), which we will introduce later in Section 6.3. This property enables RL with temporal logic objectives and is essential for our algorithm proposed in Chapter 6.

2.3.3 Reward Machines

Another popular approach to specify high-level objectives is through reward machines (Icarte et al., 2018), which are finite state automata that define how rewards are assigned in a RL environment based on the trajectory history of the agent. Instead of hardcoding the reward function into the RL environment, a reward machine defines its transitions based on the atomic propositions associated with the environment states (recall Section 2.1.1.1), while its states specify the corresponding rewards.

Comparing to LTL, which provides declarative high-level objectives specifying what should happen, a reward machine provides a procedural representation of the high-level objective that defines how the agent should be rewarded. Functionally, it plays a similar role to an ω-automaton, which also provides a procedural specification of the objective using finite state automata. However, reward machines are less expressive than ω-automata and LTL because they cannot express specifications over an infinite sequence. In fact, reward machines are equivalent to regular expressions in terms of the expressivity of objectives. In contrast, LTL not only allows for more expressive objectives, but also provides a more concise and intuitive declarative objective for RL.

Chapter 3

Literature Review

Contents

3.1 Reinforcement Learning 41

3.1.1 RL with Hidden Confounders 41

3.1.2 Imitation Learning 43

3.1.3 Learning High-Level Objectives with RL 45

3.1.4 Robust and Safe RL 47

3.1.5 Causal Reinforcement Learning 48

3.2 Hidden Confounders 51

3.2.1 Instrumental Variable Regression via Conditional Moment Restriction Estimation 51

3.2.2 Proximal Causal Learning 53

3.2.3 Double Machine Learning for Hidden Confounders 54

3.3 Summary 56

In this chapter, we provide a comprehensive review of the literature on reinforcement learning, hidden confounders, and the connection between the two, in order to provide further context for the research questions that we address in the upcoming chapters.

3.1 Reinforcement Learning

In this section, we discuss the literature on RL that is closely related to this thesis. Section 3.1.1 discusses RL with hidden confounders, which is related to Chapter 4 and Chapter 5. Section 3.1.2 discusses imitation learning with and without hidden confounders, which is related to Chapter 5. Section 3.1.3 discusses the problem of learning high-level objectives with RL, which is related to Chapter 6. In Section 3.1.4, we provide a brief review of the literature on robust and safe RL, which, although not directly related to the individual chapters, is generally pertinent to the overarching theme of this thesis.

3.1.1 RL with Hidden Confounders

As discussed in Section 2.2, since the hidden confounder problem is fundamentally a causal inference problem, most of the RL literature does not consider hidden confounders in the environment. For the works that lie at the intersection of hidden confounders and RL, the first group of papers learns a policy from scratch using the least number of online interactions, with the help of a static offline dataset (Zhang & Bareinboim, 2020; Subramanian & Ravindran, 2022; Lu et al., 2020b; Wang et al., 2020a; Gasse et al., 2023). The second group focuses on learning policy purely from a confounded offline dataset, which is more relevant to this thesis and will be our primary focus.

For confounded offline RL, most work performs causal inference (e.g., instrumental variable regression (Murphy & Topel, 1985)) to mitigate the confounding effect and maximise the RL objective. Liao et al. (2024) considered a confounded Markov decision process (MDP), where the next state, current state, and action are confounded, with IVs observed at each time step. They applied IV regression (specifically Dual IV (Muandet et al., 2020), which is discussed in Section 3.2.1) to learn the transition dynamics of the MDP, and then they used the classical value iteration algorithm on the learnt transition dynamics to obtain a policy. Li et al. (2021) also considered RL with a confounded dataset, where

only the reward is confounded with the state-action pair, and IVs are observed at each time step. They argued that the bias due to this confounding effect (named R-bias, which stands for reinforcement-bias) is amplified during the interaction of the agent with the environment. They proposed to mitigate this bias through IV regression, namely the classic 2SLS algorithm (Angrist, 1990) to learn the reward function before applying actor-critic methods to learn a policy. Xu et al. (2023) and Fu et al. (2022) considered the off-policy policy evaluation (OPE) problem with hidden confounders, where the current state, reward, action, and next state are all confounded, with IVs observed at each time step. They both derived doubly robust estimators for the policy values when instrumental variables exist. Xu et al. (2023) did this by designing efficient influence functions as estimator adjustments, and Fu et al. (2022) achieved double robustness by combining the value function estimator with the marginalised importance sampling estimator. Chen et al. (2021b) considered the OPE problem in a standard MDP without hidden confounders. Their novel contribution is to consider the previous state-action pair as an instrument for the Bellman residual estimation problem of the current state-action pair. Hence, they can employ any IV regression method for this purpose, and in particular, they applied DFIV (Xu et al., 2020), Deep IV (Hartford et al., 2017), and DeepGMM (Bennett et al., 2019), among others. Bennett et al. (2021) considered the OPE problem in a specific setting of an infinite-horizon ergodic MDP that is confounded by hidden confounders. They proposed a method that uses states and actions as proxies for the hidden confounders to identify policy values. Their formulation involves the estimation of the stationary density ratio, and therefore it is not directly generalisable to the general nonlinear IV regression problem. When exact causal inference is unattainable due to the lack of IVs or proxies and the causal effect remains unidentifiable, Pace et al. (2024) developed a pessimistic algorithm based on Delphic uncertainty. This algorithm quantifies the bias arising from hidden confounders by considering variations across plausible world models consistent with observed data. This pessimistic RL algorithm did demonstrate empirically that it accounts for the uncertainty caused by hidden confounders,

but without identifiability there are no theoretical guarantees on the performance of the learnt policy.

In Chapter 4, we propose a novel DML-CMR estimator that solves IV regression with theoretical convergence rate guarantees. We point out that DML-CMR can be directly applied in many of the aforementioned methods (e.g., Liao et al. 2024; Li et al. 2021; Chen et al. 2021b) by replacing the IV regression component in their algorithms with DML-CMR, granting them the convergence rate guarantees of DML-CMR.

3.1.2 Imitation Learning

Learning from demonstrations has been studied in applications such as autonomous driving (Pomerleau, 1988; Lecun et al., 2005) and robotics (Schaal, 1999). Standard imitation learning (IL) methods include Behaviour Cloning (Pomerleau, 1988), Inverse RL (Russell, 1998), and adversarial methods (Ho & Ermon, 2016). IL can suffer from a mismatch between the expert trajectory distribution and the imitator roll-out distribution due to the accumulation of errors. This is first formalised by Ross et al. (2011), who demonstrated that interactive experts that suggest actions in states generated by imitator policy roll-outs allow the imitator to recover from mistakes. Swamy et al. (2021) provided theoretical upper and lower bounds on the imitation gap of IL algorithms that only have access to the trajectories (off-policy), and compared their performance against algorithms that additionally have access to an interactive expert and/or a simulator (on-policy). As expected, on-policy algorithms can produce better imitator policies.

3.1.2.1 Causal and Confounded Imitation Learning

Recently, IL from offline trajectories has been shown to suffer from the existence of latent variables (Ortega et al., 2021), which cause causal delusion. This issue can be resolved by learning an interventional policy. Following this discovery, various methods (Vuorio et al., 2022; Swamy et al., 2022a) considered IL when the expert has access to the full hidden

context that is fixed throughout each episode, but the imitator does not observe the hidden context. These methods learn an interventional policy through on-policy IL algorithms that require an interactive demonstrator and/or an interactive simulator (e.g., DAgger (Ross et al., 2011)).

Orthogonal to these works, Swamy et al. (2022b) considered latent variables unknown to the expert that act as confounding noise that affects the expert policy, but not the transition dynamics. To address this challenge, the problem is then cast into an IV regression problem. In Chapter 5, we combine and generalise the above works (Vuorio et al., 2022; Swamy et al., 2022a,b) to allow the latent variables to be only partially known to the expert, evolving over time in each episode and directly affecting both the expert policy and the transition dynamics. Solving this generalisation implies solving the above problems simultaneously.

Causal confusion (de Haan et al., 2019; Pfrommer et al., 2023) considers the situation where the expert's actions are spuriously correlated with non-causal features of the previous observable states. While it is implicitly assumed that there are no latent variables present in the environment, we can still model this spurious correlation as the existence of hidden confounders that affect both previous states and current expert actions. Slight variations of this setting have been studied in Wen et al. (2020); Spencer et al. (2021); Codevilla et al. (2019).

From the perspective of causal inference (Kumor et al., 2021; Zhang et al., 2020c), there have been studies of the theoretical conditions on the causal graph such that the imitator can exactly match the performance of the expert through backdoor adjustments (*imitability*). Similarly, Ruan et al. (2023) extended imitation conditions and backdoor adjustments to inverse RL. Beyond backdoor adjustments, imitability has also been theoretically studied using context-specific independence relations (Jamshidi et al., 2023).

3.1.3 Learning High-Level Objectives with RL

As mentioned above, Linear Temporal Logic (LTL) is a temporal logic language that can specify high-level objectives and is widely used for robotics and multi-agent systems. For unknown MDP environments, it is desirable for an agent to learn policies with respect to these high-level specifications, thus adopting RL. However, it is not straightforward to use RL to learn the optimal policy that maximises the probability of satisfying the LTL specifications (Alur et al., 2022) as discussed in Chapter 1.

Regarding LTL learning with RL, most existing approaches learn a policy within a product MDP, that is, a product of the environment MDP and an automaton translated from the LTL specification. In earlier work, Sadigh et al. (2014) used a deterministic Rabin automaton to create the product and a simple discounted reward design to learn LTL, but it was later proven (Hahn et al., 2019) that this approach does not provide optimality guarantees. Subsequent work is motivated by a new automaton design, called the limit-deterministic Büchi automaton (LDBA) (Sickert et al., 2016), which is more compatible with reinforcement learning because the limited nondeterminism can be handled by adding additional actions for the agent. Hahn et al. (2019) adopted a product MDP with LDBA and augmented it with sink states to reduce the LTL satisfaction problem into a limit-average reward problem with optimality guarantees. Hahn et al. (2020) later modified this approach by including two discount factors with similar optimality guarantee results, and these algorithms are later integrated into the Mungojerrie (Hahn et al., 2021) tool. Bozkurt et al. (2019) proposed a discounted reward learning algorithm on the product MDP with optimality guarantees, where the discount factor is chosen based on certain assumptions of the unknown environment MDP. Hasanbeig et al. (2019) proposed an accepting frontier function for the product MDP as the reward for the agent, which works best with multiple goal states that must be visited in succession. Cai et al. (2021) extended this reward frontier to continuous motion control tasks. In Chapter 6, we propose a novel and generalised product MDP with an RL algorithm that can learn LTL objectives with optimality guarantees

through a more in-depth theoretical analysis.

Due to the difficulty of learning the infinite-horizon properties of full LTL language, many approaches focus on learning restricted finite variants of the LTL language. De Giacomo & Vardi (2013) and De Giacomo et al. (2019) introduced the LTLf variant and a corresponding reinforcement learning algorithm for it. Littman et al. (2017) proposed a learning algorithm for the GLTL variant, Aksaray et al. (2016) proposed an algorithm to learn Signal Temporal Logic (STL), Kapoor et al. (2020) developed a model-based approach using model predictive control to learn STL, and Li et al. (2016) introduced a truncated LTL variant for robotics applications. Cohen & Belta (2021) developed a method to learn approximately optimal policies for co-safe LTL formulae in a continuous-time uncertain control system using a Lyapunov-based method to provide guarantees. Tractable extensions to learn STL objectives from RL include He et al. (2024), which optimises the value function space, and Wang et al. (2024), which incorporates the idea of counterfactual experience replay inspired by Shao & Kwiatkowska (2023)'s counterfactual imagining to improve tractability. These finite variants of LTL are still very expressive but are unable to represent properties of an infinite proposition sequence.

Another related line of work leverages automata to deal with non-Markovian rewards. Icarte et al. (2022, 2018) proposed a reward machine automaton to represent high-level non-Markovian rewards, and, by exploiting the structure of the known reward machine, better sample efficiency is achieved. Camacho et al. (2019) introduced a method to learn finite LTL specifications by transforming them into reward machines. Gaon & Brafman (2020) used L* to learn a standard DFA from non-Markovian rewards to facilitate learning. Note that the expressiveness of reward machines and DFAs is equivalent to that of regular expressions, which can only express finite-horizon properties and is strictly weaker than LTL. However, recent theoretical work (Le et al., 2024) showed that it is possible to translate LTL objectives into a limit average reward paradigm when the reward can be specified with finite memory reward machines. The limit average reward paradigm is different from

the classic RL setting, where discounted reward is considered, but it opened a new avenue for future theoretical and empirical work on learning LTL with RL.

Moreover, many works exploit high-level logic specifications, such as LTL, to guide hierarchical and compositional learning, which require less complex samples. Andreas et al. (2016) proposed to modularise the task into subtasks and learn subpolicies for them separately. Jiang et al. (2021) proposed to use reward shaping (Ng et al., 1999) based on LTL specifications to guide learning. Jothimurugan et al. (2020) and Jothimurugan et al. (2021) first proposed a logic language to specify high-level tasks, and used this language to guide the learning of compositional tasks represented on a high-level graph. Jothimurugan et al. (2022) further extended this specification-guided learning to the multi-agent setting, learning joint policies that form a Nash equilibrium. Multi-task RL where each task is specified by an LTL specification is also considered by (Vaezipoor et al., 2021), where they exploited the compositional syntax of LTL to efficiently generalise to a large number of tasks. Jackermeier & Abate (2025) improved multi-task RL with LTL specifications by proposing a novel algorithm that can exploit the structure of Büchi automata, instead of greedily or short-sightedly completing the LTL tasks.

3.1.4 Robust and Safe RL

In RL, safety concerns preventing the agent from taking actions that lead to undesirable or catastrophic outcomes, while robustness refers to maintaining reliable performance under model uncertainty or adversarial perturbations. Both are essential for the deployment of RL in safety-critical domains such as healthcare and autonomous driving.

Early works in robust RL has demonstrated that RL agents are vulnerable to adversarial attacks (Lin et al., 2017), which add small perturbations to the state and reward the agent sees. Huang et al. (2017) also showed that, by adversarially perturbing the policy, the performance of the policy can experience dramatic drops. To combat this, a group of works

focuses on learning safe policies using RL. Mason et al. (2017) proposed assured RL, which is a framework that constrains agent behaviour using formally verified policies to achieve safety and optimality requirements. Berkenkamp et al. (2017) proposed to constrain the policy to stay within pre-identified safe regions in the state-action space with ensured Lyapunov stability. Chow et al. (2018) further developed an approach to construct Lyapunov functions as constraints under the constrained Markov decision problem framework to effectively guarantee global safety. Lütjens et al. (2019) proposed to compute guaranteed lower bounds on state-action values to choose optimal actions under worst-case adversarial perturbations in the input space, whilst Zhang et al. (2020b) further developed a theoretically principled policy regularisation to ensure safety for a wider range of RL algorithms. Bethell et al. (2024) proposed adaptive post-shielding with a contrastive autoencoder to identify and prevent unsafe agent actions.

Another line of work directly verifies whether a given policy is safe in the RL environment. Wang et al. (2020b) proposed to integrate neural network certification tools with robust control theory to find invariant sets in the dynamical control loop system that the agent will never escape. Gupta & Hwang (2020) proposed a safety verification framework for model-based RL policies by finding all reachable states under the policy and verifying if the reachable set is safe. Bacci et al. (2021) proposed a method to verify the RL agent up to infinite-horizon through abstract interpretations of the state space. Reachable set analysis was performed to ensure that the agent will remain in the safe regions forever. Bacci & Parker (2022) further extended the abstract interpretation approach to verify probabilistic policies for deep RL with probabilistic guarantees on the execution of that policy.

3.1.5 Causal Reinforcement Learning

Beyond RL with hidden confounders, there is also a variety of research on causal RL, which is an area of research that generally aims to leverage and understand the causal mechanisms and consequences of actions to make better and informed decisions. Usually,

additional assumptions or prior knowledge are required for causal RL methods to account for causal relationships beyond associative correlations in the data. Note that causal RL is not a specific problem formulation, but rather a cluster of RL methods with a causality mindset, aimed at solving various problems. Broadly speaking, there are three general challenges that causal RL aims to solve: spurious correlations and hidden confounders, generalisability and knowledge transfer, and efficient online learning. A review of RL methods dealing with spurious correlations and hidden confounders is already provided in Section 3.1.2.1 and Section 3.1.1. In this section, we will introduce causal RL methods that focus on the other challenges. For these methods, it is generally assumed that no hidden confounders are present.

Firstly, we consider causal RL algorithms that leverage causal inference to generalise the knowledge acquired in the training environment to unseen scenarios and tasks. By learning the causal relationships between variables and identifying the invariance in the environment, agents are able to ignore irrelevant variables and generalise to varying environment dynamics. Zhang et al. (2020a) proposed to use invariant causal prediction to learn causal representation under a block MDP framework that allows the agent to generalise to diverse observation spaces. A similar approach (Bica et al., 2021) proposed to use invariant causal representation to generalise IL across environments. Wang et al. (2022) focused on learning causal dynamics in the environment, which helps to eliminate irrelevant variables and dependencies. This allows the policy to focus on the true causal dynamics, which is generalisable. Saengkyongam et al. (2023) proposed to iteratively check invariance conditions for subsets of variables to find a subset of environment-invariant variables. There also exist algorithms (Sontakke et al., 2021; Lee et al., 2021; Ding et al., 2022) that generalise to environments with parts of the causal mechanism changed, such as different gravitational constants. They all try to identify the relevant causal variables and discard irrelevant variables, either through causal representation learning or causal reasoning. For causal RL algorithms that generalise to different tasks, i.e. reward functions, Eghbal-Zadeh

et al. (2021) proposed to use an attention module for the agent to incorporate disentangled environmental features into the causal representations as signals for adapting to new tasks. Pitis et al. (2022) considered learning a causal dynamic model to generate counterfactual transitions in a dataset, which are transitions predicted by a causal model (e.g., SCM) when the agent is put into some new environment counterfactually.

Secondly, we discuss causal RL algorithms that improve sample efficiency. The algorithms in this category mainly use causal representation learning, directed exploration, and counterfactual data augmentation. Causal representation learning aims to learn a good representation of the underlying causal structure of the environment, which can help the agent to identify significant and causal features while avoiding irrelevant ones, thereby improving its sample efficiency. Therefore, algorithms (Sontakke et al., 2021; Lee et al., 2021; Wang et al., 2022; Huang et al., 2022) that use causal representation learning for generalisation also benefit from improved sample efficiency for this reason. For directed exploration, Zhang & Bareinboim (2017) proposed to derive upper and lower bounds for causal effects when the true causal effects are unidentifiable and use the bounds to guide exploration towards more promising actions. Seitzer et al. (2021) first quantifies the causal influence of actions from environment interactions and uses this information for guided exploration. For counterfactual data augmentation, as the name suggests, it includes methods that use counterfactual reasoning to generate additional data to improve sample efficiency. Buesing et al. (2019) introduced counterfactually-guided policy search, which uses an SCM framework to evaluate the outcomes of counterfactual actions using online interactions. Similarly, Lu et al. (2020a) used an SCM framework to evaluate the potential outcome of counterfactual actions for agents, which reduces the number of interactions with the environment. Pitis et al. (2020) proposed to use a locally factored dynamics model, where it is assumed that the state-action space can be partitioned into disjoint subsets, to generate counterfactual transitions.

3.2 Hidden Confounders

The problem of *hidden confounders* (Pearl, 2000) has been studied in many areas of research. In this section, we first focus on reviewing the literature that identify the causal effect between action and outcome in the presence of hidden confounders through instrumental variables (Section 3.2.1) and proximal causal learning (Section 3.2.2), which is closely related to the work in Chapter 4 and Chapter 5. Next, in Section 3.2.3, we provide an overview of the work related to double machine learning, which is a statistical technique that we use to tackle the problem of hidden confounders in Chapter 4.

3.2.1 Instrumental Variable Regression via Conditional Moment Restriction Estimation

Instrumental variable (IV) regression is a typical example of a conditional moment restriction (CMR) problem, and CMR estimators can be generally applied to solve IV regression directly. Therefore, we first introduce CMR estimators and then introduce algorithms that are proposed specifically for IV regression.

The classic framework of the generalised method of moments (GMM) (Hansen, 1982) was first proposed to translate conditional moment restrictions into unconditional moments, which can thus be estimated under the GMM framework. However, finding the set of unconditional moments that fully capture the conditional moments is challenging. Sometimes, for nonlinear CMRs and functions of interest f, an infinite set of unconditional moments may be required to represent the conditional moments, and a misspecification of the unconditional moments can bias the results (Domínguez & Lobato, 2004). To mitigate some of these issues, Angrist et al. (1996) considered linear functions of interest and proposed the classic two-stage least squares (2SLS) algorithm as a special case of GMMs. Following this, many efforts extend 2SLS to solve CMRs, where the function of interest is nonlinear or nonparametric. One common approach is the sieve method, which uses nonlin-

ear basis functions. Sieve minimum distance (SMD) estimator (Newey & Powell, 2003; Ai & Chen, 2003) performs regression in two stages using an increasing set of nonlinear basis functions as the number of samples increases. Later, a penalised version of SMD (Chen & Pouzo, 2012) was proposed to generalise SMD by allowing non-smooth residuals and high-dimensional function spaces. These sieve-GMM methods and later works (Blundell et al., 2007; Chen & Christensen, 2018; Singh et al., 2019; Muandet et al., 2020) that consider different dictionaries of basis functions enjoy strong consistency and efficiency guarantees, but their flexibility is limited by the set of basis functions, and they can be sensitive to the choice of such functions and regularisation hyperparameters. Note that these works also perform the estimation in two stages.

More recently, various CMR estimation methods based on DNNs have been proposed, since machine learning methods can model highly nonlinear and high-dimensional relationships with greater flexibility. DeepIV (Hartford et al., 2017) extended the classic 2SLS algorithm to the nonlinear setting by adopting DNNs for both stages with conditional density estimation (Darolles et al., 2011) in the first stage. GMM methods are also extended to use DNNs (Bennett et al., 2019; Liao et al., 2020; Dikkala et al., 2020; Bennett & Kallus, 2020), where a minimax criterion is optimised adversarially. Minimax approaches solve a two-player zero-sum game where the players play adversarially. Specifically, player one chooses a hypothesis function to minimise moment violation, and player two chooses a test function that maximises moment violation. These DNN-based GMM methods require minimax optimisation, which is similar to the training of Generative Adversarial Networks (Goodfellow et al., 2014) and could be experimentally unstable. In Chapter 4, we propose a novel CMR estimator, DML-CMR, that uses the double machine learning (Chernozhukov et al., 2018) framework, which we will introduce in Section 4.1.1, and DNNs to provide fast convergence rate guarantees for the estimator.

For algorithms that are developed specifically for IV regression, DFIV (Xu et al., 2020) proposed to use basis functions parameterised by DNNs, which remove restrictions on the

functional form. Kernel IV (Singh et al., 2019) and Dual IV (Muandet et al., 2020) used different dictionaries of basis functions in reproducing kernel Hilbert spaces (RKHS) to solve the IV regression problem. DeepGMM (Bennett et al., 2019) is a DNN-based method that was inspired by GMM to solve IV regression using a minimax approach. Kremer & Schölkopf (2024) improved GMM-based IV regression methods in settings where the data manifold is not uniform through data-derivative information. Chen et al. (2023) presented two semiparametric efficient procedures to estimate a weighted average derivative, and therefore nonparametric IV regression. Their procedures use neural network sieves to approximate the unknown IV functions. They also showed that, with a good choice of sieves, their approach can achieve the semiparametric convergence rate. In addition, other scenarios where the hidden confounder is correlated with some observed confounders, or context, that also affect both the action and the outcome, have been studied by Wu et al. (2022). The correlation will cause the second stage in two-stage IV regression to generate imbalanced observed confounders, and they propose a method to balance the generated confounders that can improve the consistency of the algorithm. Extending this setting, Cheng et al. (2024) considered nonlinear IV regression with IVs that can be correlated with the observed confounder, called conditional IVs. They proposed an IV regression approach leveraging conditional IVs through confounding balancing and representation learning.

3.2.2 Proximal Causal Learning

Proximal Causal Learning (PCL), which considers the problem of learning causal effects with hidden confounders, is also an instance of CMR problems. PCL was first proposed by Miao et al. (2018) to leverage two *proxy variables* for causal identification in estimating the causal function. This was extended by Shi et al. (2020) to a general semiparametric framework, where Tchetgen et al. (2024) introduced a two-stage procedure for linear causal models based on ordinary least squares regression. Mastouri et al. (2021) resolved how to handle nonlinear causal models by replacing linear regression with kernel

ridge regression. To extend the kernel-based PCL methods, Xu et al. (2021) used DNNs as feature maps instead of fixed kernels. This improves the flexibility of the method, especially for highly nonlinear models. Kompa et al. (2022) proposed a single-stage PCL method based on maximum moment restrictions, where they train neural networks to minimise a loss function derived to satisfy the maximum moment restrictions. Cui et al. (2023) introduced a treatment bridge function and incorporated it into the Proximal Inverse Probability Weighting (PIPW) estimator. They considered only binary treatments and derived the Proximal Doubly Robust (PDR) estimator via influence functions. A similar approach by Wu et al. (2024) derived a doubly robust estimator for PCL with continuous treatment through influence functions.

The algorithms proposed specifically for IV regression and PCL often require additional problem-specific assumptions about variables and the functional form. We instead provide a general method for solving CMRs, in Chapter 4, that can be directly applied to a range of problems, including IV regression and PCL.

3.2.3 Double Machine Learning for Hidden Confounders

Double machine learning (DML) was originally proposed for semiparametric regression (Robinson, 1988). It relies on the derivation of a Neyman orthogonal (Neyman & Scott, 1965) score function that describes the regression problem as the learning objective. DML was then extended by adopting DNNs for generalised linear regressions (Chernozhukov et al., 2021). Its strength is that it provides unbiased estimates for two-stage estimations (Jung et al., 2021; Chernozhukov et al., 2022b) under certain identifiability conditions and offers $N^{-1/2}$ convergence rate guarantees, where N is the sample size.

There are previous works on combining DML with CMR estimation, specifically for the IV regression problem, but they are mainly focused on linear and partially linear functions of interest. Belloni et al. (2012) proposed a method to use Lasso and Post-Lasso methods for the first-stage estimation of linear IV to estimate the optimal instruments. To avoid

selection biases, Belloni et al. (2012) leveraged techniques from weak identification robust inference. In addition, Chernozhukov et al. (2015) proposed a Neyman-orthogonalised score for the linear IV problem with control and instrument selection to potentially be robust to regularisation and selection biases of Lasso as a model selection method. Neyman orthogonality for partially linear models with IVs was mainly discussed in the work of Chernozhukov et al. (2018). For an additional discussion, we refer to the book (Chernozhukov et al., 2024).

DML for semiparametric models (Chernozhukov et al., 2022a; Ichimura & Newey, 2022) has been previously applied to solve the nonparametric IV (NPIV) problem. However, their methods require that the average moment of the Neyman orthogonal score is affine (linear) in the nuisance parameters. Therefore, when applied to solve NPIV, functional assumptions regarding the IV set and the residual function were made. Such assumptions are not required in Chapter 4 since we are considering a different problem setting and we formulate a novel Neyman orthogonal score. Furthermore, in contrast to our work in Chapter 4, no implementation and empirical evaluation are performed by these works, and thus it is unclear whether any practical estimator that enjoys these theoretical properties can be derived from their score.

3.2.3.1 Double Machine Learning for Causal Inference

For completeness, we also briefly survey works that use DML for estimating the causal effect without hidden confounders. Doubly robust estimation for causality problems revolved predominantly around the estimation of average treatment effects (ATE) (Robins et al., 1994; Funk et al., 2011; Benkeser et al., 2017; Bang & Robins, 2005; Słoczyński & Wooldridge, 2018). Recently, there has been a surge in doubly robust identification of causal structures beyond the ATE settings. Soleymani et al. (2022); Quinzan et al. (2023) focus on finding direct causes of the target variable using orthogonalised scores. Furthermore, DML techniques for identifying the local average treatment effects (LATE) for non-

linear models with a binary instrument and treatment (action) have been explored before (Chernozhukov et al., 2024).

3.3 Summary

In this section, we summarise the literature reviewed in this chapter and discuss the strengths and limitations of the current state-of-the-art (SOTA) methods. We then position the contributions of this thesis within the existing literature, highlighting the gaps in the literature that this thesis addresses.

The works on online and offline RL, introduced in Section 2.1.2.2 and Section 2.1.3.1, respectively, have demonstrated significant progress in developing practical and high-performing policy learning algorithms with theoretical advancements. However, these approaches do not consider the presence of hidden confounders in the RL environment. They are included in this thesis to provide an overview of the policy learning literature and background for policy learning under hidden confounders. Similarly, the literature on robust RL and causal RL, discussed in Section 3.1.4 and Section 3.1.5, respectively, is included for completeness due to its relevance to this thesis's aim of developing practical and optimal policies with theoretical guarantees.

The works pertaining to RL with hidden confounders, reviewed in Section 3.1.1, are related to Chapter 4. However, as explained in Section 3.1.1, these works focus on applying existing IV regression and causal inference techniques to the confounded RL setting, which involves sequential decision-making. In contrast, Chapter 4 proposes a novel CMR estimator, DML-CMR, that solves the IV regression and the offline IV bandit problem, which is a single-step decision-making problem with hidden confounders and IVs.

In Section 3.2, we reviewed the literature on CMR estimation, IV regression, and PCL. As discussed in that section, classical approaches that use the GMM framework with sieve and kernel methods offer strong theoretical guarantees, particularly in terms of consistency.

However, their practicality is often limited when dealing with high-dimensional or complex function classes, leading to degraded estimation performance or intractable computational costs. Modern CMR estimators leverage DNNs to address these challenges, but often lack theoretical guarantees on consistency, especially on convergence rates. In Chapter 4, our proposed DML-CMR estimator mitigates both theoretical and estimation challenges by employing DNNs for flexible function estimation while providing a fast convergence rate of $O(N^{-1/2})$ through the DML framework. To the best of our knowledge, no existing work adopts the DML framework with DNNs for nonlinear IV regression or general CMR problems. Furthermore, many methods developed specifically for IV regression and PCL are not generalised to solve the full class of CMR problems. In contrast, DML-CMR provides a solution applicable to general CMR problems.

In Section 3.1.2, we reviewed the literature on IL, with a particular focus on confounded IL in Section 3.1.2.1. While prior work has explored IL with confounded datasets, as discussed, these approaches only address isolated aspects of the problem under specific assumptions and settings. A holistic treatment of the issue, along with a discussion of the connections between these settings, is still lacking in the literature. In Chapter 5, we propose a unifying framework that generalises existing methods and allows the solution of a broader class of confounded IL problems with optimality guarantees.

In Section 3.1.3, we reviewed RL approaches for learning high-level objectives, with recent works showing good empirical results for LTL and other logic-based specifications. However, many of these methods lack optimality guarantees or are restricted to finite variants of LTL. Among the works that provide optimality guarantees for full LTL, as discussed in Section 3.1.3, their theoretical results rely on key hyperparameter choices, either by proving the existence of such hyperparameters without any constructive methods to obtain them or by providing unnecessarily harsh bounds. This presents a practical challenge — without appropriate guidance for selecting these hyperparameters, user misspecification can result in suboptimal policies or inefficient learning. In Chapter 6, we propose a novel RL algo-

rithm for learning full LTL objectives that not only provides optimality guarantees, but also achieves substantially improved sample efficiency over existing methods. Moreover, our improved theoretical analysis includes an explicit procedure for selecting key hyperparameters to ensure optimality in practice.

Chapter 4

Learning Policies from Confounded Datasets with Instrumental Variables

Contents

4.1 Introduction **60**

4.1.1 Double Machine Learning 62

4.2 Problem Formulation **64**

4.2.1 Contextual IV Setting 64

4.2.2 Offline IV Bandit 65

4.3 Solving Conditional Moment Restrictions with DML **66**

4.3.1 Neyman Orthogonal Score 67

4.3.2 A DML Estimator for Solving CMR Problems — DML-CMR . 68

4.3.3 Theoretical Analysis 70

4.3.4 A Computationally Efficient Variant of DML-CMR 77

4.3.5 DML Identifiability Condition and Ill-posedness 78

4.4 Suboptimality for Offline IV Bandit **79**

4.5 Experimental Results **80**

4.5.1 IV Regression 81

4.5.2 Offline IV Bandit 84

4.5.3 Proximal Causal Learning 85

4.6 Discussion 88

4.1 Introduction

In this chapter, we focus on learning sample-efficient and optimal policies from offline datasets in the presence of hidden confounders, where IVs are observed in the dataset. Specifically, we learn the causal function using IVs (also known as *IV regression*), in order to learn a decision policy that maximises the expected reward (which we refer to as the *offline IV bandit* problem, described in Section 4.2.2) with suboptimality guarantees. As introduced in Section 2.2.2, IV regression involves solving a CMR problem, but solving it analytically is ill-posed (Nashed & Wahba, 1974; Kress, 1999) because it is an inverse problem. Therefore, various techniques have been proposed to estimate the solutions to CMR problems.

Recall from Section 3.2 that most existing CMR methods, including those that adopt DNNs, take a two-stage approach (Angrist et al., 1996; Newey & Powell, 2003; Chen & Christensen, 2018; Singh et al., 2019; Muandet et al., 2020). In the first stage, they estimate some nuisance parameters, which are parameters or infinite-dimensional functions of no direct interest, but are necessary for the second-stage estimation. However, in these settings, regularisation is often employed to trade off overfitting with the induced regularisation bias, especially for high-dimensional inputs. This is problematic because both regularisation and overfitting can cause heavy bias (Chernozhukov et al., 2018) in two-stage estimations when the first-stage estimator is naively plugged in for the second-stage estimation, which results in a slow convergence rate of the estimator.

To address this challenge, we take inspiration from *double/debiased machine learn-*

ing (Chernozhukov et al., 2018) (DML), which is a statistical framework that provides unbiased estimators with strong convergence rate guarantees for general two-stage regression problems. DML relies on having a Neyman orthogonal (Neyman & Scott, 1965) score function, which describes the estimation problem, to deal with regularisation bias. In addition, it uses cross-fitting, that is, an efficient form of (randomised) data splitting, to address overfitting bias. The DML framework is well-suited for CMR estimation as it allows the use of flexible and powerful estimators such as DNNs, while still providing fast convergence guarantees, addressing key limitations in the literature (recall Section 3.3).

In this chapter, we propose DML-CMR, a novel CMR estimator with fast convergence rate guarantees based on the DML framework. We derive a novel Neyman orthogonal score for CMR problems and design a cross-fitting regime such that, under mild regularity conditions, our estimator is guaranteed to converge at the rate of $N^{-1/2}$, where N is the sample size. We then adopt DML-CMR to solve the offline IV bandit problem, where we derive a policy from the DML-CMR estimator and provide a $O(N^{-1/2})$ suboptimality bound with high probability that matches the suboptimality bounds of *unconfounded* offline bandit algorithms (Jin et al., 2021; Nguyen-Tang et al., 2022). Finally, we evaluate DML-CMR on multiple benchmarks for IV regression, offline IV bandit, and proximal causal learning (PCL).

Main Contributions of this Chapter.

- We propose DML-CMR, a novel CMR estimator that leverages the DML framework to provide unbiased estimates of the solutions to CMR problems, which we use to solve the offline IV bandit problem.
- We derive a novel, Neyman orthogonal, score function for CMR problems in Section 4.3.1, and design a cross-fitting regime for the DML-CMR estimator to mitigate the bias in Section 4.3.2.
- In Section 4.3.3, we show an asymtoptic convergence rate for the DML-CMR estimator

at the rate of $N^{-1/2}$, which is minimax optimal under parameterisation and mild regularity conditions, leading to $O(N^{-1/2})$ suboptimality for the derived policy proved in Section 4.4.

- On a range of IV regression, offline IV bandit, and PCL benchmarks, including two real-world datasets, we empirically demonstrate that DML-CMR outperforms other SOTA methods in Section 4.5.

Work reported in this chapter first appeared in Shao et al. (2024, 2025b).

4.1.1 Double Machine Learning

Before introducing the problem studied in this chapter, we first describe the DML framework. DML considers the problem of estimating a function of interest f as a solution to an equation of the form

$$\mathbb{E}[\psi(\mathcal{D}; f_0, \eta)] = 0, \tag{4.1}$$

where ψ is referred to as a score function and f_0 is the true function. Here, η is a nuisance parameter, which can be of parametric form or infinite-dimensional functions. It is of no direct interest, but must be estimated to obtain an estimate of f_0. For example, in a two-stage CMR estimator, nuisance parameters such as conditional density are estimated in the first stage, and in the second stage, they are used to estimate f_0. DML provides a set of tools to derive an unbiased estimator of f_0 with convergence rate guarantees, even when the nuisance parameter η suffers from regularisation, overfitting, and other types of biases present in the training of ML models, which typically cause slow convergence when learning f_0.

In order to estimate f_0, DML reduces biases by using score functions ψ that are Neyman orthogonal (Neyman & Scott, 1965) in η. This requires the Gateaux derivative, which defines the directional derivative for functionals, of the score function ψ w.r.t. the nuisance

parameters at f_0, η_0 to be zero:

$$\frac{\partial}{\partial r}\Big|_{r=0} \mathbb{E}[\psi(\mathcal{D}; f_0, \eta_0 + r\eta)] = 0 \quad \text{for all } \eta. \tag{4.2}$$

Here, f_0 and η_0 are the true parameters where $\mathbb{E}[\psi(\mathcal{D}; f_0, \eta_0)] = 0$. Intuitively, the condition in Equation (4.2) is met if small changes in the nuisance parameter do not significantly affect the score function around the true function f_0. Neyman orthogonality is key in DML, as it allows fast convergence for estimating f_0, even if the estimator for the nuisance parameter η is biased. For score functions that are Neyman orthogonal, we define DML with *K-fold cross-fitting* as follows.

Definition 4.1 (DML, Definition 3.2 (Chernozhukov et al., 2018))**.** Given a dataset $\mathcal{D}$ of N observations, consider a score function ψ as in Equation (4.1), and suppose that ψ is Neyman orthogonal that satisfies Equation (4.2). Take a *K-fold* random partition $\{I_k\}_{k=1}^K$ of observation indices $[N]$ each with size $n = N/K$, and let $\mathcal{D}_{I_k}$ be the set of observations $\{d_i : i \in I_k\}$. Furthermore, define $I_k^c := [N] \setminus I_k$ for each fold k, and construct estimators $\widehat{\eta}_k$ of the nuisance parameter using $\mathcal{D}_{I_k^c}$. Then, construct an estimator $\widehat{f}$ as a solution to the following equation,

$$\frac{1}{K}\sum_{k=1}^{K} \widehat{\mathbb{E}}_k[\psi(\mathcal{D}_{I_k}; \widehat{f}, \widehat{\eta}_k)] = 0, \tag{4.3}$$

where $\widehat{\mathbb{E}}_k$ is the empirical expectation over $\mathcal{D}_{I_k}$.

In the above definition, $\widehat{f}$ is defined as an exact solution to the empirical expectation equation in Equation (4.3). In practice, we can also define the estimator $\widehat{f}$ as an approximate solution to Equation (4.3).[1]

[1]This approximation error is different from the estimation error. The estimation error measures the difference between $\widehat{f}$ and f_0, whereas the approximation error concerns the error of minimising the empirical risk. In fact, the approximation error contributes to the estimation error, which is analysed in Section 4.3.3.

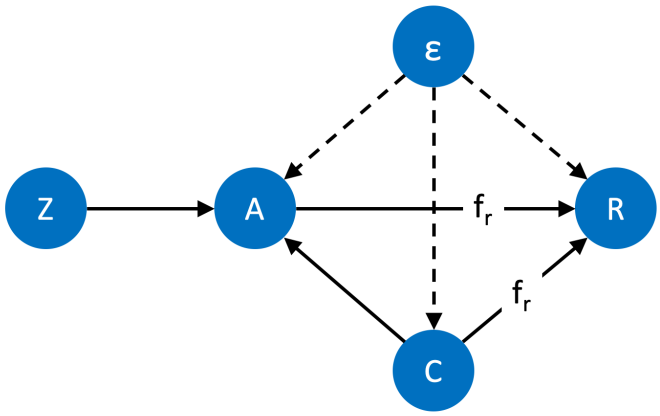


Figure 4.1: The causal graph of the contextual IV setting, where $R = f_r(C, A) + \epsilon$ and Z is an *instrumental variable* that affects R only through A.

4.2 Problem Formulation

We can now formally state the problem considered in this chapter. We first introduce the contextual IV setting with the corresponding offline IV bandit problem and discuss their connection to CMR problems.

4.2.1 Contextual IV Setting

We begin with a description of the contextual IV setting (Hartford et al., 2017) that is considered in this chapter. We observe an *action* $A \in \mathcal{A} \subseteq \mathbb{R}^{d_A}$, a *context* $C \in \mathcal{C} \subseteq \mathbb{R}^{d_C}$, an *IV* $Z \in \mathcal{Z} \subseteq \mathbb{R}^{d_Z}$ and an *outcome* $R \in \mathbb{R}$, where there exist *unobserved confounders* that affect all of A, C and R through a hidden variable (or *noise*) ϵ. IV directly affects the action A, does not directly affect the outcome R, and is not correlated with the hidden confounder ϵ. These causal relationships are illustrated in Figure 4.1 and are represented by the following structural causal model (recall Section 2.2.1):

$$R := f_r(C, A) + \epsilon, \quad \mathbb{E}[\epsilon] = 0, \quad \mathbb{E}[\epsilon | A, C] \neq 0, \tag{4.4}$$

where f_r is an unknown, continuous, and potentially nonlinear causal function. Denote the set of observations $\{(c_i, z_i, a_i, r_i)\}_{i\in[N]}$ generated from this model as the *offline dataset* $\mathcal{D}$ of size N. In this chapter, we consider the problem of learning the *counterfactual prediction function* (Hartford et al., 2017),

$$f_0(C, A) := f_r(C, A) + \mathbb{E}[\epsilon|C] = \mathbb{E}[R|do(A), C] \tag{4.5}$$

The term $\mathbb{E}[\epsilon|C]$ is typically nonzero[2], but learning f_0 still allows us to compare between different actions when given a context as $f_0(C, a_1) - f_0(C, a_2) = f_r(C, a_1) - f_r(C, a_2)$ for all $a_1, a_2 \in \mathcal{A}$, and in particular, $\arg\max_a f_0(C, a) = \arg\max_a f_r(C, a)$.

As discussed in Section 2.2.3, learning f_0 in this setting is an IV regression problem with observed context C, provided that Z satisfies the IV conditions in Definition 2.5. Furthermore, it can be formulated as a CMR problem of the form $\mathbb{E}[R - f_0(C, A)|Z, C] = 0$. Therefore, we will be able to use the novel CMR estimator with fast convergence rate guarantees proposed in Section 4.3 to learn the counterfactual prediction function f_0.

4.2.2 Offline IV Bandit

The learnt estimator of f_0 from the offline dataset $\mathcal{D}$ can be used to solve the *offline bandit problem in the contextual IV setting* (Zhang et al., 2022), that is, to identify a (deterministic) policy $\pi : \mathcal{C} \to \mathcal{A}$ that maximises the value,

$$V(\pi) := \mathbb{E}_{c\sim\mathbb{P}_{\text{test}}}[R|do(A = \pi(c)), c] = \mathbb{E}_{c\sim\mathbb{P}_{\text{test}}}[f_0(c, \pi(c))],$$

which is the expected outcome when performing actions following π. $\mathbb{P}_{\text{test}}$ is a test context distribution that can potentially differ from the distribution of $\mathcal{D}$. The optimal policy π^* should satisfy $V(\pi^*) = \max_\pi V(\pi)$, and suboptimality of a policy π is defined as

[2]In the setting where $\mathbb{E}[\epsilon|C] = 0$ is assumed (Bennett et al., 2019; Xu et al., 2020), $f_0 = f_r$ and all our results apply.

subopt$(\pi) := V(\pi^*) - V(\pi)$. Furthermore, the optimal policy π^* can be retrieved from f_0 by selecting $\pi^*(c) = \arg\max_{a \in \mathcal{A}} f_0(c, a)$.

4.3 Solving Conditional Moment Restrictions with DML

We now present the main contribution of this chapter — an estimator that solves CMR problems using the DML framework. We first propose a novel Neyman orthogonal score for the estimator and a novel two-stage algorithm that estimates the solutions to CMR problems. Our estimator can use DNN estimators in both stages and provides guarantees on the convergence rate by leveraging the DML framework. Then, we adopt this CMR estimator to solve the offline IV bandit problem.

To solve the CMR problem defined in Equation (2.9) using the DML framework, we first need a Neyman orthogonal score. As introduced in Section 2.2.2, the optimisation objective for many two-stage CMR estimators (Angrist et al., 1996; Hartford et al., 2017; Singh et al., 2019) is $\hat{f} \in \arg\min_{f \in \mathcal{F}} \mathbb{E}[(Y - \mathbb{E}[f(X)|C])^2]$ (recall Equation (2.10)). Let $g_0(f, c) := \mathbb{E}[f(X)|c]$ and $\mathcal{G}$ be some function space that includes g_0 and its potential estimators $\widehat{g}$. Then, these two-stage CMR estimators estimate g_0 with $\widehat{g}$ in the first stage, and then use $\widehat{g}$ to optimise the following loss function, $\ell = (Y - \widehat{g}(f, c))^2$, in the second stage. However, as we show in Proposition 4.2 with proof deferred to Appendix A.1, this objective, or the score function, is not Neyman orthogonal.

Proposition 4.2. *The score (or objective) function for standard two-stage CMR estimators, $\ell = (Y - \widehat{g}(f, c))^2$, is not Neyman orthogonal at (f_0, g_0).*

This means that small misspecifications or estimation biases of $\widehat{g}$ can lead to significant changes to the score function, and there are no guarantees on the convergence rate if the first-stage estimator $\widehat{g}$ is naively plugged into the second stage to estimate f_0. To address this, we first derive a novel Neyman orthogonal score function for the CMR problem and then design a CMR algorithm with K-fold cross-fitting that uses the DML framework.

4.3.1 Neyman Orthogonal Score

Typically, to construct a Neyman orthogonal score from a non-orthogonal score, additional nuisance parameters need to be estimated (Chernozhukov et al., 2018). These additional nuisance terms adjust the score in a way that makes it orthogonal, where the error in estimating f_0 due to errors in the nuisance parameters becomes second order in the Taylor expansion (Foster & Syrgkanis, 2019). In our case, to construct a Neyman orthogonal score for CMR problems from the standard objective in Equation (2.10), we first select relevant functions that should be estimated as nuisance parameters. Following two-stage IV regression approaches (Hartford et al., 2017), estimating g_0 is essential for identifying f_0, so we will estimate it as a nuisance parameter. We find that, by additionally estimating $s_0(c) \coloneqq \mathbb{E}[Y|c]$ inside some function space $\mathcal{S}$, we can construct the score function:

$$\psi(\mathcal{D}; f, (s, g)) = (s(c) - g(f, c))^2. \tag{4.6}$$

Here, the nuisance parameters are $\eta = (s, g)$. For this to be a valid Neyman orthogonal score function, we check, with the following theorem, that

- $\mathbb{E}[\psi(\mathcal{D}; f_0, (s_0, g_0))] = 0$ with the true functions (s_0, g_0);
- The Gateaux derivative of $\psi(\mathcal{D}; f, (s, g))$ vanishes at $(f_0, (s_0, g_0))$.

The proof is deferred to Appendix A.2.1.

Theorem 4.3 (Neyman orthogonality)**.** *The score function* $\psi(\mathcal{D}; f, (s, g)) = (s(c) - g(f, c))^2$ *obeys the Neyman orthogonality conditions at* $(f_0, (s_0, g_0))$.

This Neyman orthogonal score function is abstract in the sense that it allows for general estimation methods for g_0 and s_0, as long as they satisfy certain convergence conditions, which are introduced in the next section. In addition, having a Neyman orthogonal score is useful in general to debias two-stage estimators (Foster & Syrgkanis, 2019), beyond the DML framework.

4.3.2 A DML Estimator for Solving CMR Problems — DML-CMR

With the Neyman orthogonal score, we now propose a novel DML estimator, DML-CMR, that solves Equation (2.9). Note that, in general, f_0 is allowed to be infinite-dimensional, as commonly seen in the nonparametric IV literature (Newey & Powell, 2003). We also allow f_0 to be infinite-dimensional for the Neyman orthogonal score introduced in Section 4.3.1. For the theoretical analysis of DML-CMR, while it is possible to provide a general analysis following Foster & Syrgkanis (2019) for nonparametric f_0 with the Neyman orthogonal score, the analysis would require more assumptions and the convergence rate will depend on the complexity of the function classes involved in the estimation. For our analysis, since we propose a concrete estimator, we would like to provide a concrete analysis following the DML framework (Chernozhukov et al., 2018), which is designed for semiparametric estimation, to show an optimal parametric rate for DML-CMR. Therefore, we assume that f_0 is finite-dimensional and parameterised for the theoretical analysis of DML-CMR.

Assumption 4.4 (Parameterisation)**.** Let $f_0 = f_{\theta_0}$ and $\Theta \subseteq \mathbb{R}^{d_\theta}$ be a compact space of parameters of f, where the true parameter $\theta_0 \in \Theta$ is in the interior of Θ.

From this assumption, we can define $\mathcal{F} := \{f_\theta : \theta \in \Theta\}$ as the function space of f.

The DML-CMR Estimator. The procedure of our DML-CMR estimator with K-fold cross-fitting is outlined in Algorithm 1. Given a dataset $\mathcal{D} = (y_i, x_i, c_i)_{i\in[N]}$ of size N, we first split the dataset using a random partition $\{I_k\}_{k=1}^K$ of dataset indices $[N]$ such that the size of each fold I_k is N/K, and let $\mathcal{D}_{I_k}$ denote the set of observations $\{d_i : i \in I_k\}$.

As introduced in Section 4.1.1, our DML estimator will be a two-stage procedure. In the first stage (lines 4-7 in Algorithm 1), for each fold $k \in [K]$, we learn $\widehat{s}_k$ and $\widehat{g}_k$ using data $\mathcal{D}_{I_k^c}$ with indices $I_k^c := [N] \setminus I_k$. Then $\widehat{s}_k \approx \mathbb{E}[Y|C]$ can be learnt through standard supervised learning using a neural network with inputs C and labels Y. For $\widehat{g}_k$, we follow Hartford et al. (2017) to estimate $F_0(X|C)$, the conditional distribution of X given C,

Algorithm 1 DML-CMR with K-fold cross-fitting

1: **Input:** Dataset $\mathcal{D}$ of size N, number of folds K for cross-fitting, mini-batch size n_b
2: Get a partition $(I_k)_{k=1}^K$ of dataset indices $[N]$
3: **for** $k = 1$ **to** K **do**
4: $\quad I_k^c := [N] \setminus I_k$
5: $\quad$ Learn $\widehat{s}_k$ and $\widehat{g}_k$ using $\{(\mathcal{D}_i) : i \in I_k^c\}$
6: **end for**
7: Initialise $f_{\widehat{\theta}}$
8: **repeat**
9: $\quad$ **for** $k = 1$ **to** K **do**
10: $\quad\quad$ Sample n_b data c_i^k from $\{(\mathcal{D}_i) : i \in I_k\}$
11: $\quad\quad \mathcal{L} = \widehat{\mathbb{E}}_{c_i^k}\left[(\widehat{s}_k(c) - \widehat{g}_k(f_{\widehat{\theta}}, c))^2\right]$
12: $\quad\quad$ Update $\widehat{\theta}$ to minimise loss $\mathcal{L}$
13: $\quad$ **end for**
14: **until** convergence
15: **Output:** The DML-CMR estimator $f_{\widehat{\theta}}$

with $\widehat{F}$, and then estimate $\widehat{g}$ via

$$\widehat{g}(f_\theta, c) = \sum_{\dot{X} \sim \widehat{F}(X|C)} f_\theta(\dot{X}) \approx \int f_\theta(X)\widehat{F}(X|C = c)dX \approx \mathbb{E}[f_\theta(X)|c].$$

For example, if the action space is discrete, $\widehat{F}$ can be a categorical model, e.g., a DNN with softmax output. For a continuous action space, a mixture of Gaussian models can be adopted to estimate the distribution $F_0(X|C)$, where a DNN is used to predict the mean and standard deviation of the Gaussian distributions.

In the second stage (lines 8-15 in Algorithm 1), we estimate $\widehat{\theta}$ using our Neyman orthogonal score function ψ in Equation (4.6). The key is to optimise $\widehat{\theta}$ with data from the k-th fold $\mathcal{D}_{I_k}$ using nuisance parameters $\widehat{s}_k$, $\widehat{g}_k$ that are trained with $\mathcal{D}_{I_k^c}$, the complement of $\mathcal{D}_{I_k}$. This is important to fully debias the estimator $\widehat{\theta}$. The DML estimator $f_{\widehat{\theta}}$ is then defined as

$$f_{\widehat{\theta}} := \min_{f_\theta \in \mathcal{F}} \frac{1}{K} \sum_{k=1}^{K} \widehat{\mathbb{E}}_k[(\widehat{s}_k(c) - \widehat{g}_k(f_{\widehat{\theta}}, c)^2], \tag{4.7}$$

where $\widehat{\mathbb{E}}_k$ is the empirical expectation over $\mathcal{D}_{I_k}$. In practice, we alternate between the K folds while sampling a mini-batch c_i^k of size n_b from each fold $\mathcal{D}_{I_k}$ to update $\widehat{\theta}$ by minimising the empirical loss on the mini-batch following our Neyman orthogonal score ψ,

$$\mathcal{L} = \widehat{\mathbb{E}}_{c_i^k}\left[(\widehat{s}_k(c) - \widehat{g}_k(f_{\widehat{\theta}}, c))^2\right] = \sum_{c_i^k} \frac{1}{n_b}\left((\widehat{s}_k(c) - \widehat{g}_k(f_{\widehat{\theta}}, c))^2\right).$$

When the second stage converges, we return the DML-CMR estimator $f_{\widehat{\theta}}$.

The two hyperparameters for this algorithm are the number of folds K and the mini-batch size n_b. In theory, the convergence guarantee derived below holds for any choice of $K \in [2...N]$, where $K = N$ would be leave-one-out cross-fitting. However, there exists a trade-off between estimation error and computational complexity. While a larger K allows more data to train the nuisance parameters $(\widehat{s}_k, \widehat{g}_k)$ for each fold, which implies a lower estimation error, the nuisance parameters must be trained K times, increasing the computation cost. Empirically, good performance can often be observed from $K = 5$ to $K = 10$. The mini-batch size n_b is a standard hyperparameter in deep learning that can be determined based on the training dataset, the neural network architecture, and the computation hardware.

4.3.3 Theoretical Analysis

In this section, we provide a theoretical analysis on the convergence of DML-CMR. The key benefit of DML is its debiasing effect for two-stage regressions, and crucially, it is possible to leverage the DML framework (Chernozhukov et al., 2018) to show a fast asymptotic convergence rate of $O(N^{-1/2})$, i.e., the DML estimator $\widehat{\theta}$ converges to the true parameters θ_0 at the rate of $O(N^{-1/2})$ with high probability. To provide a road map of this section, we first list all the technical conditions required for a general DML estimator to converge at the fast rate of $O(N^{-1/2})$ following Chernozhukov et al. (2018). Later, in Theorem 4.8, we will prove that all these conditions hold for our DML-CMR estimator. The full proofs in

this section are deferred to Appendix A.2.1.

Condition 4.5 (Technical conditions of DML $N^{-1/2}$ rate proved later in Theorem 4.8)**.** For sample size $N \geq 3$:

(a) The map $(\theta, (s, g)) \mapsto \mathbb{E}[\psi(\mathcal{D}; f_\theta, (s, g))]$ is twice continuously Gateaux-differentiable.

(b) The score ψ obeys the Neyman orthogonality conditions in Equation (4.2).

(c) The true parameter θ_0 obeys $\mathbb{E}[\psi(\mathcal{D}; f_{\theta_0}, (s_0, g_0))] = 0$ and Θ contains a ball of radius $c_1 N^{-1/2} \log N$ centered at θ_0.

(d) For all $\theta \in \Theta$, the identification relationship

$$2\|\mathbb{E}[\psi(\mathcal{D}; f_\theta, (s_0, g_0))]\| \gtrsim \|J_0(\theta - \theta_0)\|$$

is satisfied, where $J_0 := \partial_{\theta'}\{\mathbb{E}[\psi(\mathcal{D}; f_{\theta'}, (s_0, g_0))]\}|_{\theta'=\theta_0}$ is the Jacobian matrix, with singular values bounded between $c_0 > 0$ and $c_1 > 0$.

(e) Let K be a fixed integer. Given a random partition $\{I_k\}_{k=1}^K$ of indices $[N]$, each of size $n = N/K$, the nuisance parameter estimator $\widehat{s}_k$ and $\widehat{g}_k$ learnt using data with indices I_k^c belongs to shrinking realisation sets $\mathcal{S}_N$ and $\mathcal{G}_N$, respectively, and the nuisance parameters should be estimated at the $o(N^{-1/4})$ rate, e.g., $\|\widehat{s} - s_0\|_2 = o(N^{-1/4})$.

Among these conditions, (a), (b), and (c) are conditions regarding the Neyman orthogonal score ψ. The Neyman orthogonality in (b) is shown in Theorem 4.3 and the other conditions (b) and (c) are mild regularity conditions, standard for moment problems. (d) is an identification condition that ensures sufficient identifiability of θ_0. This condition also implies bounded ill-posedness, which we will discuss in detail in Section 4.3.5. Finally, (e) is a key condition that states the nuisance parameters should converge to their true values

at the crude rate of $o(N^{-1/4})$, where a shrinking realisation set $\mathcal{S}_N$ is a decreasing set of possible estimators $\widehat{s}$ as the sample size N increases.

In Chernozhukov et al. (2018), an additional condition that all eigenvalues of the matrix $\mathbb{E}[\psi(\mathcal{D}; f_{\theta_0}, (s_0, g_0))\psi(\mathcal{D}; f_{\theta_0}, (s_0, g_0))^T]$ are strictly positive (bounded away from zero) is required, which means that the score function has nondegenerate variance at true parameters. This additional condition allows them to show that $\widehat{\theta}$ is asymptotically normal around θ_0, in addition to the $N^{-1/2}$ convergence rate. However, since our score function's variance at true parameters is 0 almost surely, we omit this condition and instead provide $N^{-1/2}$ convergence rate guarantee with high probability.

In Lemma 4.7, following recent works (Chernozhukov et al., 2018, 2021, 2022b), we show that the convergence condition for the nuisance parameters in Condition 4.5 (e) can be transformed into a condition on the critical radius (Bartlett et al., 2005) of the realisation sets, which is a quantity that describes the complexity of estimation and has been widely studied for various function classes. We start by assuming the realisability of the true functions g_0, s_0 and f_0 in their corresponding function classes, and further assuming that they are bounded. We formalise these assumptions in Assumption 4.6.

Assumption 4.6 (Realisable and bounded function classes)**.** We assume that g_0, s_0, f_0 are realisable in the function classes $\mathcal{G}, \mathcal{S}, \mathcal{F}$, that is, $g_0, s_0, f_0 \in \mathcal{G}, \mathcal{S}, \mathcal{F}$, respectively, and furthermore, $\|f\|_\infty, \|s\|_\infty \leq B$ for all $f, s \in \mathcal{F}, \mathcal{S}$, where B is a positive constant. Moreover, we assume that the random variable $|Y| \leq B$ *almost surely*.

As discussed in Section 4.3.1, DML-CMR has two nuisance parameters that are required to be estimated: $\widehat{s} \in \mathcal{S}$ and $\widehat{g} \in \mathcal{G}$. As we saw in Section 4.3.2, the estimation of $\widehat{s}$ is made through standard supervised learning algorithms that we can directly analyse. However, the estimation of $\widehat{g}$ has two steps: (i) we estimate the conditional distribution

$\widehat{F}(X|C) \in \mathcal{P}$, where the density sieve $\mathcal{P}$ is defined as

$$\mathcal{P} \subset \left\{ F : \int F(x|C = c)dx = 1 \quad \forall c \in \mathcal{C} \right\};$$

and (ii) we plug in the functional f_θ into the conditional expectation estimator,

$$\widehat{g}_{\widehat{F}}(f_\theta, c) := \int f_\theta(x)\widehat{F}(x|C = c)dx,$$

for all candidate test functions $f_\theta \in \mathcal{F} = \{f_\theta \ : \ \theta \in \Theta\}$. From the realisability assumption of $g_0 \in \mathcal{G}$ in Assumption 4.6, it follows that $F_0(X|C) \in \mathcal{P}$, and the hypothesis space for the estimand $\widehat{g}$ and the true parameter g_0 is defined as $\mathcal{G} := \{g_F \ : \ F \in \mathcal{P}\}$.

Lemma 4.7 (Convergence of nuisance parameters)**.** *Under Assumption 4.6, let $\mathcal{S}_N^*$ be the star-hull of the realisation set $\mathcal{S}_N$ of function class $\mathcal{S}$ centred at s_0, defined as*

$$\mathcal{S}_N^* = \{C \mapsto \gamma(s(C) - s_0(C)) \ : \ s \in \mathcal{S}_N, \gamma \in [0, 1]\},$$

$\mathcal{P}_N^*$ be the star-hull of the realisation set $\mathcal{P}_N$ of the function class $\mathcal{P}$ centred at F_0,

$$\mathcal{P}_N^* = \{C \mapsto \gamma(F(\cdot|C) - F_0(\cdot|C)) \ : \ F \in \mathcal{P}_N, \gamma \in [0, 1]\},$$

and $\mathcal{G}_N^*$ be the star-hull of the realisation set $\mathcal{G}_N$ of the function class $\mathcal{G}$ centred at g_0,

$$\mathcal{G}_N^* = \{C, f \mapsto \gamma(g(C, f) - g_0(C, f)) \ : \ g \in \mathcal{G}_N, \gamma \in [0, 1]\},$$

where $\mathcal{S}_N$, $\mathcal{P}_N$ and $\mathcal{G}_N$ are properly shrinking neighbourhoods of the true functions s_0, F_0 and g_0. Then, there exist universal constants c_1 and c_2, for which we have that, with

probability at least $1-\xi$*, the estimation errors are bounded as*

$$\|\widehat{s}-s_0\|_2^2 \leq c_1\left(\delta_N(\mathcal{S}_N^*)^2+\sqrt{\frac{\log(1/\zeta)}{N}}\right);$$
$$\|\widehat{g}-g_0\|_2^2 \leq c_2\left(\delta_N(\mathcal{P}_N^*)^2+\sqrt{\frac{\log(1/\zeta)}{N}}\right),$$

where $\delta_N(\mathcal{S}_N^*)$ *denotes the critical radius of* $\mathcal{S}_N^*$*. The critical radius is defined as the minimum* δ *that satisfies* $\mathcal{G}(\mathcal{S}_N^*,\delta)\leq\delta^2/2$*, where*

$$\mathcal{G}(\mathcal{S}_N^*,\delta) \coloneqq \mathbb{E}_\epsilon\left[\sup_{h\in\mathcal{H}^*:\|h\|_N\leq\delta}\langle\epsilon,h\rangle\right]$$

This lemma shows that, if we can upper bound the critical radii $\delta_N(\mathcal{S}^*)$ and $\delta_N(\mathcal{P}^*)$ by $o(N^{-1/4})$, then $\|\widehat{s}-s_0\|_2=o(N^{-1/4})$ and $\|\widehat{g}-g_0\|_2=o(N^{-1/4})$, meaning that nuisance parameters converge to their true values at the rate of $o(N^{-1/4})$ as required by Condition 4.5 (e).

For the critical radius, it is typically shown that $\delta_N=O(d_N^{1/2}N^{-1/2})$ (Chernozhukov et al., 2022b, 2021), where d_N is the effective dimension of the hypothesis space, which is another notion to bound the estimation error in the regression problem (for further details on the effective dimension, see Appendix A.2.3.). This, together with Lemma 4.7, implies that $\|\widehat{s}-s_0\|_2=O(d_N(\mathcal{S}^*)^{1/2}N^{-1/2})$. Therefore, we can also see that, if the effective dimension satisfies $d_N(\mathcal{S}^*)=o(N^{1/4})$, then $\|\widehat{s}-s_0\|_2=o(N^{-1/4})$ as required by Condition 4.5 (e) (and similarly for $\widehat{g}$ and $d_N(\mathcal{P}^*)$).

Therefore, we can refer to results in the literature that analyse the effective dimension and critical radius of various estimators to provide examples of estimators that satisfy Condition 4.5 (e). For the estimation of $\widehat{s}$, we have a regression problem and Condition 4.5 (e) is satisfied by many supervised learning estimators such as parametric generalised linear models (Van Der Vaart et al., 1996), Lasso (Bickel et al., 2009), random forests (Syrgkanis & Zampetakis, 2020), boosting (Luo et al., 2016), Sobolev kernel regression with α-

smooth RKHS ($\alpha > d/2$, where d is the dimension of X) (Caponnetto & De Vito, 2007; Christmann & Steinwart, 2008) and neural networks (Chen & White, 1999; Yarotsky, 2018; Schmidt-Hieber, 2020; Farrell et al., 2021). For the conditional density estimator $\widehat{g}$, the above estimators also satisfy Condition 4.5 (e) if the conditional distribution can be parameterised accordingly. Otherwise, this condition is also satisfied by Gaussian mixtures density networks (Ho et al., 2022), polynomial sieve with Hölder smoothness $\alpha > \frac{d+1}{2}$ (Ai & Chen, 2003), and categorical-logistic models (Zhao et al., 2022), among others.

Leveraging Lemma 4.7, we derive the following theorem on the convergence of the DML-CMR estimator.

Theorem 4.8 (Convergence of DML-CMR)**.** *Let $f_{\theta_0} \in \mathcal{F}$ be a solution that satisfies the CMRs in Equation* (2.9)*, let ψ be the Neyman orthogonal score defined in Equation* (4.6) *and let $J_0 := \partial_{\theta'}\{\mathbb{E}[\psi(\mathcal{D}; f_{\theta'}, (s_0, g_0))]\}|_{\theta'=\theta_0}$ be the Jacobian matrix of $\mathbb{E}[\psi]$ w.r.t. θ. Suppose that the upper bound of the critical radius $\delta_N = o(N^{-1/4})$ for $\widehat{s}$, $\widehat{g}$, and J_0 has bounded singular values. Then, if Assumption 4.4 and 4.6 hold, our DML estimator $f_{\widehat{\theta}}$ satisfies that $\widehat{\theta}$ is concentrated in a $N^{-1/2}$ neighbourhood of θ_0 with probability $1 - o(1)$. Specifically, for all $\zeta \in (0, 1]$, there exist $K > 0$ such that for all integer $N \geq K$,*

$$\|\widehat{\theta} - \theta_0\| = O(N^{-1/2})$$

with probability $1 - \zeta$.

Proof sketch. From Proposition A.1, we have that if all technical conditions stated in Condition 4.5 are satisfied, the DML estimator converges at the required rate (formally stated and proved in Proposition A.1). Therefore, we need to verify that, under Assumption 4.6, all components of Condition 4.5 are satisfied. Condition (a) is satisfied since $(s - g)^2$ is twice continuously differentiable with respect to s and g. Condition (b) and (c) are satisfied by Theorem 4.3 and Assumption 4.4, since the true parameter $\theta_0 \in \Theta$ is in the interior of Θ.

Condition (d) and (e) are technical conditions that ensure sufficient identifiability and non-degeneracy. Condition (d) can be proved to hold using the first-order Taylor series for the score function $\mathbb{E}[\psi(\mathcal{D}; f_\theta, (s_0, g_0))]$ around the point θ_0 and the fact that $J_0 J_0^T$ is non-singular since the singular value of J_0 is bounded, as stated in the Theorem. Finally, condition (e) is satisfied by Lemma 4.7 and the fact that the critical radius $\delta_N = o(N^{-1/4})$. Therefore, all components of Condition 4.5 are satisfied, and the proof follows by Proposition A.1. □

Theorem 4.8 states that, with adequately trained nuisance parameter estimators in terms of their critical radius and identifiability conditions in terms of the non-singularity of the Jacobian matrix J_0, the estimator error $\widehat{\theta} - \theta_0$ is normally distributed, where its variance shrinks at the rate of $N^{-1/2}$. This implies that $\widehat{\theta}$ converges to θ_0 at the rate $O(N^{-1/2})$ with high probability, which allows us to bound the estimation error $\|f_{\widehat{\theta}} - f_{\theta_0}\|_2$ of the DML-CMR estimator with high probability, under a Lipschitz condition of f_θ.

Corollary 4.9 (Estimation error bounds)**.** *Let $f_{\widehat{\theta}}$ be the DML estimator for CMRs. If all assumptions for Theorem 4.8 hold and there exists a constant $L > 0$ such that $\|f_\theta(x) - f_{\theta_0}(x)\|_2 \leq L\|\theta - \theta_0\|_2$ for all $x \in \mathcal{X}$ and $\theta \in \Theta$, then for all $\zeta \in (0, 1]$, we have that for all $\zeta \in (0, 1]$, there exist $K > 0$ such that for all integer $N \geq K$,*

$$\|f_{\widehat{\theta}} - f_{\theta_0}\|_2 = O\left(L \cdot N^{-1/2}\right),$$

with probability $1 - \zeta$.

Proof. From Theorem 4.8, we have that the parameters $\widehat{\theta}$ for our DML estimator $f_{\widehat{\theta}}$ learnt from a dataset of size N satisfy that for all $\zeta \in (0, 1]$, there exist $M, K > 0$ such that for all integer $N \geq K$,

$$\mathbb{P}\left(\|\widehat{\theta} - \theta_0\| \leq M \cdot N^{-1/2}\right) \geq 1 - \zeta.$$

If we let L to be a constant such that $\|f_\theta(x) - f_{\theta_0}(x)\| \leq L\|\theta - \theta_0\|$ for all $x \in \mathcal{X}$ and $\theta \in \Theta$, we have that, for all $\zeta \in (0, 1]$, there exist $M, K > 0$ such that for all integer $N \geq K$,

$$\mathbb{P}\left(\|f_{\widehat{\theta}}(x) - f_{\theta_0}(x)\| \leq L \cdot M \cdot N^{-1/2}\right) \geq 1 - \zeta,$$

which implies $\|f_{\widehat{\theta}} - f_{\theta_0}\| = O\left(L \cdot N^{-1/2}\right)$ with probability 1-ζ as required. □

Here, we assume a local Lipschitz condition of f_θ w.r.t. θ around θ_0: $\|f_\theta(x) - f_{\theta_0}(x)\|_2 \leq L\|\theta - \theta_0\|_2$ for all $x \in \mathcal{X}$ and $\theta \in \Theta$ for some $L > 0$. Since Θ is compact, we see that it is enough for this Lipschitz condition to hold locally in a neighbourhood of θ_0.

4.3.4 A Computationally Efficient Variant of DML-CMR

In this section, we propose a computationally efficient variant of DML-CMR. The standard DML-CMR with K-fold cross-fitting (Algorithm 1) trains the nuisance parameters $\widehat{s}$ and $\widehat{g}$ K times on different subsets of the dataset to mitigate overfitting bias. While this approach improves statistical robustness and stability, it is computationally expensive. To address this, we propose CE-DML-CMR, a variant of DML-CMR described in Algorithm 2 that does not use K-fold cross-fitting. It trains $\widehat{s}$ and $\widehat{g}$ only once (instead of K times) using the entire dataset, significantly reducing computational cost. CE-DML-CMR uses the same Neyman orthogonal score as the standard DML-CMR, so it still enjoys the partial debiasing effect (Mackey et al., 2018) from the Neyman orthogonal score. However, without K-fold cross-fitting, it lacks the theoretical convergence rate guarantees provided by Theorem 4.8. Nonetheless, CE-DML-CMR can be viewed as a trade-off between computational complexity and theoretical guarantees, and it also serves as an ablation study on K-fold cross-fitting.

Algorithm 2 Computationally Efficient CE-DML-CMR

Input: Dataset $\mathcal{D}$ with size N, mini-batch size n_b
Learn $\widehat{s}$ and $\widehat{g}$ using $\mathcal{D}$
Initialise $f_{\widehat{\theta}}$
repeat
  Sample n_b data c_i from $\mathcal{D}$
  $\mathcal{L} = \widehat{\mathbb{E}}_{c_i}[(\widehat{s}(c) - \widehat{g}(f_\theta, c))^2]$
  Update $\widehat{\theta}$ to minimise loss $\mathcal{L}$
until convergence
Output: The CE-DML-CMR estimator $f_{\widehat{\theta}}$

4.3.5 DML Identifiability Condition and Ill-posedness

In this section, we provide a discussion regarding the relationship between our identifiability condition, Condition 4.5 (d), and a more common notion of identifiability for CMR problems, the *ill-posedness* (Chen & Pouzo, 2012). To begin with, we define the ill-posedness of CMR problems.

Definition 4.10 (Ill-posedness (Chen & Pouzo, 2012; Dikkala et al., 2020))**.** Given a CMR problem as in Equation (2.9), the *ill-posedness* ν of the function space $\mathcal{F}$ is given by

$$\nu = \sup_{f \in \mathcal{F}} \frac{\|f_0 - f\|_2}{\|\mathbb{E}[f_0(X) - f(X)|C]\|_2}.$$

Intuitively, ill-posedness describes how well a small CMR error (the projected error under conditional expectation) implies a small L_2 error (root mean squared error) for f_0. For identifiability, it is usually assumed that the ill-posedness ν is bounded. Otherwise, even solving the CMRs with very small error does not guarantee a solution $\widehat{f}$ that is close to f_0. In our case, we demonstrate that the identification condition of a DML estimator actually implies bounded ill-posedness. Specifically, Condition 4.5 (d) implies that the ill-posedness is bounded, as shown by the following proposition.

Proposition 4.11 (Identifiability and Ill-posedness)**.** *For all $\theta \in \Theta$, if there exists a constant $L > 0$ such that $\|f_\theta(x) - f_{\theta_0}(x)\|_2 \leq L\|\theta - \theta_0\|_2$ for all $x \in \mathcal{X}$ and $\theta \in \Theta$, then the*

identifiability condition in Condition 4.5 (d), which states

$$2\|\mathbb{E}[\psi(\mathcal{D}; f_\theta, (s_0, g_0))]\| \geq \|J_0(\theta - \theta_0)\|$$

and the Jacobian matrix J_0 have singular values bounded between $c_0 > 0$ and $c_1 > 0$, implies that the ill-posedness is bounded by $\nu \leq L/\sqrt{c_0}$.

The proof of Proposition 4.11 is deferred to Appendix A.2.2. This interesting finding explains why, in Theorem 4.8, there are no explicit assumptions about the ill-posedness of the problem. The only identifiability assumption for Theorem 4.8 is the non-singularity of the Jacobian matrix J_0, which for DML-CMR is sufficient to ensure Condition 4.5 (d) holds. Therefore, by Proposition 4.11, this ensures a bounded ill-posedness of the problem, which in turn allows us to identify the solution f_0 with a small error.

4.4 Suboptimality for Offline IV Bandit

In this section, we apply DML-CMR (Algorithm 1) to solve the offline IV bandit problem and provide an upper bound on the suboptimality of the learnt policy. Recall from Section 4.2.1 that the CMR problem we solve for the offline IV bandit problem is $\mathbb{E}[R - f_0(C, A)|Z, C] = 0$. Using DML-CMR, we can provide an estimator $f_{\widehat{\theta}}(C, A)$ for f_0, and we retrieve (an estimate of) the induced optimal policy as $\widehat{\pi}(c) := \arg\max_a h_{\widehat{\theta}}(c, a)$. The suboptimality of a policy is subopt$(\widehat{\pi}) := V(\pi^*) - V(\widehat{\pi})$. Next, we show a suboptimality bound for the DML-CMR policy in terms of the sample size N based on Theorem 4.8.

Corollary 4.12 (Suboptimality Bounds)**.** *Let the learnt policy from a dataset of size N be $\widehat{\pi}(c) := \arg\max_a f_{\widehat{\theta}}(c, a)$, where $\widehat{\theta}$ is the DML-CMR estimator. Let L be a constant such that $\|f_\theta(C, A) - f_{\theta_0}(C, A)\|_2 \leq L\|\theta - \theta_0\|_2$ for all C in the support of $\mathbb{P}_{test}$, $A \in \mathcal{A}$, and $\theta \in \Theta$. Then, for all $\zeta \in (0, 1]$, we have that there exist $K > 0$ such that for all integer*

$N \geq K$, the suboptimality of $\widehat{\pi}$ satisfies

$$\textit{subopt}(\widehat{\pi}_N) = O(L \cdot N^{-1/2})$$

with probability $1 - \zeta$.

The proof is deferred to Appendix A.2.4. To the best of our knowledge, this is the first time that the suboptimality bounds of $O(N^{-1/2})$ have been proved for confounded IV bandit, matching the suboptimality bounds of the unconfounded bandit.

4.5 Experimental Results

In this section, we empirically evaluate DML-CMR for IV regression, offline IV bandit, and PCL tasks. We also evaluate the computationally efficient CE-DML-CMR, finding CE-DML-CMR empirically performs as well as standard DML-CMR on low-dimensional datasets, where overfitting bias is not prevalent.

Experimental Setup The experiments are carried out on a Linux server (Ubuntu 18.04.2) with two Intel Xeon Gold 6252 CPUs and six NVIDIA GeForce RTX 2080 Ti GPUs. We use DNN estimators for both stages of DML-CMR (Algorithm 1), with network architectures and hyperparameters detailed in Appendix A.4. Our evaluation considers both low- and high-dimensional synthetic datasets, as well as semi-synthetic real-world datasets. The synthetic datasets are based on an airline ticket demand scenario introduced in Chapter 1. The semi-synthetic datasets are from the healthcare domain, studying the effect of treatments on infants and the impact of air pollution on human health. Details of the datasets are provided in the following sections and in Appendix A.3. We run each method 20 times and plot the median, 25th, and 75th percentiles of the results. All algorithms are implemented using PyTorch (Paszke et al., 2019), and the code is available on GitHub[3].

[3] `https://github.com/shaodaqian/DML-CMR`

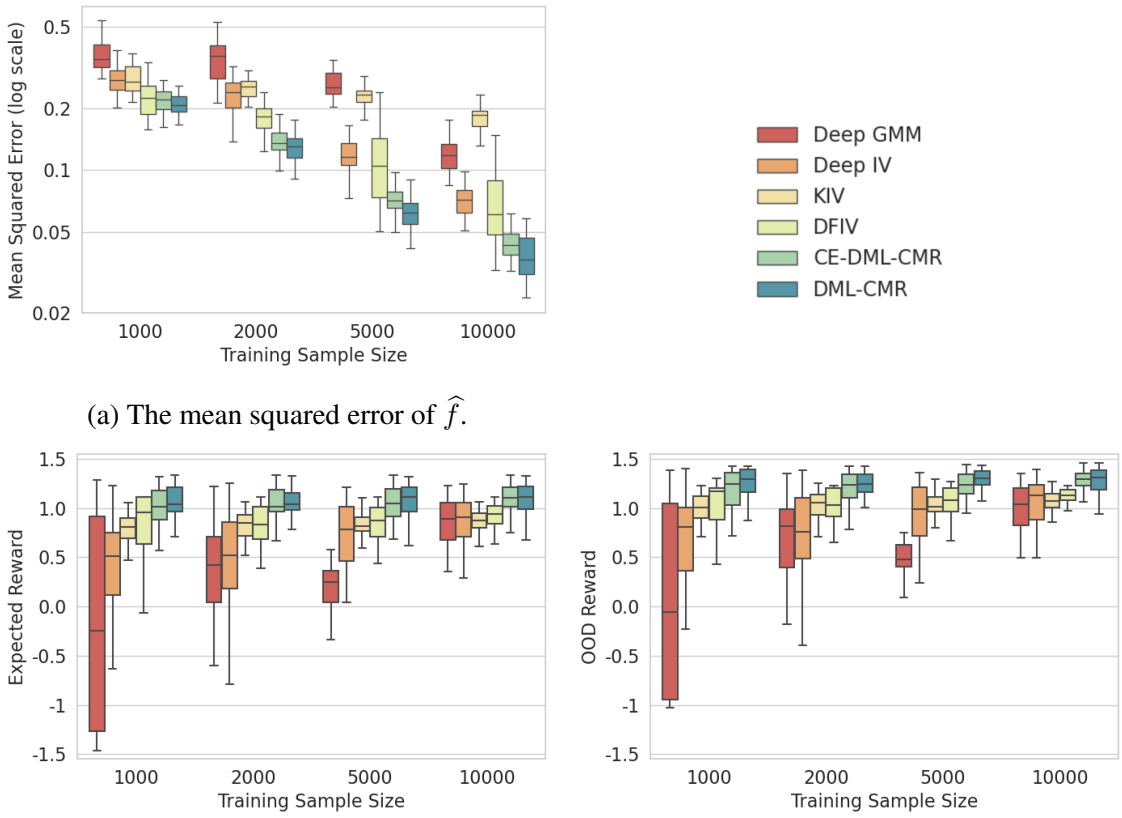
4.5.1 IV Regression

For IV regression, we compare our methods against leading modern approaches: Deep IV (Hartford et al., 2017), DeepGMM (Bennett et al., 2019), KIV (Singh et al., 2019) and DFIV (Xu et al., 2020). As reviewed in Section 3.2.1, Deep IV, DeepGMM, and DFIV are DNN-based estimators, whereas KIV is a kernel-based estimator that leverages the GMM framework. Therefore, this selection of methods provides a representative and diverse benchmark, covering both neural and kernel-based methods, and reflects the current best-performing algorithms commonly used as baselines in the IV regression literature.

Results of DML-CMR using tree-based estimators such as Random Forests and Gradient Boosting are provided in Appendix A.6.2, where comparable performance to DNN-based DML-CMR is demonstrated. In addition, we provide a sensitivity analysis against hyperparameter changes in Appendix A.6.3 and an evaluation of algorithms when the IV is weakly correlated with the treatment, representing higher ill-posedness of the CMR problem, in Appendix A.6.1. The training time per run for DML-CMR with $K = 10$ is less than one hour, whereas CE-DML-CMR takes less than 10 minutes. The runtimes of other algorithms range from minutes to tens of minutes.

4.5.1.1 Ticket Demand Dataset

We first conduct experiments for IV regression on the ticket demand dataset, which is a synthetic dataset introduced by Hartford et al. (2017) that is now a standard benchmark for nonlinear IV methods. In this dataset, the goal is to understand how ticket prices p affect ticket sales r. We observe two context variables, which are the time of year $t \in [0, 10]$ and customer type $s \in [7]$ variables, the latter categorised by the level of price sensitivity. Price and context affect sales through $f_0((t, s), p) = 100 + (10 + p) \cdot s \cdot \psi(t) - 2p$, where $\psi(t)$ is a complex nonlinear function. However, the noise of r and p is correlated, which indicates the existence of unobserved confounders. The fuel price z is introduced as an instrumental variable. Details of this dataset are included in Appendix A.3.1.



(a) The mean squared error of $\widehat{f}$.

(b) The average reward following the policy $\widehat{\pi}$ derived from $\widehat{f}$.

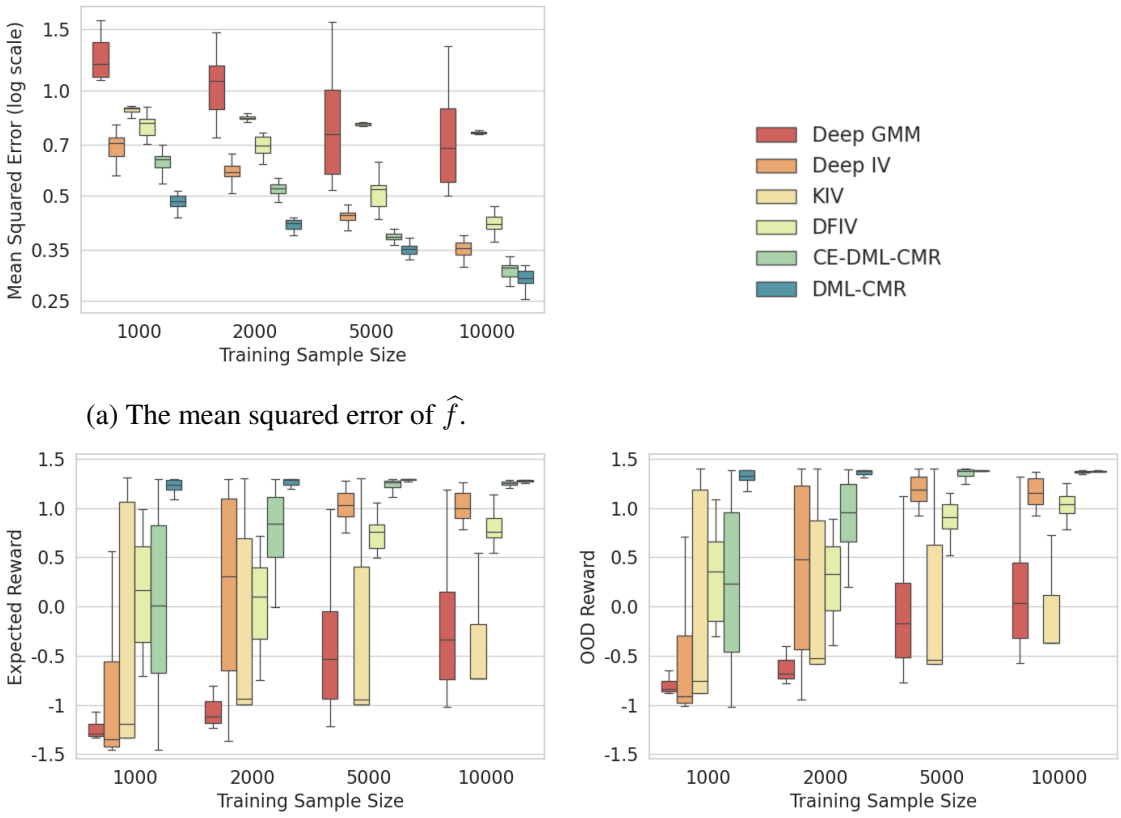
(c) The average reward following $\widehat{\pi}$ with out of training distribution context.

Figure 4.2: Results on the ticket demand dataset with low-dimensional context.

The mean squared errors (MSE) for learning f_0 with this dataset of various sizes are provided in Figure 4.2a. It can be seen that DML-CMR performs better than other IV regression methods for all dataset sizes. CE-DML-CMR, which requires significantly less computation, matches the performance of DML-CMR in this case.

4.5.1.2 High-Dimensional Dataset

In real applications, we typically do not observe variables such as the customer type as explicit categories. Therefore, we follow Hartford et al. (2017) and consider the case where the customer type $s \in [7]$ is replaced by images of the corresponding handwritten digits from the MNIST dataset (LeCun & Cortes, 2010) to evaluate our methods with high-dimensional (28^2=784 dimensions) inputs. The task remains to learn f_0, but the al-



(a) The mean squared error of $\widehat{f}$.

(b) The average reward following the policy $\widehat{\pi}$ derived from $\widehat{f}$.

(c) The average reward following $\widehat{\pi}$ with out of training distribution context.

Figure 4.3: Results on the ticket demand dataset with high-dimensional context.

gorithms are no longer explicitly given the 7 customer types, and instead have to infer the relationship between the image data and the outcome. Results for IV regression are plotted in Figure 4.3a, where DML-CMR and CE-DML-CMR outperform all other methods. In these high-dimensional settings, regularisation is heavily used to avoid overfitting. DML-CMR demonstrates the benefits of using DML to reduce both the regularisation and overfitting bias caused by learning the nuisance parameters.

4.5.1.3 Real-World Datsets

Next, we test the performance of DML-CMR on real-world datasets. The true counterfactual prediction function is rarely available for real-world data. Therefore, in line with previous approaches (Shalit et al., 2017; Wu et al., 2023b; Schwab et al., 2019; Bica et al.,

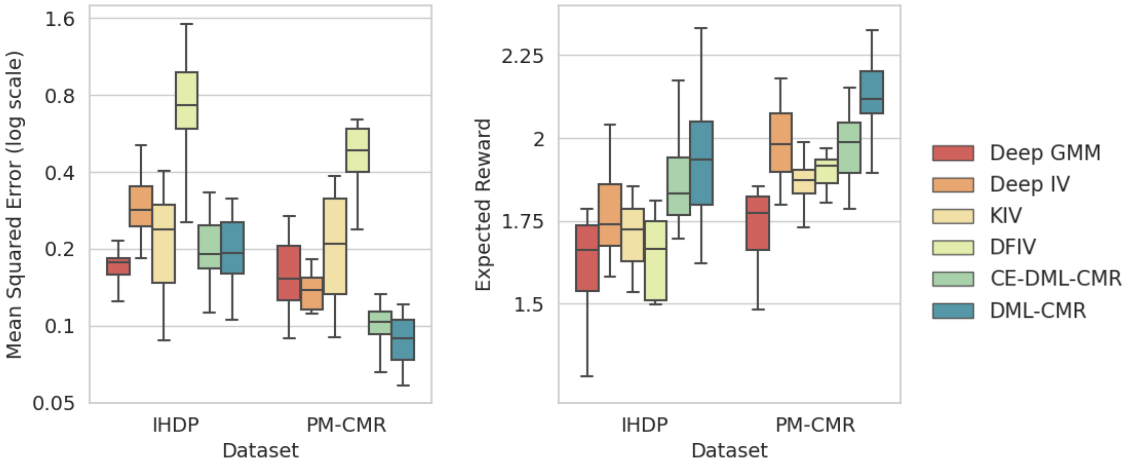


Figure 4.4: The mean squared error of $\widehat{f}$ and average reward following $\widehat{\pi}$ for the real-world datasets.

2020), we instead consider two semi-synthetic real-world datasets IHDP[4] (Hill, 2011) and PM-CMR[5] (Wyatt et al., 2020). We directly use the continuous variables from IHDP and PM-CMR as context variables and generate the outcome variable with a nonlinear synthetic function following Wu et al. (2023b). There are 470 and 1350 training samples in IHDP and PM-CMR, respectively (for details see Appendix A.3.1). As shown in Figure 4.4, DML-CMR and CE-DML-CMR demonstrate comparable, if not lower, MSE of fitting $\widehat{f}$ than the other methods. This shows that our algorithm can reliably learn the counterfactual prediction function from real-world data.

4.5.2 Offline IV Bandit

We also evaluate the ability of DML-CMR to learn effective policies in the offline IV bandit setting, where the goal is to maximise expected reward. This evaluation is conducted using the same datasets and benchmarking methods as in Section 4.5.1. From the learnt $\widehat{f}$, for each context sampled from the test distribution, we retrieve the best action by uniformly sampling actions from the action space $\mathcal{A}$ and selecting the action for which $\widehat{f}$ returns the

[4]IHDP: `https://www.fredjo.com/`.
[5]PM-CMR: `https://doi.org/10.23719/1506014`.

highest value. Using this induced policy $\widehat{\pi}$, we compare the expected reward following $\widehat{\pi}$ over the test distribution.

For the low-dimensional ticket demand dataset, we first set the test distribution to be the same as the training distribution and plot the average rewards in Figure 4.2b. In Figure 4.2c, we shift the test distribution out of the training distribution by incrementing the distribution of t by 1. For the high-dimensional setting, Figure 4.3b and Figure 4.3c demonstrate the expected rewards for test distributions in and out of the training distribution, respectively. There is a clear trend that a better fitted (low MSE) $\widehat{f}$ leads to an induced policy with higher expected reward. In all cases, DML-CMR outperforms all other methods, especially in the high-dimensional setting, where DML-CMR consistently learns the near-optimal policy with only 2000 samples. CE-DML-CMR, on the other hand, only matches the performance of DML-CMR for the low-dimensional setting, but still outperforms the other methods in the high-dimensional setting. For the real-world dataset, DML-CMR also outperforms all other methods in Figure 4.4.

We only compare with other IV regression methods because there are no offline bandit methods that consider the IV setting, and standard offline bandit algorithms (e.g., (Valko et al., 2013; Jin et al., 2021; Nguyen-Tang et al., 2022)) fail to learn meaningful policies when the dataset is confounded, as demonstrated in Appendix A.5.

4.5.3 Proximal Causal Learning

For the PCL task, recall that the CMR problem to solve is $\mathbb{E}[Y - h(A, W)|A, V] = 0$ as shown in Section 2.2.4. We compare our methods with modern PCL methods: CEVAE (Im et al., 2021), PMMR (Mastouri et al., 2021), KPV (Mastouri et al., 2021), DFPV (Xu et al., 2021), NMMR U (Kompa et al., 2022), NMMR V (Kompa et al., 2022) and PKDR (Wu et al., 2024). As reviewed in Section 3.2.2, these methods also include both DNN-based and kernel-based methods, forming a high-performing and diverse benchmark. As discussed in Section 2.2.4, in PCL, we can only estimate the average treatment effect $\mathbb{E}[R|do(A)]$ rather

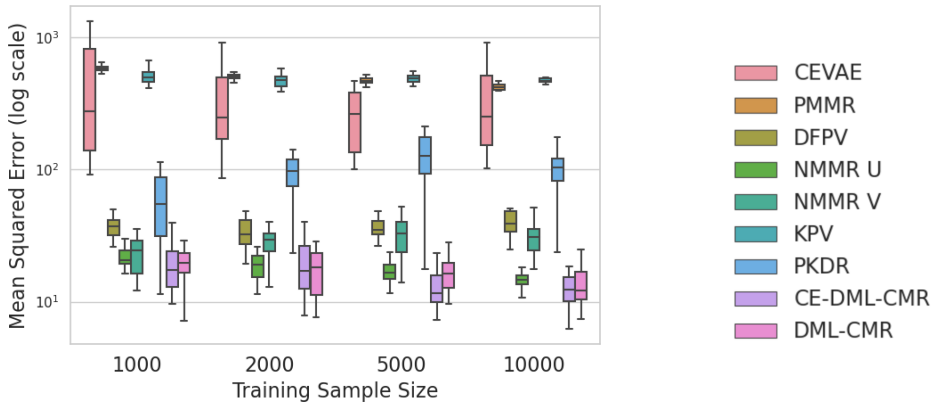


Figure 4.5: The mean squared error of $\widehat{f}$ on the ticket demand dataset for the PCL task.

than the conditional average treatment effect $\mathbb{E}[R|do(A), C]$, as in the case of IV regression. Therefore, a context-dependent policy cannot be derived using PCL. Nonetheless, we can still evaluate DML-CMR for solving the CMRs and estimating the average treatment effect in PCL. The training time per run for DML-CMR with $K = 10$ and CE-DML-CMR is also around one hour and 10 minutes, respectively. The runtimes of other algorithms range from minutes to tens of minutes.

4.5.3.1 Ticket Demand Dataset

Similarly to IV regression, we start with the ticket demand dataset (Hartford et al., 2017) which has been adopted to the PCL setting (Xu et al., 2021), where the goal is to understand how ticket prices affect ticket sales and learn the causal function f_0. The hidden confounder in this case is the varying demand U, while the cost of fuel V is the treatment proxy, which directly impacts the ticket price, and the number of views on the airline's reservation website W is the outcome proxy. Details of this dataset are included in Appendix A.3.2.

The results for learning f_0 with this dataset of various sizes are provided in Figure 4.5. It can be seen that DML-CMR and CE-DML-CMR achieved state-of-the-art (SOTA) performance with very similar performance to each other. NMMR U can match DML-CMR at smaller dataset sizes, but DML-CMR achieves lower MSE at 5000 and 7500 sample sizes.

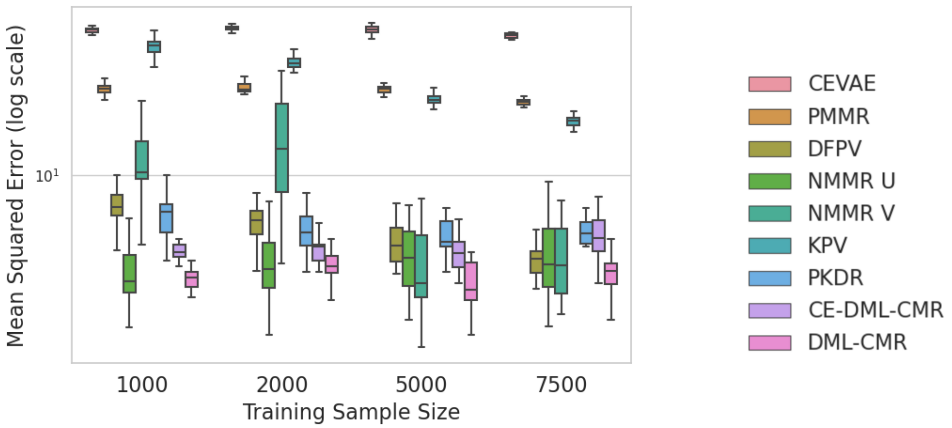


Figure 4.6: The mean squared error of $\widehat{f}$ on the dSprites dataset with high dimensional treatment for the PCL task.

The performance gap between CE-DML-CMR and DML-CMR is small in this case, as expected, because the variables are low-dimensional.

4.5.3.2 High-dimensional dSprites Dataset

For a high-dimensional dataset, we adopt the dSprites dataset (Matthey et al., 2017) for PCL, first introduced by Xu et al. (2021). dSprites is an image (64×64) dataset where each image is described by five parameters: *shape*, *scale*, *rotation*, *posX* and *poxY*. The treatments are these high-dimensional dSprites images, the hidden confounder is posY, the proxies are noisy observations of scale, rotation, and posX, and the outcome is defined by a nonlinear causal function. Details of this dataset are provided in Appendix A.3.2.

The results are presented in Figure 4.6. DML-CMR achieved similar performance to the SOTA methods while outperforming CE-DML-CMR. In addition, a lower variance can be observed when using DML-CMR compared to other methods, especially for smaller data sizes. The NMMR methods, in some cases, outperform CE-DML-CMR. This is in line with our previous observations that, without K-fold validation, performance can be worse for high-dimensional datasets and the full debiasing effect of the DML framework is required to achieve the best results.

4.6 Discussion

In this chapter, we proposed a novel estimator for solving CMRs, DML-CMR. Using the DML framework and our novel Neyman orthogonal score, DML-CMR can effectively estimate solutions to CMR problems with fast convergence rate guarantees of $O(N^{-1/2})$ by mitigating the regularisation and overfitting biases in two-stage estimations. We theoretically analysed DML-CMR and proved a convergence rate of $O(N^{-1/2})$ with high probability under mild regularity and parametric assumptions. We also demonstrated interesting connections between the notion of ill-posedness for CMRs and DML's identifiability condition. Using DML-CMR, we are able to learn causal functions and effective decision policies for the offline IV bandit problem with suboptimality guarantees. We evaluated DML-CMR on IV regression and offline IV bandit problems with standard benchmarks and semi-synthetic real-world datasets, experimentally showing that it outperforms SOTA IV regression methods with lower estimation error and better stability. In addition, we evaluated DML-CMR on proximal causal learning, which also demonstrated SOTA performance.

The main limitation of this chapter is the requirement that IVs need to be observed in the dataset for each action in order to identify the causal effect of actions. However, finding IVs in real-world applications is a well-studied problem in the literature and it is well documented that IVs can be observed in many real-world scenarios including econometrics, drug testing, social sciences, and recommender systems (see Wu et al. (2023a) Section 8 for more real-world datasets and applications). Moreover, even when IVs are not identified in a dataset by experts, there exist empirical methods that can test and find good candidates for IVs in the dataset (e.g., Yuan et al. 2021; Wu et al. 2023a Section 6). That being said, it is not guaranteed that IVs can be found in all real-world datasets, so in Chapter 5, we propose a method to imitate expert demonstrations with hidden confounders without explicit IVs being observed in the dataset.

Another limitation is the parametric assumption on f_0 for the theoretical analysis fol-

lowing the DML framework. As discussed in Section 4.3.3, a more generic analysis can be performed for infinite-dimensional f_0 following Foerster et al. (2018), but we provide a concrete analysis with an explicit convergence rate for DML-CMR in this chapter. Nevertheless, it would be interesting future work to analyse our Neyman orthogonal score for estimating nonparametric functions of interest. In addition, our offline IV bandit setting can be extended to sequential decision settings (Namkoong et al., 2020; Liao et al., 2024) with IVs at each time step, which is an interesting future direction to explore.

Chapter 5

Imitating Expert Policies from Confounded Datasets

Contents

5.1 Introduction 91

5.2 Causal Imitation Learning 93

5.3 A Unifying Framework for Causal Imitation Learning 94

5.3.1 MDPs with Hidden Confounders. 94

5.3.2 Causal Imitation Learning in Our Framework 95

5.3.3 Reducing Our Framework to Prior Work 97

5.4 Causal IL as a CMR Problem 100

5.4.1 Derivation of the CMR Problem 101

5.4.2 Practical Algorithms for Solving the CMR Problem 102

5.4.3 Theoretical Analysis 104

5.5 Experimental Results 108

5.5.1 Ticket Pricing Environment 109

5.5.2 Mujoco Environments 111

5.5.3 Adopting Other CMR Estimators 112

5.6 Discussion 113

5.1 Introduction

In this chapter, we consider the problem of imitation learning (IL) in the presence of hidden confounders with suboptimality guarantees, without the explicit assumption relied upon in Chapter 4 that IVs can be observed in addition to the MDP trajectories in the dataset. While classical IL theory implies that, with infinite data, the IL error should converge to zero and the imitator should be value-equivalent to the expert (Ross et al., 2011; Spencer et al., 2021), it has been observed in practice that IL algorithms often produce incorrect estimates of the expert policy, leading to suboptimal and unsafe behaviours (Lecun et al., 2005; Codevilla et al., 2019; Bansal et al., 2018; Kuefler et al., 2017).

Most existing work attributes the underlying causes of this problem to hidden confounders and spurious correlations. However, as discussed in Section 3.1.2.1, they only consider specific aspects of the problem in different settings, and a holistic treatment of the problem is lacking. In this chapter, we propose a unifying framework that generalises existing work by explicitly formulating hidden confounders, which, in the context of IL, refers to variables present in the environment but not recorded in the demonstrations. Crucially, we distinguish between hidden confounders that are observable to the expert and those that are not[1]. When the expert can observe the hidden confounders, they represent privileged information. When the expert cannot observe the hidden confounders, they act as confounding noise that contaminates the demonstrations, causing spurious correlations and causal delusions.

Building on our unifying framework, we develop algorithms that can correctly and efficiently imitate the expert policy in the presence of hidden confounders. To break the spu-

[1]We always assume that the imitator does not observe the hidden confounders.

rious correlations introduced by the expert-unobservable hidden confounders without explicit IVs at each time step, our key observation is that the trajectory histories themselves can serve as IVs for state-action pairs within our confounded IL framework. Moreover, by conditioning on the trajectory history, we can infer information about observable hidden confounders and learn a history-dependent policy that accurately mimics the expert's behaviour. Importantly, we demonstrate that, within our general framework, IL under hidden confounders can be reformulated as a conditional moment restriction (CMR) problem. This reformulation allows us to design practical algorithms based on CMR estimators (e.g., DML-CMR proposed in Chapter 4) with theoretical guarantees on the imitation gap.

Main Contributions of this Chapter.

- We introduce a unifying framework for causal IL (Section 5.3) incorporating both expert-observable and expert-unobservable confounding variables to unify and generalise prior work (e.g., Swamy et al. 2022b,a; Ortega et al. 2021; Vuorio et al. 2022).
- We reformulate the problem of confounded IL in our unifying framework as solving a CMR problem as in Chapter 4, where we learn a history-dependent policy by leveraging trajectory histories as instruments to break the confounding (Section 5.4).
- We propose DML-IL, a novel IL algorithm in our framework, for which we prove an upper bound on the imitation gap that recovers the theoretical results in prior work as special cases (Theorem 5.8).
- We empirically validate our algorithm in both custom and MuJoCo environments with both expert-observable and expert-unobservable confounders and demonstrate that DML-IL outperforms existing causal IL baselines (Section 5.5). This highlights the need to explicitly account for both types of hidden confounders.

Work reported in this chapter first appeared in Shao et al. (2025a).

5.2 Causal Imitation Learning

Before formalising our framework and the corresponding IL problem in Section 5.3, we first discuss existing causal IL formulations and clarify the objective of this chapter. As discussed above, previous work on IL with hidden confounders, which we refer to as causal IL, extends standard IL (recall Section 2.1.4) under varying assumptions in different settings, each adopting its own specific formalism.

Previous work has studied specific problem settings in isolation, including spurious correlations without explicit hidden confounders (de Haan et al., 2019; Codevilla et al., 2019; Pfrommer et al., 2023), temporal confounding noise (Swamy et al., 2022b), expert-specific privileged information (Swamy et al., 2022a; Vuorio et al., 2022; Chen et al., 2019; Choudhury et al., 2017), causal delusions (Ortega & Braun, 2008; Ortega et al., 2021), and covariate shifts (Spencer et al., 2021). However, in real-world scenarios, these problems often exist simultaneously (e.g., confounding noise and expert-privileged information co-occurring), rendering partial or independent treatments insufficient, as demonstrated in Section 5.5. Our framework proposed in Section 5.3 unifies these prior settings (see Section 5.3.3) and allows a much larger family of problems that are more realistic in practice to be considered.

More specifically, some of these works (Swamy et al., 2022a; Vuorio et al., 2022; Swamy et al., 2022b) assume that the imitator can interactively query the expert for additional demonstrations, and interactive IL algorithms (recall Section 2.1.4), such as DAgger (Ross et al., 2011), are used to address the hidden confounders. However, such an assumption is impractical in many real-world scenarios, where only a fixed set of demonstrations is available. For this reason, we focus on developing methods that mitigate the effects of hidden confounders while relying solely on a fixed set of non-interactive expert demonstrations.

5.3 A Unifying Framework for Causal Imitation Learning

In this section, we introduce a unifying framework for causal IL in the presence of hidden confounders and formally state the problem setting. We then show in Section 5.3.3 that this framework subsumes and recovers the specific settings considered in prior work.

5.3.1 MDPs with Hidden Confounders.

We begin by introducing a Markov decision process (MDP) framework with hidden confounders, $(\mathcal{S}, \mathcal{A}, \mathcal{U}, \mathcal{P}, r, \mu_0, T)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, and $\mathcal{U}$ is the confounder space. Importantly, parts of the hidden confounder u_t at time t may be available to the expert but not to the imitator due to imperfect data logging or expert knowledge. We model this by segmenting the hidden confounder at time t into two parts $u_t = (u_t^o, u_t^\epsilon)$, where u_t^o is observable to the expert and u_t^ϵ is not. Intuitively, u_t^o corresponds to the additional information that only the expert observes, and u_t^ϵ acts as confounding noise in the environment that affects both the state and action.[2]

As a result, the transition function $\mathcal{P}(\cdot \mid s_t, a_a t, (u_t^o, u_t^\epsilon))$ at time t depends on both hidden confounders, but the reward function $r(s_t, a_t, u_t^o)$ only depends on the state, action, and the observable confounder u_t^o since the confounding noise only directly affects the state and actions. Finally, μ_0 is the initial state distribution, and T is the time horizon. A causal graph illustrating these relationships is provided in Figure 5.1. This nuanced distinction between u_t^o and u_t^ϵ is crucial for determining the appropriate method for IL, and we begin with an example to motivate our setting and illustrate the importance of considering $u_t = (u_t^o, u_t^\epsilon)$.

Example 5.1. Consider an airline ticket pricing scenario (Wright, 1928), where the goal is to learn a pricing policy by imitating actual airline pricing based on expert-set profit margins. Suppose that seasonal patterns and external events are known only to experts,

[2] In our framework, we allow the actual actions taken in the environment to be affected by the noise. Noise that only perturbs data records can be considered as a special case of our framework.

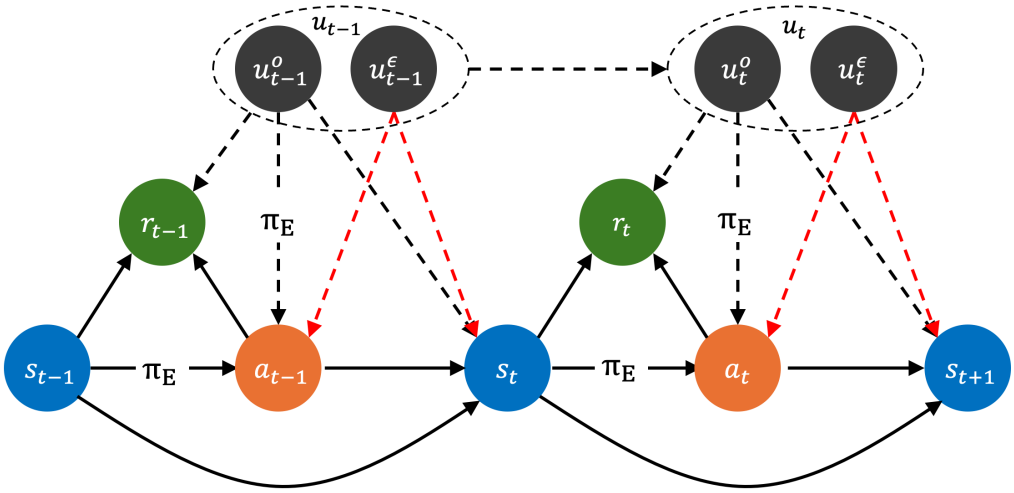


Figure 5.1: A causal graph of MDPs with hidden confounders $u_t = (u^o_t, u^\epsilon_t)$. The black dotted lines represent the causal effect of the expert-observable confounder u^o_t, which directly affects the expert action a_t. It also directly affects s_{t+1} and r_t. The red dotted lines represent the causal effect of the expert-unobservable u^ϵ_t, which acts as confounding noise and directly affects the states and actions. u^ϵ_t does not directly affect r_t (following Swamy et al. (2022b)) because the expert policy does not take u^ϵ_t into account, and letting u^ϵ_t directly affect r_t would only add noise to the expected return.

but are absent from the dataset. Hence, these latent variables serve as expert-observable confounders u^o_t. Meanwhile, actual airline prices are confounded (additively) by fluctuating operating costs, which are unknown to the experts when they set the profit margin and are not included in the dataset. Consequently, such fluctuating operating costs act as confounding noise u^ϵ_t. We conduct experiments in a toy environment inspired by this example in Section 5.5, and demonstrate that IL algorithms that do *not* distinguish between u^o_t and u^ϵ_t fail to correctly imitate the expert.

5.3.2 Causal Imitation Learning in Our Framework

We assume that an expert is demonstrating a task following some expert policy π_E (which we will specify later) and we observe a set of $N \geq 1$ expert demonstrations $\{d_1, d_2, ..., d_N\}$. Each demonstration is a state-action trajectory $(s_1, a_1, ..., s_T, a_T)$, where, at each time step t, we observe the state s_t and the action a_t taken in the environment. The next state is then

sampled from the transition function $\mathcal{P}(\cdot \mid s_t, a_t, (u_t^o, u_t^\epsilon))$.

Let $h_t = (s_1, a_1, ..., s_{t-1}, a_{t-1}, s_t) \in \mathcal{H}$ denote the trajectory history at time t, where $\mathcal{H} \subseteq \bigcup_{i=0}^{T-1} (\mathcal{S} \times \mathcal{A})^i \times \mathcal{S}$ is the set of all possible trajectory histories. Importantly, we observe neither the reward nor the confounders (u_t^o, u_t^ϵ) at time t. Given the observed trajectories, our goal is to learn a history-dependent policy $\pi_h : \mathcal{H} \to \Delta(A)$, where we assume that our policy class Π is convex and compact. The Q-function of a policy $\pi_h \in \Pi$ is given by $Q_{\pi_h}(s_t, a_t, u_t^o) = \mathbb{E}_{\tau \sim \pi_h}[\sum_{t'=t}^{T} r(s_{t'}, a_{t'}, u_{t'}^o)]$, the value of a policy is $J(\pi_h) = \mathbb{E}_{\tau \sim \pi_h}[\sum_{t'=1}^{T} r(s_{t'}, a_{t'}, u_{t'}^o)]$, and the imitation gap between π_h and the expert π_E is defined as $J(\pi_E) - J(\pi_h)$.

In order to learn a policy π_h that matches the performance of π_E, we need to break the spurious correlation between states and expert actions by inferring what the expert would do if we intervened and placed them in state s_t when observing u_t^o. However, the causal inference literature (Shpitser & Pearl, 2008) tells us that, without further assumptions, it is generally impossible to identify π_E. To determine the minimal assumptions that allow π_E to be identifiable, we first observe that u_t^ϵ can be correlated for all time steps t, making it impossible to distinguish between the intended actions of the expert and the confounding noise. However, in practice, the confounding noise at far-apart time steps is often independent. For example, the effect of the confounding noise u_t^ϵ at time t on future states and actions often diminishes over time. This is typically the case for random environment noise such as wind, which acts as confounding noise u_t^ϵ. In addition, when the confounding noise u_t^ϵ at time t becomes observable at a future time t', e.g., previous operating costs are eventually observed as in Example 5.1, the unobservable confounding noise at times t and t' become independent. If this condition holds, it becomes possible to decouple the spurious correlation between the state and action pairs. Motivated by this observation, we formalise the following assumption on the *confounding noise horizon* k.

Assumption 5.2 (Confounding noise horizon)**.** For every t, the confounding noise u_t^ϵ has a

horizon of k where $1 \leq k < T$. More formally, $u_t^\epsilon \perp\!\!\!\perp u_{t-k}^\epsilon \; \forall t > k$.

We also assume that the confounding noise is additive to the action, which is standard in causal inference (Pearl, 2000; Shao et al., 2024). Without this assumption, the causal effect becomes unidentifiable (see, e.g., (Balke & Pearl, 1994)) and the best we can do is to upper/lower bound it.

Assumption 5.3 (Additive noise)**.** The structural equation that generates the actions in the observed trajectories is

$$a_t = \pi_E(s_t, u_t^o) + u_t^\epsilon, \tag{5.1}$$

where w.l.o.g. $\mathbb{E}[u_t^\epsilon] = 0$ as nonzero expectation of u_t^ϵ can be included as a constant in π_E.

In Section 5.4, we show that, with the above two assumptions, it becomes possible to identify the true causal relationship between states and expert actions, and to imitate π_E.

5.3.3 Reducing Our Framework to Prior Work

In this section, we discuss how the various previous works can be obtained as special cases of our unifying framework.

5.3.3.1 Temporally Correlated Noise

The temporally correlated noise (TCN) proposed in Swamy et al. (2022b) is a special case of our setting where $u^o = 0$ and only the confounding noise u^ϵ is present. Following Equation 14-17 of Swamy et al. (2022b), their setting can be summarised as

$$\begin{aligned} s_t &= \mathcal{T}(s_{t-1}, a_{t-1}) \\ &= \mathcal{T}(s_{t-1}, \pi_E(s_{t-1}) + u_{t-1} + u_{t-2}); \\ a_t &= \pi_E(s_t) + u_t + u_{t-1}, \end{aligned}$$

where $\mathcal{T}$ is the transition function and u_t represents the TCN. It can be seen that TCN acts as the confounding noise u^ϵ since the expert policy does not take it into account, and it affects (or confounds) both the state and action. Therefore, this is a special case of our framework when $u_t^o = 0$, where $a_t = \pi_E(s_t) + \epsilon(u_t^\epsilon)$ from Equation (5.1), and more specifically when the confounding noise horizon $k = 2$. In addition, the theoretical results in Swamy et al. (2022b) can be deduced from our main results, as shown later in Corollary 5.10.

5.3.3.2 Unobserved Contexts

The setting considered by Swamy et al. (2022a) is a special case of our setting when $u^\epsilon = 0$ and only u^o are present. Following Section 3 of Swamy et al. (2022a), their setting can be summarised as

$$\begin{aligned}
&\mathcal{T} : \mathcal{S} \times \mathcal{A} \times C \to D(S);\\
&\nabla : \mathcal{S} \times \mathcal{A} \times C \to [-1, 1];\\
&a_t = \pi_E(s_t, c),
\end{aligned}$$

where $c \in C$ is the context, which is assumed to be fixed throughout an episode. There are no hidden confounders in this setting and the context c is included in u^o under our framework. In addition, our framework allows u^o to vary throughout an episode. Furthermore, the theoretical results in Swamy et al. (2022a) can be deduced from our main results, as shown later in Corollary 5.9.

5.3.3.3 Imitation Learning with Latent Confounders

The setting considered by Vuorio et al. (2022) is also a special case of our setting when $u^\epsilon = 0$ and only u^o are present, which is very similar to Swamy et al. (2022a). In Section 2.2 of Vuorio et al. (2022), they introduced a latent variable $\theta \in \Theta$ that is fixed throughout an episode and $a_t = \pi_E(s_t, \theta)$. Therefore, the latent variable θ is included in u^o in our

framework. In Vuorio et al. (2022), no theoretical imitation gap bounds are provided. However, Corollary 5.9 can be applied directly to their setting to bound their imitation gap.

5.3.3.4 Causal Delusion and Confusion

The concepts of *causal delusion* (Ortega et al., 2021) and *causal confusion* have been extensively studied from various perspectives (de Haan et al., 2019; Pfrommer et al., 2023; Spencer et al., 2021; Wen et al., 2020). A classic example arises in autonomous driving: when the states include dashboard and road images, the brake indicator light acts as a confounding variable since it correlates with the braking action in subsequent steps. As a result, the imitator might incorrectly learn to brake whenever the brake indicator light is already on, effectively latching onto spurious correlations between the current action and the trajectory history.

In Ortega et al. (2021), such causal delusion is explicitly modelled through hidden confounders, which correspond to the confounding noise u^ϵ in our framework. Other works (de Haan et al., 2019; Pfrommer et al., 2023; Spencer et al., 2021; Wen et al., 2020) do not explicitly model unobserved confounders, but the spurious correlation between previous states and actions can be modelled as the existence of confounding noise u^o in our framework. In the autonomous driving example, the actual road hazard serves as u^o that simultaneously causes the expert to brake and activates the brake indicator light.

However, while these settings can be viewed as special cases of our framework, we stress that the practical problems studied in de Haan et al. (2019); Pfrommer et al. (2023); Spencer et al. (2021); Wen et al. (2020) differ from ours. In particular, they implicitly assume that the hidden confounders are either embedded in the observations or directly observable, and focus on designing practical algorithms that identify and exploit such confounders to improve IL performance.

5.4 Causal IL as a CMR Problem

In this section, we demonstrate that performing causal IL in our framework is possible using trajectory histories as instruments. We show that the problem can be reformulated as a CMR problem and propose an effective algorithm to solve it.

The typical target for IL would be the expert policy π_E itself. However, since the expert has access to privileged information, namely u_t^o, which the imitator does not, the best thing an imitator can do is to learn a history-dependent policy π_h that is the closest to the expert. A natural choice for a learning objective is the conditional expectation of $\pi_E(s_t, u_t^o)$ on the history h_t:

$$\pi_h(h_t) \coloneqq \mathbb{E}_{\mathbb{P}(u_t^o|h_t)}[\pi_E(s_t, u_t^o)] = \mathbb{E}[\pi_E(s_t, u_t^o) \mid h_t],$$

because the conditional expectation minimises the least squares criterion (Hastie et al., 2001) and π_h is the best predictor of π_E given h_t. In π_h, the distribution $\mathbb{P}(u_t^o \mid h_t)$ captures the information about u_t^o that can be inferred from trajectory histories.

Remark 5.4. *Learning π_h is not trivial. Policies learnt naively using behaviour cloning (i.e., $\mathbb{E}[a_t \mid h_t]$) fail to match π_E. To see this, note that, in view of Equation* (5.1)*, we have*

$$\begin{aligned}\mathbb{E}[a_t \mid h_t] &= \mathbb{E}[\pi_E(s_t, u_t^o) \mid h_t] + \mathbb{E}[u_t^\epsilon \mid h_t] \\ &= \pi_h(h_t) + \mathbb{E}[u_t^\epsilon \mid h_t], \end{aligned} \tag{5.2}$$

where $\mathbb{E}[u_t^\epsilon \mid h_t] \neq 0$ due to the spurious correlation between u_t^ϵ and the trajectory history h_t. As a result, $\mathbb{E}[a_t \mid h_t]$ becomes biased, which can lead to arbitrarily worse performance compared to π_E.

5.4.1 Derivation of the CMR Problem

Leveraging the confounding horizon from Assumption 5.2, we are able to break the spurious correlation using the independence of u_t^ϵ and u_{t-k}^ϵ. We propose to use the k-step trajectory history $h_{t-k} = (s_1, a_1, ..., s_{t-k})$ as an instrument for the current state s_t. Taking the expectation conditional on h_{t-k} for both sides of Equation (5.2) yields:

$$\begin{aligned}
\mathbb{E}[a_t \mid h_{t-k}] = \mathbb{E}\left[\mathbb{E}[a_t \mid h_t] \mid h_{t-k}\right] &= \mathbb{E}[\pi_h(h_t) \mid h_{t-k}] + \mathbb{E}[\mathbb{E}[u_t^\epsilon \mid h_t] \mid h_{t-k}] \\
&= \mathbb{E}[\pi_h(h_t) \mid h_{t-k}] + \mathbb{E}[u_t^\epsilon \mid h_{t-k}] \\
&= \mathbb{E}[\pi_h(h_t) \mid h_{t-k}] + \mathbb{E}[u_t^\epsilon] = \mathbb{E}[\pi_h(h_t) \mid h_{t-k}],
\end{aligned}$$

where we use the fact that h_{t-k} is $\sigma(h_t)$-measurable because $h_{t-k} \subseteq h_t$, $u_t^\epsilon \perp\!\!\!\perp u_{t-k}^\epsilon$ and $\mathbb{E}[u_t^\epsilon] = 0$ by Assumption 5.2. As a result, the problem of learning π_h reduces to solving for π_h that satisfies the following identity:

$$\mathbb{E}[a_t - \pi_h(h_t) \mid h_{t-k}] = 0, \tag{5.3}$$

which is a CMR problem as defined in Section 2.2.2. In this case, both a_t and h_t are observed in the confounded expert demonstrations, and h_{t-k} acts as the instrument.

To ensure that the instrument h_{t-k} is valid, we verify the three conditions of IV in Section 2.2.3: firstly, $u_t^\epsilon \perp\!\!\!\perp h_{t-k}$ as explained above. Secondly, the environment and expert policy are nontrivial since $\mathbb{P}(h_t \mid h_{t-k})$ is not constant in h_{t-k}. Finally, h_{t-k} affects a_t only through s_t by the Markov property. However, the strength of the instrument h_{t-k}, representing its correlation with h_t, influences how well $\pi_h(h_t)$ can be identified by solving the CMR in Equation (5.3). As the confounding horizon k increases, this correlation weakens, making h_{t-k} a less effective instrument. This relationship is formally analysed in Proposition 5.6 and validated in experiments in Section 5.5.

Algorithm 3 DML-IL

1: **Input:** Dataset $\mathcal{D}_E$ of expert demonstrations, confounding noise horizon k
2: Initialise the roll-out model $\widehat{M}$ as a Gaussian mixture model
3: **repeat**
4: Sample (h_t, a_t) from data $\mathcal{D}_E$
5: Fit the roll-out model $(h_t, a_t) \sim \widehat{M}(h_{t-k})$ to maximise the log likelihood
6: **until** convergence
7: Initialise the expert model $\widehat{\pi}_h$ as a neural network
8: **repeat**
9: Sample h_{t-k} from $\mathcal{D}_E$
10: Generate $\widehat{h}_t$ and $\widehat{a}_t$ using the roll-out model $\widehat{M}$
11: Update $\widehat{\pi}_h$ to minimise the loss $\ell := \|\widehat{a}_t - \widehat{\pi}_h(\widehat{h}_t)\|_2$
12: **until** convergence
13: **Output:** A history-dependent imitator policy $\widehat{\pi}_h$

5.4.2 Practical Algorithms for Solving the CMR Problem

There are various techniques (Bennett et al., 2019; Xu et al., 2020; Shao et al., 2024) for solving the CMR $\mathbb{E}[a_t|h_{t-k}] = \mathbb{E}[\pi_h(h_t)|h_{t-k}]$. Here, the *CMR error* that we minimise is given by

$$\sqrt{\mathbb{E}\big[\mathbb{E}[a_t - \widehat{\pi}_h(h_t)|h_{t-k}]^2\big]} = \|\mathbb{E}[a_t - \widehat{\pi}_h(h_t)|h_{t-k}]\|_2.$$

In Algorithm 3, we introduce DML-IL, an algorithm that applies our CMR estimator DML-CMR, proposed in Chapter 4, to solve the CMR problem we derived for causal IL. The first part of the algorithm (lines 3-7) learns a Gaussian mixture roll-out model $\widehat{M}$ that predicts the transition dynamics and generates a trajectory k steps ahead given h_{t-k}. Then, the DNN-based expert model $\widehat{\pi}_h$ takes the generated trajectory $\widehat{h}_t$ from $\widehat{M}(h_{t-k})$ as input and minimises the mean square error to the next action $\widehat{a}_t$, that is, $\ell := \|\widehat{a}_t - \widehat{\pi}_h(\widehat{h}_t)\|_2$, until convergence (lines 8-13). Using generated trajectories $\widehat{h}_t$ is crucial in breaking the spurious correlation caused by u_t^ϵ, and the trajectory history before h_{t-k} allows the imitator to infer information about u_t^o.

Note that Algorithm 3 requires the confounding noise horizon k as input. Although the

Algorithm 4 DML-IL with K-fold cross-fitting

Input: Dataset $\mathcal{D}_E$ of expert demonstrations, Confounding noise horizon k, number of folds K for cross-fitting
Get a partition $(I_i)_{i=1}^K$ of dataset indices $[N]$ of trajectories
for $i = 1$ **to** K **do**
  $I_i^c := [N] \setminus I_i$
  Initialise the roll-out model $\widehat{M_i}$ as a mixture of Gaussians model
  repeat
    Sample (h_t, a_t) from data $\{(\mathcal{D}_{E,j}) : j \in I_i^c\}$
    Fit the roll-out model $(h_t, a_t) \sim \widehat{M_i}(h_{t-k})$ to maximise log likelihood
  until convergence
end for
Initialise the expert model $\widehat{\pi}_h$ as a neural network
repeat
  for $i = 1$ **to** K **do**
    Sample h_{t-k} from $\{(\mathcal{D}_{E,j}) : j \in I_i\}$
    Generate $\widehat{h_t}$ and $\widehat{a}_t$ using the roll-out model $\widehat{M_i}$
    Update $\widehat{\pi}_h$ to minimise the loss $\ell := \|\widehat{a}_t - \widehat{\pi}_h(\widehat{h_t})\|_2$
  end for
until convergence
Output: A history-dependent imitator policy $\widehat{\pi}_h$

exact value of k can be difficult to obtain in reality, any upper bound $\bar{k}$ of k is sufficient to guarantee the correctness of Algorithm 3, since $h_{t-\bar{k}}$ is also a valid instrument. Ideally, we would like a data-driven approach to determine k. However, it is generally intractable to empirically verify whether h_{t-k} is a valid instrument from a static dataset, especially the unconfounded instrument condition (i.e., $h_{t-k} \perp\!\!\!\perp u_t^e$). Therefore, we rely on the user to provide a reasonable choice of $\bar{k}$ based on the environment that does not substantially overestimate k.

DML-IL can also be implemented with K-fold cross-fitting (recall Algorithm 1, Chapter 4) to reduce estimation bias and improve training stability. As shown in Algorithm 4, the expert dataset $\mathcal{D}_E$ with indices $[N]$ is partitioned into K equal-sized folds $\{I_i\}_{i=1}^K$. For each fold $i \in [1..K]$, we use the remaining folds with indices $I_i^c := [N] \setminus I_i$ to train a roll-out model $\widehat{M_i}$. Then, a single expert model $\widehat{\pi}_h$ is trained using samples from each fold I_i, where the trajectory rollouts $\widehat{h_t}$ are generated using the corresponding roll-out model $\widehat{M_i}$,

by minimising the same mean squared error objective as in Algorithm 3. The CMR estimator we use—DML-CMR with K-fold cross-fitting—is theoretically shown in Chapter 4 to converge at the rate of $O(N^{-1/2})$ under standard regularity conditions. Consequently, DML-IL with K-fold cross-fitting will also inherit this theoretical guarantee.

5.4.3 Theoretical Analysis

In this section, we derive theoretical guarantees for our algorithm, focusing on the imitation gap and its relationship to existing work. All proofs in this section are deferred to Appendix B.1.

On a high level, in order to bound the imitation gap of the learnt policy $\widehat{\pi}_h$, i.e., $J(\pi_E) - J(\widehat{\pi}_h)$, we need to control:

(i) The amount of information about the hidden confounders that can be inferred from trajectory histories;

(ii) The ill-posedness (or identifiability) of the induced CMR problem, which intuitively measures the strength of the instrument h_{t-k};

(iii) The disturbance of the confounding noise to the states and actions at test time.

These factors are all determined by the environment and the expert policy. To control (i), we measure how much information about u_t^o is captured by the trajectory history h_t by analysing the total variation (TV) distance between the distribution of u_t^o and $\mathbb{E}[u_t^o|h_t]$ along the trajectories of π_E. To control (ii) and (iii), we need to introduce the following two key concepts.

Definition 5.5 (The ill-posedness of Confounded IL)**.** Given the derived CMR problem in Equation (5.3), the *ill-posedness* $\nu(\Pi, k)$ of the policy space Π with confounding noise

horizon k is

$$\nu(\Pi, k) = \sup_{\pi \in \Pi} \frac{\|\pi_E - \pi\|_2}{\|\mathbb{E}[a_t - \pi(h_t)|h_{t-k}]\|_2}.$$

The ill-posedness $\nu(\Pi, k)$ measures the strength of the instrument, where a higher $\nu(\Pi, k)$ indicates a weaker instrument. It bounds the ratio between the L_2 error of the imitator to the expert policy, and the learning error of the imitator following our CMR objective.

As discussed previously, intuitively, the strength of the instrument would decrease as the confounding horizon k increases. This is confirmed by the following proposition.

Proposition 5.6 (Monotonicity of the ill-posedness)**.** *$\nu(\Pi, k)$ is monotonically increasing as the confounded horizon k increases.*

Next, we introduce the notion of c-total variation stability for our theoretical analysis.

Definition 5.7 (c-total variation stability (Bassily et al., 2021; Swamy et al., 2022b))**.** Let $P(X)$ be the distribution of a random variable $X : \Omega \to \mathcal{X}$. $P(X)$ is c-TV stable if for $a_1, a_2 \in \mathcal{X}$ and $\Delta > 0$,

$$\|a_1 - a_2\| \leq \Delta \implies \delta_{TV}(a_1 + X, a_2 + X) \leq c\Delta,$$

where $\|\cdot\|$ is some norm defined on $\mathcal{X}$ and δ_{TV} is the TV distance.

A wide range of distributions are c-TV stable. For example, standard normal distributions are $\frac{1}{2}$-TV stable. We apply this notion to the distribution over u_t^ϵ to bound the disturbance it induces in the trajectory and the expected return.

With the notion of ill-posedness and c-TV stability, we can now analyse and upper bound the imitation gap $J(\pi_E) - J(\widehat{\pi}_h)$ by controlling the three previously discussed components (i) – (iii).

Theorem 5.8 (Imitation Gap Bound)**.** *Let $\widehat{\pi}_h$ be the learnt policy with CMR error ϵ and let $\nu(\Pi, k)$ be the ill-posedness of the problem. Assume that $\delta_{TV}(u_t^o, \mathbb{E}_{\pi_E}[u_t^o|h_t]) \leq \delta$ for $\delta \in \mathbb{R}^+$, $P(u_t^\epsilon)$ is c-TV stable and π_E is deterministic. Then, the imitation gap is upper bounded by*

$$J(\pi_E) - J(\widehat{\pi}_h) \leq T^2\big(c\epsilon\nu(\Pi, k) + 2\delta\big) = \mathcal{O}\big(T^2(\delta + \epsilon)\big).$$

Proof sketch. We begin by using the Performance Difference Lemma (Kakade & Langford, 2002) to separate the imitation gap into two parts. Let $Q_{\widehat{\pi_h}}(s_t, a_t, u_t^o)$ be the Q-function of $\widehat{\pi_h}$, then

$$J(\pi_E) - J(\widehat{\pi_h}) = \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\pi_E}[\tilde{Q} - \mathbb{E}_{a\sim\widehat{\pi_h}}[\tilde{Q}]] + \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\pi_E}[Q_{\widehat{\pi_h}} - \tilde{Q} - \mathbb{E}_{a\sim\widehat{\pi_h}}[Q_{\widehat{\pi_h}} - \tilde{Q}]],$$

for any Q-function $\tilde{Q}(h_t, a_t)$ that takes in the trajectory history h_t and action a_t. For this proof, we choose $\tilde{Q}$ as follows:

$$\tilde{Q}(h_t, a_t) \coloneqq Q(s_t, a_t, \mathbb{E}_{\tau\sim\pi_E}[u_t^o|h_t]).$$

The first part can be bounded with the ill-posedness $\nu(\Pi, k)$ and c from the $c-$TV stability of the hidden confounder U. Specifically, $\mathbb{E}_{\tau\sim\pi_E}[\tilde{Q}_{\widehat{\pi_h}} - \mathbb{E}_{a\sim\widehat{\pi_h}}[\tilde{Q}_{\widehat{\pi_h}}]] \leq Tc\epsilon\nu(\Pi, k)$ by first applying the total variation distance bound for expectations of $\tilde{Q}_{\widehat{\pi_h}}$ over different distributions of action a_t and Jensen's inequality.

For the second part, we first bound

$$\begin{aligned}|\mathbb{E}_{\tau\sim\pi_E,\dot{a}\sim\pi}[Q(s_t, \dot{a}, u_t) - \tilde{Q}(h_t, \dot{a})]| &\leq \|\mathbb{E}_{\pi_E,\pi}[Q(s_t, \dot{a}, u_t^o)|u_t^o]\|_\infty \delta_{TV}(u_t^o, \mathbb{E}_{\pi_E}[u_t^o|h_t]) \\ &\leq T \cdot \delta_{TV}(u_t^o, \mathbb{E}_{\pi_E}[u_t^o|h_t]) \leq T\delta\end{aligned}$$

for any action $\dot{a} \sim \pi$ following some policy π (in our case, it can be π_E or $\widehat{\pi_h}$) using

the tower property of expectations and total variation distance bound. Next, we apply this inequality twice to bound $\mathbb{E}_{\tau\sim\pi_E}[Q_{\widehat{\pi_h}} - \tilde{Q}_{\widehat{\pi_h}} - \mathbb{E}_{a\sim\widehat{\pi_h}}[Q_{\widehat{\pi_h}} - \tilde{Q}_{\widehat{\pi_h}}]] \leq 2T\delta$.

Finally, we use the bounds on the two parts of the imitation gap and deduce that

$$J(\pi_E) - J(\widehat{\pi}_h) \leq T \cdot (Tc\epsilon\nu(\Pi, k) + 2T\delta) = T^2(c\epsilon\nu(\Pi, k) + 2\delta) = \mathcal{O}(T^2(\epsilon + \delta)).$$

□

This upper bound scales at the rate of T^2, which aligns with the expected behaviour of imitation learning without an interactive expert (Ross & Bagnell, 2010). Next, we show that the upper bounds on the imitation gap from prior work (Swamy et al., 2022b,a) are special cases of Theorem 5.8. The proofs are deferred to Appendix B.1.3.

Corollary 5.9. *In the special case that $u_t^o = 0$, i.e., there are no expert-observable confounders, or $u_t^o = \mathbb{E}_{\pi_E}[u_t^o|h_t]$, i.e., u_t^o is $\sigma(h_t)$ measurable (all information about u_t^o is contained in the history), the imitation gap is upper bounded by*

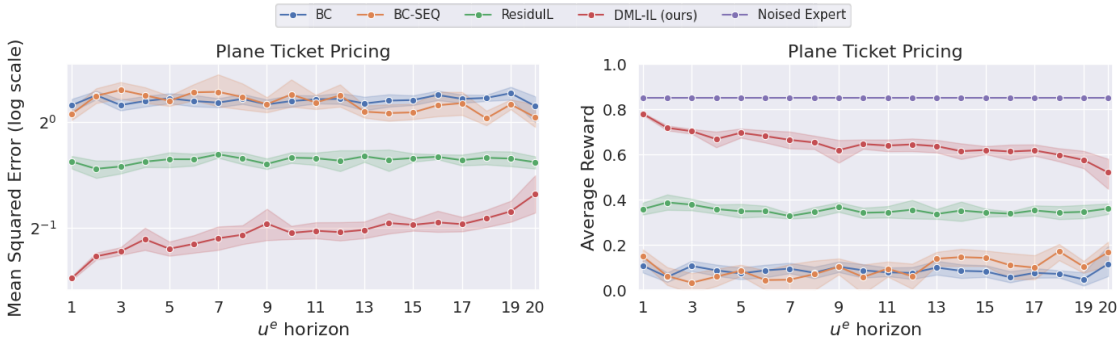
$$J(\pi_E) - J(\widehat{\pi}_h) \leq T^2\big(c\epsilon\nu(\Pi, k)\big) = \mathcal{O}\big(T^2\epsilon\big),$$

which coincides with Theorem 5.1 of Swamy et al. (2022b).

When there are no hidden confounders, i.e., $u_t^\epsilon = 0$, our framework is reduced to that of Swamy et al. (2022a). However, Swamy et al. (2022a) provided an abstract bound that directly uses the supremum of key components in the imitation gap over all possible Q-functions to bound the imitation gap. We further extend and concretise the bound using the learning error ϵ and the TV distance bound δ instead of relying on the supremum.

Corollary 5.10. *In the special case that $u_t^\epsilon = 0$, if the learnt policy has optimisation error ϵ, the imitation gap is upper bounded by*

$$J(\pi_E) - J(\widehat{\pi}_h) \leq T^2\left(\frac{2}{\sqrt{\dim(A)}}\epsilon + 2\delta\right),$$



(a) MSE in log scale, lower is better.

(b) Average reward, higher is better.

Figure 5.2: **Plane Ticket Environment** (Example 5.1): On the left, the MSE between the learnt policy and the expert. On the right, the average reward of our approach and baselines.

which is a concrete bound that extends the abstract bound in Theorem 5.4 of Swamy et al. (2022a).

Remark 5.11. *If both u_t^ϵ and u_t^o are zero, we then recover the classic setting of IL without confounders (Ross & Bagnell, 2010), and the imitation gap bound is $T^2\epsilon$, where ϵ is the optimisation error of the algorithm.*

5.5 Experimental Results

In this section, we empirically evaluate the performance of DML-IL (Algorithm 4) on the toy environment with continuous state and action spaces introduced in Example 5.1 and Mujoco environments: Ant, Half Cheetah, and Hopper. We compare with the following existing methods reviewed in Section 3.1.2: Behavioural Cloning (BC), which naively minimises $\mathbb{E}[-\log \pi(a_t|s_t)]$; BC-SEQ (Swamy et al., 2022a), which learns a history-dependent policy to handle hidden contexts observable to the expert; ResiduIL (Swamy et al., 2022b), which we adapt to our setting by providing h_{t-k} as instruments to learn a history-independent policy; and the noised expert, which is the performance of the expert when put in the confounded environment, and corresponds to the maximally achievable performance. In Section 5.5.3, we also include additional evaluations of using other CMR estimation algo-

rithms, including DFIV (Xu et al., 2020) and DeepGMM (Bennett et al., 2019), as the core CMR estimator, but we found inconsistent and subpar performance.

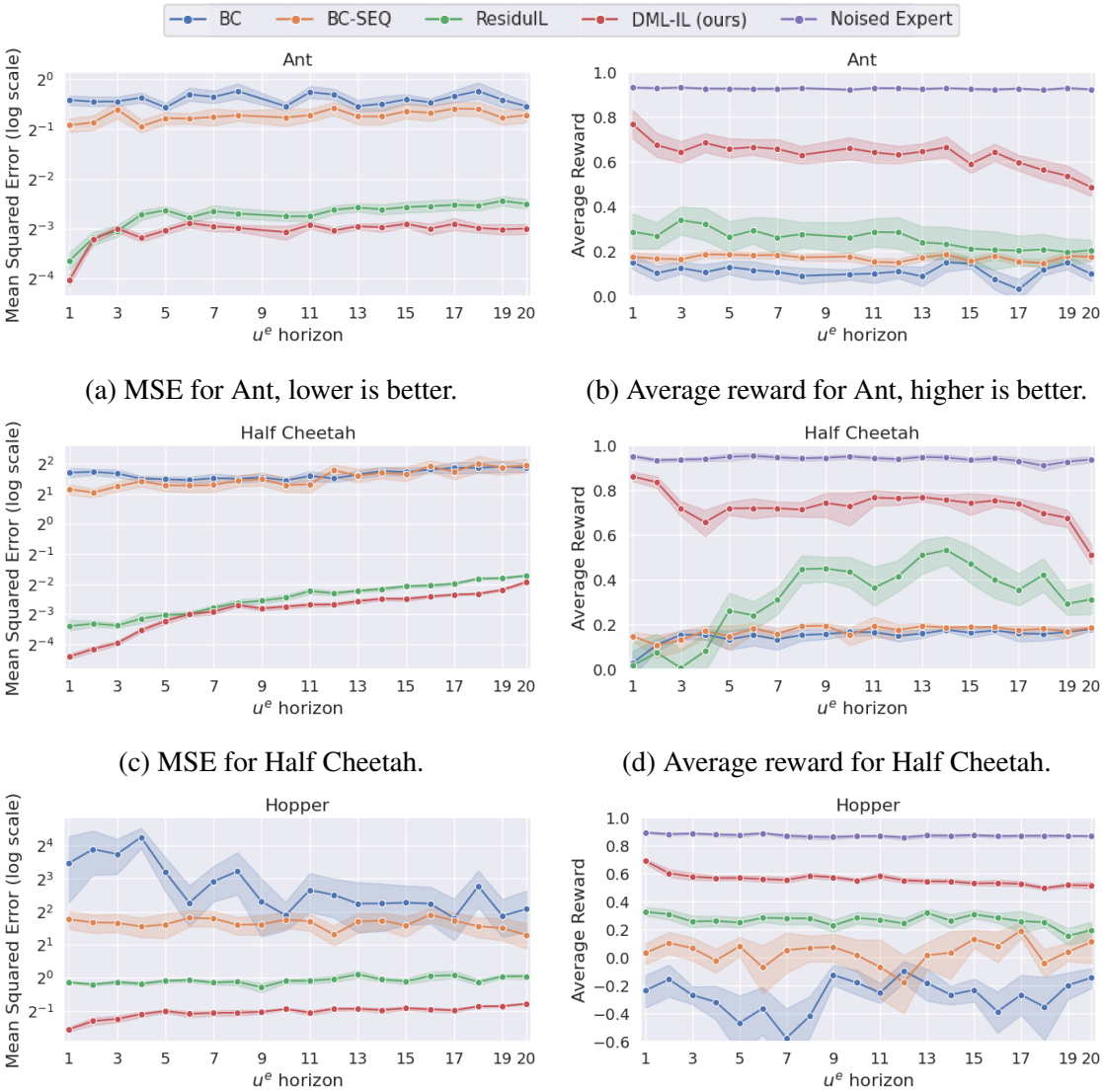
Experimental Setup The experiments are carried out on a Linux server (Ubuntu 18.04.2) with two Intel Xeon Gold 6252 CPUs and six NVIDIA GeForce RTX 2080 Ti GPUs. We train imitators with 20000 samples (40 trajectories of 500 steps each) of the expert trajectory using each algorithm and report the average reward when tested online in their respective environments. Each training run on the toy environment takes around 20 minutes and a few hours on MuJoCo tasks. The reward is scaled such that 1 is the performance of the un-noised expert, and 0 is a random policy. We also report the mean squared error (MSE) between imitator's and expert's actions. The purpose of evaluating the MSE is to assess how well the imitator learnt from the expert, and importantly whether the confounding noise problem is mitigated. When the confounding noise u^ϵ is not handled, we should expect to observe a much higher MSE. All results are plotted with one standard deviation as a shaded area. In addition, we vary the confounding noise horizon k from 1 to 20 in order to increase the difficulty of the problem with weaker instruments h_{t-k}. Further training details, network architecture, and hyperparameters can be found in Appendix B.3. All algorithms are implemented using PyTorch (Paszke et al., 2019), and the code is available on GitHub[3].

5.5.1 Ticket Pricing Environment

We first consider the plane ticket pricing environment described in Example 5.1. The confounding noise u^ϵ are operational costs and u_t^o are seasonal demand patterns and events. We set u_t^o to continuously vary with a rate of change of approximately every 30 steps. Details on this environment are provided in Appendix B.2.1.

The results are presented in Figure 5.2. DML-IL performed the best with the lowest

[3] `https://github.com/shaodaqian/Confounded-IL`



(a) MSE for Ant, lower is better.

(b) Average reward for Ant, higher is better.

(c) MSE for Half Cheetah.

(d) Average reward for Half Cheetah.

(e) MSE for Hopper.

(f) Average reward for Hopper. Note that BC performs worse in Hopper than a random policy.

Figure 5.3: **MuJoCo:** On the left, the MSE between learnt policy and the expert. On the right, the average reward in the MuJoCo environments Ant, Half Cheetah and Hopper.

MSE and the highest average reward that is closest to the expert, especially when the u_t^ϵ horizon is 1. This implies that DML-IL is successful in handling both u_t^ϵ and u_t^o. ResiduIL is able to reduce the confounding effect of u_t^ϵ, evident by the lower MSE compared to the two other methods that do not deal with u_t^ϵ. However, since it does not explicitly consider u_t^o, the imitator has no information on u_t^o, and the best it can do is to assume some average value (or expectation) of u_t^o. Therefore, while ResiduIL still achieves some reward, its

performance gap to DML-IL is due to the lack of consideration of u_t^o. Both BC and BC-SEQ fail completely in the presence of confounding noise u_t^ϵ, with much higher MSE and average reward close to a random policy. From the similar performance of BC-SEQ and BC, we see that the use of trajectory history to infer u_t^o is not helpful when the confounding noise is not handled explicitly. This demonstrates that considering the effect of u_t^ϵ and u_t^o partially is insufficient to learn a well-performing imitator under the general setting.

In addition, as the confounding noise horizon k increases (x-axis), the performance of DML-IL drops. This coincides with the fact that the instrument weakens and less information regarding u_t^o can be captured from h_{t-k} as k increases. When $k = 20$, it can be seen that the performance of DML-IL is close to that of ResiduIL, which does not consider the effect of u_t^o, because very limited information about the current expert-observable confounder u_t^o can be inferred from states 20 steps prior.

5.5.2 Mujoco Environments

In Figure 5.3, we consider the Mujoco tasks. While the original tasks do not have hidden variables, we modify the environment to introduce u_t^ϵ and u_t^o. Specifically, instead of controlling the Ant, Half Cheetah, and Hopper, respectively, to travel as fast as possible, the goal is to control the agent to travel at a target speed that is varying throughout an episode. This target speed is u_t^o, which is observed by the expert but not recorded in the dataset. In addition, we add confounding noise u_t^ϵ to s_t and a_t to mimic the confounding noise such as wind. Additional details about the modifications made to the environments are provided in Appendix B.2.2.

DML-IL outperforms other methods in all three Mujoco environments as shown in Figure 5.3. Similarly to the plane ticket environment, ResiduIL is effective in removing the confounding noise but fails to match the average reward of DML-IL due to the lack of knowledge of u_t^o. BC and BC-SEQ have much higher MSE and fail to learn meaningful policies. As u_t^ϵ horizon increases, the performance of DML-IL drops as expected due to

weaker instruments and limited inferable information regarding u_t^o, especially for the Ant and Half Cheetah tasks.

5.5.3 Adopting Other CMR Estimators

We also experimented with other CMR estimators that have been previously shown to be practical (Shao et al., 2024) for different tasks and with high-dimensional input. Specifically, we experimented with DFIV (Xu et al., 2020), which is an iterative algorithm that integrates the training of two models that depend on each other, and DeepGMM (Bennett et al., 2019), which solves a minimax game by optimising two models adversarially. Note that DeepIV (Hartford et al., 2017) can be considered a special case of DML-CMR, so we did not evaluate it.

The additional results for using DFIV and DeepGMM as the CMR estimator are provided in Figure 5.4 and Figure 5.5. It can be seen from Figure 5.4 that only DFIV achieves good performance in the airline ticket pricing environment, surpassing the performance of ResiduIL. For the Ant Mujoco task in Figure 5.5, both DFIV and DeepGMM fail to learn good policies, with only slightly lower MSE than BC and BC-SEQ. We conjecture that this is mainly due to the high-dimensional state and action spaces and the inherent instability in the DFIV and DeepGMM algorithms. For DFIV, the interleaving of training of two models causes highly nonstationary training targets for both models, and, for DeepGMM, the adversarial training procedure of two models is similar to that of generative adversarial networks (GANs), which are known to be unstable and difficult to train. In addition, when the CMR problem is weakly identifiable, as in the case of a weak instrument, the algorithms may converge to local minima that are far from the true solution in the face of instabilities in the algorithm.

We conclude that solving the CMR problem for an imitator policy can be sensitive to the choice of CMR estimator as well as to the choice of hyperparameters. In addition, some CMR estimators do not work well with high-dimensional input. Our CMR estimator

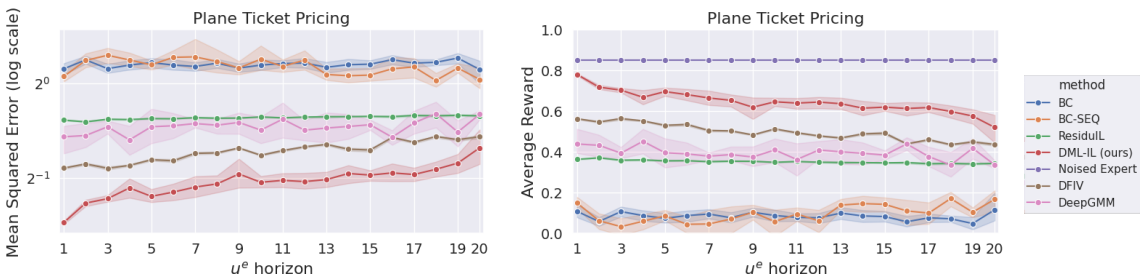


Figure 5.4: **Plane Ticket Environment**: the MSE between learnt policy and expert, and the average reward, with DFIV and DeepGMM as the CMR estimator.

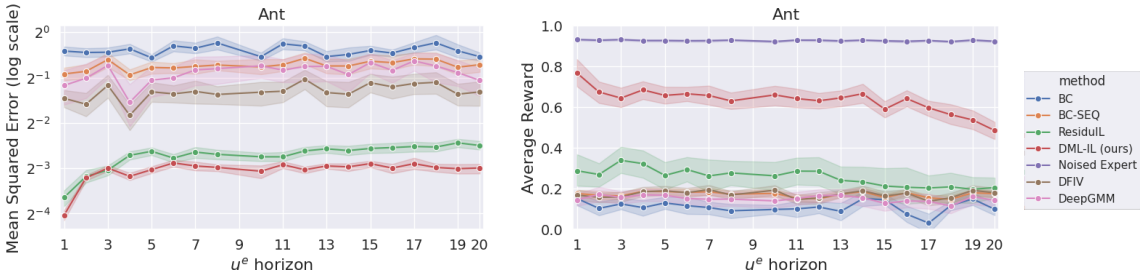


Figure 5.5: **MuJoCo**: the MSE between learnt policy and expert, and the average reward, with DFIV and DeepGMM as the CMR estimator.

of choice, DML-CMR, provides a robust base for the DML-IL algorithm that demonstrates good performance across all tasks and environments. This demonstrates the benefit of using double machine learning (recall Chapter 4), which can debias two-stage estimators and provide good empirical and theoretical convergence.

5.6 Discussion

In this chapter, we considered the problem of IL with hidden confounders that do not assume explicit IVs are observed in the dataset in addition to the trajectory histories. We proposed a unifying framework for confounded IL that unifies confounded IL settings. Specifically, we considered hidden confounders to be partially observable to the expert and demonstrated that causal IL under this framework can be reduced to a CMR problem with the trajectory histories as instruments. We proposed DML-IL, a novel algorithm that solves this CMR problem to learn an imitator policy, and provided bounds on the imitation gap for

the learnt imitator. Finally, we empirically evaluated DML-IL on multiple tasks, including Mujoco environments, and demonstrated state-of-the-art performance against other causal IL algorithms.

One limitation of our proposed framework is the explicit assumptions made in Section 5.3, which are essential for the expert policy to be identifiable. The first assumption is the confounding noise horizon, but, as discussed in Section 5.3, it often holds in practice if the effect of the confounding noise on future states and actions diminishes over time or the confounding noise becomes observable at a future time. The second assumption is additive noise, which is a standard assumption in causal inference (Pearl, 2000; Balke & Pearl, 1994) for the causal effect to be identifiable. This assumption holds for confounding noises such as wind noise or any environmental noise in robotics, and supply costs in econometrics settings such as in our ticket pricing scenario. In addition, we assume knowledge of the confounding noise horizon k or an upper bound on it for Algorithm 3 such that the trajectory history h_{t-k} is a valid IV. However, the value of k or the validity of h_{t-k} as an IV generally cannot be verified empirically, but there exist methods that, under certain assumptions, can identify good candidates for IVs (e.g., Yuan et al. 2021; Wu et al. 2023a Section 6). For DML-IL, we rely on a reasonable choice of k by the user based on the environment, but it would be interesting future work to incorporate methods that test the validity of IVs with increasing k until a good candidate h_{t-k} of IV is found.

Chapter 6

Learning Policies for High-Level Objectives

Contents

6.1 Introduction **116**

6.2 Problem Formulation **118**

6.3 ω-Automata and Limit-Deterministic Büchi Automata **118**

6.4 Methodology **121**

6.4.1 Product MDP 122

6.4.2 LTL Learning with Q-Learning 128

6.4.3 Counterfactual Imagining 129

6.5 Experimental Results **131**

6.5.1 Probabilistic Gate MDP 133

6.5.2 Frozen Lake MDP 133

6.5.3 Office World MDP 133

6.5.4 Runtime Analysis and Sensitivity Analysis 134

6.6 Discussion **135**

6.1 Introduction

Thus far, we have focused on learning optimal and sample-efficient policies that maximise expected reward. In this chapter, we focus on learning policies that satisfy high-level objectives optimally and sample-efficiently using RL. Specifically, we consider the use of linear temporal logic (LTL), which is a temporal logic language that can encode formulae regarding the properties of an infinite sequence of logic propositions, to specify high-level objectives for a system or an agent in the system (recall Section 2.3). Such systems are modelled as Markov decision processes (MDPs), where policies need to be learnt through interactions with the MDP if the states and transitions of the MDP are not known *a priori*.

Reinforcement learning (recall Section 2.1.2.2), a powerful method for policy learning that maximises rewards over time in an unknown environment, is a perfect candidate for learning policies that satisfy LTL specifications. However, as discussed in Chapter 1, using RL to learn a policy that maximises the probability of satisfying LTL specifications is non-trivial (Alur et al., 2022). Most existing work first transforms the LTL specification into an automaton, then builds a product MDP using the original environment MDP and the automaton, on which model-free RL algorithms are applied. However, a crucial obstacle still remains, that is, how to properly define the reward that leads an agent to the optimal satisfaction of the LTL specification with theoretical guarantees. Several algorithms have been proposed (Hahn et al., 2019; Bozkurt et al., 2019; Hasanbeig et al., 2019) for learning LTL specifications with optimality guarantees. However, as discussed in Section 3.3, they face challenges in determining the key hyperparameters that ensure optimality, which in practice can lead to suboptimal policies or inefficient learning. In addition, it remains unclear how to explicitly select these hyperparameters even with certain knowledge regarding the environment MDP. Furthermore, their proposed reward functions are often inflexible, making them ineffective at encouraging agent exploration, which leads to poor sample efficiency.

In this chapter, we propose a novel and more general product MDP with a flexible reward structure and discounting mechanism that, by leveraging model-free reinforcement

learning algorithms, efficiently learns the optimal policy that maximises the probability of satisfying the LTL specification with guarantees. We demonstrate improved theoretical results on the optimality of our product MDP and the reward structure, with a more stringent analysis that yields better bounds on the optimality hyperparameters. Moreover, this analysis sheds light on how to explicitly choose the optimality hyperparameters based on the environment MDP. We also adopt counterfactual imagining that takes advantage of the known high-level LTL specification to further improve the performance of our algorithm. Last but not least, we propose to use the PRISM model checker (Kwiatkowska et al., 2011) to directly evaluate the satisfaction probability of the learnt policies, providing a platform to compare algorithms and tune key hyperparameters. We conduct experiments on several common MDP environments with various challenging LTL tasks and demonstrate the improved sample efficiency and convergence of our methods.

Main Contributions of this Chapter.

- We propose a novel product MDP that incorporates an accepting states counter with a generalised reward structure (Section 6.4.1).
- We propose a novel reinforcement learning algorithm that converges to the optimal policy for satisfying LTL specifications (Section 6.4.2), with theoretical optimality guarantees and theoretical analysis results for choosing key hyperparameters (Theorem 6.10).
- We propose counterfactual imagining (Section 6.4.3), a method to take advantage of the known structure of the LTL specification by creating imagination experiences through counterfactual reasoning.
- We provide a direct evaluation of the proposed algorithms through a novel integration of probabilistic model checkers within the evaluation pipeline, with strong empirical results that demonstrate better sample efficiency and training convergence (Section 6.5).

Work reported in this chapter first appeared in Shao & Kwiatkowska (2023).

6.2 Problem Formulation

Our goal is to formulate a model-free reinforcement learning approach (recall Chapter 2) to efficiently learn an optimal policy that maximises the probability of satisfying an LTL specification with guarantees.

Given an MDP $\mathcal{M} = (\mathcal{S}, s_0, \mathcal{A}, \mathcal{P}, \mathcal{AP}, L, r, \gamma)$ with unknown states and transitions and an LTL objective φ, for any policy π of the MDP $\mathcal{M}$, let $\mathbb{P}_\pi(s \models \varphi)$ denote the probability of paths from state s following π that satisfy the LTL formula φ:

$$\mathbb{P}_\pi(s \models \varphi) = \mathbb{P}_\pi\{\sigma \in Paths_\pi^{\mathcal{M}} \mid \sigma[0] = s, \sigma \models \varphi\}.$$

Then, we would like to design a model-free RL algorithm that learns a deterministic memoryless optimal policy π_φ that maximises the probability of $\mathcal{M}$ satisfying φ:

$$\mathbb{P}_{\pi_\varphi}(s \models \varphi) = \max_\pi \mathbb{P}_\pi(s \models \varphi) \quad \forall s \in \mathcal{S}.$$

6.3 ω-Automata and Limit-Deterministic Büchi Automata

We first transform the LTL specifications into automata. The most common type of automaton is the finite automaton, which is mathematically equivalent to regular expressions and accepts only finite sequences. However, LTL is strictly more expressive than regular expressions because it can describe properties over infinite sequences, which regular expressions cannot. Fortunately, there exists a more expressive class of automata, called ω-automata, that can be used to represent LTL specifications. Unlike finite automata, ω-automata can process infinite sequences and use specific accepting conditions to determine whether an infinite sequence is accepted. The most common ω-automata include Rabin automata (Rabin, 1969) and Büchi automata (Büchi, 1966), which we will introduce next.

Definition 6.1 (DRA (Rabin, 1969))**.** A deterministic Rabin automaton (DRA) is defined as

a tuple: $\mathcal{A} = (\mathcal{AP}, \mathcal{Q}, q_0, \Delta, \mathcal{F})$, where $\mathcal{AP}$ is the set of atomic propositions, $\mathcal{Q}$ is a finite set of states, $q_0 \in \mathcal{Q}$ is the initial state, and $\mathcal{F} = \{(F_i, G_i)\}_{i=1}^{k}$ is the Rabin acceptance condition, where for each i, $F_i \subseteq \mathcal{Q}$ is a set of states that must be visited infinitely often, and $G_i \subseteq \mathcal{Q}$ is a set of states that must only be visited finitely often. Let $\Sigma = 2^{\mathcal{AP}}$ be a finite alphabet, and the deterministic transition function is given by $\Delta : \mathcal{Q} \times \Sigma \to \mathcal{Q}$.

An infinite word $\mathcal{W} \in \Sigma^\omega$, where Σ^ω is the set of all infinite words over the alphabet Σ, is accepted by a DRA $\mathcal{A}$ if and only if there exists an infinite automaton run (sequence of states) $\theta \in \mathcal{Q}^\omega$ from q_0, where $\theta[t+1] = \Delta(\theta[t], \mathcal{W}[t]), \forall t \geq 0$, such that there exists at least one pair (F_i, G_i) where some state in F_i appears infinitely often and all states in G_i appear only finitely often.

Definition 6.2 (NBA (Büchi, 1966))**.** A nondeterministic Büchi automaton (NBA), or simply a Büchi automaton, is defined as a tuple $\mathcal{A} = (\mathcal{AP}, \mathcal{Q}, q_0, \Delta, \mathcal{F})$, where $\mathcal{AP}$ is the set of atomic propositions, $\mathcal{Q}$ is a finite set of states, $q_0 \in \mathcal{Q}$ is the initial state, and $\mathcal{F} \subseteq \mathcal{Q}$ is the set of accepting states that must be visited infinitely often. Let $\Sigma = 2^{\mathcal{AP}} \cup \{\epsilon\}$ be a finite alphabet, where ϵ handles the nondeterministic transitions. The nondeterministic transition function is given by $\Delta : \mathcal{Q} \times \Sigma \to 2^{\mathcal{Q}}$, which allows the automaton to transition to multiple states due to the inherent nondeterminism.

An infinite word $\mathcal{W} \in \Sigma^\omega$, where Σ^ω is the set of all infinite words over the alphabet Σ, is accepted by a Büchi automaton $\mathcal{A}$ if and only if there exists an infinite automaton run $\theta \in \mathcal{Q}^\omega$ from q_0, where $\theta[t+1] \in \Delta(\theta[t], \mathcal{W}[t]), \forall t \geq 0$, such that $\inf(\theta) \cap \mathcal{F} \neq \varnothing$, where $\inf(\theta)$ is the set of automaton states that are visited infinitely often in the run θ. This means that for an accepting infinite word $\mathcal{W}$, by following the transition function of the NBA, there must exist an infinite automaton run in which at least one of the accepting states is visited infinitely often. Since NBA is nondeterministic, an infinite word $\mathcal{W}$ can produce many infinite automaton runs following the transition function. However, the accepting condition only requires the existence of one such automaton run to visit some accepting states infinitely often.

Both DRA and NBA can be used to translate from LTL specifications, but they each present challenges when applied in reinforcement learning contexts. For DRA, the accepting condition is notably complex, encompassing a set of states that must be visited infinitely often alongside another set of states that must be visited only finitely often. This makes it difficult, if not impossible (Hahn et al., 2022), to translate the Rabin accepting condition into a reward that the RL agent maximises. For NBA, the accepting condition is simple, but the nondeterminism can significantly increase branching in the search space when identifying an accepting run. This leads to inefficiencies when used to build a product MDP for RL.

In this chapter, we adopt a variant of Büchi automata known as limit-deterministic Büchi automata (LDBA) (Sickert et al., 2016), which has the same accepting condition as a standard Büchi automaton but with limited nondeterminism. Specifically, through the construction of an LDBA, the nondeterminism is restricted to an initial component $\mathcal{Q}_{\mathcal{N}}$ of the automaton, and after reaching any accepting state, the automaton becomes deterministic.

Definition 6.3 (LDBA)**.** A Büchi automaton is limit-deterministic if $\mathcal{Q}$ can be partitioned into a deterministic set and a nondeterministic set, that is, $\mathcal{Q} = \mathcal{Q}_{\mathcal{N}} \cup \mathcal{Q}_{\mathcal{D}}$, where $\mathcal{Q}_{\mathcal{N}} \cap \mathcal{Q}_{\mathcal{D}} = \varnothing$, such that

1. $\mathcal{F} \subseteq \mathcal{Q}_{\mathcal{D}}$ and $q_0 \in \mathcal{Q}_{\mathcal{N}}$;

2. $|\Delta(q, \nu)| \leq 1$ for all $q \in \mathcal{Q}_{\mathcal{N}}$ and $\nu \neq \epsilon \in \Sigma$;

3. $\Delta(q, \nu) \subseteq \mathcal{Q}_{\mathcal{D}}$ and $|\Delta(q, \nu)| \leq 1$ for all $q \in \mathcal{Q}_{\mathcal{D}}$ and $\nu \in \Sigma$;

An LDBA starts in a nondeterministic initial component and then transitions into a deterministic accepting component $\mathcal{Q}_{\mathcal{D}}$ through ϵ-transitions after reaching an accepting state, where all transitions after this point are deterministic. We follow the formulation of Bozkurt et al. (2019) to extend the alphabets with Σ an ϵ-transition as defined in Definition 6.2 that handles all nondeterminism, which means only ϵ can transition the automaton

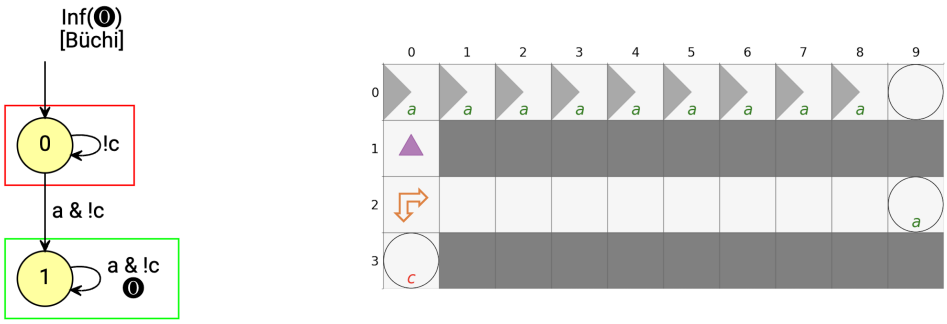


Figure 6.1: An LDBA for "FG a & G !c" (left), and a probabilistic gate MDP (right) motivating the K counter (see Example 6.5).

state to more than one state: $|\Delta(q, \epsilon)| > 1$. This allows the MDP to synchronise with the automaton, which we will discuss in detail in Section 6.4.1.

LDBAs are as expressive as the LTL language, and the satisfaction of any given LTL specification φ can be evaluated on the LDBA derived from φ. We use Rabinizer 4 (Křetínský et al., 2018) to transform LTL formulae into LDBAs. In Figure 6.1 (left) we present an example of the LDBA derived from the LTL formula "FG a & G !c", where state 1 is the accepting state. LDBAs differ from reward machines (Icarte et al., 2022) in that they can express properties satisfiable by infinite paths, which are strictly more expressive than reward machines, and they have different accepting conditions (see Section 2.3.1 for details).

6.4 Methodology

We now provide an overview of our method. Firstly, we transform the LTL objective φ into a limit-deterministic Büchi automaton. Then, we introduce a novel product MDP in Section 6.4.1 and define a generalised reward structure over it, which serves as a general framework that can be instantiated into specific reward functions. With this reward structure, we propose a Q-learning algorithm that adopts a collapsed Q function to learn the optimal policy with optimality guarantees. Lastly, we enhance our algorithm with counterfactual imagining in Section 6.4.3, which takes advantage of the automaton structure to improve performance while maintaining optimality.

6.4.1 Product MDP

In this section, we propose a novel product MDP of the environment MDP, an LDBA, and an integer counter, where the transitions for each component are synchronised. Contrary to the standard product MDP used in the literature (Bozkurt et al., 2019; Hahn et al., 2019; Hasanbeig et al., 2020), this novel product MDP incorporates a counter that counts the number of accepting states visited by paths starting at the initial state.

Definition 6.4 (Product MDP)**.** Given an MDP $\mathcal{M} = (\mathcal{S}, s_0, \mathcal{A}, \mathcal{P}, \mathcal{AP}, L, r, \gamma)$, an LDBA $\mathcal{A} = (\mathcal{AP}, \mathcal{Q}, q_0, \Delta, \mathcal{F})$ and $K \in \mathbb{N}$, we construct the product MDP as follows:

$$\mathcal{M}^\times = \mathcal{M} \times \mathcal{A} \times [0..K] = (S^\times, s_0^\times, A^\times, T^\times, \mathcal{F}^\times, r^\times, \gamma^\times),$$

where the product states $S^\times = \mathcal{S} \times \mathcal{Q} \times [0..K]$, the initial state $s_0^\times = (s_0, q_0, 0)$, the product actions $A^\times = \mathcal{A} \cup \{\epsilon_q \mid q \in \mathcal{Q}\}$, the accepting set $\mathcal{F}^\times = \mathcal{S} \times \mathcal{F} \times [0..K]$, and the product transitions $T^\times : S^\times \times A^\times \times S^\times \to [0, 1]$, which are defined as:

$$T^\times((s, q, n), a, (\hat{s}, \hat{q}, n)) = \begin{cases} \mathcal{P}(s, a, \hat{s}) & \text{if } a \in \mathcal{A} \text{ and } \hat{q} \in \Delta(q, L(s)) \setminus \mathcal{F}; \\ 1 & \text{if } a = \epsilon_{\hat{q}}, \hat{q} \in \Delta(q, \epsilon) \text{ and } \hat{s} = s; \\ 0 & \text{otherwise,} \end{cases}$$

$$T^\times((s, q, n), a, (\hat{s}, \hat{q}, \min(n+1, K))) = \begin{cases} \mathcal{P}(s, a, \hat{s}) & \text{if } a \in \mathcal{A} \text{ and } \hat{q} \in \Delta(q, L(s)) \cap \mathcal{F}; \\ 0 & \text{otherwise,} \end{cases}$$

where all other transitions are equal to 0. The product reward $r^\times : S^\times \times A^\times \times S^\times \to \mathbb{R}$ and the product discount function $\gamma^\times : S^\times \to (0, 1]$ that are suitable for LTL learning are defined later in Definition 6.7.

Furthermore, an infinite path σ of $\mathcal{M}^\times$ satisfies the Büchi condition $\varphi_\mathcal{F}$ if $\inf(\sigma) \cap \mathcal{F}^\times \neq \varnothing$. With a slight abuse of notation, we denote this condition in LTL language as $\sigma \models \mathsf{GF}\ \varphi_\mathcal{F}$, meaning that, for all $M \in \mathbb{N}$, there always exists $s^\times \in \mathcal{F}^\times$ that will be visited

in $\sigma[M :]$.

When an MDP action $a \in \mathcal{A}$ is taken in the product MDP $\mathcal{M}^{\times}$, the alphabet used to transition the LDBA is deduced by applying the label function to the current state of the environment MDP: $L(s) \in 2^{\mathcal{AP}}$. In this case, the LDBA transition $\delta(q, L(s))$ is deterministic. Otherwise, if an ϵ-action $\epsilon_{\hat{q}} \in \{\epsilon_q \mid q \in \mathcal{Q}\}$ is taken, LDBA is transitioned with an ϵ-transition, and the nondeterminism of $\delta(q, \epsilon)$ is resolved by transitioning the automaton state to $\hat{q}$. The K counter value is initially set to 0, and each time an accepting state is reached, the counter value increases by one until it is capped at K.

Example 6.5. We motivate our product MDP structure of Definition 6.4 through an example. In Figure 6.1 (right), we have a grid environment where the agent can decide to move up, down, left, or right. The task is to visit states labelled "a" infinitely often without visiting "c" as described by the LDBA in Figure 6.1 (left). The MDP starts at (1,0), with walls denoted by solid grey squares. The states in the first row only allow action right, as denoted by the right-pointing triangle, which leads to a sink at (0,9). There is also a probabilistic gate at (2,0) that randomly transitions the agent down or right, and if the agent reaches (2,1), the accepting sink at (2,9) becomes reachable. Therefore, the optimal policy is to move down from the start and stay in (2,9) if the probabilistic gate at (2,0) transitions the agent to the right. The probability of satisfying this task is the probability of the gate sending the agent to the right. Intuitively, this environment has some initial accepting states that are easy to explore but lead to non-accepting sinks, whereas the true optimal path requires more exploration. If we set $K = 10$ in the product MDP for this task, we can assign very small rewards to the initially visited accepting states and gradually increase the reward as more accepting states are visited to encourage exploration and guide the agent towards the optimal policy.

Next, we present a theorem which states that the product MDP with Büchi condition $\varphi_{\mathcal{F}}$ is equivalent, in terms of the optimal policy, to the original MDP with LTL specification φ. The proof of this theorem is provided in Appendix C.1.1.

Theorem 6.6 (Satisfiability Equivalence)**.** *For any product MDP $\mathcal{M}^\times$ that is induced from LTL formula φ, we have that*

$$\sup_{\pi} \mathbb{P}_{\pi}(s_0 \models \varphi) = \sup_{\pi^\times} \mathbb{P}_{\pi^\times}(s_0^\times \models \mathsf{GF}\ \varphi_{\mathcal{F}}).$$

Furthermore, a deterministic memoryless policy that maximises the probability of satisfying the Büchi condition $\varphi_{\mathcal{F}}$ on the product MDP $\mathcal{M}^\times$, starting from the initial state, induces a deterministic finite-memory optimal policy that maximises the probability of satisfying φ on the original MDP $\mathcal{M}$ from the initial state.

6.4.1.1 Reward Structure for LTL Learning

We first define a generalised reward structure in $\mathcal{M}^\times$ for learning policies that satisfy LTL specifications, and then prove the equivalence between acquiring the highest expected discounted reward and achieving the highest probability of satisfying $\varphi_{\mathcal{F}}$ using this reward structure.

Definition 6.7 (Reward Structure)**.** Given a product MDP $\mathcal{M}^\times$ and a policy π, the product reward function $r^\times : S^\times \times A^\times \times S^\times \to \mathbb{R}$ is suitable for LTL learning if

$$r^\times((s, q, n), a^\times, (s', q', n')) = \begin{cases} R_n & \text{if } q' \in \mathcal{F}; \\ 0 & \text{otherwise,} \end{cases}$$

where $R_n \in (0, U]$ are constants for $n \in [0..K]$ and $U \in (0, 1]$ is an upper bound on the rewards. The rewards are nonzero only for accepting automaton states and depend on the value of the K counter.

Then, given a discount factor $\gamma \in (0, 1]$, we define the product discount function $\gamma^\times$:

$S^\times \to (0, 1]$ as

$$\gamma^\times(s_j^\times) = \begin{cases} 1 - r_j^\times & \text{if } r_j^\times > 0; \\ \gamma & \text{otherwise,} \end{cases}$$

and the expected discounted reward following policy π starting at $s^\times$ and time step t is

$$G_t^\pi(s^\times) = \mathbb{E}_\pi[\sum_{i=t}^{\infty}(\prod_{j=t}^{i-1}\gamma^\times(s_j^\times)) \cdot r^\times(s_i^\times, a_i^\times, s_{i+1}^\times) \mid s_t^\times = s^\times].$$

The highest K value reached in a path, that is, the number of accepting states visited in the path, acts as a measure of how promising that path is for satisfying $\varphi_\mathcal{F}$. Depending on the current K value, we can assign varying rewards to accepting states to guide the agent, as discussed in the motivating example in Section 6.4.1.

Next, we provide an optimality guarantee that the policy which maximises the expected discounted reward also maximises the probability of satisfying the Büchi condition, and thus the LTL specification. To achieve this, we assume that the state and action spaces of the MDP are discrete and finite.

Assumption 6.8 (Finiteness of MDP)**.** For the environment MDP $\mathcal{M} = (\mathcal{S}, s_0, \mathcal{A}, \mathcal{P}, \mathcal{AP}, L, r, \gamma)$, we assume that $\mathcal{S}$ is a finite set of states and $\mathcal{A}$ is a finite set of actions.

This assumption is common in the literature (Bozkurt et al., 2019; Hahn et al., 2019; Hasanbeig et al., 2019) of learning LTL objectives with RL to obtain optimality guarantees. It is necessary because we need to analyse the bottom strongly connected components (BSCCs, see Section 2.1.1) of the product MDP and the rewards accumulated within them. If we have continuous state and action spaces, it may take an infinite number of steps to reach a BSCC, and no guarantees regarding the behaviour of infinite paths (i.e., LTL satisfaction) can be made based on finite trajectories explored by RL agents. It is an open problem to provide guarantees for learning LTL objectives in continuous state and action spaces. For more discussion, see Section 6.6 and Section 7.1.

Next, we provide a lemma stating the properties of the product MDP regarding the satisfaction of the Büchi condition $\varphi_{\mathcal{F}}$.

Lemma 6.9. *Given a product MDP $\mathcal{M}^{\times}$ with its corresponding LTL formula φ and a policy π, we write $\mathcal{M}^{\times}_{\pi}$ for the induced Markov chain from π. Let $B^{\times}_{\mathcal{F}}$ denote the set of states that belong to accepting BSCCs of $\mathcal{M}^{\times}_{\pi}$, and $B^{\times}_{\varnothing}$ denote the set of states that belong to rejecting BSCCs:*

$$B^{\times}_{\mathcal{F}} \coloneqq \{s^{\times} \mid s^{\times} \in B \in BSCC(\mathcal{M}^{\times}_{\pi}), B \cap \mathcal{F}^{\times} \neq \varnothing\};$$
$$B^{\times}_{\varnothing} \coloneqq \{s^{\times} \mid s^{\times} \in B \in BSCC(\mathcal{M}^{\times}_{\pi}), B \cap \mathcal{F}^{\times} = \varnothing\},$$

where $BSCC(\mathcal{M}^{\times}_{\pi})$ is the set of all BSCCs of $\mathcal{M}^{\times}_{\pi}$. We further define more general accepting and rejecting sets:

$$B_{\mathcal{F}} \coloneqq \{(s, q, n) \mid \exists n' \in [0..K] : (s, q, n') \in B^{\times}_{\mathcal{F}}\}; \tag{6.1}$$
$$B_{\varnothing} \coloneqq \{(s, q, n) \mid \exists n' \in [0..K] : (s, q, n') \in B^{\times}_{\varnothing}\}. \tag{6.2}$$

If Assumption 6.8 holds, then $B_{\varnothing} \cap \mathcal{F}^{\times} = \varnothing$, $\forall s^{\times} \in B_{\mathcal{F}} : \mathbb{P}_{\pi}(s^{\times} \models \mathsf{GF}\ \varphi_{\mathcal{F}}) = 1$, and $\forall s^{\times} \in B_{\varnothing} : \mathbb{P}_{\pi}(s^{\times} \models \mathsf{GF}\ \varphi_{\mathcal{F}}) = 0$. Furthermore, $B_{\mathcal{F}}$ and $B_{\varnothing}$ are sink sets, which means that, once the set is reached, no states outside the set can be reached.

The proof of this lemma is provided in Appendix C.1.2. Using this lemma, we can now state and prove the main theorem of this chapter.

Theorem 6.10 (Optimality Guarantee)**.** *Given an LTL formula φ and a product MDP $\mathcal{M}^{\times}$, if Assumption 6.8 holds, then there exists an upper bound $U \in (0, 1]$ for rewards and a discount factor $\gamma \in (0, 1]$ such that, for all product rewards $r^{\times}$ and product discount functions $\gamma^{\times}$ satisfying Definition 6.7, the optimal deterministic memoryless policy π_r that maximises the expected discounted reward $G^{\pi_r}_0(s^{\times}_0)$ is also an optimal policy π_{φ} that maximises the probability of satisfying the Büchi condition $\mathbb{P}_{\pi_{\varphi}}(s^{\times}_0 \models \mathsf{GF}\ \varphi_{\mathcal{F}})$ on the product MDP $\mathcal{M}^{\times}$.*

Proof sketch. We now present a sketch of the proof to provide intuition for the main steps and the selection of key hyperparameters. The complete proof is provided in Appendix C.1.3.

To ensure optimality, given a policy π with the product MDP $\mathcal{M}^{\times}$ and the LTL formula φ, we want to demonstrate a tight bound between the expected discounted reward following π and the probability of π satisfying φ, such that maximising one quantity is equivalent to maximising the other.

At a high level, we want to select the two key hyperparameters, the reward upper bound $U \in (0, 1]$ and the discount factor $\gamma \in (0, 1]$, to adequately bound: (i) the rewards given for paths that eventually reach rejecting BSCCs (thus not satisfying the LTL specification); and (ii) the discount of rewards received from rejecting states for paths that eventually reach accepting BSCCs.

We informally denote by $C^{\pi}_{\varnothing}$ the expected number of visits to accepting states before reaching a rejecting BSCC, and (i) can be sufficiently bounded by selecting $U = 1/C^{\pi}_{\varnothing}$. Next, we informally write $C^{\pi}_{\mathcal{F}}$ for the expected number of rejecting states visited before reaching an accepting BSCC, and denote by N^{π} the expected steps between visits of accepting states in the accepting BSCC. Intuitively, for (ii), we bound the amount of discount before reaching the accepting BSCC using $C^{\pi}_{\mathcal{F}}$, and we bound the discount after reaching the BSCC using N^{π}, yielding $\gamma = 1 - 1/(C^{\pi}_{\varnothing} * N^{\pi} + C^{\pi}_{\mathcal{F}})$.

In practice, using upper bounds of $C^{\pi}_{\varnothing}, C^{\pi}_{\mathcal{F}}$ and N^{π} instead also ensures optimality, and these bounds can be deduced from assumptions about, or the knowledge of the MDP. □

As shown in the proof sketch, selecting $U = 1/C^{\pi}_{\varnothing}$ and $\gamma = 1 - 1/(C^{\pi}_{\varnothing} * N^{\pi} + C^{\pi}_{\mathcal{F}})$ is sufficient to ensure optimality. Using the example of the probabilistic gate MDP in Figure 6.1 (right), we have $C^{\pi}_{\varnothing} \approx C^{\pi}_{\mathcal{F}} \leq 10$ and $N^{\pi} = 1$, so choosing $U = 0.1$ and $\gamma = 0.95$ is sufficient to guarantee optimality. For more general MDPs, under the common assumption that the number of states $|\mathcal{S}|$ and the minimum nonzero transition probability $p_{\min} := \min_{s,a,s'}\{\mathcal{P}(s, a, s') > 0\}$ are known, $C^{\pi}_{\varnothing}$ and $C^{\pi}_{\mathcal{F}}$ can be upper bounded by

Algorithm 5 KC Q-learning from LTL

1: **Input:** environment MDP $\mathcal{M}$, LTL formula φ
2: translate φ into an LDBA $\mathcal{A}$
3: construct product MDP $\mathcal{M}^\times$ using $\mathcal{M}$ and $\mathcal{A}$
4: initialise Q for each s and q pair
5: **for** $l \leftarrow 0$ to max_episode **do**
6: initialise $(s, q, n) \leftarrow (s_0, q_0, 0)$
7: **for** $t \leftarrow 0$ to max_timestep **do**
8: get policy π derived from Q (*e.g.,* ϵ-greedy)
9: take action $a^\times \leftarrow \pi((s, q, n))$ in $\mathcal{M}^\times$
10: get next product state (s', q', n')
11: $r \leftarrow r^\times((s, q, n), a^\times, (s', q', n'))$
12: $\gamma \leftarrow \gamma'(s, q, n)$
13: $Q((s, q), a^\times) \overset{\alpha}{\leftarrow} r + \gamma \max_{a \in A^\times} Q((s', q'), a)$
14: update $(s, q, n) \leftarrow (s', q', n')$
15: **end for**
16: **end for**
17: get greedy policy π_φ from Q
18: **Output:** induced policy on $\mathcal{M}$ by removing ϵ-actions

$|\mathcal{S}|/p_{\min}$, while N^π can be upper bounded by $|\mathcal{S}|$.

6.4.2 LTL Learning with Q-Learning

Using the product MDP $\mathcal{M}^\times$ of Definition 6.4 and its reward structure as in Definition 6.7, we present Algorithm 5 (KC), a model-free Q-learning algorithm for LTL specifications using the K counter product MDP. In line 10, the product MDP is constructed on the fly as we explore: for action $a^\times \in \mathcal{A}$, observe the next environment state s' by taking action $a^\times$ in environment state s. Then, we obtain the next automaton state q' using the transition function $\Delta(q, L(s))$ and counter n depending on whether $n \leq K$ and $q' \in \mathcal{F}$. If $a^\times \in \{\epsilon_q \mid q \in \mathcal{Q}\}$, update q' using the ϵ-transition and leave the environment state s and counter n unchanged. However, directly adopting Q-learning on this product MDP produces a Q function defined on the entire product state space $S^\times$, which means that the agent needs to learn the Q function for each K value. To improve efficiency, we propose to define the Q function only on the environment states $\mathcal{S}$ and the automaton states $\mathcal{Q}$, and

for a path σ of $\mathcal{M}^\times$, the update rule for the Q function at time step t is:

$$Q_{t+1}((s_t, q_t), a_t^\times) \overset{\alpha}{\leftarrow} r^\times(s_t^\times, a_t^\times, s_{t+1}^\times) + \gamma'(s_t^\times) \max_{a^\times \in A^\times} Q_t((s_{t+1}, q_{t+1}), a^\times),$$

where $s_t^\times = (s_t, q_t, n_t)$, $a_t^\times$ is the action taken at time step t, $s_{t+1}^\times = (s_{t+1}, q_{t+1}, n_{t+1})$ is the next product state, and α is the learning rate. We assert that, with this collapsed Q function, the algorithm returns the optimal policy for satisfying φ because the optimal policy is independent of the K counter, with the proof provided in Appendix C.1.4.

Theorem 6.11 (Convergence of KC)**.** *Given an environment MDP $\mathcal{M}$ and an LTL specification φ with an appropriate discount factor γ and a reward function $r^\times$ satisfying Theorem 6.10, then if Assumption 6.8 holds, Q-learning for LTL described in Algorithm 5 converges to an optimal policy π_φ that maximises the probability of satisfying φ in $\mathcal{M}$.*

6.4.3 Counterfactual Imagining

We now propose a method to exploit the structure of the product MDP, specifically the LDBA, to improve sample efficiency. While KC (Algorithm 5) is guaranteed to converge to the optimal policy, it requires the RL agent to explore an expanded state space in the product MDP, which increases the sample complexity. To address this, we leverage counterfactual reasoning to generate synthetic imaginations — from a given state in the environment MDP, we generate synthetic trajectories by imagining being at each possible automaton state while taking the same actions. This enables more efficient learning by augmenting the training data without additional environment interaction.

If the agent is at product state (s, q, n) and an action $a^\times \in \mathcal{A}$ is chosen, for each $\bar{q} \in \mathcal{Q}$, the next state $(s', \bar{q}', n')$ by taking an action $a^\times$ from $(s, \bar{q}, n)$ can be computed by first taking the action in the environment state, and then computing the next automaton state $\bar{q}' = \Delta(\bar{q}, L(s))$ and the next K value n'. The reward for the agent is $r^\times(s, \bar{q}, \bar{n})$, and therefore we can update the Q function with this enriched set of experiences. These experiences

Algorithm 6 CF+KC Q-learning from LTL

1: **Input:** environment MDP $\mathcal{M}$, LTL formula φ
2: translate φ into an LDBA $\mathcal{A}$
3: construct product MDP $\mathcal{M}^{\times}$ using $\mathcal{M}$ and $\mathcal{A}$
4: initialise Q for each s and q pair
5: **for** $l \leftarrow 0$ to max_episode **do**
6: initialise $(s, q, n) \leftarrow (s_0, q_0, 0)$
7: **for** $t \leftarrow 0$ to max_timestep **do**
8: get policy π derived from Q (e.g., ϵ-greedy)
9: get action $a^{\times} \leftarrow \pi((s, q, n))$ in $\mathcal{M}^{\times}$
10: **for** $\bar{q} \in \mathcal{Q}$ **do**
11: get counterfactual imagination $(s', \bar{q}', n')$ by taking action $a^{\times}$ at $(s, \bar{q}, n)$
12: $r \leftarrow r^{\times}((s, \bar{q}, n), a^{\times}, (s', \bar{q}', , n'))$
13: $\gamma \leftarrow \gamma'(s, \bar{q}, n)$
14: $Q((s, \bar{q}), a^{\times}) \overset{\alpha}{\leftarrow} r + \gamma \max_{a \in A^{\times}} Q((s', \bar{q}'), a)$
15: **end for**
16: obtain (s', q', n') using action $a^{\times}$ at (s, q, n)
17: update $(s, q, n) \leftarrow (s', q', n')$
18: **end for**
19: **end for**
20: get greedy policy π_{φ} from Q
21: **Output:** induced policy on $\mathcal{M}$ by removing ϵ-actions

produced by counterfactual imagining are still sampled from $\mathcal{M}^{\times}$ following the transition function $T^{\times}$, and hence, when used in conjunction with any off-policy learning algorithms, such as Q-learning, the optimality guarantees of the algorithm are preserved.

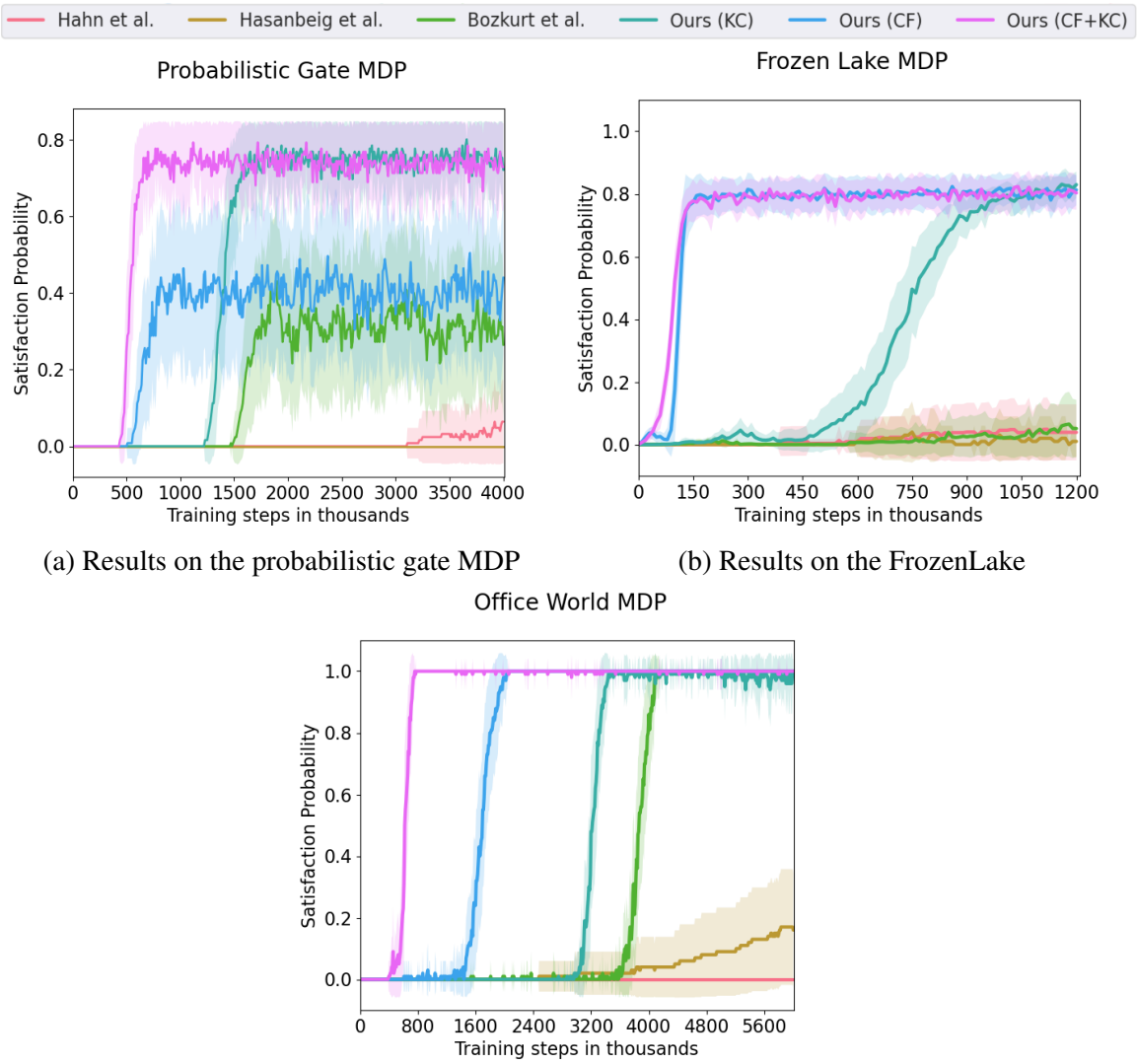
As shown in Algorithm 6 (CF-KC), counterfactual imagining (CF) can be incorporated into our KC Q-learning algorithm by altering a few lines (lines 9-12 in Algorithm 6) of code, and it can also be used in combination with other RL algorithms for LTL learning that use automata-based product MDP. Note that the idea of counterfactual imagining is similar to that proposed by Icarte et al. (2022), but our approach has adopted LDBAs in the product MDPs for LTL specification learning.

6.5 Experimental Results

We evaluate our algorithms in various MDP environments, including the more realistic and challenging stochastic MDP environments[1]. We use tabular Q-learning as the core off-policy learning method to implement our three algorithms: Q-learning with K counter reward structure (KC), Q-learning with K counter reward structure and counterfactual imagining (CF+KC), and Q-learning with only counterfactual imagining (CF), in which we set $K = 0$. We compare the performance of our methods against the methods proposed by Bozkurt et al. (2019), Hahn et al. (2019) and Hasanbeig et al. (2020). Note that our KC algorithm, in the special case that $K = 0$ with no counterfactual imagining, is algorithmically equivalent to Bozkurt et al. (2019) when setting their hyperparameter $\gamma_B = 1 - U$. Our methods differ from all other existing methods to the best of our knowledge.

Experimental Setup The experiments were carried out on a Linux server (Ubuntu 18.04.2) with two Intel Xeon Gold 6252 CPUs and six NVIDIA GeForce RTX 2080 Ti GPUs. We select three stochastic environments for evaluation: the probabilistic gate (Example 6.5), the frozen lake (Brockman et al., 2016), and the office world (Icarte et al., 2022). We set the learning rate $\alpha = 0.1$ and $\epsilon = 0.1$ for exploration. We also set a relatively loose upper bound on rewards $U = 0.1$ and the discount factor $\gamma = 0.99$ for all experiments to ensure optimality. Note that the optimality of our algorithms holds for a family of reward structures defined in Definition 6.7, and for experiments we opt for a specific reward function that linearly increases the reward for accepting states as the value of K increases, namely $r_n = U \cdot n/K \ \forall n \in [0..K]$, to facilitate training and exploration. The Q function is optimistically initialised by setting the Q value for all available state-action pairs at $2U$. We propose to directly evaluate the probability of satisfying LTL specifications by employing the probabilistic model checker PRISM (Kwiatkowska et al., 2011). We build the induced MC $\mathcal{M}_\pi$ from the environment MDP and the policy in PRISM format, and adopt PRISM to

[1]The implementation of our algorithms and experiments can be found on GitHub: `https://github.com/shaodaqian/rl-from-ltl`



(a) Results on the probabilistic gate MDP

(b) Results on the FrozenLake

(c) Results on the office world (Icarte et al., 2022).

Figure 6.2: The experimental results on various MDPs and LTL tasks.

compute the exact satisfaction probability of the given LTL specification. All experiments are run 100 times, where we plot the average satisfaction probability with half the standard deviation in the shaded area. Further details regarding the experimental setup are provided in Appendix C.2.

6.5.1 Probabilistic Gate MDP

First, we conduct experiments on the probabilistic gate MDP described in Example 6.5 with task "FG a & G !c", which means reaching only states labelled "a" in the future while never reaching "c" labelled states. We set $K = 10$ for this task, and in Figure 6.2 (left), compared to the other three methods, our method KC achieved better sample efficiency and convergence, and CF demonstrates better sample efficiency while still lacking training stability. The best performance is achieved by CF+KC, while other methods exhibit slower convergence (Bozkurt et al., 2019; Hahn et al., 2019) or do not converge (Hasanbeig et al., 2020) due to the lack of theoretical optimality guarantees.

6.5.2 Frozen Lake MDP

The second MDP environment is the 8×8 frozen lake environment of OpenAI Gym (Brockman et al., 2016). This environment consists of frozen lake tiles, where the agent has a 1/3 chance of moving in the intended direction and 1/3 of going sideways each, with details provided in Appendix C.2.2. The task is "(GF a | GF b) & G !h", meaning to always reach lake camp "a" or lake camp "b" while never falling into holes "h". We set $K = 10$ for this task, and in Figure 6.2 (middle), we observe significantly better sample efficiency for all our methods, especially for CF+KC and CF, which converge to the optimal policy at around 150k training steps. The other three methods, on the other hand, barely start to converge at 1200k training steps. CF performs especially well in this task because the choice of always reaching "a" or "b" can be considered simultaneously during each time step, reducing the sample complexity to explore the environment.

6.5.3 Office World MDP

Lastly, we experiment on a slight modification of the more challenging office world environment proposed by Icarte et al. (2022), with details provided in Appendix C.2.3. We

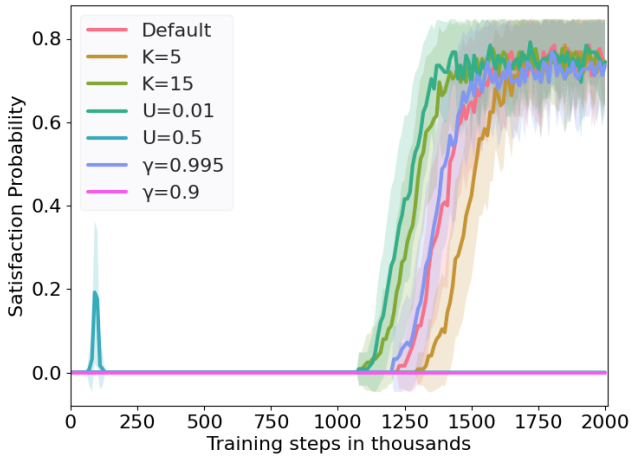


Figure 6.3: Sensitivity analysis on key hyperparameters.

include patches of icy surfaces in the office world, with the task to either patrol in the corridor between "a" and "b", or write letters at "l" and then patrol between getting tea "t" and the workplace "w", while never hitting obstacles "o". $K = 5$ is set for this task for a steeper increase in reward, since the long distance between patrolling states makes visiting many accepting states in each episode time consuming. Figure 6.2 (right) presents again the performance benefit of our methods, with CF+KC performing the best and CF and KC second and third, respectively. For this challenging task in a large environment, the method of Hahn et al. (2019) requires the highest number of training steps to converge.

Overall, the results demonstrate that KC improves both sample efficiency and training stability, especially for challenging tasks. In addition, CF greatly improves the sample efficiency, which, in combination with KC, achieves the best results.

6.5.4 Runtime Analysis and Sensitivity Analysis

It is worth mentioning that, for counterfactual imagining, multiple updates to the Q function are performed at each step in the environment. This increases computational complexity, but the additional inner loop updates on the Q function will only marginally affect the overall computation time if the environment steps are computationally expensive. Taking

the office world task as an example, the average time to perform 6 million training steps is 170.9s, 206.3s, and 236.1s for KC, CF, and CF+KC, respectively. However, the time until convergence to the optimal policies is 96.8s, 70.2s, and 27.5s for KC, CF, and CF+KC, respectively.

For sensitivity analysis of key hyperparameters, we conduct experiments on the probabilistic gate MDP task with different hyperparameters against the default values of $U = 0.1, \gamma = 0.99$ and $K = 10$. As shown in Figure 6.3, if $U(= 0.5)$ is chosen too high or $\gamma(= 0.9)$ is chosen too low, the algorithm does not converge to the optimal policy as expected. However, looser hyperparameters $U = 0.01$ and $\gamma = 0.995$ do not harm performance, which means that, even with limited knowledge of the underlying MDP, our algorithm still performs well with loose hyperparameters. Optimality is not affected by the K value, while performance is only slightly affected by different K values.

6.6 Discussion

In this chapter, we considered the problem of learning policies that satisfy high-level LTL objectives. We presented a novel model-free reinforcement learning algorithm to learn the optimal policy for satisfying LTL specifications in an unknown stochastic MDP environment with optimality guarantees. We proposed a novel product MDP, a generalised reward structure, and an RL algorithm that ensures convergence to the optimal policy with the appropriate hyperparameters. Furthermore, we incorporated counterfactual imagining, which exploits the LTL specification to create counterfactual experiences. Lastly, using PRISM (Kwiatkowska et al., 2011), we directly evaluated the performance of our methods and demonstrated superior performance in various MDP environments and LTL tasks.

The reward structure we defined with the product MDP is general in the sense that it allows for a family of specific reward functions that depend on the counter value K, all of which enjoy the optimality guarantees. In this chapter, we explored a reward function

that increases linearly with the K value, since it encourages exploration. It would also be interesting future work to explore other specific reward functions under our generalised reward structure, such as exponentially increasing or decreasing rewards, and to evaluate the different learning dynamics induced by these reward functions.

The main limitation of the method presented in this chapter is the assumption regarding the discrete state and action spaces of the MDP. It is necessary for our analysis to provide theoretical guarantees on the optimality of the learnt policy due to the infinite MDP path required for an LTL specification to be satisfied. It will be a challenging future task to tackle the open problem (Alur et al., 2022) of dropping all assumptions regarding the underlying MDP, and extending the theoretical framework to continuous state and action environment MDPs, which might be addressed through a finite abstraction of the state space. In addition, using potential-based reward shaping (Ng et al., 1999; Devlin & Kudenko, 2012) to exploit the semantic class structures of the LTL specifications and transferring similar temporal logic knowledge of the agent (Xu & Topcu, 2019) between environments are also interesting future directions.

Chapter 7

Conclusion

This thesis studied the problem of learning decision policies with optimality and sample efficiency guarantees that improve the practicality of AI-driven decisions in real-world applications. Specifically, we considered the problem of learning policies from offline datasets in the presence of hidden confounders and learning policies that satisfy high-level LTL objectives.

We first considered learning policies from a static offline dataset that is confounded by hidden confounders. Under the RL paradigm, we learn the causal effect of actions on the expected return, which can then be used to learn an optimal policy. Specifically, we used instrumental variables, which are variables independent of the hidden confounders, to learn the causal effect of actions through IV regression. The problem of IV regression is a CMR problem, and we proposed a novel CMR estimator, DML-CMR, using the DML framework that provides fast convergence rate guarantees. The learnt policy using DML-CMR also enjoys optimality guarantees, where strong empirical results are demonstrated. However, this approach requires that IVs are observed in the dataset for each action to allow the causal effect of actions to be identifiable. Although IVs are commonly observed in many applications, they are not guaranteed to exist for all real-world applications, and identifying them in the dataset often requires expert knowledge. To this end, we studied the problem of

learning policies without explicit IVs in the confounded dataset and found that it is possible to imitate an expert from a dataset of expert demonstrations, even in the presence of hidden confounders. This describes the problem of confounded imitation learning, where we proposed a framework for confounded IL in Chapter 5 that unifies and generalises previous work in confounded IL. We showed that, by considering the hidden confounders to be partially observable to the expert, this framework subsumes a wide range of settings in previous works. We then reduced the problem of confounded IL in our framework into a CMR problem, where any CMR estimator in principle can be adopted to solve it. Specifically, we adopted our DML-CMR estimator and proposed DML-IL, a confounded IL algorithm in our framework that inherits the convergence rate guarantees of DML-CMR.

Finally, we studied the problem of learning policies that satisfy high-level objectives under the classic online RL framework. To this end, we considered LTL objectives for their expressiveness and widespread usage in robotics and formal verification. We proposed to first translate the LTL specification into a limit-deterministic Büchi automaton, where an accepting run in the automaton also satisfies the LTL specification. Then, we proposed a product MDP that is the product of the automaton, the environment MDP, and a counter that counts the number of accepting states visited. This counter allows a generalised reward structure to be defined in our product MDP, which enables sample-efficient learning by encouraging the agent to explore more efficiently in the environment. We proved that our algorithm will learn an optimal policy that satisfies the LTL specification with the highest probability and demonstrated significantly better sample efficiency against state-of-the-art algorithms.

7.1 Discussion

We now provide a general discussion of the results derived in this thesis, including the practicality of the proposed algorithms, and the strengths and weaknesses of the methods.

7.1.1 Practicality in Real-World Applications

In this thesis, we proposed policy-learning algorithms that are practical in real-world applications. Specifically, in Chapter 4 and Chapter 5, we considered the problem of learning from offline datasets, which are abundant in practice, as opposed to online interactions, which are often unsafe or impractical. We address the hidden confounder problem, which is present in many real-world datasets and often leads to suboptimal policies, by enabling policy learning from confounded offline datasets. However, the ability to handle hidden confounders comes at a cost, namely, the assumptions required to identify the true causal effect between states and actions. In Chapter 4, we assumed that IVs are observed in the offline dataset, such that the causal effect of actions can be estimated. In Chapter 5, we weakened the IV assumption and instead assumed that the confounding noise has a maximum horizon of k, such that the confounding noise k steps prior is independent from the current confounding noise. This allows us to correctly imitate the intended actions of the expert. Although these assumptions limit practicality, we offered solutions for different types of assumptions such that, depending on the specific real-world scenario, users can choose to adopt the appropriate algorithms. In addition, as discussed in the chapters, these assumptions often hold in real-world applications, for example, in robotics, econometrics, and healthcare.

In Chapter 6, we considered learning policies that satisfy LTL specifications. This allows policies to be learnt directly from high-level objectives, removing the need to manually design complex reward functions and reducing the risks of reward misspecification. Our proposed algorithm is, in general, very practical since it adopts model-free RL. Although the optimality guarantee requires the state and action space to be discrete, the algorithm and the product MDP can be adapted to environments with continuous states and actions in practice following Hasanbeig et al. (2020).

Finally, we point out that all the algorithms proposed in this thesis are sample-efficient while providing optimality guarantees. This addresses two important challenges that pre-

vent RL- and DL-driven decision policies from being deployed in the real world. We prove optimality or provide an upper bound on the suboptimality, which guarantees that the algorithms will learn good policies that achieve the desired behaviour. In addition, sample efficiency is important both in offline and online settings, especially because data collection is difficult and costly. This allows a better policy to be learnt given the same amount of data in practice.

7.1.2 Strengths and Weaknesses of the Methods

The main strength of this thesis is the versatile suite of algorithms proposed for various problem settings that improve practicality with provable guarantees, as discussed above. In addition, we proposed novel frameworks and methods that are very general. The method developed in Chapter 4, DML-CMR, can be adopted to solve general CMR problems, which include a large family of important problems in econometrics, statistical learning, causal inference, and mathematics. In Chapter 5, we proposed a unifying framework for confounded IL that unifies and generalises previous work. The algorithm we proposed for confounded IL, DML-IL, also addresses a wider range of problems described by our unifying framework. In Chapter 6, we proposed a generalised product MDP and a reward structure admitting a more flexible and general class of reward functions. In addition, each of these reward functions enjoys optimality guarantees thanks to the generalised product MDP and our theoretical framework.

Regarding the weaknesses, we first highlight the assumptions we made for the algorithms and the theoretical analysis in this thesis. In Chapter 4, we assumed that IVs can be observed in the dataset as the basis for causal identification, and for the theoretical analysis, we assumed some regularity and identifiability conditions. In Chapter 5, we assumed that the confounding noise horizon k is bounded and that an upper bound on it is known for the DML-IL algorithm to imitate the expert. This is also a basic assumption for the expert policy to be identifiable, where we discussed the practicality of these assumptions in detail

in Chapter 5. In Chapter 6, for the theoretical analysis, we assumed that the state and action spaces are discrete. This is a strong assumption to make since many real-world applications have continuous state spaces. However, the discrete assumption is required to obtain the theoretical guarantees, and we point out that in practice, discrete abstractions often exist for continuous state and action spaces. The other weakness of this thesis is that the empirical evaluation is done on synthetic academic benchmarks and semi-synthetic datasets. This is due to the nature of academic research, which favours controlled experiments and reproducibility. In addition, it is challenging to handle real-world data due to privacy concerns and to deploy algorithms in real-world scenarios, both of which are beyond the scope of this thesis. Although it would strengthen the thesis to apply the proposed algorithms in the real world, the empirical results in this thesis provide a strong proof-of-concept assurance for practitioners to use them in practice.

7.2 Open Problems and Future Work

Unsurprisingly, the work in this thesis gives rise to open problems. In Chapter 4, an interesting future work is to extend our DML framework to nonparametric estimation. A possible way to do this is to follow Foster & Syrgkanis (2019), who analysed nonparametric estimation with Neyman orthogonal score functions. In Chapter 5, an open problem is to explore confounded imitation learning in real-world applications such as robotics. Moreover, the setting in which limited queries can be made to the expert is of interest, as it is realistic. The challenge here would be to find the optimal way to query the expert. In Chapter 6, an open problem is to provide optimality guarantees for RL with LTL objectives in continuous state and action spaces. In addition, it would also be interesting to explore offline RL with LTL objectives. One challenge is to design effective offline RL algorithms for LTL objectives, while another challenge is to derive guarantees for the learnt policy.

More broadly and ambitiously, a challenging yet important open problem is to relax the causal identification assumptions when dealing with hidden confounders. Instead, we can consider a setting in which the causal effect can only be partially identified, such as in cases where only the upper and lower bounds of the causal effect can be obtained, which is often considered in the IV literature (Balke & Pearl, 1994; Swanson et al., 2018). Without full identification, we cannot exactly infer the causal effect of actions, but it may be possible to provide pessimistic and robust algorithms based on the uncertainty (e.g., Pace et al. 2024) arising from partial identification. In addition, it is desirable to develop algorithms that remain robust to imperfect or weak IV, possibly by leveraging partial identification and bounding techniques, using uncertainty quantification techniques and instrument synthesis techniques to infer pseudo-instruments (Yuan et al., 2022).

Furthermore, it would be interesting to develop benchmarking datasets for confounded offline RL problems to standardise evaluation. Currently, algorithms tackling confounded offline RL are typically evaluated on toy causal graphs, MuJoCo or grid-world environments with various types of synthetic confounding, and small semi-synthetic healthcare datasets. Existing publications typically use different confounding mechanisms and dataset setups to showcase various aspects of the problem that suit their algorithm, making it difficult for the researcher to understand all the minor nuanced differences in each experimental setup and compare between the methods. Therefore, it would be beneficial to establish a benchmarking dataset for confounded RL in order to standardise the evaluation, improve reproducibility, and enable fair comparison between algorithms. The potential steps towards this goal would be to formally define a general confounded MDP framework, build confounded simulators which can be adopted from D4RL (Fu et al., 2020) environments, establish evaluation metrics, and integrate existing baseline algorithms.

Bibliography

Ai, C. and Chen, X. Efficient estimation of models with conditional moment restrictions containing unknown functions. *Econometrica*, 71:1795–1843, 11 2003.

Aksaray, D., Jones, A., Kong, Z., Schwager, M., and Belta, C. Q-learning for robust satisfaction of signal temporal logic specifications. *2016 IEEE 55th Conference on Decision and Control, CDC 2016*, pp. 6565–6570, 9 2016.

Alur, R., Bansal, S., Bastani, O., and Jothimurugan, K. A framework for transforming specifications in reinforcement learning. *Principles of Systems Design*, 13660 LNCS: 604–624, 2022.

Amos, B., Rodriguez, I. D. J., Sacks, J., Boots, B., and Zico Kolter, J. Differentiable MPC for end-to-end planning and control. *Advances in Neural Information Processing Systems*, 2018-December:8289–8300, 10 2018.

Andreas, J., Klein, D., and Levine, S. Modular multitask reinforcement learning with policy sketches. *34th International Conference on Machine Learning, ICML 2017*, 1: 229–239, 11 2016.

Andrews, I., Stock, J. H., and Sun, L. Weak instruments in instrumental variables regression: Theory and practice. *Annual Review of Economics*, 11:727–753, 8 2019.

Angrist, J. D. Lifetime earnings and the vietnam era draft lottery: Evidence from social security administrative records. *The American Economic Review*, 80:1284–1286, 1990.

Angrist, J. D. and Pischke, J.-S. *Mostly Harmless Econometrics*. Princeton University Press, 2 2009.

Angrist, J. D., Imbens, G. W., and Rubin, D. B. Identification of causal effects using instrumental variables. *Journal of the American Statistical Association*, 91:444–455, 6 1996.

Bacci, E. and Parker, D. Verified probabilistic policies for deep reinforcement learning. *NASA Formal Methods*, 13260 LNCS:193–212, 2022.

Bacci, E., Giacobbe, M., and Parker, D. Verifying reinforcement learning up to infinity. *Proceedings of the International Joint Conference on Artificial Intelligence*, 2021.

Baier, C. and Katoen, J.-P. *Principles Of Model Checking*, volume 950. MIT Press, 2008.

Balke, A. and Pearl, J. Counterfactual probabilities: Computational methods, bounds and applications. *Uncertainty Proceedings 1994*, pp. 46–54, 1 1994.

Bang, H. and Robins, J. M. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4):962–973, 2005.

Bansal, M., Krizhevsky, A., and Ogale, A. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *Robotics: Science and Systems*, 5:5986, 12 2018.

Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., Dhruva, T. B., Muldal, A., Heess, N., and Lillicrap, T. Distributed distributional deterministic policy gradients. *Proceedings of the International Conference on Learning Representations, ICLR 2018*, 4 2018.

Bartlett, P. L., Bousquet, O., and Mendelson, S. Local rademacher complexities. *The Annals of Statistics*, 33:1497-1537, 2005.

Bassily, R., Steinke, T., Nissim, K., Stemmer, U., Smith, A., and Ullman, J. Algorithmic stability for adaptive data analysis. *SIAM Journal on Computing*, pp. 1046–1059, 11 2021.

Bellman, R. A Markovian decision process. *Journal of Mathematics and Mechanics*, 1957.

Belloni, A., Chen, D., Chernozhukov, V., and Hansen, C. Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrica*, 80(6): 2369–2429, 2012.

Benkeser, D., Carone, M., Laan, M. V. D., and Gilbert, P. B. Doubly robust nonparametric inference on the average treatment effect. *Biometrika*, 104(4):863–880, 2017.

Bennett, A. and Kallus, N. The variational method of moments. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85:810–841, 12 2020.

Bennett, A., Kallus, N., and Schnabel, T. Deep generalized method of moments for instrumental variable analysis. *Advances in Neural Information Processing Systems*, 32, 2019.

Bennett, A., Kallus, N., Li, L., and Mousavi, A. Off-policy evaluation in infinite-horizon reinforcement learning with latent confounders. *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pp. 1999–2007, 3 2021.

Berkenkamp, F., Turchetta, M., Schoellig, A. P., and Krause, A. Safe model-based reinforcement learning with stability guarantees. *Advances in Neural Information Processing Systems*, 2017-December:909–919, 5 2017.

Bethell, D., Gerasimou, S., Calinescu, R., and Imrie, C. Safe reinforcement learning in black-box environments via adaptive shielding. *arXiv:2405.18180*, 5 2024.

Bica, I., Jordon, J., and van der Schaar, M. Estimating the effects of continuous-valued

interventions using generative adversarial networks. *Advances in Neural Information Processing Systems*, 2020-December, 2 2020.

Bica, I., Jarrett, D., and van der Schaar, M. Invariant causal imitation learning for generalizable policies. *Proceedings of the Conference on Neural Information Processing Systems*, 11 2021.

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37:1705–1732, 8 2009.

Blundell, R., Chen, X., and Kristensen, D. Semi-nonparametric IV estimation of shape-invariant engel curves. *Econometrica*, 75:1613–1669, 11 2007.

Bound, J., Jaeger, D. A., and Baker, R. M. Problems with instrumental variables estimation when the correlation between the instruments and the endogeneous explanatory variable is weak. *Journal of the American Statistical Association*, 90:443, 6 1995.

Bozkurt, A. K., Wang, Y., Zavlanos, M. M., and Pajic, M. Control synthesis from linear temporal logic specifications using model-free reinforcement learning. *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 10349–10355, 2019.

Brandfonbrener, D., Whitney, W. F., Ranganath, R., and Bruna, J. Offline RL without off-policy evaluation. *Advances in Neural Information Processing Systems*, 7:4933–4946, 6 2021.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Openai, W. Z. OpenAI Gym. *arXiv:1606.01540*, 6 2016.

Buckman, J., Hafner, D., Tucker, G., Brevdo, E., and Lee, H. Sample-efficient reinforcement learning with stochastic ensemble value expansion. *Advances in Neural Information Processing Systems*, 2018-December:8224–8234, 7 2018.

Buesing, L., Weber, T., Zwols, Y., Racanière, S., Guez, A., Lespiau, J. B., and Heess, N. Woulda, coulda, shoulda: Counterfactually-guided policy search. *7th International Conference on Learning Representations*, 11 2019.

Büchi, J. R. On a decision method in restricted second order arithmetic. *Studies in Logic and the Foundations of Mathematics*, 44:1–11, 1 1966.

Cai, M., Hasanbeig, M., Xiao, S., Abate, A., and Kan, Z. Modular deep reinforcement learning for continuous motion planning with temporal logic. *IEEE Robotics and Automation Letters*, 6(4):7973–7980, 2021.

Camacho, A., Icarte, R. T., Klassen, T. Q., Valenzano, R., and McIlraith, S. A. LTL and beyond: Formal languages for reward function specification in reinforcement learning. *IJCAI International Joint Conference on Artificial Intelligence*, 2019-August:6065–6073, 2019.

Caponnetto, A. and De Vito, E. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7:331–368, 2007.

Carrasco, M., Florens, J. P., and Renault, E. Linear inverse problems in structural econometrics estimation based on spectral decomposition and regularization. *Handbook of Econometrics*, 6:5633–5751, 1 2007.

Chebotar, Y., Handa, A., Makoviychuk, V., MacKlin, M., Issac, J., Ratliff, N., and Fox, D. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. *Proceedings - IEEE International Conference on Robotics and Automation*, 2019-May:8973–8979, 10 2018.

Chen, D., Zhou, B., Koltun, V., and Krähenbühl, P. Learning by cheating. *Proceedings of Machine Learning Research*, 100:66–75, 12 2019.

Chen, J., Chen, X., and Tamer, E. Efficient estimation of average derivatives in npiv models: Simulation comparisons of neural network estimators. *Journal of Econometrics*, 235: 1848–1875, 8 2023.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Advances in Neural Information Processing Systems*, 18:15084–15097, 6 2021a.

Chen, X. and Christensen, T. M. Optimal sup-norm rates and uniform inference on nonlinear functionals of nonparametric IV regression. *Quantitative Economics*, 9:39–84, 3 2018.

Chen, X. and Pouzo, D. Estimation of nonparametric conditional moment models with possibly nonsmooth generalized residuals. *Econometrica*, 80:277–321, 1 2012.

Chen, X. and White, H. Improved rates and asymptotic normality for nonparametric neural network estimators. *IEEE Transactions on Information Theory*, 45(2):682–691, 1999.

Chen, Y., Xu, L., Gulcehre, C., Paine, T. L., Gretton, A., de Freitas, N., and Doucet, A. On instrumental variable regression for deep offline policy evaluation. *Journal of Machine Learning Research*, 23, 5 2021b.

Cheng, C. A., Yan, X., and Boots, B. Trajectory-wise control variates for variance reduction in policy gradient methods. *Proceedings of Machine Learning Research*, 8 2019.

Cheng, D., Xu, Z., Li, J., Liu, L., Liu, J., and Duy, T. L. Conditional instrumental variable regression with representation learning for causal inference. *Proceedings of the International Conference on Learning Representations*, 10 2024.

Chernozhukov, V., Hansen, C., and Spindler, M. Post-selection and post-regularization inference in linear models with many controls and instruments. *American Economic Review*, 105(5):486–490, 2015.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018.

Chernozhukov, V., Newey, W. K., Quintas-Martinez, V., and Syrgkanis, V. Automatic debiased machine learning via neural nets for generalized linear regression. *arXiv:2104.14737*, 4 2021.

Chernozhukov, V., Escanciano, J. C., Ichimura, H., Newey, W. K., and Robins, J. M. Locally robust semiparametric estimation. *Econometrica*, 90(4):1501–1535, 7 2022a.

Chernozhukov, V., Newey, W., Quintas-Martínez, V., and Syrgkanis, V. RieszNet and ForestRiesz: Automatic debiased machine learning with neural nets and random forests. *Proceedings of Machine Learning Research*, 162:3901–3914, 10 2022b.

Chernozhukov, V., Hansen, C., Kallus, N., Spindler, M., and Syrgkanis, V. Applied causal inference powered by ml and ai. *CausalML-book.org*, 12(1):338, 2024.

Choudhury, S., Bhardwaj, M., Arora, S., Kapoor, A., Ranade, G., Scherer, S., and Dey, D. Data-driven planning via imitation learning. *International Journal of Robotics Research*, 37:1632–1672, 11 2017.

Chow, Y., Nachum, O., Duenez-Guzman, E., and Ghavamzadeh, M. A lyapunov-based approach to safe reinforcement learning. *Advances in Neural Information Processing Systems*, 2018-December:8092–8101, 5 2018.

Christmann, A. and Steinwart, I. *Support vector machines*. Springer, 2008.

Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in Neural Information Processing Systems*, 2018-December:4754–4765, 5 2018.

Clavera, I., Rothfuss, J., Schulman, J., Fujita, Y., Asfour, T., and Abbeel, P. Model-based reinforcement learning via meta-policy optimization. *Conference on Robot Learning*, 9 2018.

Codevilla, F., Santana, E., Lopez, A., and Gaidon, A. Exploring the limitations of behavior cloning for autonomous driving. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:9328–9337, 4 2019.

Cohen, M. H. and Belta, C. Model-based reinforcement learning for approximate optimal control with temporal logic specifications. *HSCC 2021 - Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control (part of CPS-IoT Week)*, pp. 11, 5 2021.

Cui, Y., Pu, H., Shi, X., Miao, W., and Tchetgen, E. T. Semiparametric proximal causal inference. *Journal of the American Statistical Association*, 119:1348–1359, 11 2023.

Darolles, S., Fan, Y., Florens, J. P., and Renault, E. Nonparametric instrumental regression. *Econometrica*, 79:1541–1565, 9 2011.

De Giacomo, G. and Vardi, M. Y. Linear temporal logic and linear dynamic logic on finite traces. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2013.

De Giacomo, G., Luca Iocchi, Marco Favorito, and Fabio Patrizi. Foundations for restraining bolts: Reinforcement learning with LTLf/LDLf restraining specifications. In *Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling*, 2019.

de Haan, P., Jayaraman, D., and Levine, S. Causal confusion in imitation learning. *Advances in Neural Information Processing Systems*, 32, 5 2019.

Devlin, S. and Kudenko, D. Dynamic potential-based reward shaping. *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, 2012.

Dikkala, N., Lewis, G., Mackey, L., and Syrgkanis, V. Minimax estimation of conditional moment models. *Advances in Neural Information Processing Systems*, 2020-December, 6 2020.

Ding, W., Lin, H., Li, B., and Zhao, D. Generalizing goal-conditioned reinforcement learning with variational causal reasoning. *Advances in Neural Information Processing Systems*, 35, 7 2022.

Domínguez, M. A. and Lobato, I. N. Consistent estimation of models defined by conditional moment restrictions. *Econometrica*, 72:1601–1615, 9 2004.

Eghbal-Zadeh, H., Henkel, F., and Widmer, G. Learning to infer unseen contexts in causal contextual reinforcement learning. *In Self-supervision for Reinforcement Learning (SSL-RL) Workshop, ICLR*, 2021.

Ernst, D., Geurts, P., and Wehenkel, L. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

Farrell, M. H., Liang, T., and Misra, S. Deep neural networks for estimation and inference. *Econometrica*, 89(1):181–213, 2021.

Feinberg, V., Wan, A., Stoica, I., Jordan, M. I., Gonzalez, J. E., and Levine, S. Model-based value estimation for efficient model-free reinforcement learning. *arXiv:1803.00101*, 2 2018.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. *32nd AAAI Conference on Artificial Intelligence*, pp. 2974–2982, 5 2018.

Foster, D. J. and Syrgkanis, V. Orthogonal statistical learning. *Annals of Statistics*, 51(3): 879–908, 1 2019.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv:2004.07219*, 4 2020.

Fu, Z., Qi, Z., Wang, Z., Yang, Z., Xu, Y., and Kosorok, M. R. Offline reinforcement learning with instrumental variables in confounded Markov decision processes. *arXiv:2209.08666v1*, 2022.

Funk, M. J., Westreich, D., Wiesen, C., Stürmer, T., Brookhart, M. A., and Davidian, M. Doubly robust estimation of causal effects. *American journal of epidemiology*, 173(7): 761–767, 2011.

Gaon, M. and Brafman, R. I. Reinforcement learning with non-Markovian rewards. *34th AAAI Conference on Artificial Intelligence*, pp. 3980–3987, 12 2020.

Gasse, M., Grasset, D., Gaudron, G., and Oudeyer, P.-Y. Using confounded data in latent model-based reinforcement learning. *Transactions on Machine Learning Research*, 2023.

Gelada, C. and Bellemare, M. G. Off-policy deep reinforcement learning by bootstrapping the covariate shift. *33rd AAAI Conference on Artificial Intelligence*, pp. 3647–3655, 1 2019.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63:139–144, 6 2014.

Gu, S., Lillicrap, T., Sutskever, U., and Levine, S. Continuous deep Q-learning with model-based acceleration. *33rd International Conference on Machine Learning, ICML 2016*, 6:4135–4148, 3 2016.

Gu, S., Holly, E., Lillicrap, T., and Levine, S. Deep reinforcement learning for robotic

manipulation with asynchronous off-policy updates. In *Proceedings - IEEE International Conference on Robotics and Automation*, 2017.

Gupta, A. and Hwang, I. Safety verification of model based reinforcement learning controllers. *arXiv:2010.10740*, 10 2020.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *35th International Conference on Machine Learning, ICML 2018*, 5:2976–2989, 1 2018.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:4528–4547, 11 2018.

Hafner, D., Research, G., Lillicrap, T., Norouzi, M., and Ba, J. Mastering atari with discrete world models. *Proceedings of the International Conference on Learning Representations 2021*, 2020.

Hahn, E. M., Perez, M., Schewe, S., Somenzi, F., Trivedi, A., and Wojtczak, D. Omega-regular objectives in model-free reinforcement learning. *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 11427 LNCS:395–412, 2019.

Hahn, E. M., Perez, M., Schewe, S., Somenzi, F., Trivedi, A., and Wojtczak, D. Faithful and effective reward schemes for model-free reinforcement learning of omega-regular objectives. *Automated Technology for Verification and Analysis*, 12302 LNCS:108–124, 2020.

Hahn, E. M., Perez, M., Schewe, S., Somenzi, F., Trivedi, A., and Wojtczak, D. Mungojerrie: Reinforcement learning of linear-time objectives. *arXiv:2106.09161*, 2021.

Hahn, E. M., Perez, M., Schewe, S., Somenzi, F., Trivedi, A., and Wojtczak, D. An impossibility result in automata-theoretic reinforcement learning. *Automated Technology for Verification and Analysis*, 13505 LNCS:42–57, 2022.

Hallak, A. and Mannor, S. Consistent on-line off-policy evaluation. *34th International Conference on Machine Learning, ICML 2017*, 3:2197–2214, 2 2017.

Hansen, L. P. Large sample properties of generalized method of moments estimators. *Econometrica*, 50:1029, 7 1982.

Hartford, J., Lewis, G., Leyton-Brown, K., and Taddy, M. Deep IV: a flexible approach for counterfactual prediction. *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Harutyunyan, A., Bellemare, M. G., Stepleton, T., and Munos, R. Q(λ) with off-policy corrections. *Algorithmic Learning Theory*, 9925 LNAI:305–320, 2016.

Hasanbeig, M., Kantaros, Y., Abate, A., Kroening, D., Pappas, G. J., and Lee, I. Reinforcement learning for temporal logic control synthesis with probabilistic satisfaction guarantees. *Proceedings of the IEEE Conference on Decision and Control*, 2019-December: 5338–5343, 9 2019.

Hasanbeig, M., Kroening, D., and Abate, A. Deep reinforcement learning with temporal logics. *Proceedings of the International Conference on Formal Modeling and Analysis of Timed Systems*, 12288 LNCS:1–22, 2020.

Hasselt, V. H., Doron, Y., Strub, F., Hessel, M., Sonnerat, N., and Modayil, J. Deep reinforcement learning and the deadly triad. *arXiv:1812.02648*, 12 2018.

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

Hausknecht, M. and Stone, P. Deep recurrent Q-learning for partially observable MDPs. *AAAI Fall Symposium - Technical Report*, FS-15-06:29–37, 7 2015.

He, Y., Liu, P., and Ji, Y. Scalable signal temporal logic guided reinforcement learning via value function space optimization. *arXiv:2408.01923*, 8 2024.

Hill, J. L. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20:217–240, 3 2011.

Ho, J. and Ermon, S. Generative adversarial imitation learning. *Advances in Neural Information Processing Systems*, pp. 4572–4580, 6 2016.

Ho, N., Yang, C.-Y., and Jordan, M. I. Convergence rates for Gaussian mixtures of experts. *Journal of Machine Learning Research*, 23(323):1–81, 2022.

Honerkamp, J. and Weese, J. Tikhonovs regularization method for ill-posed problems - a comparison of different methods for the determination of the regularization parameter. *Continuum Mechanics and Thermodynamics*, 2:17–30, 3 1990.

Hoque, R., Balakrishna, A., Novoseller, E., Wilcox, A., Brown, D. S., and Goldberg, K. Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning. *Proceedings of Machine Learning Research*, 164:598–608, 9 2021.

Huang, B., Lu, C., Leqi, L., Hernandez-Lobato, J. M., Glymour, C., Scholkopf, B., and Zhang, K. Action-sufficient state representation learning for control with structural constraints. *Proceedings of the International Conference on Machine Learning*, 162:9260–9279, 10 2022.

Huang, J. and Jiang, N. From importance sampling to doubly robust policy gradient. *37th International Conference on Machine Learning, ICML 2020*, pp. 4384–4393, 10 2019.

Huang, S., Papernot, N., Goodfellow, I., Duan, Y., and Abbeel, P. Adversarial attacks on neural network policies. *arXiv:1702.02284*, 2017.

Huang, Z., Van Der Aalst, W. M., Lu, X., and Duan, H. Reinforcement learning based resource allocation in business process management. *Data & Knowledge Engineering*, 70(1):127–145, 1 2011.

Icarte, R. T., Klassen, T. Q., Valenzano, R., and McIlraith, S. A. Using reward machines for high-level task specification and decomposition in reinforcement learning. *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Icarte, R. T., Klassen, T. Q., Valenzano, R., and McIlraith, S. A. Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*, 73:173–208, 2022.

Ichimura, H. and Newey, W. K. The influence function of semiparametric estimators. *Quantitative Economics*, 13:29–61, 1 2022.

Im, D. J., Cho, K., and Razavian, N. Causal effect variational autoencoder with uniform treatment. *arXiv:2111.08656*, 11 2021.

Jackermeier, M. and Abate, A. DeepLTL: Learning to efficiently satisfy complex LTL specifications. *Proceedgins of the International Conference on Learning Representations*, 10 2025.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(51):1563–1600, 2010.

Jamshidi, F., Akbari, S., and Kiyavash, N. Causal imitability under context-specific independence relations. *Proceedings of the Conference on Neural Information Processing Systems, NeurIPS*, 6 2023.

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. *Advances in Neural Information Processing Systems*, 32, 6 2019.

Janner, M., Li, Q., and Levine, S. Offline reinforcement learning as one big sequence modeling problem. *Advances in Neural Information Processing Systems*, 2:1273–1286, 6 2021.

Jaques, N., Ghandeharioun, A., Shen, J. H., Ferguson, C., Lapedriza, A., Jones, N., Gu, S., and Picard, R. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv:1907.00456*, 6 2019.

Jiang, Y., Bharadwaj, S., Wu, B., Shah, R., Topcu, U., and Stone, P. Temporal-logic-based reward shaping for continuing reinforcement learning tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):7995–8003, 5 2021.

Jin, Y., Yang, Z., and Wang, Z. Is pessimism provably efficient for offline RL? *International Conference on Machine Learning*, 2021.

Jothimurugan, K., Alur, R., and Bastani, O. A composable specification language for reinforcement learning tasks. *Advances in Neural Information Processing Systems*, 32, 8 2020.

Jothimurugan, K., Bansal, S., Bastani, O., and Alur, R. Compositional reinforcement learning from logical specifications. *Advances in Neural Information Processing Systems*, 12: 10026–10039, 6 2021.

Jothimurugan, K., Bansal, S., Bastani, O., and Alur, R. Specification-guided learning of nash equilibria with high social welfare. *Computer Aided Verification*, 13372 LNCS: 343–363, 2022.

Jung, Y., Tian, J., and Bareinboim, E. Estimating identifiable causal effects through double machine learning. *AAAI Conference on Artificial Intelligence*, 2021.

Kahn, G., Villaflor, A., Abbeel, P., Levine, S., and Research, B. A. Composable action-

conditioned predictors: Flexible off-policy learning for robot navigation. *Conference on Robot Learning*, 10 2018.

Kahn, G., Abbeel, P., and Levine, S. BADGR: an autonomous self-supervised learning-based navigation system. *IEEE Robotics and Automation Letters*, 6(2):1312–1319, 2 2020.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. *International Conference on Machine Learning*, 2002.

Kakade, S. M. A natural policy gradient. *Advances in Neural Information Processing Systems*, 14, 2001.

Kakade, S. M. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, University College London, 2003.

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., and Levine, S. QT-Opt: scalable deep reinforcement learning for vision-based robotic manipulation. *Conference on Robot Learning*, 6 2018.

Kapoor, P., Balakrishnan, A., and Deshmukh, J. V. Model-based reinforcement learning from signal temporal logic specifications. *arXiv:2011.04950*, 11 2020.

Kelly, M., Sidrane, C., Driggs-Campbell, K., and Kochenderfer, M. J. HG-DAgger: interactive imitation learning with human experts. *Proceedings - IEEE International Conference on Robotics and Automation*, 2019-May:8077–8083, 10 2018.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. MOReL: model-based offline reinforcement learning. *Advances in Neural Information Processing Systems*, 2020-December, 2020.

Kloek, T. and van Dijk, H. K. Bayesian estimates of equation system parameters: An application of integration by monte carlo. *Econometrica*, 46(1):1, 1 1978.

Kompa, B., Bellamy, D. R., Kolokotrones, T., Robins, J. M., and Beam, A. L. Deep learning methods for proximal inference via maximum moment restriction. *Advances in Neural Information Processing Systems*, 35, 5 2022.

Konda, V. R. and Tsitsiklis, J. N. Actor-critic algorithms. *Advances in Neural Information Processing Systems*, 2000.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit Q-learning. *Proceedings of The International Conference on Learning Representations*, 10 2022.

Kremer, H. and Schölkopf, B. Geometry-aware instrumental variable regression. *Proceedings of the International Conference on Machine Learning*, 5 2024.

Kress, R. *Linear integral equations*, volume 82. Springer, 1999.

Křetínský, J., Meggendorfer, T., Sickert, S., and Ziegler, C. Rabinizer 4: From LTL to your favourite deterministic automaton. *Proceedings of the International Conference on Computer Aided Verification*, 10981 LNCS:567–577, 2018.

Kuefler, A., Morton, J., Wheeler, T., and Kochenderfer, M. Imitating driver behavior with generative adversarial networks. *IEEE Intelligent Vehicles Symposium, Proceedings*, 5: 204–211, 1 2017.

Kumar, A., Fu, J., Tucker, G., and Levine, S. Stabilizing off-policy Q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 6 2019.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative Q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 2020-December, 6 2020.

Kumar, N., Agrawal, P., Ramponi, G., Levy, K. Y., and Mannor, S. Improved sample complexity for global convergence of actor-critic algorithms. *arXiv:2410.08868*, 10 2024.

Kumor, D., Zhang, J., and Bareinboim, E. Sequential causal imitation learning with unobserved confounders. *Proceedings of the 35th Conference on Neural Information Processing Systems*, 8 2021.

Kurutach, T., Clavera, I., Duan, Y., Tamar, A., and Abbeel, P. Model-ensemble trust-region policy optimization. *International Conference on Learning Representations, ICLR 2018*, 2 2018.

Kwiatkowska, M., Norman, G., and Parker, D. PRISM 4.0: Verification of probabilistic real-time systems. *Computer Aided Verification*, 6806 LNCS:585–591, 2011.

Lagoudakis, M. G. and Parr, R. Least-squares policy iteration. *Journal of Machine Learning Research*, 4(Dec):1107–1149, 2003.

Le, X.-B., Wagner, D., Witzman, L., Rabinovich, A., and Ong, L. Reinforcement learning with ltl and omega-regular objectives via optimality-preserving translation to average rewards. *Conference on Neural Information Processing Systems*, 2024.

LeCun, Y. and Cortes, C. Mnist handwritten digit database, 2010.

Lecun, Y., Muller, U., Ben, J., Cosatto, E., and Flepp, B. Off-road obstacle avoidance through end-to-end learning. *Advances in Neural Information Processing Systems*, 18, 2005.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015.

Lee, D. and He, N. Stochastic primal-dual Q-learning algorithm for discounted mdps. *Proceedings of the American Control Conference*, 2019-July:4897–4902, 7 2019.

Lee, T. E., Zhao, J., Sawhney, A. S., Girdhar, S., and Kroemer, O. Causal reasoning in simulation for structure and transfer learning of robot manipulation policies. *Proceedings*

- *IEEE International Conference on Robotics and Automation*, 2021-May:4776–4782, 3 2021.

Levine, S. and Koltun, V. Guided policy search. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 1–9. PMLR, 5 2013.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17:1–40, 4 2015.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv:2005.01643*, 2020.

Li, G., Wei, Y., Chi, Y., and Chen, Y. Breaking the sample size barrier in model-based reinforcement learning with a generative model. *Operations Research*, 72:203–221, 5 2023.

Li, J., Luo, Y., Zhang, X., Ai, C., Chernozhukov, V., Dai, J., Fernandez-Val, I., Forneron, J.-J., Jiang, W., and Kaido, H. Causal reinforcement learning: An instrumental variable approach. *SSRN Electronic Journal*, 3 2021.

Li, X., Vasile, C. I., and Belta, C. Reinforcement learning with temporal logic rewards. *IEEE International Conference on Intelligent Robots and Systems*, 2017-September: 3834–3839, 12 2016.

Liao, L., Chen, Y. L., Yang, Z., Dai, B., Wang, Z., and Kolar, M. Provably efficient neural estimation of structural equation model: An adversarial approach. *Advances in Neural Information Processing Systems*, 2020-December, 7 2020.

Liao, L., Fu, Z., Yang, Z., Wang, Y., Ma, D., Kolar, M., and Wang, Z. Instrumental variable value iteration for causal offline reinforcement learning. *Journal of Machine Learning Research*, 25:1–55, 2 2024.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *Proceedings of the International Conference on Learning Representations*, 9 2016.

Lin, L.-J. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning 1992 8:3*, 8(3):293–321, 5 1992.

Lin, Y. C., Hong, Z. W., Liao, Y. H., Shih, M. L., Liu, M. Y., and Sun, M. Tactics of adversarial attack on deep reinforcement learning agents. In *IJCAI International Joint Conference on Artificial Intelligence*, pp. 3756–3762, 2017.

Littman, M. L., Topcu, U., Fu, J., Isbell, C., Wen, M., and MacGlashan, J. Environment-independent task specifications via GLTL. *arXiv:1704.04341*, 4 2017.

Liu, C. and Liu, G. Jointppo: Diving deeper into the effectiveness of PPO in multi-agent reinforcement learning. *arXiv:2404.11831*, 4 2024.

Liu, Q., Tang, Z., Li, L., and Zhou, D. Breaking the curse of horizon: Infinite-horizon off-policy estimation. *Advances in Neural Information Processing Systems*, 2018-December:5356–5366, 10 2019.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *7th International Conference on Learning Representations, ICLR 2019*, 11 2017.

Lu, C., Huang, B., Wang, K., Jo´se, J., Hernández-Lobato, M., Zhang, K., and Schölkopf, B. Sample-efficient reinforcement learning via counterfactual-based data augmentation. *Offline Reinforcement Learning Workshop at Neurips*, 12 2020a.

Lu, Y., Meisami, A., Tewari, A., and Yan, Z. Regret analysis of bandit problems with causal background knowledge. *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence*, 10 2020b.

Luo, Y., Spindler, M., and Kück, J. High-dimensional L_2 boosting: Rate of convergence. *arXiv:1602.08927*, 2016.

Lütjens, B., Everett, M., and How, J. P. Certified adversarial robustness for deep reinforcement learning. *Proceedings of the Conference on Robot Learning*, 10 2019.

Mackey, L., Syrgkanis, V., and Zadik, D. Orthogonal machine learning: Power and limitations. *35th International Conference on Machine Learning, ICML 2018*, 13:9112–9124, 11 2018.

Majeed, S. J. and Hutter, M. On Q-learning convergence for non-Markov decision processes. *IJCAI International Joint Conference on Artificial Intelligence*, 2018-July:2546–2552, 2018.

Mandlekar, A., Xu, D., Martín-Martín, R., Zhu, Y., Fei-Fei, L., and Savarese, S. Human-in-the-loop imitation learning using remote teleoperation. *arXiv:2012.06733*, 12 2020.

Mason, G. R., Calinescu, R. C., Kudenko, D., and Banks, A. Assured reinforcement learning with formally verified abstract policies. In *9th International Conference on Agents and Artificial Intelligence*, 2017.

Mastouri, A., Zhu, Y., Gultchin, L., Korba, A., Silva, R., Kusner, M. J., Gretton, A., and Muandet, K. Proximal causal learning with kernels: Two-stage estimation and moment restriction. *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Matthey, L., Higgins, I., Hassabis, D., and Lerchner, A. dSprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Miao, W., Geng, Z., and Tchetgen, E. J. T. Identifying causal effects with proxy variables of an unmeasured confounder. *Biometrika*, 105:987–993, 9 2018.

Milosevic, N., Muller, J., and Scherf, N. Embedding safety into RL: A new take on trust region methods. *arXiv:2411.02957*, 11 2024.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *NeurIPS Deep Learning Workshop*, 12 2013.

Mnih, V., Badia, A. P., Mirza, L., Graves, A., Harley, T., Lillicrap, T. P., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. *33rd International Conference on Machine Learning, ICML 2016*, 4:2850–2869, 2 2016.

Muandet, K., Mehrjou, A., Lee, S. K., and Raj, A. Dual instrumental variable regression. *Advances in Neural Information Processing Systems*, 2020-December, 10 2020.

Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. G. Safe and efficient off-policy reinforcement learning. *Advances in Neural Information Processing Systems*, pp. 1054–1062, 6 2016.

Murphy, K. M. and Topel, R. H. Estimation and inference in two-step econometric models. *Journal of Business and Economic Statistics*, 3(4), 1985.

Nachum, O. and Dai, B. Reinforcement learning via Fenchel-Rockafellar duality. *arXiv:2001.01866*, 1 2020.

Nachum, O., Chow, Y., Dai, B., and Li, L. DualDICE: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in Neural Information Processing Systems*, 32, 6 2019.

Nagabandi, A., Kahn, G., Fearing, R. S., and Levine, S. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 7579–7586, 8 2017.

Nair, A., Gupta, A., Dalal, M., and Levine, S. AWAC: Accelerating online reinforcement learning with offline datasets. *arXiv:2006.09359*, 6 2020.

Namkoong, H., Keramati, R., Yadlowsky, S., and Brunskill, E. Off-policy policy evaluation for sequential decisions under unobserved confounding. *Advances in Neural Information Processing Systems*, 33:18819–18831, 2020.

Nashed, M. Z. and Wahba, G. Generalized inverses in reproducing kernel spaces: An approach to regularization of linear operator equations. *SIAM Journal on Mathematical Analysis*, 5, 1974.

Newey, W. K. and Powell, J. L. Instrumental variable estimation of nonparametric models. *Econometrica*, 71:1565–1578, 9 2003.

Neyman, J. and Scott, E. L. Asymptotically optimal tests of composite hypotheses for randomized experiments with noncontrolled predictor variables. *Journal of the American Statistical Association*, 60:699–721, 1965.

Ng, A. Y., Ng, A. Y., Harada, D., and Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287, 1999.

Nguyen-Tang, T., Gupta, S., Nguyen, A. T., and Venkatesh, S. Offline neural contextual bandits: Pessimism, optimization and generalization. *Proceeding of the International Conference on Learning Representations*, 2022.

Nowozin, S., Cseke, B., and Tomioka, R. f-GAN: Training generative neural samplers using variational divergence minimization. *Advances in Neural Information Processing Systems*, pp. 271–279, 6 2016.

O'Donoghue, B., Osband, I., Munos, R., and Mnih, V. The uncertainty bellman equation

and exploration. *35th International Conference on Machine Learning, ICML 2018*, 9: 6154–6173, 9 2017.

Oren, Y., Zanger, M. A., van der Vaart, P. R., Spaan, M. T. J., and Bohmer, W. Value improved actor critic algorithms. *arXiv:2406.01423*, 6 2024.

Ortega, P. A. and Braun, D. A. A minimum relative entropy principle for learning and acting. *Journal of Artificial Intelligence Research*, 38:475–511, 10 2008.

Ortega, P. A., Kunesch, M., Delétang, G., Genewein, T., Grau-Moya, J., Veness, J., Buchli, J., Degrave, J., Piot, B., Perolat, J., Everitt, T., Tallec, C., Parisotto, E., Erez, T., Chen, Y., Reed, S., Hutter, M., de Freitas, N., and Legg, S. Shaking the foundations: delusions in sequence models for interaction and control. *arXiv:2110.10819*, 10 2021.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped DQN. *Advances in Neural Information Processing Systems*, pp. 4033–4041, 2 2016.

Pace, A., Yèche, H., Schölkopf, B., Rätsch, G., and Tennenholtz, G. Delphic offline reinforcement learning under nonidentifiable hidden confounding. *12th International Conference on Learning Representations*, 6 2024.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 12 2019.

Pearl, J. Causality: models, reasoning, and inference. *Econometric Theory*, 2000.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv:1910.00177*, 10 2019.

Peshkin, L. and Shelton, C. R. Learning from scarce experience. *Proceedings of the International Conference on Machine Learning*, 4 2002.

Pfrommer, S., Bai, Y., Lee, H., and Sojoudi, S. Initial state interventions for deconfounded imitation learning. *arXiv:2307.15980*, 7 2023.

Piot, B., Geist, M., and Pietquin, O. Bridging the gap between imitation learning and inverse reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 28:1814–1826, 8 2017.

Pitis, S., Creager, E., and Garg, A. Counterfactual data augmentation using locally factored dynamics. *Advances in Neural Information Processing Systems*, 2020-December, 7 2020.

Pitis, S., Creager, E., Mandlekar, A., and Garg, A. Mocoda: Model-based counterfactual data augmentation. *Advances in Neural Information Processing Systems*, 35, 10 2022.

Pomerleau, D. A. Alvinn: An autonomous land vehicle in a neural network. *Advances in Neural Information Processing Systems*, 1, 1988.

Powell, W. B. *Approximate Dynamic Programming: Solving the Curses of Dimensionality: Second Edition*. Wiley Blackwell, 9 2007.

Precup, D., Sutton, R. S., and Singh, S. P. Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 759–766, 2000.

Precup, D., Sutton, R. S., and Dasgupta, S. Off-policy temporal difference learning with function approximation. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 417–424, 2001.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 1st edition, 1994.

Quinzan, F., Soleymani, A., Jaillet, P., Rojas, C. R., and Bauer, S. Drcfs: Doubly robust causal feature selection. In *International Conference on Machine Learning*, pp. 28468–28491, 2023.

Rabin, M. O. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1, 7 1969.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

Reiersöl, O. Confluence analysis by means of instrumental sets of variables. *astronomi och fysik*, 1945.

Riedmiller, M. Neural fitted Q iteration-first experiences with a data efficient neural reinforcement learning method. *European Conference on Machine Learning*, 2005.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 9 1951.

Robins, J. M., Rotnitzky, A., and Zhao, L. P. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association*, 89(427):846–866, 1994.

Robinson, P. M. Root-n-consistent semiparametric regression. *Econometrica*, 56:931, 7 1988.

Ross, S. and Bagnell, J. A. Efficient reductions for imitation learning. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010.

Ross, S., Gordon, G. J., and Bagnell, J. A. A reduction of imitation learning and structured prediction to no-regret online learning. *Journal of Machine Learning Research*, 15:627–635, 11 2011.

Ruan, K., Zhang, J., Di, X., and Bareinboim, E. Causal imitation learning via inverse reinforcement learning. *Proceedings at the International Conference on Learning Representations*, 2023.

Russell, S. Learning agents for uncertain environments. *In The Eleventh Annual Conference on Computational Learning Theory*, 1998.

Sadeghi, F. and Levine, S. CAD2RL: Real single-image flight without a single real image. *Robotics: Science and Systems*, 13, 11 2016.

Sadigh, D., Kim, E. S., Coogan, S., Sastry, S. S., and Seshia, S. A. A learning based approach to control synthesis of Markov decision processes for linear temporal logic specifications. *Proceedings of the IEEE Conference on Decision and Control*, 2015-February(February):1091–1096, 2014.

Saengkyongam, S., Thams, N., Peters, J., and Pfister, N. Invariant policy learning: A causal perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:8606–8620, 6 2023.

Schaal, S. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3:233–242, 6 1999.

Schiewer, R., Subramoney, A., and Wiskott, L. Exploring the limits of hierarchical world models in reinforcement learning. *Scientific Reports 2024 14:1*, 14:1–24, 11 2024.

Schmidt-Hieber, J. Nonparametric regression using deep neural networks with relu activation function. *The Annals of Statistics*, 48(4):1875, 2020.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., and Silver, D. Mastering atari, Go, chess and shogi by planning with a learned model. *Nature 2020 588:7839*, 588(7839): 604–609, 12 2020.

Schulman, J., Levine, S., Moritz, P., Jordan, M., and Abbeel, P. Trust region policy optimization. *32nd International Conference on Machine Learning, ICML 2015*, 3:1889–1897, 2 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Openai, O. K. Proximal policy optimization algorithms. *arXiv:1707.06347*, 7 2017.

Schwab, P., Linhardt, L., Bauer, S., Buhmann, J. M., and Karlen, W. Learning counterfactual representations for estimating individual dose-response curves. *34th AAAI Conference on Artificial Intelligence*, pp. 5612–5619, 2 2019.

Seitzer, M., Schölkopf, B., and Martius, G. Causal influence detection for improving efficiency in reinforcement learning. *Advances in Neural Information Processing Systems*, 27:22905–22918, 6 2021.

Shalit, U., Johansson, F. D., and Sontag, D. Estimating individual treatment effect: generalization bounds and algorithms. *34th International Conference on Machine Learning, ICML 2017*, 6:4709–4718, 6 2017.

Shao, D. and Kwiatkowska, M. Sample efficient model-free reinforcement learning from LTL specifications with optimality guarantees. *Proceedings of the International Joint Conference on Artificial Intelligence*, 5 2023.

Shao, D., Fesser, L., and Kwiatkowska, M. STR-Cert: Robustness certification for deep text recognition on deep learning pipelines and vision transformers. *arXiv:2401.05338*, 11 2023.

Shao, D., Soleymani, A., Quinzan, F., and Kwiatkowska, M. Learning decision policies with instrumental variables through double machine learning. *Proceedings of the International Conference on Machine Learning*, 2024.

Shao, D., Buening, T. K., and Kwiatkowska, M. A unifying framework for causal imitation learning with hidden confounders. *International Conference on Learning Representations, Workshop on Spurious Correlation and Shortcut Learning: Foundations and Solutions*, 2 2025a.

Shao, D., Soleymani, A., Quinzan, F., and Kwiatkowska, M. Double machine learning for conditional moment restrictions: Iv regression, proximal causal learning and beyond. *arXiv:2506.14950*, 6 2025b.

Shi, X., Miao, W., Nelson, J. C., and Tchetgen, E. J. T. Multiply robust causal inference with double negative control adjustment for categorical unmeasured confounding. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 82:521–540, 8 2020.

Shpitser, I. and Pearl, J. Complete identification methods for the causal hierarchy. *Journal of Machine Learning Research*, 9:1941–1979, 2008.

Sickert, S., Esparza, J., Jaax, S., and Křetínský, J. Limit-deterministic Büchi automata for linear temporal logic. *Computer Aided Verification*, 9780:312–332, 2016.

Siegel, N. Y., Springenberg, J. T., Berkenkamp, F., Abdolmaleki, A., Neunert, M., Lampe, T., Hafner, R., Heess, N., and Riedmiller, M. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *8th International Conference on Learning Representations*, 2 2020.

Silver, D., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. *Proceedings of the 31 st International Conference on Machine Learning*, 2014.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., Van Den Driess-

che, G., Graepel, T., and Hassabis, D. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 10 2017.

Singh, R., Sahani, M., and Gretton, A. Kernel instrumental variable regression. *Advances in Neural Information Processing Systems*, 32, 6 2019.

Słoczyński, T. and Wooldridge, J. M. A general double robustness result for estimating average treatment effects. *Econometric Theory*, 34(1):112–133, 2018.

Soleymani, A., Raj, A., Bauer, S., Schölkopf, B., and Besserve, M. Causal feature selection via orthogonal search. *Transactions on Machine Learning Research*, 2022.

Sontakke, S. A., Mehrjou, A., Itti, L., and Schölkopf, B. Causal curiosity: RL agents discovering self-supervised experiments for causal representation learning. *Proceedings of the 38th International Conference on Machine Learning*, 139:9848–9858, 10 2021.

Spencer, J., Choudhury, S., Venkatraman, A., Ziebart, B., and Bagnell, J. A. Feedback in imitation learning: The three regimes of covariate shift. *arXiv:2102.02872*, 2 2021.

Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Schölkopf, B., and Lanckriet, G. R. G. On integral probability metrics, phi-divergences and binary classification. *arXiv:0901.2698*, 1 2009.

Subramanian, C. and Ravindran, B. Causal contextual bandits with targeted interventions. In *International Conference on Learning Representations*, 1 2022.

Sutton, R. S. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163, 7 1991.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An Introduction*, volume 3. The MIT Press, 2018.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12, 1999.

Sutton, R. S., Mahmood, A. R., and White, M. An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Research*, 17, 3 2015.

Swamy, G., Choudhury, S., Bagnell, J. A., and Wu, Z. S. Of moments and matching: A game-theoretic framework for closing the imitation gap. *Proceedings of Machine Learning Research*, 139:10022–10032, 3 2021.

Swamy, G., Choudhury, S., Bagnell, J. A., and Wu, Z. S. Sequence model imitation learning with unobserved contexts. *Advances in Neural Information Processing Systems*, 35, 8 2022a.

Swamy, G., Choudhury, S., Bagnell, J. A., and Wu, Z. S. Causal imitation learning under temporally correlated noise. *Proceedings of Machine Learning Research*, 162:20877–20890, 2 2022b.

Swanson, S. A., Hernán, M. A., Miller, M., Robins, J. M., and Richardson, T. S. Partial identification of the average treatment effect using instrumental variables: Review of methods for binary instruments, treatments, and outcomes. *Journal of the American Statistical Association*, 113:933, 4 2018.

Syrgkanis, V. and Zampetakis, M. Estimation and inference with trees and forests in high dimensions. In *Conference on learning theory*, pp. 3453–3454. PMLR, 2020.

Tassa, Y., Erez, T., and Todorov, E. Synthesis and stabilization of complex behaviors through online trajectory optimization. *IEEE International Conference on Intelligent Robots and Systems*, pp. 4906–4913, 2012.

Tchetgen, E. J. T., Ying, A., Cui, Y., Shi, X., and Miao, W. An introduction to proximal causal learning. *Statistical Science*, 39:375–390, 8 2024.

Thomas, P. S. Bias in natural actor-critic algorithms. In *Proceedings of Machine Learning Research*, pp. 441–448. PMLR, 1 2014.

Thomas, P. S. and Brunskill, E. Data-efficient off-policy policy evaluation for reinforcement learning. *33rd International Conference on Machine Learning*, 5:3158–3167, 4 2016.

Thomas, P. S., Theocharous, G., and Ghavamzadeh, M. High-confidence off-policy evaluation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1):3000–3006, 2 2015.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. *IEEE International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

Vaezipoor, P., Li, A. C., Icarte, R. T., and Mcilraith, S. LTL2Action: generalizing LTL instructions for multi-task RL. *Proceedings of the International Conference on Machine Learning*, 2021.

Valko, M., Korda, N., Munos, R., Flaounas, I., and Cristianini, N. Finite-time analysis of kernelised contextual bandits. *Uncertainty in Artificial Intelligence - Proceedings of the 29th Conference*, pp. 654–663, 9 2013.

Van Der Vaart, A. W., Wellner, J. A., van der Vaart, A. W., and Wellner, J. A. *Weak convergence*. Springer, 1996.

Van Handel, R. Probability in high dimension. *Lecture Notes (Princeton University)*, 2014.

Vapnik, V. N. *Statistical learning theory*. Wiley, 1998.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-December:5999–6009, 6 2017.

Vuorio, R., Brehmer, J., Ackermann, H., Dijkman, D., Cohen, T., and de Haan, P. Deconfounded imitation learning. *arXiv:2211.02667*, 11 2022.

Wainwright, M. J. High-dimensional statistics: A non-asymptotic viewpoint. *Cambridge University Press*, pp. 1–552, 1 2019.

Wang, L., Yang, Z., and Wan, Z. Provably efficient causal reinforcement learning with confounded observational data. *Advances in Neural Information Processing Systems*, 25:21164–21175, 2020a.

Wang, S., Yin, X., Li, S., and Yin, X. Tractable reinforcement learning for signal temporal logic tasks with counterfactual experience replay. *IEEE Control Systems Letters*, 8:616–621, 2024.

Wang, Y. S., Weng, T. W., and Daniel, L. Verification of neural network control policy under persistent adversarial perturbation. In *Proceedings of the 37th International Conference on Machine Learning*, 2020b.

Wang, Y. X., Agarwal, A., and Dudík, M. Optimal and adaptive off-policy evaluation in contextual bandits. *34th International Conference on Machine Learning*, 7:5472–5494, 12 2016.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H. V., Lanctot, M., and Frcitas, N. D. Dueling network architectures for deep reinforcement learning. *33rd International Conference on Machine Learning, ICML 2016*, 4:2939–2947, 11 2015.

Wang, Z., Xiao, X., Xu, Z., Zhu, Y., and Stone, P. Causal dynamics learning for task-

independent state abstraction. *Proceedings of Machine Learning Research*, 162:23151–23180, 6 2022.

Watkins, C. J. C. H. and Dayan, P. Q-learning. *Machine Learning 1992 8:3*, 8(3):279–292, 5 1992.

Weisstein, E. W. Asymptotic notation, 2023.

Wen, C., Lin, J., Darrell, T., Jayaraman, D., and Gao, Y. Fighting copycat agents in behavioral cloning from observation histories. *Advances in Neural Information Processing Systems*, 2020-December, 10 2020.

Wright, P. G. The tariff on animal and vegetable oils. *https://doi.org/10.1086/254144*, 38: 619–620, 10 1928.

Wu, A., Kuang, K., Li, B., and Wu, F. Instrumental variable regression with confounder balancing. *Proceedings of the International Conference on Machine Learning,*, 2022.

Wu, A., Kuang, K., Xiong, R., and Wu, F. Instrumental variables in causal inference and machine learning: A survey. *ACM Computing Surveys*, 9 2023a.

Wu, A., Kuang, K., Xiong, R., Zhu, M., Liu, Y., Li, B., Liu, F., Wang, Z., and Wu, F. Learning instrumental variable from data fusion for treatment effect estimation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 8 2023b.

Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. *arXiv:1911.11361*, 11 2019.

Wu, Y., Fu, Y., Wang, S., and Sun, X. Doubly robust proximal causal learning for continuous treatments. *Proceedings of the International Conference on Learning Representations*, 9 2024.

Wyatt, L. H., Peterson, G. C. L., Wade, T. J., Neas, L. M., and Rappold, A. G. Annual pm2.5 and cardiovascular mortality rate data: Trends modified by county socioeconomic status in 2,132 us counties. *Data in brief*, 30, 6 2020.

Xu, L., Chen, Y., Srinivasan, S., de Freitas, N., Doucet, A., and Gretton, A. Learning deep features in instrumental variable regression. *9th International Conference on Learning Representations*, 10 2020.

Xu, L., Kanagawa, H., and Gretton, A. Deep proxy causal learning and its application to confounded bandit policy evaluation. *Advances in Neural Information Processing Systems*, 31:26264–26275, 6 2021.

Xu, Y., Zhu, J., Shi, C., Luo, S., and Song, R. An instrumental variable approach to confounded off-policy evaluation. *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Xu, Z. and Topcu, U. Transfer of temporal logic formulas in reinforcement learning. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

Yang, Y. and Barron, A. Information-theoretic determination of minimax rates of convergence. *Annals of Statistics*, pp. 1564–1599, 1999.

Yao, Y., Liu, Z., Cen, Z., Zhu, J., Yu, W., Zhang, T., and Zhao, D. Constraint-conditioned policy optimization for versatile safe reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 10 2023.

Yarotsky, D. Optimal approximation of continuous functions by very deep ReLU networks. In *Conference on learning theory*, pp. 639–649. PMLR, 2018.

Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., and Wu, Y. The surprising

effectiveness of PPO in cooperative, multi-agent games. *Advances in Neural Information Processing Systems*, 35, 3 2021a.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J., Levine, S., Finn, C., and Ma, T. MOPO: Model-based offline policy optimization. *Proceedings of Conference on Neural Information Processing Systems*, 2020.

Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., and Finn, C. COMBO: Conservative offline model-based policy optimization. *Advances in Neural Information Processing Systems*, 35:28954–28967, 2 2021b.

Yuan, J., Kuang, K., Li, B. O., Wu, A., Li, B., Wu, R., Wu, F., and Lin, L. Auto IV: Counterfactual prediction via automatic instrumental variable decomposition. *ACM Transactions on Knowledge Discovery from Data*, 16:1–20, 7 2021.

Yuan, J., Wu, A., Kuang, K., Li, B., Wu, R., Wu, F., and Lin, L. Auto iv: Counterfactual prediction via automatic instrumental variable decomposition. *ACM Transactions on Knowledge Discovery from Data*, 16:1–20, 1 2022.

Zhang, A., Lyle, C., Sodhani, S., Filos, A., Kwiatkowska, M., Pineau, J., Gal, Y., and Precup, D. Invariant causal prediction for block MDPs. *Proceedings of the International Conference on Machine Learning*, 2020a.

Zhang, H., Chen, H., Xiao, C., Li, B., Liu, M., Boning, D., and Hsieh, C. J. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in Neural Information Processing Systems*, 2020-December, 3 2020b.

Zhang, J. and Bareinboim, E. Transfer learning in multi-armed bandits: A causal approach. *Proceedings of the International Joint Conference on Artificial Intelligence*, 2017.

Zhang, J. and Bareinboim, E. Designing optimal dynamic treatment regimes: A causal

reinforcement learning approach. *Proceedings of the 37th International Conference on Machine Learning*, pp. 119, 2020.

Zhang, J., Kumor, D., and Bareinboim, E. Causal imitation learning with unobserved confounders. *Proceedings of the 34th Conference on Neural Information Processing Systems*, 8 2020c.

Zhang, J., Chen, Y., Allen, P. G., and Singh, A. Causal bandits: Online decision-making in endogenous settings. *A causal view on dynamical systems workshop at NeurIPS 2022*, 2022.

Zhao, Q., Sur, P., and Candes, E. J. The asymptotic distribution of the MLE in high-dimensional logistic models: Arbitrary covariance. *Bernoulli*, 28(3):1835–1861, 2022.

Appendix A

Learning Policies from Confounded Datasets with Instrumental Variables

A.1 The Score Function for Standard Two-Stage CMR Estimators

In this section, we show that the learning objective, or score function, for standard two-stage CMR estimators (Angrist et al., 1996; Hartford et al., 2017; Singh et al., 2019) is not Neyman orthogonal and thus cannot be used to create a DML estimator for the CMR problem.

Proposition 4.2. *The score (or objective) function for standard two-stage CMR estimators* $\ell = (Y - \widehat{g}(f, c))^2$ *is not Neyman orthogonal at* (f_0, g_0).

Proof. The score $\ell = (Y - \widehat{g}(f, c))^2$ is not Neyman orthogonal because, first of all, $\mathbb{E}[(Y - g_0(f_0, c))^2] = \mathbb{E}[(Y - \mathbb{E}[Y|C])^2] \neq 0$ since $\mathbb{E}[f_0(X)|C] = \mathbb{E}[Y|C]$ and $Y - \mathbb{E}[Y|C] \neq 0$ due to the noise on Y. This violates the basic condition for a Neyman orthogonal score that the score function equals zero with the true functions f_0 and g_0.

Secondly, the Gateaux derivative against small changes in g for score $\mathbb{E}[(Y - g_0(f_0, c))^2]$

at (f_0, g_0) is

$$\begin{aligned}
&\frac{\partial}{\partial r}\mathbb{E}\Big[(Y - g_0(f_0, C) - r \cdot g(f_0, C))^2\Big] \\
&\quad = \frac{\partial}{\partial r}\mathbb{E}\Big[(Y - g_0(f_0, C))^2 - 2r \cdot (Y - g_0(f_0, C))g(f_0, C) + r^2 \cdot g(f_0, C)^2\Big] \\
&\quad = \mathbb{E}\Big[2(Y - g_0(f_0, C))g(f_0, C) + 2r \cdot g(f_0, C)^2\Big],
\end{aligned}$$

and, when $r = 0$, this derivative evaluates to

$$\mathbb{E}[2(Y - g_0(f_0, c))g(f_0, c)] = \mathbb{E}[2(Y - \mathbb{E}[Y|C])g(f_0, c)],$$

which does not equal 0 for general $g \in \mathcal{G}$ since generally $g(f_0, c)$ and the residual $(Y - \mathbb{E}[Y|C])$ are correlated. Therefore, this standard score function for two-stage CMR estimation is not Neyman orthogonal at (f_0, g_0). □

A.2 Proofs

In this section, we restate all the conditions required to prove the $N^{-1/2}$ convergence rate guarantees for the DML-CMR estimator, and provide the omitted proofs in Chapter 4 for Theorem 4.3, Lemma 4.7, Theorem 4.8, Corollary 4.9, and Corollary 4.12.

A.2.1 DML-CMR $N^{-1/2}$ Convergence Rate Guarantees

To obtain $N^{-1/2}$ convergence rate guarantees of the DML-CMR estimator, recall that the following conditions must be satisfied.

Condition 4.5 [Conditions for $N^{-1/2}$ convergence of DML, Assumption 3.3 and 3.4 in Chernozhukov et al. (2018)]

For sample size $N \geq 3$:

(a) The map $(\theta, (s, g)) \mapsto \mathbb{E}[\psi(\mathcal{D}; f_\theta, (s, g))]$ is twice continuously Gateaux-differentiable.

(b) The score ψ obeys the Neyman orthogonality conditions.

(c) The true parameter θ_0 obeys $\mathbb{E}[\psi(\mathcal{D}; f_{\theta_0}, (s_0, g_0))] = 0$ and Θ contains a ball of radius $c_1 N^{-1/2} \log N$ centered at θ_0.

(d) For all $\theta \in \Theta$, the identification relationship

$$2\|\mathbb{E}[\psi(\mathcal{D}; f_\theta, (s_0, g_0))]\| \gtrsim \|J_0(\theta - \theta_0)\|$$

is satisfied, where $J_0 := \partial_{\theta'}\{\mathbb{E}[\psi(\mathcal{D}; f_{\theta'}, (s_0, g_0))]\}|_{\theta'=\theta_0}$ is the Jacobian matrix, with singular values bounded between $c_0 > 0$ and $c_1 > 0$.

(e) Let K be a fixed integer. Given a random partition $\{I_k\}_{k=1}^K$ of indices $[N]$ each of size $n = N/K$, the nuisance parameter estimator $\widehat{s}_k$ and $\widehat{g}_k$ learnt using data with indices I_k^c belongs to shrinking realisation sets $\mathcal{S}_N$ and $\mathcal{G}_N$ respectively, and the nuisance parameters should be estimated at the $o(N^{-1/4})$ rate, e.g., $\|\widehat{s} - s_0\|_2 = o(N^{-1/4})$.

To formalise the convergence rate guarantees in relation to the technical conditions, we present the following proposition as a direct consequence of Lemma 6.3 of Chernozhukov et al. (2018).

Proposition A.1 (DML convergence rate of $N^{-1/2}$)**.** *If all conditions in Condition 4.5 hold, then the DML estimator $\widehat{\theta}$ as defined in Definition 4.1 is concentrated in a $1/\sqrt{N}$ neighbourhood of θ_0 with probability $1 - o(1)$:*

$$\sqrt{N}\|\widehat{\theta} - \theta_0\| = O_p(\rho_N)$$

where $\{\rho_N\}_{N\geq 1} = o(1)$ is some sequence of positive constants converging to zero converges to 0. This implies that, for all $\zeta \in (0, 1]$, there exists $K > 0$ such that, for all $N \geq K$, $\|\widehat{\theta} - \theta_0\| = O(N^{-1/2})$ with probability $1 - \zeta$.

Proof. This proposition is a slight variant of Lemma 6.3 from Chernozhukov et al. (2018). We will reuse partial results from Lemma 6.3, and specifically the following two technical bounds on some quantity of interest that we will use for this proof. We denote by $\eta = (s, g)$ the nuisance parameters and $\mathcal{T}_N = (\mathcal{S}_N, \mathcal{G}_N)$ the realisation set of the nuisance parameters.

From step 1 of Lemma 6.3 (Chernozhukov et al., 2018), we have that there exists a sequence of positive constants $\{\tau_N\}_{N\geq 1}$ that converges to zero, such that $\|\widehat{\theta} - \theta_0\| \leq \tau_N$ with probability $1 - o(1)$.

From step 4 of Lemma 6.3 (Chernozhukov et al., 2018), with probability $1-o(1)$, there exists $q > 2$ such that

$$\mathcal{I}_3 := \inf_{\theta\in\Theta} \sqrt{n}\, \|\mathbb{E}_n[\psi(\mathcal{D}; \theta, \widehat{\eta}_0)]\| = O\big(n^{-1/2+1/q} \log n + o(N^{-1/4})^2 \sqrt{n}\big)\,. \tag{A.1}$$

From step 3 of Lemma 6.3 (Chernozhukov et al., 2018), with probability $1 - o(1)$, we have that

$$\mathcal{I}_5 := \sup_{\|\theta-\theta_0\|\leq\tau_N} \big\|\mathbb{G}_n\big(\psi(\mathcal{D}; \theta, \widehat{\eta}_0) - \psi(\mathcal{D}; \theta_0, \eta_0)\big)\big\| = O(n^{-1/2+1/q} \log n). \tag{A.2}$$

For a set of data indices I, we define the empirical expectation as:

$$E_n(\psi(\mathcal{D})) := \frac{1}{n} \sum_{i\in I} \psi(\mathcal{D}_i);$$

and define the empirical process $G_n(\psi(\mathcal{D}))$ as a linear operator acting on measurable functions $\psi : \mathcal{D} \to \mathbb{R}$ such that $\|\psi\|_{P,2} < \infty$:

$$\begin{aligned} G_n(\psi(\mathcal{D})) &:= \frac{1}{\sqrt{n}} \sum_{i\in I} \psi(\mathcal{D}_i) - \int \psi(w) dP(w). \\ &:= E_n(\psi(\mathcal{D})) - E_{P_N}(\psi(\mathcal{D})). \end{aligned}$$

Now, we prove the claim of the proposition. First, by definition of $\widehat{\theta}_0$ and the compact-

ness of Θ, we have

$$\sqrt{n}\left\|\mathbb{E}_n[\psi(\mathcal{D};\widehat{\theta}_0,\widehat{\eta}_0)]\right\| = \inf_{\theta\in\Theta}\sqrt{n}\left\|\mathbb{E}_n[\psi(\mathcal{D};\theta,\widehat{\eta}_0)]\right\|.$$

Moreover, for any $\theta \in \Theta$ and $\eta \in \mathcal{T}_N$, we have

$$\begin{aligned}\sqrt{n}\mathbb{E}_n[\psi(\mathcal{D};\theta,\eta)] &= \sqrt{n}\mathbb{E}_n[\psi(\mathcal{D};\theta_0,\eta_0)] \\ &\quad + G_n[\psi(\mathcal{D};\theta,\eta)-\psi(\mathcal{D};\theta_0,\eta_0)] + \sqrt{n}\big(\mathbb{E}_{P_N}[\psi(\mathcal{D};\theta,\eta)]\big), \qquad \text{(A.3)}\end{aligned}$$

where we are using that $\mathbb{E}_{P_N}[\psi(\mathcal{D};\theta_0,\eta_0)] = 0$. Finally, by Taylor's expansion of the function $r \mapsto \mathbb{E}_{P_N}[\psi(\mathcal{D};\theta_0+r(\theta-\theta_0),\eta_0+r(\eta-\eta_0))]$ and the fact that $\mathbb{E}_{P_N}[\psi(\mathcal{D};\theta_0,\eta_0)] = 0$, we have

$$\begin{aligned}\mathbb{E}_{P_N}[\psi(\mathcal{D};\theta,\eta)] &= J_0(\theta-\theta_0) + \partial_\eta\mathbb{E}_{P_N}\psi(\mathcal{D};\theta_0,\eta_0)[\eta-\eta_0] \\ &\quad + 0.5\int_0^1 \partial_r^2\mathbb{E}_{P_N}[\mathcal{D};\theta_0+r(\theta-\theta_0),\eta_0+r(\eta-\eta_0)]dr. \qquad \text{(A.4)}\end{aligned}$$

Therefore, since $\|\partial_\eta\mathbb{E}_{P_N}[\psi(\mathcal{D};\theta_0,\eta_0)]\|\|\widehat{\eta}_0-\eta_0\| = 0$ by Neyman orthogonality, we apply (A.3) with $\theta = \tilde{\theta}_0$ and $\eta = \widehat{\eta}_0$, and substitute in (A.4) to obtain that

$$\sqrt{n}\left\|\mathbb{E}_n[\psi(\mathcal{D};\theta_0,\eta_0)] + J_0(\tilde{\theta}_0-\theta_0)\right\| \le \epsilon_N\sqrt{n} + \mathcal{I}_3 + \mathcal{I}_4 + \mathcal{I}_5 \quad \text{with probability } 1-o(1),$$

where we have that

$$\begin{aligned}\mathcal{I}_4 &:= \sqrt{n}\sup_{\|\theta-\theta_0\|\le\tau_N,\eta\in\mathcal{T}_N}\left\|\int_0^1 \partial_r^2\mathbb{E}_{P_N}\psi[\mathcal{D};\theta_0+r(\theta-\theta_0),\eta_0+r(\eta-\eta_0)]dr\right\| \\ &\le o(\|\eta-\eta_0\|)^2\sqrt{n} = o(N^{-1/4})^2\sqrt{n},\end{aligned}$$

by Condition 4.5 (e) and the fact that the curvature of the score function is bounded within $\mathcal{T}_N$.

Therefore, since all singular values of J_0 are bounded below from zero by Condition 4.5 (d), it follows that

$$\left\| J_0^{-1}\sqrt{n}\mathbb{E}_n[\psi(\mathcal{D};\theta_0,\eta_0)] + \sqrt{n}(\tilde{\theta}_0 - \theta_0) \right\| = O\left(n^{-1/2+1/q}\log n + o(n^{-1/4})^2\sqrt{n}\right) \quad \text{(A.5)}$$

$$= O\left(o(1) + o(n^{-1/2})n^{1/2}\right) = o(1). \quad \text{(A.6)}$$

Since our score function $\psi(\mathcal{D};\theta_0,\eta_0) = \mathbb{E}[Y - f_{\theta_0}(X)|C] = 0$ almost surely at true parameters (θ_0,η_0) for all $c \in \mathcal{C}$, the empirical expectation $\mathbb{E}_n[\psi(\mathcal{D};\theta_0,\eta_0)] = 0$, and thus, with probability $1 - o(1)$,

$$\left\| \sqrt{n}(\tilde{\theta}_0 - \theta_0) \right\| = o(1), \quad \text{(A.7)}$$

as required. □

We will show that all these conditions are satisfied in the proof of Theorem 4.8. To begin with, we prove Theorem 4.3, which shows that our score function ψ is Neyman orthogonal.

Theorem 4.3 (Neyman orthogonality)**.** *The score function* $\psi(\mathcal{D}; f, (s,g)) = (s(c) - g(f,c))^2$ *obeys the Neyman orthogonality conditions at* $(f_0, (s_0, g_0))$.

Proof. Firstly, by Equation (2.14), we have $s_0(C) = g_0(f_0, C)$, thus

$$\psi(\mathcal{D}; f_0, (s_0, g_0)) = \mathbb{E}\Big[(s_0(C) - g_0(f_0, C))^2\Big] = 0.$$

Then, we compute the derivative w.r.t. small changes in the nuisance parameters. For all

$s, g \in \mathcal{S}, \mathcal{G}$,

$$\begin{aligned}
&\frac{\partial}{\partial r}\mathbb{E}\Big[(s_0(C) + r \cdot s(C) - g_0(f_0, C) - r \cdot g(f_0, C))^2\Big] \\
&= \frac{\partial}{\partial r}\mathbb{E}\Big[2r(s_0(C) - g_0(f_0, C))(s(C) - g(f_0, C)) + r^2(s(C) - g(f_0, C))^2\Big] \\
&= \mathbb{E}\Big[2(s_0(C) - g_0(f_0, C))(s(C) - g(f_0, C)) + 2r(s(C) - g(f_0, C))^2\Big],
\end{aligned}$$

and, when at $r = 0$, the derivative evaluates to

$$\mathbb{E}\Big[2(s_0(C) - g_0(f_0, C))(s(C) - g(f_0, C))\Big] = \mathbb{E}\Big[0 \times (s(C) - g(f_0, C))\Big] = 0 \; \forall s, g \in \mathcal{S}, \mathcal{G},$$

since $s_0(C) = \mathbb{E}[Y|C] = \mathbb{E}[f_0(X)|C] = g_0(f_0, C)$. Therefore, our moment function ψ is Neyman orthogonal at $(f_0, (s_0, g_0))$. □

In turn, we state and prove the convergence of the nuisance parameters $\widehat{s}$ and $\widehat{g}$ with the machinery developed for the analysis of the excess risk in constrained ERM in Appendix A.2.3.

Lemma 4.7 (Convergence of nuisance parameters)**.** *Under Assumption 4.6, let $\mathcal{S}_N^*$ be the star-hull of the realisation set $\mathcal{S}_N$ of function class $\mathcal{S}$,*

$$\mathcal{S}_N^* = \{C \mapsto \gamma(s(C) - s_0(C)) \; : \; s \in \mathcal{S}_N, \gamma \in [0, 1]\},$$

$\mathcal{P}_N^*$ be the star-hull of the realisation set $\mathcal{P}_N$ of the function class $\mathcal{P}$,

$$\mathcal{P}_N^* = \{C \mapsto \gamma(F(\cdot|C) - F_0(\cdot|C)) \; : \; F \in \mathcal{P}_N, \gamma \in [0, 1]\},$$

and $\mathcal{G}_N^*$ be the star-hull of the realisation set $\mathcal{G}_N$ of the function class $\mathcal{G}$,

$$\mathcal{G}_N^* = \{C, f \mapsto \gamma(g(C, f) - g_0(C, f)) \; : \; g \in \mathcal{G}_N, \gamma \in [0, 1]\},$$

where $\mathcal{S}_N$, $\mathcal{P}_N$ and $\mathcal{G}_N$ are properly shrinking neighbourhoods of the true functions s_0, F_0 and g_0. Then, there exist universal constants c_1 and c_2, for which we have that with probability at least $1-\xi$, the estimation errors are bounded as

$$\|\widehat{s}-s_0\|_2^2 \leq c_1\left(\delta_N(\mathcal{S}_N^*)^2+\sqrt{\frac{\log(1/\zeta)}{N}}\right);$$
$$\|\widehat{g}-g_0\|_2^2 \leq c_2\left(\delta_N(\mathcal{P}_N^*)^2+\sqrt{\frac{\log(1/\zeta)}{N}}\right).$$

Proof. The bound on estimation error for $\widehat{s}$ is straightforward and follows directly from Theorem A.5. In order to prove that, we only need to show that the choice of loss function for the ERM estimator $\widehat{s}$ is Lipschitz in its first argument. To this end, we formulate the ERM as

$$\widehat{s}=\arg\min_{s\in\mathcal{S}}\mathcal{L}_N(s), \quad \text{where } \mathcal{L}_N(s)=\frac{1}{N}\sum_{i=1}^{N}\ell(s;y_i,c_i) \text{ and } \ell(s;y,c)=(y-s(c))^2.$$

Thus, we have,

$$\begin{aligned}|\ell(s_2;y,c_2)-\ell(s_1;y,c_1)| &= \left|(y-s_2(c_2))^2-(y-s_1(c_1))^2\right| \\ &= |(2y+s_2(c_2)+s_1(c_1))(s_2(c_2)-s_1(c_1))| \\ &\leq 4B|s_2(c_2)-s_1(c_1)|, \end{aligned} \tag{A.8}$$

where we used the fact that $y, \|s\|_\infty \leq B$ by Assumption 4.6. Thus, $\ell(s;y,c)$ is $4B$-Lipschitz in its argument and thus, by Theorem A.5, for a universal constant c_1,

$$\|\widehat{s}-s_0\|_2^2 \leq c_1\left(\delta_N(\mathcal{S}_N^*)^2+\sqrt{\frac{\log(1/\zeta)}{N}}\right).$$

For the estimation error of $\widehat{g}$, we recall that for any $F(\cdot|C) \in \mathcal{P}$,

$$g_F(f_\theta, c) := \int f_\theta(x) F(x|C = c) dx.$$

Thus, we can connect the estimation error of $\widehat{g}_{\widehat{F}}$ to the estimation error of the conditional density estimator F by showing

$$\begin{aligned} \|\widehat{g} - g_0\|_2 &= \|\widehat{g}_{\widehat{F}} - g_{F_0}\|_2 \\ &\leq B\|\widehat{F} - F_0\|_2 \end{aligned} \tag{A.9}$$

To prove (A.9), we observe that for any C and any test function $f \in \mathcal{F}$, we have that,

$$\begin{aligned} |g_{F_1}(C, f) - g_{F_2}(C, f)| &= \left| \int f(x)[F_1 - F_2](dx|C) \right| \\ &\leq \int |f(x)||F_1 - F_2|(dx|C) \\ &\leq B \int |F_1 - F_2|(dx|C), \end{aligned} \tag{A.10}$$

since $\|f\|_\infty \leq B$ by Assumption 4.6. Thus, by integrating w.r.t. joint law of (C, f), and the (A.10), we can show that (A.9) holds since

$$\begin{aligned} \|g_{F_1} - g_{F_2}\|_2^2 &= \mathbb{E}_C\left[(g_{F_1}(C, f) - g_{F_2}(C, f))^2\right] \\ &\leq B^2 \mathbb{E}_C\left[\left(\int |F_1 - F_2|(dx|C)\right)^2\right] \\ &\leq B^2 \mathbb{E}_C\left[\int [F_1 - F_2]^2 (dx|C)\right] \\ &= B^2 \|F_1 - F_2\|_2^2, \end{aligned} \tag{A.11}$$

for any $F_1, F_2 \in \mathcal{P}$, where in (A.11), we used Cauchy–Schwarz. Having (A.9) at hand, it suffices to prove the upper bound on the estimation error of $\widehat{F}$. To this end, we observe that

Squared CDF error,

$$\ell(F; x, c) = [\mathbb{1}_{\{X \leq x \mid C=c\}} - \mathscr{F}(x|c)]^2,$$

is Lipschitz by the similar argument as (A.8), where $\mathscr{F}(x|c)$ is the conditional CDF induced by the conditional density F. It is not hard to observe that *clipped* versions of other losses for density estimation such as integrated squared error (ISE), negative log-likelihood, and Hellinger-squared also satisfy the Lipschitz condition in the first argument. Thus, by Theorem A.5, for a universal constant c_3,

$$\|\widehat{F} - F_0\|_2^2 \leq c_3 \left(\delta_N(\mathcal{P}_{N^*})^2 + \sqrt{\frac{\log(1/\zeta)}{N}} \right).$$

Choosing $c_2 = B^2 c_3$ and (A.9) completes the proof. □

Now, we are ready to prove Theorem 4.8, which is the main theorem that states the $N^{-1/2}$ convergence rate guarantees for our DML estimator.

Theorem 4.8 (Convergence of the DML estimator for CMRs)**.** *Let $f_{\theta_0} \in \mathcal{F}$ be a solution that satisfies the CMRs in Equation* (2.9)*, let ψ be the Neyman orthogonal score defined in Equation* (4.6) *and let $J_0 := \partial_{\theta'}\{\mathbb{E}[\psi(\mathcal{D}; f_{\theta'}, (s_0, g_0))]\}|_{\theta'=\theta_0}$ be the Jacobian matrix of $\mathbb{E}[\psi]$ w.r.t. θ. Suppose that the upper bound of the critical radius $\delta_N = o(N^{-1/4})$ for $\widehat{s}$, $\widehat{g}$, and J_0 has bounded singular values. Then, if Assumption 4.4 and 4.6 hold, our DML estimator $f_{\widehat{\theta}}$ satisfies that $\widehat{\theta}$ is concentrated in a $N^{-1/2}$ neighbourhood of θ_0 with probability $1 - o(1)$. Specifically, for all $\zeta \in (0, 1]$, there exists $K > 0$ such that for all integer $N \geq K$,*

$$\|\widehat{\theta} - \theta_0\| = O(N^{-1/2})$$

with probability $1 - \zeta$.

Proof. Following Proposition A.1, we need to check whether, under Assumption 4.6, all of

Condition 4.5 for the $N^{-1/2}$ convergence rate for DML is satisfied. Condition (a) is satisfied since $(s-g)^2$ is twice continuously differentiable with respect to s and g. Condition (b) is satisfied by Theorem 4.3. Condition (c) is satisfied since f_{θ_0} satisfies the CMRs and, from Theorem 4.3, we have that $\mathbb{E}[\psi(\mathcal{D}; f_{\theta_0}, (s_0, g_0))] = 0$. In addition, from Assumption 4.4, since the true parameter $\theta_0 \in \Theta$ is in the interior of Θ, Θ contains some neighbourhood centered at the true parameter θ_0.

Condition (d) is a sufficient identifiability condition, which states that the closeness of the score function at point θ to zero implies the closeness of θ to θ_0. This condition is standard in condition moment problems and implies that the *ill-posedness* (see Definition 4.10) of the CMR problem is bounded, as shown in Section 4.3.5. To check condition (d), we first point out that, under analytical assumptions for s, g, and h, we can write down first order Taylor series for the score function $\mathbb{E}[\psi(\mathcal{D}; f_\theta, (s_0, g_0))]$ around the point θ_0,

$$\mathbb{E}[\psi(\mathcal{D}; f_\theta, (s_0, g_0))] = \mathbb{E}[\psi(\mathcal{D}; f_{\theta_0}, (s_0, g_0))] + J_0(\theta - \theta_0) + O(\|\theta - \theta_0\|^2).$$

Plugging in the validity of the score function $\psi(\mathcal{D}; f_\theta, (s_0, g_0))$, i.e., $\mathbb{E}[\psi(\mathcal{D}; f_{\theta_0}, (s_0, g_0))] = 0$, we infer that

$$\|\mathbb{E}[\psi(\mathcal{D}; f_\theta, (s_0, g_0))]\| \gtrsim \|J_0(\theta - \theta_0)\|.$$

Now for identifiability, we only need to check that $J_0 J_0^T$ is non-singular, which is guaranteed by the bounded singular value of J_0 as stated in the Theorem.

Condition (e) is satisfied since we have the critical radius $\delta_N = o(N^{-1/4})$, and together with Lemma 4.7, the nuisance parameters converge sufficiently quickly to ensure $\|\widehat{s} - s_0\|_2 \leq O(\delta_N + N^{-1/2}) = O(o(N^{-1/4}) + N^{-1/2}) = o(N^{-1/4})$ and similarly $\|\widehat{g} - g_0\|_2 \leq O(\delta_N + N^{-1/2}) = o(N^{-1/4})$.

Therefore, all the conditions in Condition 4.5 are satisfied, which concludes the proof by Proposition A.1. □

Corollary 4.9 (Estimation error bounds)**.** *Let $f_{\widehat{\theta}}$ be the DML estimator for CMRs. If all assumptions for Theorem 4.8 hold and there exists a constant $L > 0$ such that $\|f_\theta(x) - f_{\theta_0}(x)\|_2 \leq L\|\theta - \theta_0\|_2$ for all $x \in \mathcal{X}$ and $\theta \in \Theta$, then, for all $\zeta \in (0, 1]$, we have that, for all $\zeta \in (0, 1]$, there exists $K > 0$ such that for all integer $N \geq K$,*

$$\|f_{\widehat{\theta}} - f_{\theta_0}\|_2 = O\left(L \cdot N^{-1/2}\right),$$

with probability $1 - \zeta$.

Proof. From Theorem 4.8, we have that the parameters $\widehat{\theta}$ for our DML estimator $f_{\widehat{\theta}}$ learnt from a dataset of size N satisfy that, for all $\zeta \in (0, 1]$, there exist $M, K > 0$ such that for all integer $N \geq K$,

$$\mathbb{P}\left(\|\widehat{\theta} - \theta_0\| \leq M \cdot N^{-1/2}\right) \geq 1 - \zeta.$$

If we let L to be a constant such that $\|f_\theta(x) - f_{\theta_0}(x)\| \leq L\|\theta - \theta_0\|$ for all $x \in \mathcal{X}$ and $\theta \in \Theta$, we have that, for all $\zeta \in (0, 1]$, there exist $M, K > 0$ such that for all integer $N \geq K$,

$$\mathbb{P}\left(\|f_{\widehat{\theta}}(x) - f_{\theta_0}(x)\| \leq L \cdot M \cdot N^{-1/2}\right) \geq 1 - \zeta,$$

which implies $\|f_{\widehat{\theta}} - f_{\theta_0}\| = O\left(L \cdot N^{-1/2}\right)$ with probability 1-ζ as required. □

A.2.2 Ill-posedness and DML Identification

Proposition 4.11. *For all $\theta \in \Theta$, if there exists a constant $L > 0$ such that $\|f_\theta(x) - f_{\theta_0}(x)\|_2 \leq L\|\theta - \theta_0\|_2$ for all $x \in \mathcal{X}$ and $\theta \in \Theta$, then Condition 4.5 (d), which states*

$$2\|\mathbb{E}[\psi(\mathcal{D}; f_\theta, (s_0, g_0))]\| \geq \|J_0(\theta - \theta_0)\|$$

and the Jacobian matrix J_0 have singular values bounded between $c_0 > 0$ and $c_1 > 0$, implies the ill-posedness is bounded by $\nu \leq L/\sqrt{c_0}$.

Proof. Recall that our orthogonal score function is $\psi(\mathcal{D}; f_\theta, (s, g)) = (s(c) - g(f, c))^2$, where $\psi(\mathcal{D}; f_\theta, (s_0, g_0)) = (s_0(c) - g_0(f, c))^2 = (\mathbb{E}[Y - f(X)|C])^2$. Under a finite-dimensional parameterised setting, we have that from $2\|\mathbb{E}[\psi(\mathcal{D}; f_\theta, (s_0, g_0))]\| \gtrsim \|J_0(\theta - \theta_0)\|$,

$$\begin{aligned}
\|\mathbb{E}[f_{\theta_0}(X) - f_\theta(X)|C]\|_2^2 =& \|\mathbb{E}[(\mathbb{E}[f_{\theta_0}(X) - f_\theta(X)|C])^2]\| \\
=& \|\mathbb{E}[(\mathbb{E}[Y - f_\theta(X)|C])^2]\| \\
=& \|\mathbb{E}[\psi(\mathcal{D}; f_\theta, (s_0, g_0))]\| \\
\geq& \frac{1}{2}\|J_0(\theta - \theta_0)\| \\
=& \sqrt{(\theta - \theta_0)^T (J_0^T J_0)(\theta - \theta_0)} \\
\geq& \frac{1}{2}\sqrt{c_0^2 \|(\theta - \theta_0)\|_2^2} \geq \frac{1}{2} c_0 \|(\theta - \theta_0)\|_2 \geq \frac{1}{2} c_0 \|(\theta - \theta_0)\|_2^2
\end{aligned} \tag{A.12}$$

for $\|(\theta - \theta_0)\| \leq 1$ and the singular value lower bound $c_0 > 0$ of J_0. With a local Lipschitz condition of f_θ around θ_0: $\|f_\theta(x) - f_{\theta_0}(x)\|_2 \leq L\|\theta - \theta_0\|_2$ for all $x \in \mathcal{X}$ and $\theta \in \Theta$, we have that

$$\begin{aligned}
\|f_{\theta_0} - f_\theta\|_2^2 &= \mathbb{E}[(f_{\theta_0}(x) - f_\theta(x))^2] \\
&\leq \mathbb{E}[(L\|\theta_0 - \theta\|)^2] \\
&\leq L^2 \|\theta_0 - \theta\|^2; \\
\implies \|\theta_0 - \theta\|^2 &\geq \frac{\|f_{\theta_0} - f_\theta\|_2^2}{L^2}.
\end{aligned} \tag{A.13}$$

Therefore, from Equation (A.12) and Equation (A.13), we have that

$$\|\mathbb{E}[f_{\theta_0}(X) - f_\theta(X)|C]\|_2^2 \geq c_0\|(\theta - \theta_0)\|_2^2 \geq \frac{c_0\|f_{\theta_0} - f_\theta\|_2^2}{L^2}$$
$$\implies \|\mathbb{E}[f_{\theta_0}(X) - f_\theta(X)|C]\|_2 \geq \sqrt{c_0}\|(\theta - \theta_0)\|_2 \geq \frac{\sqrt{c_0}\|f_{\theta_0} - f_\theta\|_2}{L},$$

which bounds the ill-posedness by

$$\nu = \sup_{f \in \mathcal{F}} \frac{\|f_{\theta_0} - f_\theta\|_2}{\|\mathbb{E}[f_{\theta_0}(X) - f_\theta(X)|C]\|_2} \leq \frac{L}{\sqrt{c_0}}. \tag{A.14}$$

□

A.2.3 Constrained Empirical Risk Minimisation Bounds

In this section, we introduce basic concepts from empirical process theory and further discuss bounds on the excess risk of general empirical risk minimizers (ERM) in the style of Wainwright (2019); Foster & Syrgkanis (2019).

Definition A.2. The critical radius denoted by $\delta_N(\mathcal{H}^*)$ is defined as the minimum δ that satisfies the following upper bound on the local Gaussian complexity of a star-shaped function class $\mathcal{H}^*$[1], $\mathcal{G}(\mathcal{H}^*, \delta) \leq \delta^2/2$, where local Gaussian complexity is defined as

$$\mathcal{G}(\mathcal{H}^*, \delta) = \mathbb{E}_\epsilon \left[\sup_{h \in \mathcal{H}^* : \|h\|_N \leq \delta} \langle \epsilon, h \rangle \right], \tag{A.15}$$

with ϵ being a random i.i.d. zero-mean Gaussian vector.

The critical radius is a standard notion to bound the estimation error in the regression problem. Since local Gaussian complexity can be viewed as an expected value of a supremum of a stochastic process indexed by g, we can apply empirical process theory tools, namely the Dudley's entropy integral (Wainwright, 2019; Van Handel, 2014), to provide a

[1] A function class $\mathcal{H}$ is star-shaped if for every $h \in \mathcal{H}$ and $\alpha \in [0, 1]$, we have $\alpha h \in \mathcal{H}$.

bound on the critical radius,

$$\mathcal{G}(\mathcal{H}^*, \delta) \leq \inf_{\alpha \geq 0} \left\{ \alpha + \frac{1}{\sqrt{N}} \int_{\alpha/4}^{\delta} \sqrt{\log \mathcal{N}(\mathcal{H}^*, L^2(\mathbb{P}_N), \epsilon)}\, d\epsilon \right\},$$

where $\mathcal{N}(\mathcal{H}^*, L^2(\mathbb{P}_N), \epsilon)$ is the ϵ-covering number of the function class $\mathcal{H}^*$ in $L^2(\mathbb{P}_N)$ norm. Now, by placing $\alpha = 0$, when the integral is a single scale value of $\sqrt{\log \mathcal{N}(\mathcal{H}^*, L^2(P_n), \epsilon)}$, we infer that

$$\mathcal{G}(\mathcal{H}^*, \delta) \leq \frac{\delta}{\sqrt{N}} \sqrt{\log \mathcal{N}(\mathcal{F}^*, L^2(\mathbb{P}_N), \epsilon)}.$$

Thus, the critical radius of $\mathcal{H}^*$ will be upper bounded by

$$\delta_N(\mathcal{H}^*) \lesssim \frac{\sqrt{\log \mathcal{N}(\mathcal{H}^*, L^2(\mathbb{P}_N), \epsilon)}}{\sqrt{N}} = O(d_N(\mathcal{H}^*)^{1/2} N^{-1/2}),$$

where Chernozhukov et al. (2022b, 2021) referred to

$$d_N(\mathcal{H}^*) := \inf \left\{ d > 0 : \log \mathcal{N}(\mathcal{H}^*, L^2(\mathbb{P}_N), \epsilon) \leq d \log \left(\frac{C}{\epsilon} \right) \ \forall \epsilon \in (0,1) \text{ and } C \text{ is a constant.} \right\},$$

as the effective dimension of the hypothesis space. Note that this rate matches the minimax lower bound of fixed design estimation for this setting (Yang & Barron, 1999).

Given the dataset $\mathcal{D} = \{z_i \in \mathcal{Z}\}_{i=1}^N$ consisting of i.i.d. data points z_i drawn from distribution $\mathbb{P}$, and a function class $\mathcal{H}$, we define the realisation of a function space by subscript N, e.g., $\mathcal{H}_N$ is the realisation of $\mathcal{H}$ in the N observed data points. Since the definition of critical radius is for star-shaped function classes, we equip ourselves with the star-hull notation, where $\mathcal{H}_N^*$ is the star-hull of the function class $\mathcal{H}_N$ centred at the true function h_0, defined as

$$\mathcal{H}_N^* := \{Z \mapsto \gamma(h(X) - h_0(Z)) \ : \ f \in \mathcal{H}_N, \gamma \in [0,1]\},$$

and denote its critical radius by $\delta_N(\mathcal{H}_N^*)$, or simply $\delta_N(\mathcal{H})$. In statistical learning, we are given a loss function $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$, and we have the risk and empirical risk defined accordingly as:

$$\mathcal{L}(h) = \mathbb{E}_z[\ell(h; z)] \quad \text{and} \quad \mathcal{L}_N(h) = \frac{1}{N}\sum_{i=1}^{N} \ell(h; z_i).$$

Let us denote the ground truth function by h_0, i.e., the minimizer of the risk, as:

$$h_0 = \arg\min_{h \in \mathcal{H}} \mathcal{L}(h).$$

The ERM algorithm proposes the estimator $\widehat{h}$ that minimises the empirical risk of the observed N data points,

$$\widehat{h} = \arg\min_{h \in \mathcal{H}} \mathcal{L}_N(h).$$

Theorem A.3 (Foster & Syrgkanis (2019, Lemma 7))**.** *Consider a function class $\mathcal{H}$ and its star-hull $\mathcal{H}^*$, with $\sup_{h \in \mathcal{H}^*} \|h\|_\infty \leq 1$, critical radius $\delta_N(\mathcal{H}^*)$, and any choice of δ such that,*

$$\delta^2 \geq \max\left\{\delta_N(\mathcal{H}^*)^2, \frac{4\log(41\log(2c_1 N))}{c_1 N}\right\},$$

for a constant c_1. Moreover, assume that the loss function ℓ is L-Lipschitz in its first argument with respect to ℓ_2 norm. Then there exist universal constants c_2 and c_3 such that with probability at least $1 - c_2 \exp\{c_3 N \delta^2\}$,

$$|(\mathcal{L}_N(h) - \mathcal{L}_N(h_0)) - (\mathcal{L}(h) - \mathcal{L}(h_0))| \leq 18L\delta\{\|h - h_0\|_2 + \delta\}, \qquad \forall h \in \mathcal{H}.$$

We mention an immediate corollary of Theorem A.3 for functions h that are bounded by B, which follows by rescaling arguments.

Corollary A.4. *Consider a function class $\mathcal{H}$ and its star-hull $\mathcal{H}^*$, with $\sup_{h\in\mathcal{H}^*}\|h\|_\infty \le B$, critical radius $\delta_N(\mathcal{H}^*)$, and any choice of δ such that,*

$$\delta^2 \ge \max\left\{\delta_N(\mathcal{H}^*)^2, \frac{4\log(41\log(2c_1N))}{c_1N}\right\},$$

for a constant c_1. Moreover, assume that the loss function ℓ is L-Lipschitz in its first argument with respect to ℓ_2 norm. Then there exist universal constants c_2 and c_3 such that with probability at least $1 - c_2\exp\{c_3N\delta^2/\max\{1,B\}^2\}$,

$$|(\mathcal{L}_N(h) - \mathcal{L}_N(h_0)) - (\mathcal{L}(h) - \mathcal{L}(h_0))| \le \frac{18L\delta}{\max\{1,B\}}\{\|h-h_0\|_2 + \delta\}, \qquad \forall h \in \mathcal{H}.$$

Proof. If $B \le 1$, the proof follows trivially from Theorem A.3, thus let us assume otherwise. Define the B-scaled function class $\widetilde{\mathcal{H}}$,

$$\widetilde{\mathcal{H}} := \left\{\widetilde{h} \ : \ B\widetilde{h} \in \mathcal{H}\right\}.$$

Then, the loss function $\ell(h,z) = \ell(B\widetilde{h}, z)$ is LB-Lipschitz in $\widetilde{h}$ and by homogeneity of the local Gaussian averages, i.e.,

$$\mathcal{G}(\widetilde{\mathcal{H}}^*, r) = \frac{1}{B}\mathcal{G}(\mathcal{H}^*, Br),$$

wee see that $\widetilde{\delta} := \delta/B$ satisfies the fixed point condition (A.15). Putting these pieces together and invoking Theorem A.3 for $\widehat{\mathcal{H}}$ completes the proof. □

We can equivalently write Corollary A.4 in the failure probability format. That is, for a target failure probability $0 < \xi < 1$, define

$$\delta(\xi) := \delta_N(\mathcal{H}^*) + \max\{1,B\}\sqrt{\frac{1}{c_3N}\log(1/\xi)},$$

then with probability at least $1-\xi$,

$$|(\mathcal{L}_N(h)-\mathcal{L}_N(h_0))-(\mathcal{L}(h)-\mathcal{L}(h_0))| \leq \frac{18L\delta(\xi)}{\max\{1,B\}}\{\|h-h_0\|_2+\delta(\xi)\}, \qquad \forall h \in \mathcal{H}. \tag{A.16}$$

Now, we are ready to state and prove the following master theorem for the excess risk of the constrained ERM. For the analysis, in line with Chernozhukov et al. (2021), we require that the population risk has positive curvature for identifiability purposes. Then, the generalisation bound in terms of excess risk can be converted into estimation error.

Theorem A.5 (Estimation Error of Constrained ERM)**.** *Assume that the population risk $\mathcal{L}$ has a positive curvature, i.e., for a positive number λ,*

$$\mathcal{L}(h)-\mathcal{L}(h_0) \geq \frac{\lambda}{2}\|h-h_0\|_2^2 \qquad \forall h \in \mathcal{H}, \tag{A.17}$$

and the loss function $|\ell(h;z)| \leq M$ is bounded and L-Lipschitz in its first argument w.r.t. ℓ_2 norm. Then, the solution to the ERM algorithm:

$$\widehat{h} := \arg\min_{h\in\mathcal{H}} \mathcal{L}_N(h), \tag{A.18}$$

has the following estimation error with probability at least $1-\xi$,

$$\|\widehat{h}-h_0\|_2^2 \leq C\left[(\mathcal{L}(h^*)-\mathcal{L}(h_0))+\delta_N(\mathcal{H}^*)^2+\sqrt{\frac{\log(1/\xi)}{N}}\right].$$

for a universal constant C, where $h^* = \arg\inf_{h\in\mathcal{H}} \mathcal{L}(h)$. In addition, if h_0 is realisable, i.e., $h_0 \in \mathcal{H}$, then,

$$\|\widehat{h}-h_0\|_2^2 \leq C\left[\delta_N(\mathcal{H}^*)^2+\sqrt{\frac{\log(1/\xi)}{N}}\right].$$

Proof. First, we want to upper bound the population excess risk, $\mathcal{L}(\widehat{h})-\mathcal{L}(h_0)$, and then,

by the curvature of $\mathcal{L}$, we can convert this bound into an upper bound on the estimation error $\|\widehat{h} - h_0\|_2$.

To begin with, we have,

$$\mathcal{L}(\widehat{h}) - \mathcal{L}(h_0) = \underbrace{\Big(\mathcal{L}(\widehat{h}) - \mathcal{L}_N(\widehat{h})\Big)}_{\text{(I)}} + \underbrace{\Big(\mathcal{L}_N(\widehat{h}) - \mathcal{L}_N(h^*)\Big)}_{\text{(II)}} + \underbrace{\Big(\mathcal{L}_N(h^*) - \mathcal{L}(h_0)\Big)}_{\text{(III)}}.$$

By ERM algorithm (A.18), we know that,

$$\mathcal{L}_N(\widehat{h}) \leq \mathcal{L}_N(h^*),$$

Hence, (II) ≤ 0. For the term (I),

$$\begin{aligned}
\text{(I)} &= \mathcal{L}(\widehat{h}) - \mathcal{L}_N(\widehat{h}) \\
&= (\mathcal{L} - \mathcal{L}_N)(\widehat{h} - h_0) + (\mathcal{L} - \mathcal{L}_N)(h_0),
\end{aligned}$$

and for the term (III), we know that,

$$\begin{aligned}
\text{(III)} &= \mathcal{L}_N(h^*) - \mathcal{L}(h_0) \\
&= (\mathcal{L}(h^*) - \mathcal{L}(h_0)) + (\mathcal{L}_N - \mathcal{L})(h^*) \\
&= (\mathcal{L}(h^*) - \mathcal{L}(h_0)) + (\mathcal{L}_N - \mathcal{L})(h^* - h_0) + (\mathcal{L}_N - \mathcal{L})(h_0).
\end{aligned}$$

Summing these terms from (I), (II) and (III) yields,

$$\mathcal{L}(\widehat{h}) - \mathcal{L}(h_0) \leq (\mathcal{L}(h^*) - \mathcal{L}(h_0)) + \underbrace{(\mathcal{L} - \mathcal{L}_N)(\widehat{h} - h_0)}_{\epsilon_1(\widehat{h})} + \underbrace{(\mathcal{L}_N - \mathcal{L})(h^* - h_0)}_{\epsilon_2}.$$

From Corollary A.4, especially the formulation in (A.16), we know that with probability at

least $1-\xi/2$,

$$\sup_{h\in\mathcal{H}}|(\mathcal{L}_N-\mathcal{L})(h-h_0)|\leq\frac{18L\delta(\xi/2)}{\max\{1,B\}}(\|h-h_0\|_2+\delta(\xi/2)),$$

where,

$$\delta(\xi/2)=\delta_N(\mathcal{H}^*)+\max\{1,B\}\sqrt{\frac{1}{c_3N}\log(2/\xi)}. \tag{A.19}$$

Thus, we can upper bound $\epsilon_1(\widehat{h})$ as,

$$\epsilon_1(\widehat{h})=(\mathcal{L}-\mathcal{L}_N)(\widehat{h}-h_0)\leq\frac{18L\delta(\xi/2)}{\max\{1,B\}}\Big(\|\widehat{h}-h_0\|_2+\delta(\xi/2)\Big).$$

To bound the term ϵ_2, define the random variable,

$$X_i=\ell(h^*;z_i)-\ell(h_0;z_i),$$

where X_i is bounded as $|X_i|\leq 2M$. Then, by Hoeffding's inequality, with probability at least $1-\xi/2$, we infer that

$$\epsilon_2=(\mathcal{L}_N-\mathcal{L})(h^*-h_0)\leq 2\sqrt{2}M\sqrt{\frac{\log(4/\xi)}{N}}.$$

Thus, by union bound and (A.17), we conclude that with probability at leat $1-\xi$

$$\begin{aligned}
\frac{\lambda}{2}\|\widehat{h}-h_0\|_2^2 &\leq (\mathcal{L}(h^*)-\mathcal{L}(h_0)) + \frac{18L\delta(\xi/2)}{\max\{1,B\}}\Big(\|\widehat{h}-h_0\|_2+\delta(\xi/2)\Big) + 2\sqrt{2}M\sqrt{\frac{\log(4/\xi)}{N}} \\
&= (\mathcal{L}(h^*)-\mathcal{L}(h_0)) + \frac{18L\delta(\xi/2)}{\max\{1,B\}}\|\widehat{h}-h_0\|_2 + \frac{18L\delta(\xi/2)^2}{\max\{1,B\}} + 2\sqrt{2}M\sqrt{\frac{\log(4/\xi)}{N}} \\
&\leq (\mathcal{L}(h^*)-\mathcal{L}(h_0)) + \frac{\lambda}{4}\|\widehat{h}-h_0\|_2^2 + \frac{1}{\lambda}\left(\frac{18L\delta(\xi/2)}{\max\{1,B\}}\right)^2 \\
&\qquad + \frac{18L\delta(\xi/2)^2}{\max\{1,B\}} + 2\sqrt{2}M\sqrt{\frac{\log(4/\xi)}{N}} \qquad \text{(A.20)}
\end{aligned}$$

where (A.20) follows by Young's inequality, i.e., for any positive number $\lambda > 0$,

$$ax \leq \frac{\lambda}{4}x^2 + \frac{a^2}{\lambda}.$$

Therefore, by cancelling out $\frac{\lambda}{4}\|\widehat{h}-h_0\|_2^2$ from both sides,

$$\begin{aligned}
\|\widehat{h}-h_0\|_2^2 \leq C_1\Bigg[\frac{1}{\lambda}(\mathcal{L}(h^*)-\mathcal{L}(h_0)) + \left(\frac{L^2}{\lambda\max\{1,B\}^2} + \frac{L}{\max\{1,B\}}\right)\delta(\xi/2)^2 \\
+ M\sqrt{\frac{\log(1/\xi)}{N}}\Bigg],
\end{aligned}$$

for a constant C_1. In turn, by (A.19) and Young's inequality, we know that,

$$\delta(\xi/2)^2 \leq C_2\left[\delta_N(\mathcal{H}^*)^2 + \frac{\max\{1,B\}^2}{N}\log(1/\xi)\right],$$

for a constant C_2. Hence,

$$\begin{aligned}
\|\widehat{h} - h_0\|_2^2 \leq C_3 \Bigg[& \frac{1}{\lambda}(\mathcal{L}(h^*) - \mathcal{L}(h_0)) \\
& + \left(\frac{L^2}{\lambda \max\{1, B\}^2} + \frac{L}{\max\{1, B\}} \right) \left[\delta_N(\mathcal{H}^*)^2 + \max\{1, B\}^2 \frac{\log(1/\xi)}{N} \right] \\
& + M \sqrt{\frac{\log(1/\xi)}{N}} \Bigg] \\
\leq C_4 \Bigg[& \frac{1}{\lambda}(\mathcal{L}(h^*) - \mathcal{L}(h_0)) + \frac{\max\{L^2, L\}}{\min\{\lambda, 1\} \max\{1, B\}} \delta_N(\mathcal{H}^*)^2 \\
& + \frac{\max\{L^2, L\} \max\{1, B\}}{\min\{\lambda, 1\}} \frac{\log(1/\xi)}{N} + M \sqrt{\frac{\log(1/\xi)}{N}} \Bigg],
\end{aligned} \tag{A.21}$$

for constants C_3 and C_4.

Therefore, by hiding dependence on the constants, λ, B, L and M in a universal constant C, we can summarise the result (A.21) and conclude the proof,

$$\|\widehat{h} - h_0\|_2^2 \leq C \left[(\mathcal{L}(h^*) - \mathcal{L}(h_0)) + \delta_N(\mathcal{H}^*)^2 + \sqrt{\frac{\log(1/\xi)}{N}} \right].$$

□

A.2.4 Suboptimaltiy of the Decision Policy

We now present the proof of Corollary 4.12, which bounds the suboptimality of the policy learnt using DML-CMR.

Corollary 4.12. *Let the learnt policy from a dataset of size N be $\widehat{\pi}(c) := \arg\max_a f_{\widehat{\theta}}(c, a)$, where $\widehat{\theta}$ is the DML-CMR estimator. Let L be a constant such that $\|f_\theta(C, A) - f_{\theta_0}(C, A)\|_2 \leq L\|\theta - \theta_0\|_2$ for all C in the support of $\mathbb{P}_{test}$, $A \in \mathcal{A}$, and $\theta \in \Theta$. Then, for all $\zeta \in (0, 1]$, we have that there exists $K > 0$ such that, for all integer $N \geq K$, the suboptimality of $\widehat{\pi}$*

satisfies

$$\textit{subopt}(\widehat{\pi}_N) = O(L \cdot N^{-1/2})$$

with probability $1 - \zeta$.

Proof. From Theorem 4.8, we have that the parameters $\widehat{\theta}$ for our DML estimator $f_{\widehat{\theta}}$ satisfy that, for all $\zeta \in (0, 1]$, there exist $K, M > 0$ such that for all integer $N \geq K$,

$$\mathbb{P}\left(\|\widehat{\theta} - \theta_0\| \leq M \cdot N^{-1/2}\right) \geq 1 - \zeta.$$

Since we have that $\|f_\theta(C, A) - f_{\theta_0}(C, A)\| \leq L\|\theta - \theta_0\|$ for all $C, A \in \text{supp}^M(C, A)$ and $\theta \in \Theta$, we have that, for all $\zeta \in (0, 1]$, there exist $M, K > 0$ such that for all integer $N \geq K$,

$$\mathbb{P}\left(\|h_{\widehat{\theta}}(C, A) - h_{\theta_0}(C, A)\| \leq L \cdot M \cdot N^{-1/2}\right) \geq 1 - \zeta \quad \forall C, A \in \text{supp}^M(C, A). \tag{A.22}$$

Next, we can show that the suboptimality of $\widehat{\pi}$ satisfies

$$\begin{aligned}
\text{subopt}(\widehat{\pi}) &= V(\pi^*) - V(\widehat{\pi}) \\
&= \mathbb{E}_{C \sim \mathbb{P}_{\text{test}}}[R \mid C, do(A = \pi^*(c))] - \mathbb{E}_{C \sim \mathbb{P}_{\text{test}}}[R \mid C, do(A = \widehat{\pi}(c))] \\
&= \mathbb{E}_{C \sim \mathbb{P}_{\text{test}}}[f_r(C, \pi^*(C)) - f_r(C, \widehat{\pi}(C))] \\
&= \mathbb{E}_{C \sim \mathbb{P}_{\text{test}}}[h(C, \pi^*(C)) - h(C, \widehat{\pi}(C))] \\
&\leq \max_{c \in \text{supp}(\mathbb{P}_{\text{test}})} (h(c, \pi^*(c)) - h(c, \widehat{\pi}(c))) \\
&\leq \max_{c \in \text{supp}(\mathbb{P}_{\text{test}})} |h(c, \pi^*(c)) - h_{\widehat{\theta}}(c, \pi^*(c))| + (h_{\widehat{\theta}}(c, \pi^*(c)) - h_{\widehat{\theta}}(c, \widehat{\pi}(c))) \qquad \text{(A.23)} \\
&+ |h_{\widehat{\theta}}(c, \widehat{\pi}(c)) - h(c, \widehat{\pi}(c))| \\
&\leq 2L \cdot M \cdot N^{-1/2} \quad \text{with probability } 1 - \zeta, \qquad \text{(A.24)}
\end{aligned}$$

by Equation (A.22) and the fact that $h_{\widehat{\theta}}(C, \pi^*(C)) - h_{\widehat{\theta}}(C, \widehat{\pi}(C)) \leq 0$, where $\text{supp}(\mathbb{P}_{\text{test}})$ is the support of $\mathbb{P}_{\text{test}}$. This implies that $\text{subopt}(\widehat{\pi}_N) = O(L \cdot N^{-1/2})$ with probability $1 - \zeta$ as required. □

A.3 Datasets Details

In this section, we provide details of the datasets considered in this paper for IV regression and proximal causal learning tasks.

A.3.1 IV Regression

We first provide the details for IV regression benchmarking datasets. Recall that we denote A as the action, Y as the outcome, Z as the instrument, and X as additional observed context, and the CMR problem we solve is $\mathbb{E}[Y - f(A, X)|Z, X] = 0$.

A.3.1.1 Ticket Demand Dataset

Here, we describe the ticket demand dataset for IV regression, first introduced by Hartford et al. (2017). The observable variables are generated by the following model:

$$r = f_0((t, s), p) + \epsilon, \quad \mathbb{E}[\epsilon|t, s, p] = 0;$$
$$p = 25 + (z + 3)\psi(t) + \omega,$$

where r is the ticket sales (as the outcome variable Y) and p is the ticket price (as the action variable A). (t, s) are observed context variables, where t is the time of year and s is the customer type. The fuel price z is introduced as an instrumental variable, which only affects the ticket price p. The noises ϵ and ω are correlated with correlation $\rho \in [0, 1]$,

where in our experiments we set $\rho = 0.9$. f_0 is the true causal effect function, defined as

$$f_0((t,s),p) = 100 + (10+p)\cdot s \cdot \psi(t) - 2p,$$
$$\psi(t) = 2\left(\frac{(t-5)^4}{600} + \exp(-4(t-5)^2) + \frac{t}{10} - 2\right),$$

where $\psi(t)$ is a complex non-linear function of t plotted in Figure A.1. The offline dataset is sampled with the following distributions:

$$s \sim \text{Unif}\{1, ..., 7\}$$
$$t \sim \text{Unif}(0, 10)$$
$$z \sim \mathcal{N}(0, 1)$$
$$\omega \sim \mathcal{N}(0, 1)$$
$$\epsilon \sim \mathcal{N}(\rho\omega, 1 - \rho^2).$$

From the observations (r, p, t, s, z), we estimate $\widehat{h}$ using IV regression methods, and the mean squared error between $\widehat{h}$ and the true causal function f_0 is computed on 10000 random samples from the above model. For the out-of-distribution test samples, we sample $t \sim$ Unif$(1, 11)$ instead.

We standardise the action and outcome variables p and r to centre the data around a mean of zero and a standard deviation of one following Hartford et al. (2017). This is standard practice for DNN training, which improves training stability and optimization efficiency.

A.3.1.2 Ticket Demand High-Dimensional Setting

For the high-dimensional setting, we again follow Hartford et al. (2017) to replace the customer type $s \in [7]$ in the low-dimensional setting with images of the corresponding handwritten digits from the MNIST dataset (LeCun & Cortes, 2010). For each digit $d \in [7]$,

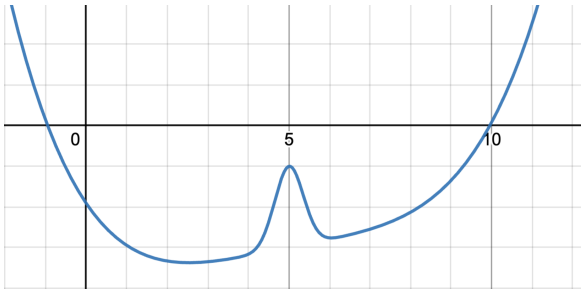


Figure A.1: A graph of the nonlinear function $\psi(t)$ in the ticket demand dataset for IV regression.

we select a random MNIST image from the digit class d as the new customer type variable s. The images are $28 \times 28 = 784$ dimensional.

A.3.1.3 Real-World Datasets

Following previously studied causal inference methods (Shalit et al., 2017; Wu et al., 2023b; Schwab et al., 2019; Bica et al., 2020), we consider two semi-synthetic real-world datasets IHDP[2] (Hill, 2011) and PM-CMR[3] (Wyatt et al., 2020) for experiments, since the true counterfactual prediction function is rarely available for real-world datasets.

IHDP, the Infant Health and Development Program (IHDP), comprises 747 units with 6 pre-treatment continuous variables, one action variable, and 19 discrete variables related to the children and their mothers, aiming at evaluating the effect of specialist home visits on the future cognitive test scores of premature infants. From the original data, we select all 6 continuous covariance variables as our context variable X.

PM-CMR studies the impact of PM2.5 particle level on the cardiovascular mortality rate (CMR) in 2132 counties in the United States using data provided by the National Studies on Air Pollution and Health (Wyatt et al., 2020). We use 6 continuous variables about CMR in each city as our context variable X.

Following Wu et al. (2023b), from the context variables X obtained from real-world datasets, we generate the instrument Z, the action A, and the outcome Y using the follow-

[2]IHDP: `https://www.fredjo.com/`.
[3]PM-CMR:`https://doi.org/10.23719/1506014`.

ing model:

$$
\begin{aligned}
&Z \sim \mathbb{P}(Z = z) = 1/K, \quad z \in [1..K]; \\
&A = \sum_{z=1}^{K} 1_{Z=z} \sum_{i=1}^{d_X} w_{iz}(X_i + 0.2\epsilon + f_z(z)) + \delta_A, \quad w_{iz} \sim \text{Unif}(-1, 1); \\
&Y = 9A^2 - 1.5A + \sum_{i=1}^{d_X} \frac{X_i}{d_X} + |X_1 X_2| - \sin(10 + X_2 X_3) + 2\epsilon + \delta_Y,
\end{aligned}
$$

where X_i denotes the i-th variable in X, f_z is a function that returns different constants depending on the input z, $\delta_Y, \delta_A \sim \mathcal{N}(0, 1)$ and $\epsilon \sim \mathcal{N}(0, 0.1)$ is the unobserved confounder. The fully generated semi-synthetic datasets IHDP and PM-CMR have 747 and 2132 samples respectively, and we randomly split them into training (63%), validation (27%), and testing (10%) following Wu et al. (2023b).

A.3.2 Proximal Causal Learning

Next, we provide details for benchmarking datasets of proximal causal learning. Recall that we denote A as the treatment, Y as the outcome, W as the outcome proxy, and V as the treatment proxy. The CMR problem we are solving is $\mathbb{E}[Y - f(A, W)|A, V]$.

A.3.2.1 Ticket Demand Dataset

The ticket demand dataset (Hartford et al., 2017) is also extended to the PCL setting, which is first introduced by Xu et al. (2021). The data generating process is described by the

following model:

$$
\begin{aligned}
U &\sim \text{Unif}(0, 10) \\
[V_1, V_2] &= [2\sin(2\pi U/10) + \epsilon_1, 2\cos(2\pi U/10) + \epsilon_2] \\
W &= 7g(U) + 45 + \epsilon_3 \\
A &= 35 + (V_1 + 3)g(U) + V_2 + \epsilon_4 \\
Y &= A \cdot \min(\exp(\frac{W - A}{10}, 5) - 5g(U) + \epsilon_5 \\
\text{with} \quad g(u) &= 2(\frac{(u-5)^4}{600} + \exp(-4(u-5)^2) + u/10 - 2) \\
\text{and} \quad \epsilon_i &\sim \mathcal{N}(0, 1),
\end{aligned}
$$

where U is the demand, which acts as the hidden confounder, V_1, V_2 are fuel prices, which act as treatment proxy, W is the web page views, which acts as outcome proxy, A is the price, and Y is the sale. Here, we can see that the outcome proxy W and the treatment proxy V are both affected by U, where W directly affects the outcome and V directly affects the treatment A.

A.3.2.2 dSprites high-dimensional Dataset

The dSprites dataset (Matthey et al., 2017) is a high-dimensional (64×64) image dataset described by five latent parameters: *shape*, *scale*, *rotation*, *posX* and *poxY*. It is proposed by Xu et al. (2021) to adopt it as a benchmark for PCL where the treatment is each figure and the hidden confounder is *posY*. For the experiments, we fix the *shape* to be heart.

The data-generating process can be described by the following steps:

1. Randomly generate values for *scale, rotation, posX* and posY:
   scale $\sim$ Unif$\{0.5, 0.6, ..., 1.0\}$, *rotation* $\sim$ Unif$(0, 2\pi)$, *posX, posY* $\sim$ Unif$\{0, ..., 31\}$.

2. Set U=*posY*

3. Set V=*(scale,rotation,posX)*

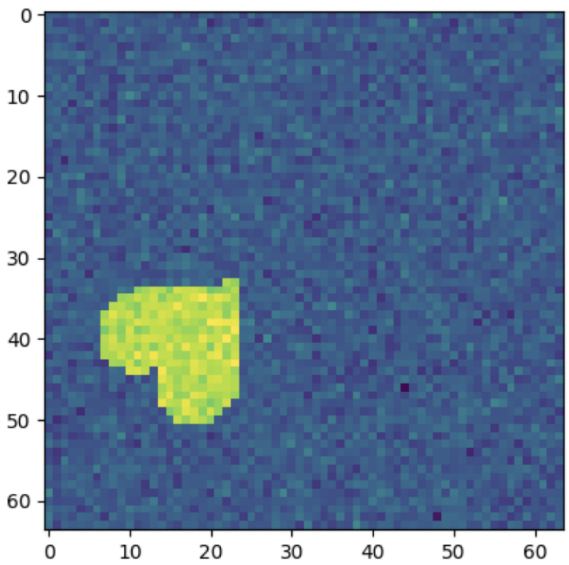


Figure A.2: An example of dSprites image, which is used as the treatment A in PCL experiments. Its scale, rotation, x position and y position are randomly generated.

4. Set A as the dSprites image with features *(scale, rotation, posX, posY)* and add Gaussian noise $\mathcal{N}(0, 0.1)$ to each pixel.

5. Set W as posY with Gaussian noise $\mathcal{N}(0, 1)$.

6. $Y = \frac{0.1\|vec(A)^T B\|_2^2 - 5000}{1000} \times \frac{(31 \times U - 15.5)^2}{85.25} + \epsilon, \epsilon \sim \mathcal{N}(0, 0.5)$, where the matrix $B \in \mathbb{R}^{64\times 64}$ is given by $B_{i,j} = |32 - j|/32$.

For the test dataset, a fixed grid of image parameters is chosen:

$$\textit{posX} \in [0, 5, 10, 15, 20, 25, 30]$$

$$\textit{posY} \in [0, 5, 10, 15, 20, 25, 30]$$

$$\textit{scale} \in [0.5, 0.8, 1.0]$$

$$\textit{rotation} \in [0, 0.5\pi, \pi, 1.5\pi],$$

which consists of 588 images to reliably evaluate different PCL algorithms.

A.4 Network Architecture and Hyperparameters

Here, we describe the network architecture and hyperparameters of all experiments. Unless otherwise specified, all neural network algorithms are optimised using AdamW (Loshchilov & Hutter, 2017) with learning rate $= 0.001$, $\beta = (0.9, 0.999)$ and $\epsilon = 10^{-8}$. In addition, we set $K = 10$ for K-fold cross-fitting in DML-CMR.

A.4.1 IV Regression

We first introduce details for the IV regression experiments.

A.4.1.1 Ticket Demand Dataset

For DML-CMR and CE-DML-CMR, we use the network architecture described in Table A.1. We use a learning rate of 0.0002 with a weight decay of 0.001 (L2 regularisation) and a dropout rate of $\frac{1000}{5000+N}$ that depends on the data size N. For DeepGMM, we use the same structure as the outcome network of DML-CMR with dropout $= 0.1$ and the same learning rate as DML-CMR. For DFIV, we follow the original structure proposed in Xu et al. (2020) with regularisers $\lambda 1$, $\lambda 2$ both set to 0.1 and weight decay of 0.001. For DeepIV, we use the same network architectures as the action network and stage 2 network for DML-CMR, with the dropout rate in Hartford et al. (2017) and weight decay of 0.001. For KIV, we use the Gaussian kernel, where the bandwidth is determined by the median trick as originally described by Singh et al. (2019), and we use the random Fourier feature trick with 100 dimensions.

A.4.1.2 Ticket Demand with MNIST

For DML-CMR and CE-DML-CMR, we use a convolutional neural network (CNN) feature extractor, which we denote as *ImageFeature*, described in Table A.2, for all networks. The full network architecture is described in Table A.3; we use weight decay of 0.05. For

Layer Type	Configuration
Input	C
FC + ReLU	in:3 out:128
Dropout	-
FC + ReLU	in:128 out:64
Dropout	-
FC + ReLU	in:64 out:32
Dropout	-
MixtureGaussian	10

(a) Action Network for $\widehat{g}$

Layer Type	Configuration
Input	C
FC + ReLU	in:3 out:128
Dropout	-
FC + ReLU	in:128 out:64
Dropout	-
FC + ReLU	in:64 out:32
Dropout	-
FC	in:32 out:1

(b) Outcome Network for $\widehat{s}$

Layer Type	Configuration
Input	C, A
FC + ReLU	in:3 out:128
Dropout	-
FC + ReLU	in:128 out:64
Dropout	-
FC + ReLU	in:64 out:32
Dropout	-
FC	in:32 out:1

(c) Stage 2 Network for $\widehat{h}$

Table A.1: Network architecture for DML-CMR and CE-DML-CMR for the ticket demand low-dimensional dataset for IV regression. For the input layer, we provide the input variables. For mixture of Gaussians output, we report the number of components. The dropout rate is given in the main text.

DeepGMM, we use the same structure as the outcome network of DML-CMR, with a dropout rate of 0.1 and weight decay of 0.05. For DFIV, we follow the original structure proposed in Xu et al. (2020) with regularisers $\lambda 1$, $\lambda 2$ both set to 0.1 and weight decay of 0.05. For DeepIV, we use the same network architecture as the action network and stage 2 network for DML-CMR, with the dropout rate in Hartford et al. (2017) and weight decay of 0.05. For KIV, we use the Gaussian kernel, where the bandwidth is determined by the median trick as originally described by Singh et al. (2019), and we use the random Fourier feature trick with 100 dimensions.

Layer Type	Configuration
Input	28×28
Conv + ReLU	$3 \times 3 \times 32$, s:1, p:0
Max Pooling	2×2, s:2
Dropout	-
Conv + ReLU	$3 \times 3 \times 64$, s:1, p:0
Max Pooling	2×2, s:2
Dropout	-
Conv + ReLU	$3 \times 3 \times 64$, s:1, p:0
Dropout	-
FC + ReLU	in: 576, out:64

Table A.2: Network architecture of the feature extractor used for the ticket demand dataset with MNIST for IV regression. For each convolution layer, we list the kernel size, input dimension and output dimension, where s stands for stride and p stands for padding. For max-pooling, we provide the size of the kernel. The dropout rate here is set to 0.3. We denote this feature extractor as *ImageFeature*.

A.4.1.3 IHDP and PM-CMR

For the two real-world datasets, we use the same network architectures described in Table A.1 as in the low-dimensional ticket demand setting, where the input dimension is increased to 7 for all networks. We use a dropout rate of 0.1 and weight decay of 0.001. For DeepGMM, we use the same structure as the outcome network of DML-CMR with dropout $= 0.1$. For DFIV, we also use the same network architectures as in the low dimensional ticket demand setting with regularisers $\lambda 1$, $\lambda 2$ both set to 0.1 and weight decay of 0.001. For DeepIV, we use the same network architectures as the action network and stage 2 network of DML-CMR, with a dropout rate of 0.1 and weight decay of 0.001. For KIV, we use the Gaussian kernel where the bandwidth is determined by the median trick as originally described by Singh et al. (2019), and we use the random Fourier feature trick with 100 dimensions.

A.4.2 Proximal Causal Learning

Next, we introduce details for the proximal causal learning experiments.

Layer Type	Configuration
Input	ImageFeature(C), Z
FC + ReLU	in:66 out:32
Dropout	-
MixtureGaussian	10

(a) Action Network for $\widehat{g}$

Layer Type	Configuration
Input	ImageFeature(C), Z
FC + ReLU	in:66 out:32
Dropout	-
FC	in:32 out:1

(b) Outcome Network for $\widehat{s}$

Layer Type	Configuration
Input	ImageFeature(C), A
FC + ReLU	in:66 out:32
Dropout	-
FC	in:32 out:1

(c) Stage 2 Network for $\widehat{h}$

Table A.3: Network architecture for DML-CMR and CE-DML-CMR for the ticket demand dataset with MNIST for IV regression. For the input layer, we provide the input variables. For a mixture of Gaussians output, we report the number of components. The dropout rate is given in the main text.

A.4.2.1 Ticket Demand Dataset

For DML-CMR and CE-DML-CMR, we use the network architecture described in Table A.4. We use a learning rate of 0.0001 with a weight decay of 0.001 (L2 regularisation) for the f network and 0.0001 for the s and g networks. The dropout rate is $\frac{400}{4000+N}$, which depends on the data size N. For the comparison methods, we use the default parameter values proposed in their original papers. For CEVAE, we use 1000 epochs, 0.0001 for weight decay, 10 learning samples, 20 hidden dimensions, and 5 for early stop. For DFPV, $\lambda 1$, $\lambda 2$ both set to 0.1, weight decay is 0.01 for all networks, stage 1 iteration is 20, and stage 2 iteration is 1. For KPV, we set $\lambda 1$ and $\lambda 2$ to be 0.001 with data split ratio 0.5. For NMMR, learning rate is 0.003, L2 penalty is 0.000003 with network depth 4 and width 80 for the U statistics estimator. For the V statistics estimator, network depth is 3 and width is 80 while other hyperparameters remain the same. For PKDR, the number of components is 50, gamma is 50, alpha is 35, and cross validation is 5. For PMMR, $\lambda 1$ and $\lambda 2$ are 0.01, with scale 0.5.

Layer Type	Configuration
Input	C
FC + ReLU	in:3 out:128
Dropout	-
FC + ReLU	in:128 out:64
Dropout	-
FC + ReLU	in:64 out:32
Dropout	-
MixtureGaussian	10

(a) Action Network for $\widehat{g}$

Layer Type	Configuration
Input	C
FC + ReLU	in:3 out:128
Dropout	-
FC + ReLU	in:128 out:64
Dropout	-
FC + ReLU	in:64 out:32
Dropout	-
FC	in:32 out:1

(b) Outcome Network for $\widehat{s}$

Layer Type	Configuration
Input	C, A
FC + ReLU	in:3 out:128
Dropout	-
FC + ReLU	in:128 out:64
Dropout	-
FC + ReLU	in:64 out:32
Dropout	-
FC	in:32 out:1

(c) Stage 2 Network for $\widehat{h}$

Table A.4: Network architecture for DML-CMR and CE-DML-CMR for the ticket demand dataset for PCL. For the input layer, we provide the input variables. For mixture of Gaussians output, we report the number of components. The dropout rate is given in the main text.

A.4.2.2 dSprites dataset

For the dSprites dataset, we adopt a CNN feature extractor to handle the image inputs for DML-CMR. The architecture of this feature extractor is provided in Table A.5. We use 10 components for the mixture of Gaussian model, dropout is 0.2, batch size is 100, weight decay is 0.05, learning rate is 0.001 with Adam, and the number of epochs is the closest integer to $(1000000/N) + 100$, which depends on the sample size N. For CEVAE, DFPV, KPV, NMMR and PMMR, we follow the hyperparameters and network architecture used in Kompa et al. (2022) to generate the experimental results. For PKDR, we follow the high-dimensional dataset hyperparameters used in the original paper (Wu et al., 2024) with

Layer Type	Configuration
Input	64×64
Conv + ReLU	$5 \times 5 \times 64$, s:1, p:0
Max Pooling	2×2, s:2
Dropout	-
Conv + ReLU	$5 \times 5 \times 128$, s:1, p:0
Max Pooling	2×2, s:2
Dropout	-
Conv + ReLU	$5 \times 5 \times 128$, s:1, p:0
Dropout	-
Max Pooling	2×2, s:2
FC + ReLU	in: 2048, out:128

Table A.5: Network architecture of the feature extractor used for the dSprites image dataset for PCL. For each convolution layer, we list the kernel size, input dimension and output dimension, where s stands for stride and p stands for padding. For max-pooling, we provide the size of the kernel. The dropout rate here is set to 0.2.

weight decay 0.0001, 4 layers, learning rate 0.0001, and 500 epochs.

A.4.3 Validation and Hyperparameter Tuning

Validation procedures are crucial for tuning DNN hyperparameters and optimizer parameters. All the DML-CMR and CE-DML-CMR training stages can be validated by simply evaluating the respective losses on held-out data, as discussed in Hartford et al. (2017). This allows independent validation and hyperparameter tuning of the two first stage networks (the action and the outcome networks), and performs second stage validation using the best network selected in the first stage. This validation procedure guards against the 'weak instruments' bias (Bound et al., 1995) that can occur when the instruments are only weakly correlated with the actions variable (see detailed discussion in Hartford et al. (2017)).

A.5 Failure of Standard Offline Bandit Algorithms

It has been demonstrated that standard supervised learning that does not take IVs into account fails to learn the causal function or the counterfactual prediction function from a

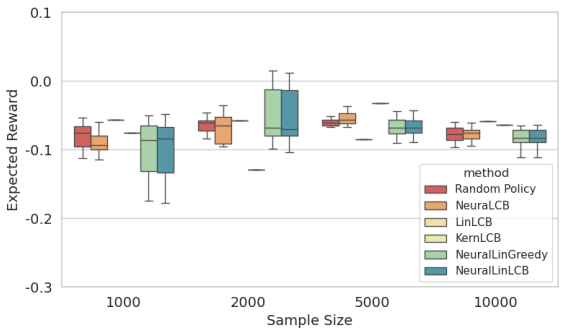


Figure A.3: Comparing the average reward obtained by policies learnt using offline bandit algorithms that do not take IVs into account with a random policy on the ticket demand dataset with low-dimensional context.

confounded offline dataset (Hartford et al., 2017). Similarly, we demonstrate here that standard offline bandit algorithms also fail to learn meaningful policies from confounded offline datasets. We evaluate PEVI, also called LinLCB (Jin et al., 2021), NeuraLCB (Nguyen-Tang et al., 2022), KernLCB (Valko et al., 2013), NeuralLinLCB (Nguyen-Tang et al., 2022) and NeuralLinGreedy (Nguyen-Tang et al., 2022) algorithms, for which we combine the context C and instrument Z variables together as the new context input for these offline bandit algorithms. For algorithms that only support discrete actions, we discretise the action space $\mathcal{A}$ into 20 discrete actions.

For all methods, we follow the network architecture and hyperparameters from the original papers, and we adopt the implementation[4] of Nguyen-Tang et al. (2022). We evaluate these methods on the ticket demand dataset described in Appendix A.3.1 and compare the average reward obtained by the learnt policies with a random policy in Figure A.3. It can be seen that all the offline bandit algorithms do not outperform a random policy while DML-IV achieves an average reward higher than 1, as shown in Figure 4.2b. This is unsurprising because these bandit methods do not exploit IVs explicitly and are unable to learn the true causal effect of actions.

[4] `https://github.com/thanhnguyentang/offline_neural_bandits`

A.6 Additional Experimental Results

In this section, we provide additional experimental results including the effects of high ill-posedness (e.g., weak IVs), performance with tree-based estimators, and a hyperparameter sensitivity analysis.

A.6.1 Effects of Weak Instruments

When the correlation between instruments and the endogenous variable (the action in our case) is weak, IV regression methods generally become unreliable (Andrews et al., 2019) because the weak correlation induces variance and bias in the first stage estimator thus induces bias in the second stage estimator, especially for non-linear IV regressions. In theory, DML-CMR should be more resistant to biases in the first stage thanks to the DML framework, as long as the causal effect is identifiable under the weak instrument. This identifiability condition is captured in Condition 4.5 for DML, and it is connected to the ill-posedness for CMR problems in general as discussed in Section 4.3.5. With identifiability, Theorem 4.8 and Corollary 4.9 all hold, and the convergence rate guarantees still apply. Intuitively, as the ill-posedness increases, worse empirical performance will be observed.

Experimentally, for the ticket demand dataset, we alter the instrument strength by changing how much the instrument z affects the price p. Recall from Appendix A.3.1 that $p = 25 + (z + 3)\psi(t) + \omega$, where ψ is a nonlinear function and ω is the noise. We add an IV strength parameter ϱ such that $p = 25 + (\varrho \cdot z + 3)\psi(t) + \omega$. In Table A.6, we present the mean and standard deviation of the MSE of $\widehat{h}$ for various IV strengths ϱ from 0.01 to 1 and sample size $N = 5000$. It is very interesting to see that DML-CMR indeed performs significantly better than SOTA nonlinear IV regression methods under weak instruments.

IV Strength	1.0	0.8	0.6	0.4	0.2	0.01
DML-CMR	**0.0676(0.0116)**	**0.0984(0.0161)**	**0.1295(0.0168)**	**0.1859(0.0376)**	**0.2899(0.0494)**	**0.4872(0.1295)**
CE-DML-CMR	**0.0765(0.0119)**	**0.1064(0.0120)**	**0.1514(0.0203)**	**0.2070(0.0329)**	**0.3194(0.0572)**	**0.5302(0.1625)**
DeepIV	0.1213(0.0209)	0.2039(0.0269)	0.3051(0.0415)	0.4476(0.0656)	0.6891(0.1210)	0.9293(0.2382)
DFIV	0.1124(0.0481)	0.1586(0.0320)	0.3080(0.1907)	0.8117(0.2779)	0.9622(0.3892)	1.6503(0.6845)
DeepGMM	0.2699(0.0522)	0.3330(0.1171)	0.4762(0.1056)	0.8666(0.2248)	1.0056(0.4334)	2.0218(0.6555)
KIV	0.2312(0.0272)	0.3149(0.0218)	0.4275(0.0368)	0.6646(0.0538)	0.8099(0.0657)	1.226(0.1014)

Table A.6: Results for the low-dimensional ticket demand dataset when the IV is weakly correlated with the action.

A.6.2 Performance of DML-CMR with Tree-Based Estimators

The DML-CMR framework allows for general estimators following the Neyman orthogonal score function. While deep learning is flexible and widely used in SOTA non-linear IV regression methods, gradient boosting and random forests are all good candidate estimators for DML-CMR. In addition, as shown in Section 4.3.3, the convergence rate and suboptimality guarantees in Corollary 4.9 and Corollary 4.12 both hold for these tree-based regressions.

Empirically, we replace the DNN estimators in DML-CMR, CE-DML-CMR, and DeepIV with Random Forests and Gradient Boosting regressors (using scikit-learn implementation). DeepIV is a good baseline for comparison, since it optimizes directly using a non-Neyman-orthogonal score and allows for direct replacement of all DNN estimators with tree-based estimators. We use 500 trees for both regressors, with minimum samples required at each leaf node of 100 for the nuisance parameters and 10 for $\widehat{h}$.

In Table A.7, we present the mean and standard deviation of the MSE of $\widehat{h}$ with Random Forests and Gradient Boosting estimators on the ticket demand dataset with various dataset sample sizes. The results demonstrate the benefits of our Neyman orthogonal score function, and interestingly, the performance of Gradient Boosting is comparable to DNN estimators.

IV Strength	Dataset Size	DNN (results in the paper)	Random Forests	Gradient Boosting
DML-CMR	2000	**0.1308(0.0206)**	0.1689(0.0172)	**0.1301(0.0112)**
CE-DML-CMR	2000	**0.1410(0.0246)**	0.1733(0.0198)	**0.1329(0.0125)**
DeepIV	2000	0.2388(0.0438)	0.2642(0.0261)	0.2052(0.0232)
DML-CMR	5000	**0.0676(0.0129)**	0.1067(0.0131)	**0.0632(0.0107)**
CE-DML-CMR	5000	**0.0765(0.0119)**	0.1154(0.0138)	**0.0699(0.0069)**
DeepIV	5000	0.1213(0.0209)	0.1626(0.0128)	0.1020(0.0091)
DML-CMR	10000	**0.0378(0.0094)**	0.0657(0.0062)	**0.0482(0.0079)**
CE-DML-CMR	10000	**0.0442(0.0070)**	0.0721(0.0039)	**0.0523(0.0059)**
DeepIV	10000	0.0714(0.0140)	0.1106(0.0080)	0.1017(0.0075)

Table A.7: Results for the low-dimensional ticket demand dataset using tree-based estimators compared to DNN estimators.

A.6.3 Sensitivity Analysis for Different Hyperparameters

The tunable hyperparameters in DML-CMR are the learning rate, network width, weight decay, and dropout rate (see Appendix A.4). As a sensitivity analysis, we provide results for the mean and standard deviation of the MSE of the DML-CMR estimator $\widehat{h}$ with different hyperparameter values for both the low-dimensional and high-dimensional datasets with sample size N=5000 in Table A.8 and Table A.9. Overall, we see that DML-CMR is not very sensitive to small changes in the hyperparameters.

Learning Rate	Weight Decay	Dropout	DNN Width	DML-CMR	CE-DML-CMR
0.0002	0.001	0.1	128	**0.0676(0.0129)**	**0.0765(0.0119)**
0.0005				0.0752(0.0122)	0.0897(0.0196)
0.0001				**0.0703(0.0195)**	**0.0794(0.0201)**
	0.0005			0.0794(0.0185)	0.0823(0.0149)
	0.005			0.0765(0.0135)	0.0809(0.0159)
	0.01			0.0820(0.0162)	0.0865(0.0174)
		0.05		**0.0715(0.0074)**	**0.0813(0.0089)**
		0.2		0.0836(0.0100)	0.0919(0.0157)
			64	0.0830(0.0162)	0.0924(0.0121)
			256	0.0943(0.0179)	0.0981(0.0126)
	0.0005	0.2		0.0805(0.0133)	0.0910(0.0106)
	0.005	0.05		**0.0672(0.0116)**	**0.0742(0.0102)**
	0.01	0.05		0.0825(0.0152)	0.0914(0.0125)
		0.2	256	0.0810(0.0129)	0.0852(0.0121)
		0.05	64	0.0907(0.0149)	0.0963(0.0161)
	0.005		256	0.0939(0.0146)	0.0991(0.0093)

Table A.8: Results for the low-dimensional ticket demand dataset for a range of hyperparameter values. The default hyperparameters in this case are: learning rate=0.0002, weight decay=0.001, dropout=0.1 and DNN width 128. The bold results are the best performing hyperparameters.

Learning Rate	Weight Decay	Dropout	CNN Channels	DML-CMR	CE-DML-CMR
0.001	0.05	0.2	64	**0.3513(0.0125)**	**0.3808(0.0150)**
0.0005				0.4063(0.0129)	0.5008(0.0369)
0.002				0.3659(0.0219)	0.4133(0.0267)
0.005				**0.3377(0.0218)**	**0.3555(0.0202)**
	0.01			0.3935(0.0176)	0.4461(0.0478)
	0.02			**0.3595(0.03013)**	**0.3851(0.0293)**
	0.1			0.4066(0.0172)	0.5160(0.0329)
		0.1		0.4136(0.0211)	0.5386(0.0398)
		0.3		0.3857(0.0171)	0.4002(0.0249)
			128	0.4176(0.01941)	0.5129(0.0630)
			256	0.4942(0.0226)	0.6180(0.0396)
	0.1	0.1		0.4163(0.0214)	0.5952(0.0343)
	0.01	0.3		0.3636(0.0186)	0.3995(0.0250)
		0.3	128	0.4006(0.0187)	0.4764(0.0216)
		0.3	256	**0.3429(0.0215)**	**0.3971(0.0264)**
	0.1		256	0.4170(0.0283)	0.5335(0.0371)

Table A.9: Results for the high-dimensional ticket demand dataset for a range of hyperparameter values. The default hyperparameters in this case are: learning rate 0.001, weight decay=0.05, dropout=0.2 and 64 CNN channels. The bold results are the best performing hyperparameters.

Appendix B

Imitating Expert Policies from Confounded Datasets

B.1 Proofs of Main Results

In this section, we provide the proofs for the main results and corollaries in Chapter 5, which we restate for convenience.

B.1.1 Proof of Propositions

Proposition 5.6. *The ill-posedness $\nu(\Pi, k)$ is monotonically increasing as the confounding horizon k increases.*

Proof. From definition, we have that,

$$\nu(\Pi, k) = \sup_{\pi \in \Pi} \frac{\|\pi_E - \pi\|_2}{\|\mathbb{E}[a_t - \pi(h_t)|h_{t-k}]\|_2}.$$

We would like to show that, for each $\pi \in \Pi$, $\frac{\|\pi_E - \pi\|_2}{\|\mathbb{E}[a_t - \pi(h_t)|h_{t-k}]\|_2}$ is increasing as k increases, which would imply that $\nu(\Pi, k)$ is increasing. For each $\pi \in \Pi$, we see that the numerator is constant w.r.t. the horizon k. Therefore, it suffices to check that for each $\pi \in \Pi$, the

denominator $\|\mathbb{E}[a_t - \pi(h_t)|h_{t-k}]\|_2$ decreases as k increases. For any two integer horizons $k_1 > k_2$,

$$\mathbb{E}[a_t - \pi(h_t)|h_{t-k_1}]^2 = \mathbb{E}[\mathbb{E}[a_t - \pi(h_t)|h_{t-k_2}]|h_{t-k_1}]^2 \tag{B.1}$$

$$\leq \mathbb{E}[\mathbb{E}[a_t - \pi(h_t)|h_{t-k_2}]^2|h_{t-k_1}] \tag{B.2}$$

$$= \mathbb{E}[a_t - \pi(h_t)|h_{t-k_2}]^2, \tag{B.3}$$

by the tower property of conditional expectation as $\sigma(h_{t-k_1}) \subseteq \sigma(h_{t-k_2})$, Jensen's inequality for conditional expectations, and the fact that $\mathbb{E}[a_t - \pi(h_t)|h_{t-k_2}]^2$ is h_{t-k_1} measurable, respectively for each line. Therefore, we have that $\mathbb{E}[a_t - \pi(h_t)|h_{t-k}]$ is decreasing, which implies that $\|\mathbb{E}[a_t - \pi(h_t)|h_{t-k}]\|_2$ is decreasing and $\nu(\Pi, k)$ is increasing as k increases, which completes the proof. □

B.1.2 Guarantees on the Imitation Gap

Theorem 5.8. *Let $\widehat{\pi}_h$ be the learnt policy with CMR error ϵ and let $\nu(\Pi, k)$ be the ill-posedness of the problem. Assume that $\delta_{TV}(u_t^o, \mathbb{E}_{\pi_E}[u_t^o|h_t]) \leq \delta$ for $\delta \in \mathbb{R}^+$, $P(u_t^\epsilon)$ is c-TV stable and π_E is deterministic. Then, the imitation gap is upper bounded by*

$$J(\pi_E) - J(\widehat{\pi}_h) \leq T^2(c\epsilon\nu(\Pi, k) + 2\delta) = \mathcal{O}(T^2(\delta + \epsilon)).$$

Proof. Recall that $J(\pi)$ is the expected reward following π, and we would like to bound the performance gap $J(\pi_E) - J(\widehat{\pi_h})$ between the expert policy π_E and the learnt history-dependent policy $\widehat{\pi_h}$. Let $Q_{\widehat{\pi_h}}(s_t, a_t, u_t^o)$ be the Q-function of $\widehat{\pi_h}$. Using the Performance Difference Lemma (Kakade & Langford, 2002), we have that, for any Q-function $\tilde{Q}(h_t, a_t)$

that takes in the trajectory history h_t and action a_t,

$$
\begin{aligned}
J(\pi_E) - J(\widehat{\pi_h}) &= \mathbb{E}_{\tau\sim\pi_E}[\sum_{t=1}^{T} Q_{\widehat{\pi_h}}(s_t, a_t, u_t^o) - \mathbb{E}_{a\sim\widehat{\pi_h}}[Q_{\widehat{\pi_h}}(s_t, a, u_t^o)]] \\
&= \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\pi_E}[Q_{\widehat{\pi_h}}(s_t, a_t, u_t^o) - \tilde{Q}(h_t, a_t) + \tilde{Q}(h_t, a_t) - \mathbb{E}_{a\sim\widehat{\pi_h}}[Q_{\widehat{\pi_h}} - \tilde{Q} + \tilde{Q}]] \\
&= \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\pi_E}[\tilde{Q} - \mathbb{E}_{a\sim\widehat{\pi_h}}[\tilde{Q}]] + \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\pi_E}[Q_{\widehat{\pi_h}} - \tilde{Q} - \mathbb{E}_{a\sim\widehat{\pi_h}}[Q_{\widehat{\pi_h}} - \tilde{Q}]].
\end{aligned}
\tag{B.4}
$$

We first bound the second part of Equation (B.4). Denote by δ_{TV} the total variation distance. For two distributions P, Q, recall the property of total variation distance for bounding the difference in expectations:

$$
|\mathbb{E}_P[f(x)] - \mathbb{E}_Q[f(x)]| \leq \|f\|_\infty \delta_{TV}(P, Q).
$$

In order to bound the second part of Equation (B.4), for any Q function, consider inferred $\tilde{Q}$ using the conditional expectation of u^o on the history h,

$$
\tilde{Q}(h_t, a_t) \coloneqq Q(s_t, a_t, \mathbb{E}_{\tau\sim\pi_E}[u_t^o | h_t]),
$$

where note that $s_t \in h_t$. We have that, when the transition trajectory $(s_t, u_t^o, u_t^\epsilon, r_t) \sim \pi_E$ follows the expert policy, for any action $\dot{a} \sim \pi$ following some policy π (in our case, it can

be π_E or $\widehat{\pi_h}$),

$$
\begin{aligned}
|\mathbb{E}_{\tau\sim\pi_E,\dot{a}\sim\pi}[Q(s_t,\dot{a},u_t)-\tilde{Q}(h_t,\dot{a})]| &= |\mathbb{E}_{\tau\sim\pi_E,\dot{a}\sim\pi}[Q(s_t,\dot{a},u_t^o)-Q(s_t,\dot{a},\mathbb{E}_{\tau\sim\pi_E}[u_t^o|h_t])]| \\
&= \big|\mathbb{E}_{u_t^o\sim\pi_E}[\mathbb{E}_{\pi_E,\pi}[Q(s_t,\dot{a},u_t^o)|u_t^o] \\
&\qquad -\mathbb{E}_{u_t^o|h_t\sim\pi_E}[\mathbb{E}_{\pi_E,\pi}[Q(s_t,\dot{a},u_t^o)|u_t^o]\big| \quad \text{(B.5)} \\
&\leq \|\mathbb{E}_{\pi_E,\pi}[Q(s_t,\dot{a},u_t^o)|u_t^o]\|_\infty \delta_{TV}(u_t^o,\mathbb{E}_{\pi_E}[u_t^o|h_t]) \\
& \qquad \text{(B.6)} \\
&\leq T\cdot\delta_{TV}(u_t^o,\mathbb{E}_{\pi_E}[u_t^o|h_t]) \quad \text{(B.7)} \\
&\leq T\delta, \quad \text{(B.8)}
\end{aligned}
$$

where Equation (B.5) uses the tower property of expectations, Equation (B.6) uses the total variation distance bound for bounded functions, Equation (B.7) uses the fact that the Q function is bounded by T and Equation (B.8) uses the condition that $\delta_{TV}(u_t^o,\mathbb{E}_{\pi_E}[u_t^o|h_t])\leq\delta$ in the theorem statement. Since Equation (B.4) holds for any choice of $\tilde{Q}$, we choose $\tilde{Q}_{\widehat{\pi_h}}(h_t,a_t) := Q_{\widehat{\pi_h}}(s_t,a_t,\mathbb{E}_{\tau\sim\pi_E}[u_t^o|h_t])$ such that we can apply Equation (B.8) twice to bound the second part of Equation (B.4):

$$
\begin{aligned}
\mathbb{E}_{\tau\sim\pi_E}[Q_{\widehat{\pi_h}}-\tilde{Q}_{\widehat{\pi_h}}-\mathbb{E}_{a\sim\widehat{\pi_h}}[Q_{\widehat{\pi_h}}-\tilde{Q}_{\widehat{\pi_h}}]] &\leq \mathbb{E}_{\tau\sim\pi_E}[Q_{\widehat{\pi_h}}-\tilde{Q}_{\widehat{\pi_h}}+|\mathbb{E}_{a\sim\widehat{\pi_h}}[Q_{\widehat{\pi_h}}-\tilde{Q}_{\widehat{\pi_h}}]|] \\
&= \mathbb{E}_{\tau\sim\pi_E}[Q_{\widehat{\pi_h}}-\tilde{Q}_{\widehat{\pi_h}}]+|\mathbb{E}_{s_t,u_t\sim\pi_E,a\sim\widehat{\pi_h}}[Q_{\widehat{\pi_h}}-\tilde{Q}_{\widehat{\pi_h}}]| \\
&\leq |\mathbb{E}_{\tau\sim\pi_E}[Q_{\widehat{\pi_h}}-\tilde{Q}_{\widehat{\pi_h}}]|+T\delta \quad \text{(B.9)} \\
&\leq 2T\delta,
\end{aligned}
$$

where Equation (B.9) holds by applying Equation (B.8) because the expectation of the trajectories (and their transitions) is over π_E, and the actions that are used only as arguments into the Q function are sampled from $\widehat{\pi_h}$.

Next, we bound the first part of Equation (B.4). Recall that the ill-posedness of the

problem for a policy class Π is

$$\nu(\Pi, k) = \sup_{\pi \in \Pi} \frac{\|\pi_E - \pi\|_2}{\|\mathbb{E}[a_t - \pi(h_t)|h_{t-k}]\|_2},$$

where $\|\pi_E - \pi\|_2$ is the RMSE and $\|\mathbb{E}[a_t - \pi(s_t)|s_{t-k}]\|_2$ is the CMR error from our algorithm. Since the learnt policy $\widehat{\pi_h}$ has a CMR error of ϵ, we have that

$$\|\pi_E - \widehat{\pi_h}\|_2 \leq \nu(\Pi, k)\|\mathbb{E}[a_t - \widehat{\pi_h}(h_t)|h_{t-k}]\|_2 \leq \nu(\Pi, k)\epsilon.$$

Next, recall that c-total variation stability of a distribution $P(u^\epsilon)$ where $u^\epsilon \in A$ for some space A implies for two elements $a_1, a_2 \in A$,

$$\|a_1 - a_2\|_2 \leq \Delta \implies \delta_{TV}(a_1 + u^\epsilon, a_2 + u^\epsilon) \leq c\Delta.$$

Since $P(u_t^\epsilon)$ is c-TV stable w.r.t. the action space A, we have that, for all history trajectories $h_t \in H$ (note that $s_t \in h_t$),

$$\delta_{TV}(\pi_E(s_t) + u_t^\epsilon, \widehat{\pi_h}(h_t) + u_t^\epsilon) \leq c\|\pi_E(s_t) - \widehat{\pi_h}(h_t)\|_2.$$

Then, we have that by Jensen's inequality,

$$\begin{aligned}
\mathbb{E}_{h_t \sim \pi_E}[\delta_{TV}(\pi_E(s_t) + u_t^\epsilon, \widehat{\pi_h}(h_t) + u_t^\epsilon)]^2 &\leq \mathbb{E}_{h_t \sim \pi_E}[\delta_{TV}(\pi_E(s_t) + u_t^\epsilon, \widehat{\pi_h}(h_t) + u_t^\epsilon)^2] \\
\implies \mathbb{E}_{h_t \sim \pi_E}[\delta_{TV}(\pi_E(s_t) + u_t^\epsilon, \widehat{\pi_h}(h_t) + u_t^\epsilon)] &\leq \sqrt{\mathbb{E}_{h_t \sim \pi_E}[\delta_{TV}(\pi_E(s_t) + u_t^\epsilon, \widehat{\pi_h}(h_t) + u_t^\epsilon)^2]} \\
&\leq \sqrt{c^2 \mathbb{E}_{h_t \sim \pi_E}[\|\pi_E(s_t) - \widehat{\pi_h}(h_t)\|_2^2]} \\
&= c\|\pi_E - \widehat{\pi_h}\|_2 \leq c\epsilon\nu(\Pi, k).
\end{aligned}$$

Therefore, by applying the total variation distance bound for expectations of $\tilde{Q}_{\widehat{\pi_h}}$ over

different distributions of action a_t, we have that

$$\begin{aligned}
\mathbb{E}_{\tau\sim\pi_E}[\tilde{Q}_{\widehat{\pi_h}} - \mathbb{E}_{a\sim\widehat{\pi_h}}[\tilde{Q}_{\widehat{\pi_h}}]] &= \mathbb{E}_{\tau\sim\pi_E}[\tilde{Q}_{\widehat{\pi_h}}(h_t, a_t) - \mathbb{E}[\tilde{Q}_{\widehat{\pi_h}}(h_t, \widehat{\pi_h}(h_t))]] \\
&= \mathbb{E}_{h_t\sim\pi_E}[\mathbb{E}[\tilde{Q}_{\widehat{\pi_h}}(h_t, \pi_E(s_t) + u_t^\epsilon)] - \mathbb{E}[\tilde{Q}_{\widehat{\pi_h}}(h_t, \widehat{\pi_h}(h_t) + u_t^\epsilon)]] \\
&\leq \|\tilde{Q}_{\widehat{\pi_h}}\|_\infty \mathbb{E}_{h_t\sim\pi_E}[\delta_{TV}(F(\pi_E(s_t) + u_t^\epsilon), F(\widehat{\pi_h}(h_t) + u_t^\epsilon))] \\
&\leq Tc\epsilon\nu(\Pi, k).
\end{aligned}$$

Combining all of the above, we see that from Equation (B.4), by selecting $\tilde{Q}_{\widehat{\pi_h}}(h_t, a_t) := Q_{\widehat{\pi_h}}(s_t, a_t, \mathbb{E}_{\tau\sim\pi_E}[u_t^o|h_t])$, the imitation gap can be bounded by

$$\begin{aligned}
J(\pi_E) - J(\widehat{\pi_h}) &= \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\pi_E}[\tilde{Q}_{\widehat{\pi_h}} - \mathbb{E}_{a\sim\widehat{\pi_h}}[\tilde{Q}_{\widehat{\pi_h}}]] \\
&\quad + \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\pi_E}[Q_{\widehat{\pi_h}} - \tilde{Q}_{\widehat{\pi_h}} - \mathbb{E}_{a\sim\widehat{\pi_h}}[Q_{\widehat{\pi_h}} - \tilde{Q}_{\widehat{\pi_h}}]] \\
&\leq \sum_{t=1}^{T} Tc\epsilon\nu(\Pi, k) + \sum_{t=1}^{T} 2T\delta \\
&\leq T \cdot (Tc\epsilon\nu(\Pi, k) + 2T\delta) \\
&= T^2(c\epsilon\nu(\Pi, k) + 2\delta) = \mathcal{O}(T^2(\epsilon + \delta)),
\end{aligned}$$

which concludes the proof. □

B.1.3 Proofs of Corollaries

Corollary 5.9. *In the special case that $u_t^o = 0$, meaning that there is no confounder observable to the expert, or $u_t^o = \mathbb{E}_{\pi_E}[u_t^o|h_t]$, meaning that u_t^o is $\sigma(h_t)$ measurable (all information regarding u_t^o is represented in the history), the imitation gap bound is $T^2(c\epsilon\nu(\Pi, k))$, which coincides with Theorem 5.1 of Swamy et al. (2022b).*

Proof. If $u_t^o = 0$, then we have $u_t^o = \mathbb{E}_{\pi_E}[u_t^o|h_t]$ since u_t^o is a constant. If $u_t^o = \mathbb{E}_{\pi_E}[u_t^o|h_t]$,

we have that,

$$\delta_{TV}(u_t^o, \mathbb{E}_{\pi_E}[u_t^o|h_t]) = \delta_{TV}(u_t^o, u_t^o) \leq 0.$$

By plugging $\delta = 0$ into Theorem 5.8, we have that $J(\pi_E) - J(\widehat{\pi_h}) \leq T^2(c\epsilon\nu(\Pi, k))$, which is the same as the imitation gap derived in Swamy et al. (2022b) and completes the proof. □

Corollary 5.10. *In the special case that $u_t^\epsilon = 0$, if the learnt policy via supervised BC has error ϵ, then the imitation gap bound is $T^2(\frac{2}{\sqrt{\dim(A)}}\epsilon + 2\delta)$, which is a concrete bound that extends the abstract bound in Theorem 5.4 of Swamy et al. (2022a).*

Proof. In Theorem 5.4 of Swamy et al. (2022a), for the offline case, which is the setting we are considering (as opposed to the online settings), they defined the following quantities for bounding the imitation gap in a very general fashion,

$$\epsilon_{\text{off}} := \sup_{\tilde{Q}} \mathbb{E}_{\tau\sim\pi_E}[\tilde{Q} - \mathbb{E}_{a\sim\widehat{\pi_h}}[\tilde{Q}]];$$

$$\delta_{\text{off}} := \sup_{Q\times\tilde{Q}} \mathbb{E}_{\tau\sim\pi_E}[Q_{\widehat{\pi_h}} - \tilde{Q} - \mathbb{E}_{a\sim\widehat{\pi_h}}[Q_{\widehat{\pi_h}} - \tilde{Q}]].$$

The imitation gap by Theorem 5.4 in Swamy et al. (2022a) under the assumption that $u_t^\epsilon = 0$ is $T^2(\epsilon_{\text{off}}+\delta_{\text{off}})$, which can also be deduced from Equation (B.4) by naively applying the above supremum. To obtain a concrete bound, we can provide a tighter bound for $\mathbb{E}_{\tau\sim\pi_E}[\tilde{Q}_{\widehat{\pi_h}} - \mathbb{E}_{a\sim\widehat{\pi_h}}[\tilde{Q}_{\widehat{\pi_h}}]]$, which is the first part of Equation (B.4), given that $u_t^\epsilon = 0$.

For two elements $a_1, a_2 \in A$, we have that by Cauchy–Schwarz,

$$\delta_{TV}(a_1 + 0, a_2 + 0) = \frac{1}{2}\|a1 - a2\|_1 \leq \frac{\sqrt{\dim(A)}}{2}\|a1 - a2\|_2.$$

Then, we have that

$$\|a_1 - a_2\|_2 \leq \Delta \implies \delta_{TV}(a_1, a_2) \leq \frac{2}{\sqrt{\dim(A)}}\Delta,$$

so that by Theorem 5.8,

$$\begin{aligned}
\mathbb{E}_{\tau\sim\pi_E}[\tilde{Q}_{\widehat{\pi_h}} - \mathbb{E}_{a\sim\widehat{\pi_h}}[\tilde{Q}_{\widehat{\pi_h}}]] &= \mathbb{E}_{\tau\sim\pi_E}[\tilde{Q}_{\widehat{\pi_h}}(h_t, a_t) - \mathbb{E}[\tilde{Q}_{\widehat{\pi_h}}(h_t, \widehat{\pi_h}(h_t))]] \\
&= \mathbb{E}_{h_t\sim\pi_E}[\mathbb{E}[\tilde{Q}_{\widehat{\pi_h}}(h_t, \pi_E(s_t))] - \mathbb{E}[\tilde{Q}_{\widehat{\pi_h}}(h_t, \widehat{\pi_h}(h_t))]] \\
&\leq \|\tilde{Q}_{\widehat{\pi_h}}\|_\infty \frac{2}{\sqrt{\dim(A)}}\|\pi_E - \pi\|_2 \\
&\leq T\frac{2}{\sqrt{\dim(A)}}\epsilon,
\end{aligned}$$

since when $u_t^\epsilon = 0$ the learning error via supervised learning is $\epsilon := \|\pi_E - \pi\|_2$. Therefore, the final imitation bound following Theorem 5.8 is

$$\begin{aligned}
J(\pi_E) - J(\widehat{\pi_h}) &= \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\pi_E}[\tilde{Q}_{\widehat{\pi_h}} - \mathbb{E}_{a\sim\widehat{\pi_h}}[\tilde{Q}_{\widehat{\pi_h}}]] \\
&+ \sum_{t=1}^{T} \mathbb{E}_{\tau\sim\pi_E}[Q_{\widehat{\pi_h}} - \tilde{Q}_{\widehat{\pi_h}} - \mathbb{E}_{a\sim\widehat{\pi_h}}[Q_{\widehat{\pi_h}} - \tilde{Q}_{\widehat{\pi_h}}]] \\
&\leq \sum_{t=1}^{T} T\frac{2}{\sqrt{\dim(A)}}\epsilon + \sum_{t=1}^{T} 2T\delta \\
&= T^2(\frac{2}{\sqrt{\dim(A)}}\epsilon + 2\delta).
\end{aligned}$$

This bound is a concrete bound, obtained through detailed analysis of the problem at hand, that coincides with the abstract bound $T^2(\epsilon_{\text{off}} + \delta_{\text{off}})$ provided in Theorem 5.4 of Swamy et al. (2022b). Note that this bound is independent of the ill-posedness $\nu(\Pi, k)$ and the c-TV stability of u_t^ϵ, which are present in the bound of Theorem 5.8, because of the lack of hidden confounders u_t^ϵ. □

B.2 Environments and Tasks

B.2.1 Dynamic Aeroplane Ticket Pricing

Here, we provide details regarding the dynamic aeroplane ticket pricing environment introduced in Example 5.1. The environment and the expert policy are defined as follows:

$$
\begin{aligned}
\mathcal{S} &:= \mathbb{R} \\
\mathcal{A} &:= [-1, 1] \\
s_t &= sign(s) \cdot u_t^o - u_t^\epsilon \\
\pi_E &= clip(-s/u_t^o, -1, 1) \\
a_t &= \pi_E + 10 \cdot u_t^\epsilon \\
u_t^o &= mean(p_t \sim \text{Unif}[-1, 1], p_{t-1}, ...p_{t-M}) \\
u_t^\epsilon &= mean(q_t \sim \text{Normal}(0, 0.1 \cdot \sqrt{k}), q_{t-1}, ..., q_{t-k+1})
\end{aligned}
$$

where M is the influence horizon of the expert-observable u^o, which we set to 30. The states s_t are the profits at each time step, and the actions a_t are the final ticket price. u_t^o represent the seasonal patterns, where the expert π_E will try to adjust the price accordingly. u_t^ϵ represent the operating costs, which are additive both to the profit and price. Both u_t^o and u_t^ϵ are the mean over a set of i.i.d. samples, q_t and p_t, and vary across the time steps by updating the elements in the set at each time step. This construction allows u_t^ϵ and u_{t-k}^ϵ to be independent since all set elements q_t will be re-sampled from time step $t - k$ to t. We multiply the standard deviation of q_t by $\sqrt{k}$ to make sure u_t^ϵ, which is the average over k i.i.d. variables, has the same standard deviation for all choices of k.

B.2.2 Mujoco Environments

We evaluate DML-IL on three Mujoco environments: Ant, Half Cheetah, and Hopper. The original tasks do not contain hidden variables, so we modify the environment to introduce u^ϵ and u^o. We use the default transition, state, and action space defined in the Mujoco environment. However, we changed the task objectives by altering the reward function and added confounding noise to both the state and action. Specifically, instead of controlling the Ant, Half Cheetah, and Hopper, respectively, to travel as fast as possible, the goal is to control the agent to travel at a target speed that is varying throughout an episode. This target speed is u^o, which is observed by the expert but not recorded in the dataset. In addition, we add confounding noise u_t^ϵ to s_t and a_t to mimic the environmental noise such as wind noise. In all cases, the target speed u_t^o, confounding noise u_t^ϵ, and the action a_t are generated as follows:

$$
\begin{aligned}
a_t &= \pi_E + 20 \cdot u_t^\epsilon \\
u_t^o &= mean(p_t \sim \text{Unif}[-2, 4], p_{t-1}, ...p_{t-M}) \\
u_t^\epsilon &= mean(q_t \sim \text{Normal}(0, 0.01 \cdot \sqrt{k}), q_{t-1}, ..., q_{t-k+1})
\end{aligned}
$$

where $M = 30$, the state transitions follow the default Mujoco environment and the expert policy π_E is learnt online in the environment. u_t^o and u_t^ϵ follow the ticket pricing environment to be the average over a queue of i.i.d. random variables. The reward is defined to be the $1_{healthy} - (\text{current velocity} - u_t^o)^2 - \text{control loss}$, where $1_{healthy}$ gives reward 1 as long as the agent is in a healthy state as defined in the Mujoco documentation. The second penalty term penalises deviation between the current agent's velocity and the target velocity u_t^o. The control loss term is also as defined in default Mujoco, which is $0.1 * \sum(a_t^2)$ at each step to regularise the size of actions.

B.2.2.1 Ant

In the Ant environment, we follow the gym implementation [1] with an 8-dimensional action space and a 28-dimensional observable state space, where the agent's position is also included in the state space. Since the target speed u_t^o is not recorded in the trajectory dataset, we scale the current position of the agent with respect to the target speed, $pos'_t = pos_{t-1} + \frac{pos_t - pos_{t-1}}{u_t^o}$, and use the new agent position pos'_t in the observed states. This allows the imitator to infer information regarding u_t^o from trajectory history, namely from the rate of change in the past positions.

B.2.2.2 Half Cheetah

In the Half Cheetah environment, we follow the gym implementation [2] with a 6-dimensional action space and an 18-dimensional observable state space, where the agent's position is also included in the state space. Similarly to the Ant environment, we scale the current position of the agent to $pos'_t = pos_{t-1} + \frac{pos_t - pos_{t-1}}{u_t^o}$ such that the imitator can infer information regarding u_t^o from trajectory history.

B.2.2.3 Hopper

In the Hopper environment, we follow the gym implementation [3] with a 3-dimensional action space and a 12-dimensional observable state space, where the agent's position is also included in the state space. Similarly to the Ant environment, we scale the current position of the agent to $pos'_t = pos_{t-1} + \frac{pos_t - pos_{t-1}}{u_t^o}$ such that the imitator can infer information regarding u_t^o from trajectory history.

[1] Ant environment: `https://www.gymlibrary.dev/environments/mujoco/ant/`

[2] Half Cheetah environment: `https://www.gymlibrary.dev/environments/mujoco/half_cheetah/`

[3] Hopper environment: `https://www.gymlibrary.dev/environments/mujoco/hopper/`

B.3 Implementation Details

Experiments are carried out on a Linux server (Ubuntu 18.04.2) with two Intel Xeon Gold 6252 CPUs, and each experiment run uses a single NVIDIA GeForce RTX 2080 Ti GPU for neural network training.

B.3.1 Expert Training

The expert in the ticket pricing environment is explicitly hand-crafted. For the Mujoco environments, we used the Stable-Baselines3 (Raffin et al., 2021) implementation of soft actor-critic (SAC) and the default hyperparameters for each task outlined by Stable-Baselines3. The expert policy is an MLP with two hidden layers of size 256 and ReLU activations, and we train the expert for 10^7 steps.

B.3.2 Imitator Training

With the expert policy π_E, we generate 40 expert trajectories, each of 500 steps, following our previously defined environments. Specifically, the confounding noise is added to the state and actions, and crucially u_t^o is not recorded in the trajectories. The naive BC directly learns $\mathbb{E}[a_t \mid s_t]$ via supervised learning. ResiduIL mainly follows the implementation of Swamy et al. (2022b), where we adapt it to allow a longer confounding horizon $k > 1$. For DML-IL and BC-SEQ, a history-dependent policy is used, where we fix the look-back length to be $k + 3$, where k is the confounding horizon. BC-SEQ then just learns $\mathbb{E}[a_t \mid h_t]$ via supervised learning, and DML-IL is implemented with K-fold following Algorithm 4. The policy network architectures for BC, BC-SEQ, and ResiduIL are 2-layer MLPs with a 256 hidden size. The policy network $\widehat{\pi}_h$ and the mixture of Gaussians roll-out model $\widehat{M}$ for DML-IL have a similar architecture, with details provided in Table B.1. We use the AdamW optimiser with a weight decay of 10^{-4} and a learning rate of 10^{-4}. The batch size is 64, and each model is trained for 150 epochs, which is sufficient for their convergence.

Layer Type	Configuration
Input	state dim $\times$ 3
FC + ReLU	Out: 256
FC + ReLU	Out: 256
MixtureGaussian	5 components; Out: state dim $\times$ k

(a) Roll-out model $\widehat{M}$

Layer Type	Configuration
Input	state dim$\times$ (k+3)
FC + ReLU	Out: 256
FC + ReLU	Out: 256
FC	Out: action dim

(b) Policy model $\widehat{\pi}_h$

Table B.1: Network architecture for DML-IL. For the mixture of Gaussians output, we report the number of components. No dropout is used.

B.3.3 Imitator Evaluation

The trained imitator is then evaluated for 50 episodes, each 500 steps in the respective confounded environments. The average reward and the mean squared error between the imitator's action and the expert's action are recorded.

Appendix C

Learning Policies for High-Level Objectives

C.1 Proofs of Main Results

In this section, we provide the proofs for the main results in Chapter 6, which we restate for convenience.

C.1.1 Proof of Theorem 6.6

Theorem 6.6. *For any product MDP $\mathcal{M}^{\times}$ that is induced from LTL formula φ, we have that*

$$\sup_{\pi} \mathbb{P}_{\pi}(s_0 \models \varphi) = \sup_{\pi^{\times}} \mathbb{P}_{\pi^{\times}}(s_0^{\times} \models \mathsf{GF}\ \varphi_{\mathcal{F}}). \tag{C.1}$$

Furthermore, a deterministic memoryless policy that maximises the probability of satisfying the Büchi condition $\varphi_{\mathcal{F}}$ on the product MDP $\mathcal{M}^{\times}$, starting from the initial state, induces a deterministic finite-memory optimal policy that maximises the probability of satisfying φ on the original MDP $\mathcal{M}$ from the initial state.

Proof. We will prove the equality by verifying both sides of the inequality and subse-

quently constructing the induced policy on $\mathcal{M}$.

For $\leq$, it has been proven (Sickert et al., 2016) that, for every accepting path of the environment MDP $\mathcal{M}$ following a policy π, there always exists a corresponding accepting run θ of the LDBA that resolves the nondeterminism. Augmenting the resolved nondeterminism as ϵ-actions to the original path yields an accepting path of $\mathcal{M}^{\times}$, since the counter will eventually reach K and the accepting states $\mathcal{F}^{\times}$ will be visited infinitely often. Therefore, we have created a policy for $\mathcal{M}^{\times}$ that is at least as good as π on $\mathcal{M}$.

For $\geq$, it is clear that any policy π on $\mathcal{M}^{\times}$ can induce a policy for $\mathcal{M}$ by eliminating the ϵ-transitions and removing the projection of the automaton and K counter. Therefore, any path following π that meets the Büchi condition $\varphi_{\mathcal{F}}$ will induce a path of $\mathcal{M}$ that is accepting by $\mathcal{A}$ induced from φ, where the nondeterminism of $\mathcal{A}$ is resolved by ϵ-transitions of π, thus satisfying φ.

□

C.1.2 Proof of Lemma 6.9

Lemma 6.9. *Given a product MDP $\mathcal{M}^{\times}$ with its corresponding LTL formula φ and a policy π, we write $\mathcal{M}^{\times}_{\pi}$ for the induced Markov chain from π. Let $B^{\times}_{\mathcal{F}}$ denote the set of states that belong to accepting BSCCs of $\mathcal{M}^{\times}_{\pi}$, and $B^{\times}_{\varnothing}$ denote the set of states that belong to rejecting BSCCs:*

$$B^{\times}_{\mathcal{F}} \coloneqq \{s^{\times} \mid s^{\times} \in B \in BSCC(\mathcal{M}^{\times}_{\pi}), B \cap \mathcal{F}^{\times} \neq \varnothing\};$$
$$B^{\times}_{\varnothing} \coloneqq \{s^{\times} \mid s^{\times} \in B \in BSCC(\mathcal{M}^{\times}_{\pi}), B \cap \mathcal{F}^{\times} = \varnothing\},$$

where $BSCC(\mathcal{M}_\pi^\times)$ is the set of all BSCCs of $\mathcal{M}_\pi^\times$. We further define more general accepting and rejecting sets:

$$B_\mathcal{F} := \{(s, q, n) \mid \exists n' \in [0..K] : (s, q, n') \in B_\mathcal{F}^\times\}; \tag{C.2}$$

$$B_\varnothing := \{(s, q, n) \mid \exists n' \in [0..K] : (s, q, n') \in B_\varnothing^\times\}. \tag{C.3}$$

If Assumption 6.8 holds, then $\mathbb{P}_\pi(s^\times \models \mathsf{GF}\ \varphi_\mathcal{F}) = 1\ \forall s^\times \in B_\mathcal{F}$, $\mathbb{P}_\pi(s^\times \models \mathsf{GF}\ \varphi_\mathcal{F}) = 0\ \forall s^\times \in B_\varnothing$, and $B_\varnothing \cap \mathcal{F}^\times = \varnothing$. Furthermore, $B_\mathcal{F}$ and $B_\varnothing$ are sink sets, which means that once the set is reached, no states outside the set can be reached.

Proof. To begin with, we recall the property of BSCCs in Markov chains (MC): for any infinite path σ of a MC, a BSCC will eventually be reached, and once reached it cannot reach any state outside the BSCC and all states within it will be visited infinitely often with probability 1. Therefore, since all $s^\times \in B_\mathcal{F}^\times$ belong to BSCCs with accepting states, all paths from $s^\times$ will reach accepting states infinitely often with probability 1, so $\mathbb{P}_\pi(s^\times \models \mathsf{GF}\ \varphi_\mathcal{F}) = 1\ \forall s^\times \in B_\mathcal{F}^\times$. Similarly, all $s^\times \in B_\varnothing^\times$ belong to BSCCs with no accepting states, so $\mathbb{P}_\pi(s^\times \models \mathsf{GF}\ \varphi_\mathcal{F}) = 0\ \forall s^\times \in B_\varnothing^\times$.

Now, we observe that, for states in $\mathcal{M}_\pi^\times$, the Markov chain transitions for environment states and automaton states are, in fact, independent of the counter value:

$$T_\pi^\times((s, q, n_1), (s', q', \min(n_1 + 1, K))) = T_\pi^\times((s, q, n_2), (s', q', \min(n_2 + 1, K))) \tag{C.4}$$

for all $n_1, n_2 \in [0..K]$, $s, s' \in \mathcal{S}$ and $q, q' \in \mathcal{Q}$. With this observation, for all $(s, q, n) \in B_\mathcal{F}$, there exists $(s, q, n') \in B_\mathcal{F}^\times$, which belongs to some BSCC by Equation (C.2). Therefore, any state (s_1, q_1, n_1) reachable from (s, q, n) by Equation (C.4) implies that there exists n_2 such that (s_1, q_1, n_2) is reachable from (s, q, n'). This, by the definition of BSCC, means $(s_1, q_1, n_2) \in B_\mathcal{F}^\times$, which implies $(s_1, q_1, n_1) \in B_\mathcal{F}$ by Equation (C.2), which shows that $B_\mathcal{F}$ is a sink set. Furthermore, since $(s, q, n') \in B_\mathcal{F}^\times$ belongs to an accepting BSCC, it can reach an accepting state $(s_\mathcal{F}, q_\mathcal{F}, n_\mathcal{F}) \in B_\mathcal{F}^\times$ in that BSCC with probability 1. By Equation (C.4),

there exists $(s_{\mathcal{F}}, q_{\mathcal{F}}, n_3) \in B_{\mathcal{F}}$ reachable from (s, q, n) with probability 1 and, by the definition of accepting states in Definition 6.4, $(s_{\mathcal{F}}, q_{\mathcal{F}}, n_3)$ is also accepting, which means accepting states can be reached from any state in $B_{\mathcal{F}}$ with probability 1. Together with the fact that $B_{\mathcal{F}}$ is a sink set, we conclude that accepting states can be reached infinitely often with probability 1 from $B_{\mathcal{F}}$, which means $\mathbb{P}_{\pi}(s^{\times} \models \mathsf{GF}\ \varphi_{\mathcal{F}}) = 1\ \forall s^{\times} \in B_{\mathcal{F}}$. Another observation is that, since the counter values are increasing for all paths and accepting states will be reached infinitely often in accepting BSCCs, all states in accepting BSCCs must have a counter value equal to K.

Using the same argument, we conclude that $B_{\varnothing}$ is also a sink set. In addition, for $(s, q, n) \in B_{\varnothing}$, there exists $(s, q, n') \in B_{\varnothing}^{\times}$ that belongs to a rejecting BSCC such that no accepting states can be reached from it. By Equation (C.4) and the accepting states definition of product MDP (Definition 6.4), we see that no accepting state can be reached from (s, q, n) either, which implies $\mathbb{P}_{\pi}(s^{\times} \models \mathsf{GF}\ \varphi_{\mathcal{F}}) = 0\ \forall s^{\times} \in B_{\varnothing}$ and $B_{\varnothing} \cap \mathcal{F}^{\times} = \varnothing$, which completes the proof. □

C.1.3 Proof of Theorem 6.10

Theorem 6.10. *Given an LTL formula φ and a product MDP $\mathcal{M}^{\times}$, if Assumption 6.8 holds, then there exists an upper bound $U \in (0, 1]$ for rewards and a discount factor $\gamma \in (0, 1]$ such that, for all product rewards $r^{\times}$ and product discount functions $\gamma^{\times}$ satisfying Definition 6.7, the optimal deterministic memoryless policy π_r that maximises the expected discounted reward $G_0^{\pi_r}(s_0^{\times})$ is also an optimal policy π_{φ} that maximises the probability of satisfying the Büchi condition $\mathbb{P}_{\pi_{\varphi}}(s_0^{\times} \models \mathsf{GF}\ \varphi_{\mathcal{F}})$ on the product MDP $\mathcal{M}^{\times}$.*

Proof. To give an outline of the proof, we would like to show that, for any deterministic memoryless policy π, the expected discounted reward $G_0^{\pi}(s^{\times})$ for all product states with counter value 0 (i.e., $s^{\times} = (s, q, 0)$) is close to the probability $\mathbb{P}_{\pi}(s^{\times} \models \mathsf{GF}\ \varphi_{\mathcal{F}})$ of satisfying $\varphi_{\mathcal{F}}$ starting from $s^{\times}$ following π. We show this by upper and lower bounding the difference between the two quantities.

From $s^\times = (s, q, 0)$, we consider the expected discounted reward conditioned on whether the infinite path following policy π satisfies the Büchi condition $\varphi_\mathcal{F}$ or not,

$$G_0^\pi(s^\times) = G_0^\pi(s^\times \mid s^\times \models \mathsf{GF}\varphi_\mathcal{F})\mathbb{P}_\pi(s^\times \models \mathsf{GF}\varphi_\mathcal{F}) \tag{C.5}$$

$$+ G_0^\pi(s^\times \mid s^\times \not\models \mathsf{GF}\varphi_\mathcal{F})\mathbb{P}_\pi(s^\times \not\models \mathsf{GF}\varphi_\mathcal{F}). \tag{C.6}$$

We first consider component (C.5). Let the stopping time of first reaching the accepting set $B_\mathcal{F}$ from $s^\times$ be $\tau_\mathcal{F}^\pi := \inf\{t > 0 \mid s_t^\times \in B_\mathcal{F}\}$ and let the first reached state in $B_\mathcal{F}$ be (s, q, n). Then, let the hitting time between accepting states in the accepting BSCC B containing $(s, q, n) \in B_\mathcal{F}^\times$ be τ_{BSCC}^π. Furthermore, let $c_{\mathcal{F}acc}^\pi := \left|t \leq \tau_\mathcal{F}^\pi : s_t^\times \in \mathcal{F}^\times\right|$ and $c_{\mathcal{F}rej}^\pi := \left|t \leq \tau_\mathcal{F}^\pi : s_t^\times \notin \mathcal{F}^\times\right|$ be the number of accepting and non-accepting states reached before reaching $B_\mathcal{F}$ from $s^\times$ respectively, where $c_{\mathcal{F}acc}^\pi + c_{\mathcal{F}rej}^\pi = \tau_\mathcal{F}^\pi$. Note that all the quantities above depend on the starting state $s^\times$, but for convenience, we omit it in the notations. Then, we have

$$\begin{aligned}
&G_0^\pi(s^\times \mid s^\times \models \mathsf{GF}\varphi_\mathcal{F}) \\
&= \mathbb{E}_\pi\left[\sum_{i=0}^{\infty}(\prod_{j=0}^{i-1}\gamma(s_j^\times)) \cdot r_i^\times \mid s_0^\times = s^\times \models \mathsf{GF}\varphi_\mathcal{F}|\right] \\
&\geq \mathbb{E}_\pi\left[\gamma^{c_{\mathcal{F}rej}^\pi}\left(\sum_{n=0}^{c_{\mathcal{F}acc}^\pi}(\prod_{j=0}^{n-1}(1-R_j))R_n\right) + \sum_{i=\tau_\mathcal{F}^\pi}^{\infty}(\prod_{j=0}^{i-1}\gamma(s_j^\times)) \cdot r_i^\times \mid s_0^\times = s^\times \models \mathsf{GF}\varphi_\mathcal{F})\right]
\end{aligned} \tag{C.7}$$

$$\begin{aligned}
&\geq \mathbb{E}_\pi\Bigg[\gamma^{c_{\mathcal{F}rej}^\pi}\Bigg(\sum_{n=0}^{c_{\mathcal{F}acc}^\pi}(\prod_{j=0}^{n-1}(1-R_j))R_n \\
&\qquad + \sum_{i=\tau_\mathcal{F}^\pi}^{\infty}(\prod_{j=0}^{c_{\mathcal{F}acc}^\pi}(1-R_j)\prod_{k=\tau_\mathcal{F}^\pi}^{i-1}\gamma(s_k^\times)) \cdot r_i^\times\Bigg) \mid s_0^\times = s^\times \models \mathsf{GF}\varphi_\mathcal{F})\Bigg]
\end{aligned} \tag{C.8}$$

$$\geq \mathbb{E}_\pi \Bigg[\gamma^{c^\pi_{\mathcal{F}rej}} \Bigg(\sum_{n=0}^{c^\pi_{\mathcal{F}acc}} (\prod_{j=0}^{n-1}(1-R_j))R_n$$

$$+ \gamma^{\tau^\pi_{BSCC}/U} \sum_{n=c^\pi_{\mathcal{F}acc}}^{\infty} (\prod_{j=0}^{n-1}(1-R_j))R_n \Bigg) \mid s_0^\times = s^\times \models \mathsf{GF}\varphi_{\mathcal{F}} \Bigg] \tag{C.9}$$

$$\geq \gamma^{C^\pi_{\mathcal{F}}+N^\pi/U} \left(\sum_{n=0}^{\infty} (\prod_{j=0}^{n-1}(1-R_j))R_n \right) \tag{C.10}$$

$$= \gamma^{C^\pi_{\mathcal{F}}+N^\pi/U} \cdot 1, \tag{C.11}$$

where $C^\pi_{\mathcal{F}} = \mathbb{E}_\pi[c^\pi_{\mathcal{F}rej} \mid s_0^\times = s^\times \models \mathsf{GF}\varphi_{\mathcal{F}})]$, $N^\pi = \mathbb{E}_\pi[\tau^\pi_{BSCC} \mid s_0^\times = s^\times \models \mathsf{GF}\varphi_{\mathcal{F}})]$, R_n is the nonzero rewards in Definition 6.7 and we slightly abuse the notation to let $R_n = R_K \ \forall n > K$. The inequality in Equation (C.7) holds because, for the first $\tau^\pi_{\mathcal{F}}$ elements in the sum, there are at most $c^\pi_{\mathcal{F}rej}$ zero reward discount γ terms in the product and taking them out of the sum leaves only the nonzero reward terms. Equation (C.8) holds similarly by first taking $c^\pi_{\mathcal{F}rej}$ discount γ terms out of the product for the rest of the sum, leaving the $(1-R_j)$ terms and the rest of the discount factors $\gamma(s_k^\times)$. Equation (C.9) holds by Lemma 6.9 and the observation that, after reaching the accepting BSCC, each nonzero reward will receive $\gamma^{\tau^\pi_{BSCC}}$ additional discount between accepting states. In addition, by summing the infinite sequences we find $U/(1-\gamma^{\tau^\pi_{BSCC}}(1-U)) \leq \gamma^{\tau^\pi_{BSCC}/U}$ for $\gamma \in (0,1)$ and $U \in (0,1)$, upper bounding the additional discount and leaving only the nonzero reward discount factors $1 - R_j$. Finally, Equation (C.11) holds by induction because the infinite geometric sum $\sum_{n=0}^{\infty}(1-R_K)^n R_K = 1$ and $R_n + (1-R_n) * 1 = 1$ for all $n < K$.

Intuitively, the expected discounted reward for paths with only accepting states is 1, and we lower bound the expected discounted reward for general paths satisfying $\varphi_{\mathcal{F}}$ by bounding the amount of discount the reward receives from nonzero reward states.

Next, we consider the second component (C.6). Let the stopping time of first reaching the rejecting set $B_\varnothing$ from $s^\times$ be $\tau^\pi_\varnothing := \inf\{t > 0 \mid s_t^\times \in B_\varnothing\}$, and let the number of accepting states reached before reaching $B_\varnothing$ from $s^\times$ be $c^\pi_{\varnothing acc} := \left|t \leq \tau^\pi_\varnothing : s_t^\times \in \mathcal{F}^\times\right|$. We

similarly omit the dependency on $s^\times$ in the notation. Then, we have

$$
\begin{aligned}
G_0^\pi(s^\times \mid s^\times \not\models \mathsf{GF}\varphi_{\mathcal{F}}) &= \mathbb{E}_\pi\left[\sum_{i=0}^{\infty}(\prod_{j=0}^{i-1}\gamma(s_j^\times))\cdot r_i^\times \;\middle|\; s_0^\times = s^\times \not\models \mathsf{GF}\varphi_{\mathcal{F}})\right] \\
&\leq \mathbb{E}_\pi\left[\sum_{n=0}^{c^\pi_{\varnothing acc}}(\prod_{j=0}^{n-1}(1-R_j))\cdot R_n \;\middle|\; s_0^\times = s^\times \not\models \mathsf{GF}\varphi_{\mathcal{F}})\right] && \text{(C.12)} \\
&= \sum_{n=0}^{C^\pi_\varnothing}(\prod_{j=0}^{n-1}(1-R_j))\cdot R_n && \text{(C.13)} \\
&\leq \sum_{n=0}^{C^\pi_\varnothing}(1-U)^n\cdot U && \text{(C.14)} \\
&= 1-(1-U)^{C^\pi_\varnothing},
\end{aligned}
$$

where $C^\pi_\varnothing = \mathbb{E}_\pi[c^\pi_{\varnothing acc} \mid s_0^\times = s^\times \not\models \mathsf{GF}\varphi_{\mathcal{F}}]$. In Equation (C.12), the inequality holds because, by Lemma 6.9, once $B_\varnothing$ is reached, no further accepting states with nonzero rewards can be reached and the total reward can be bounded above by omitting the discount factor γ from non-accepting states and summing the discounted reward only for the $c^\pi_{\varnothing acc}$ accepting states reached before $B_\varnothing$. Equation (C.13) holds by taking expectation on $\tau^\pi_\varnothing$ and Equation (C.14) holds by induction because $R_n + (1-R_n) * X_1 \leq U + (1-U) * X_2$ for all $n \in [0..K]$ if $X_1 \leq X_2$, and $U \geq R_n$ is an upper bound for all rewards. Intuitively, we have bounded the amount of nonzero reward received by paths not satisfying $\varphi_{\mathcal{F}}$.

Therefore, from Equation (C.5) and these technical results, we have a lower bound for the expected discounted reward,

$$
G_0^\pi(s^\times) \geq \gamma^{C^\pi_{\mathcal{F}}+N^\pi/U}\cdot \mathbb{P}_\pi(s^\times \models \mathsf{GF}\varphi_{\mathcal{F}}), \tag{C.15}
$$

by assuming $G_0^\pi(s^\times \mid s^\times \not\models \mathsf{GF}\varphi_{\mathcal{F}}) = 0$, and an upper bound of the expected discounted

reward:

$$G_0^\pi(s^\times) \leq \mathbb{P}_\pi(s^\times \models \mathsf{GF}\varphi_\mathcal{F}) + (1-(1-U)^{C_\varnothing^\pi}) \cdot \mathbb{P}_\pi(s^\times \not\models \mathsf{GF}\varphi_\mathcal{F}) \tag{C.16}$$

$$= \mathbb{P}_\pi(s^\times \models \mathsf{GF}\varphi_\mathcal{F}) + (1-(1-U)^{C_\varnothing^\pi}) \cdot (1-\mathbb{P}_\pi(s^\times \models \mathsf{GF}\varphi_\mathcal{F})) \tag{C.17}$$

$$= 1-(1-U)^{C_\varnothing^\pi} + \mathbb{P}_\pi(s^\times \models \mathsf{GF}\varphi_\mathcal{F}) \cdot (1-U)^{C_\varnothing^\pi}. \tag{C.18}$$

Last but not least, since the deterministic memoryless policy is considered in a finite MDP, there is a finite set of policies and we let the difference in probability of satisfying $\varphi_\mathcal{F}$ between the optimal policy and the best sub-optimal policy be $\delta \coloneqq \mathbb{P}_{\pi_\varphi}(s_0^\times \models \mathsf{GF}\ \varphi_\mathcal{F}) - \max_{\{\pi \neq \pi_\varphi\}} \mathbb{P}_\pi(s_0^\times \models \mathsf{GF}\ \varphi_\mathcal{F})$. We can let the reward upper bound $U \in (0,1)$ be small enough and $\gamma \in (0,1)$ large enough such that following the bounds of Equation (C.15) and Equation (C.18), we have that,

$$G_0^\pi(s^\times) \leq \max_{\pi \neq \pi_\varphi}\{\mathbb{P}_\pi(s_0^\times \models \mathsf{GF}\ \varphi_\mathcal{F})\} + \delta = \mathbb{P}_{\pi_\varphi}(s_0^\times \models \mathsf{GF}\ \varphi_\mathcal{F}) \leq G_0^{\pi_\varphi}(s^\times) \quad \forall \pi \neq \pi_\varphi. \tag{C.19}$$

This means the expected discounted reward for the optimal policy π_φ must be greater than the expected discounted reward received by any sub-optimal policy. Therefore, the policy π_r that maximises the total expected discounted reward $G_0^{\pi_r}(s^\times)$ is also the optimal policy π_φ that maximises the probability of satisfying the Büchi condition $\varphi_\mathcal{F}$ in the product MDP $\mathcal{M}^\times$, which completes the proof.

For choosing the key hyperparameters $U \in (0,1)$ and $\gamma \in (0,1)$ to ensure optimality, if we assume a reasonable gap $\delta \approx 0.5$ between optimal and suboptimal policies, it generally suffices to let $U = 1/C_\varnothing^\pi$ and $\gamma = 1-1/(C_\varnothing^\pi * N^\pi + C_\mathcal{F}^\pi)$. For example, with the probabilistic gate MDP in Figure 6.1, we have $C_\varnothing^\pi \approx C_\mathcal{F}^\pi \approx 10$ and $N^\pi = 1$, so choosing $U = 0.1$ and $\gamma = 0.95$ is sufficient. If the hitting times and stopping times are not obtainable, under the common assumption for MDPs that the number of states $|\mathcal{S}|$ in $\mathcal{M}$ and the minimum nonzero transition probability $p_{\min} \coloneqq \min_{s,a,s'}\{\mathcal{P}(s,a,s') > 0\}$ are known, $C_\varnothing^\pi$ and $C_\mathcal{F}^\pi$

can be upper bounded by $|\mathcal{S}|/p_{\min}$ and N^{π} can be upper bounded by $|\mathcal{S}|$. $\square$

C.1.4 Proof of Theorem 6.11

Theorem 6.11. *Given an environment MDP $\mathcal{M}$ and an LTL specification φ with an appropriate discount factor γ and a reward function $r^{\times}$ satisfying Theorem 6.10, then if Assumption 6.8 holds, Q-learning for LTL described in Algorithm 5 converges to an optimal policy π_{φ} that maximises the probability of satisfying φ in $\mathcal{M}$.*

Proof. To begin with, recall from Theorem 6.6 and Theorem 6.10 that the policy π_r maximising the expected discounted reward of $\mathcal{M}^{\times}$ induces the optimal policy for satisfying φ in $\mathcal{M}$ by removing the ϵ-actions.

Next, note that, despite the reward for Q-learning in Algorithm 5 being non-Markovian, following the proof of convergence to the optimal Q value for non-Markovian Q-learning (Majeed & Hutter, 2018), with the K counter in Definition 6.4 and the reward in Definition 6.7, we verify that the Q function in Algorithm 5 converges to the optimal Q value for all (s, q) and action $a^{\times}$, which is the expected discounted reward by taking action $a^{\times}$ from $(s, q, 0)$.

With this, recall that the environment states and automaton state transitions do not depend on the counter value by Equation (C.4) and the nonzero reward is the same for accepting states with the same counter value by Definition 6.7. We argue that, for the optimal discounted reward policy π_r, the new policy $\pi'((s, q, n)) = \pi_r((s, q, 0))\ \forall n \in [0..K]$, where the policy for each counter value is set to be the same, remains optimal. This is because if π' is suboptimal, meaning that the satisfiability of $\varphi_{\mathcal{F}}$ is lower than that of the optimal policy, it must be that π' makes a wrong decision at some counter value such that the reachability probability to some accepting BSCC B is suboptimal. This means π_r must also be suboptimal because when the counter value equals 0, by taking the same actions as π', the reachability probability to the BSCC B is suboptimal, producing a contradiction. Lastly, since π' is independent of the counter values, the policy derived from the collapsed optimal Q function learnt by Algorithm 5 is the same as the optimal π' by ignoring the

K counter, which means that Algorithm 5 converges to the optimal policy, completing the proof. □

C.2 Additional Experimental Details and Setup

The experiments are carried out on a Linux server (Ubuntu 18.04.2) with two Intel Xeon Gold 6252 CPUs and six NVIDIA GeForce RTX 2080 Ti GPUs. All of our algorithms are implemented in Python. We select three stochastic environments (probabilistic gate, frozen lake, and office world) described below with several difficult LTL tasks. For all environments and tasks, we set the learning rate $\alpha = 0.1$ and the exploration rate $\epsilon = 0.1$, with the upper bound on the rewards $U = 0.1$ and the discount factor $\gamma = 0.99$ to ensure optimality. For the method proposed by Hahn et al. (2019), we adopt their implementation from the Mungojerrie tool[1]. We set the default hyperparameter values to $\alpha = 0.1$ and $\epsilon = 0.1$, with $\zeta = 0.995$ for the probabilistic gate task and $\zeta = 0.99$ for the frozen lake and office world tasks. For the method by Hasanbeig et al. (2020), we use the implementation in their GitHub repository[2] with $\gamma = 0.99$, $\alpha = 0.1$ and $\epsilon = 0.1$. For the method of Bozkurt et al. (2019), we also adopt their GitHub repository[3] with $\gamma = 0.99$, $\gamma_B = 0.9$, $\alpha = 0.1$ and $\epsilon = 0.1$.

For the environments and tasks used for experimentation, we specifically choose commonly used environments with stochastic transitions that are more realistic and challenging, especially for infinite-horizon LTL tasks. The LTL tasks that we use are standard task specifications for robotics and autonomous systems. The tasks for the frozen lake and the office world environments are more challenging because they involve letting the agent choose between two subtasks, which better showcase the benefits of the K counter and counterfactual imagining.

[1] https://plv.colorado.edu/wwwmungojerrie/docs/v1_0/index.html
[2] https://github.com/grockious/lcrl
[3] https://github.com/alperkamil/csrl

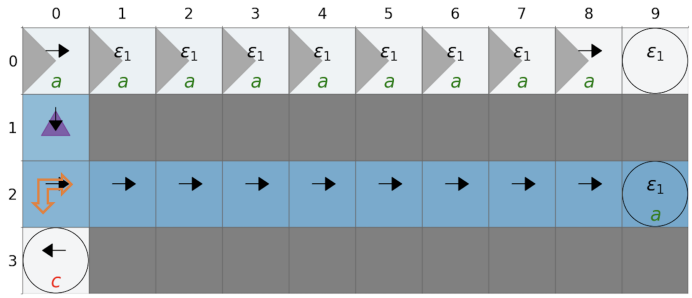
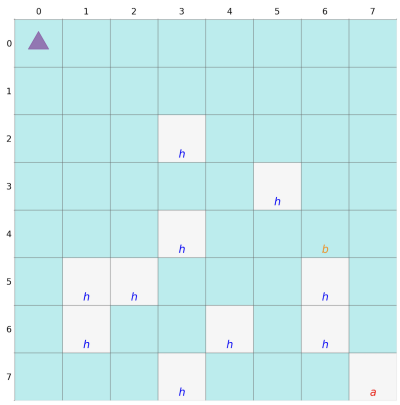
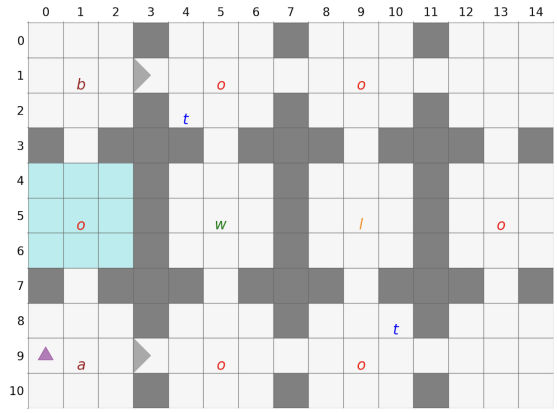


Figure C.1: The optimal policy for the example probabilistic gate MDP task in Figure 6.1



(a) The MDP environment for the frozen lake task. Blue represents ice, "h" are holes, "a" and "b" are lake camps, and the purple triangle is the start.



(b) The MDP environment for the office world task. Blue represents ice, "o" are obstacles, "w", "l" and "t" represent workplace, letter and tea respectively, and the purple triangle is the start.

Figure C.2: MDP environments used in the experiments.

C.2.1 Probabilistic Gate

The probabilistic gate MDP is described in Example 6.5, where the task is to visit states labelled "a" infinitely often without visiting states labelled "c". We set the episode length to be 100 (which is the maximum length of a path in the MDP) and the number of episodes to be 40000. In Figure C.1, the optimal policy is demonstrated by the black arrows, where the agent goes down from the start and goes all the way to the right to (2,9), which is an accepting sink state.

C.2.2 Frozen Lake

The frozen lake (Brockman et al., 2016) environment is shown in Figure C.2a. The blue states represent the frozen lake, where the agent has a 1/3 probability of moving in the intended direction and 1/3 each of going sideways (left or right). The white states with label "h" are holes and states with label "a" and "b" are lake camps. The task is "(GF a | GF b) & G !h", meaning to always reach lake camp "a" or lake camp "b" while never falling into holes "h". For this task, we set the episode length to 200 and the number of episodes to 6000.

C.2.3 Office World

The office world (Icarte et al., 2022) environment is demonstrated in Figure C.2b. We include a patch of ice labelled blue as defined in the frozen lake environment and two one-directional gates at $(1, 3)$ and $(9, 3)$, where the agent can only cross to the right, as shown by the direction of the grey triangle. The grey blocks are walls, the states labelled "o", "l", "t" and "w" are obstacles, letters, tea, and the workplace, respectively. The task for this environment is also demanding: "(GF a & GF b) | (F l & X (GF t & GF w)) & G !o". This means to either patrol in the corridor between "a" and "b", or go to write a letter at "l" and then patrol between getting tea "t" and the workplace "w", whilst never hitting obstacles "o". For this task, we set the length of the episode to 1000 and the number of episodes to 6000.

C.2.4 PRISM

We use PRISM version 4.7 (Kwiatkowska et al., 2011) to evaluate the learnt policy against the LTL tasks. The PRISM tool and the installation documentation can be obtained from their official website[4]. Each time we would like to evaluate the policy on the environment

[4]https://www.prismmodelchecker.org/download.php

MDP, our Python code automatically constructs the induced Markov chain from the MDP and the policy as a discrete-time Markov chain in the PRISM language, and evaluates this PRISM model against the LTL task specification through a call from Python. The maximum iteration parameter for PRISM is set to 100,000, and we evaluate the current policy every 10,000 training steps to plot the training graph for all our experiments.

C.2.4.1 Example PRISM model

We provide an example PRISM model of the induced Markov chain from the probabilistic gate MDP and its optimal policy.

```
dtmc
module ProductMDP
    m : [0..40] init 10;
    a : [0..3] init 0;
    [ep] (m=0)&(a=0) -> (m'=0)&(a'=1);
    [ac] (m=0)&(a=1) -> 1 : (m'=1)&(a'=1);
    [ac] (m=0)&(a=2) -> 1 : (m'=1)&(a'=2);
    [ep] (m=1)&(a=0) -> (m'=1)&(a'=1);
    [ac] (m=1)&(a=1) -> 1 : (m'=2)&(a'=1);
    [ac] (m=1)&(a=2) -> 1 : (m'=2)&(a'=2);
    [ep] (m=2)&(a=0) -> (m'=2)&(a'=1);
    [ac] (m=2)&(a=1) -> 1 : (m'=3)&(a'=1);
    [ac] (m=2)&(a=2) -> 1 : (m'=3)&(a'=2);
    [ep] (m=3)&(a=0) -> (m'=3)&(a'=1);
    [ac] (m=3)&(a=1) -> 1 : (m'=4)&(a'=1);
    [ac] (m=3)&(a=2) -> 1 : (m'=4)&(a'=2);
    [ep] (m=4)&(a=0) -> (m'=4)&(a'=1);
    [ac] (m=4)&(a=1) -> 1 : (m'=5)&(a'=1);
    [ac] (m=4)&(a=2) -> 1 : (m'=5)&(a'=2);
    [ep] (m=5)&(a=0) -> (m'=5)&(a'=1);
    [ac] (m=5)&(a=1) -> 1 : (m'=6)&(a'=1);
    [ac] (m=5)&(a=2) -> 1 : (m'=6)&(a'=2);
```

```
23    [ep] (m=6)&(a=0) -> (m'=6)&(a'=1);
24    [ac] (m=6)&(a=1) -> 1 : (m'=7)&(a'=1);
25    [ac] (m=6)&(a=2) -> 1 : (m'=7)&(a'=2);
26    [ep] (m=7)&(a=0) -> (m'=7)&(a'=1);
27    [ac] (m=7)&(a=1) -> 1 : (m'=8)&(a'=1);
28    [ac] (m=7)&(a=2) -> 1 : (m'=8)&(a'=2);
29    [ep] (m=8)&(a=0) -> (m'=8)&(a'=1);
30    [ac] (m=8)&(a=1) -> 1 : (m'=9)&(a'=1);
31    [ac] (m=8)&(a=2) -> 1 : (m'=9)&(a'=2);
32    [ep] (m=9)&(a=0) -> (m'=9)&(a'=1);
33    [ac] (m=9)&(a=1) -> 1.0 : (m'=9)&(a'=2);
34    [ac] (m=9)&(a=2) -> 1.0 : (m'=9)&(a'=2);
35    [ac] (m=10)&(a=0) -> 1 : (m'=20)&(a'=0);
36    [ac] (m=10)&(a=1) -> 1 : (m'=20)&(a'=2);
37    [ac] (m=10)&(a=2) -> 1 : (m'=20)&(a'=2);
38    [ac] (m=11)&(a=0) -> 1.0 : (m'=11)&(a'=0);
39    [ac] (m=11)&(a=1) -> 1.0 : (m'=11)&(a'=2);
40    [ac] (m=11)&(a=2) -> 1.0 : (m'=11)&(a'=2);
41    [ac] (m=12)&(a=0) -> 1.0 : (m'=12)&(a'=0);
42    [ac] (m=12)&(a=1) -> 1.0 : (m'=12)&(a'=2);
43    [ac] (m=12)&(a=2) -> 1.0 : (m'=12)&(a'=2);
44    [ac] (m=13)&(a=0) -> 1.0 : (m'=13)&(a'=0);
45    [ac] (m=13)&(a=1) -> 1.0 : (m'=13)&(a'=2);
46    [ac] (m=13)&(a=2) -> 1.0 : (m'=13)&(a'=2);
47    [ac] (m=14)&(a=0) -> 1.0 : (m'=14)&(a'=0);
48    [ac] (m=14)&(a=1) -> 1.0 : (m'=14)&(a'=2);
49    [ac] (m=14)&(a=2) -> 1.0 : (m'=14)&(a'=2);
50    [ac] (m=15)&(a=0) -> 1.0 : (m'=15)&(a'=0);
51    [ac] (m=15)&(a=1) -> 1.0 : (m'=15)&(a'=2);
52    [ac] (m=15)&(a=2) -> 1.0 : (m'=15)&(a'=2);
53    [ac] (m=16)&(a=0) -> 1.0 : (m'=16)&(a'=0);
54    [ac] (m=16)&(a=1) -> 1.0 : (m'=16)&(a'=2);
55    [ac] (m=16)&(a=2) -> 1.0 : (m'=16)&(a'=2);
```

```
56      [ac] (m=17)&(a=0) -> 1.0 : (m'=17)&(a'=0);
57      [ac] (m=17)&(a=1) -> 1.0 : (m'=17)&(a'=2);
58      [ac] (m=17)&(a=2) -> 1.0 : (m'=17)&(a'=2);
59      [ac] (m=18)&(a=0) -> 1.0 : (m'=18)&(a'=0);
60      [ac] (m=18)&(a=1) -> 1.0 : (m'=18)&(a'=2);
61      [ac] (m=18)&(a=2) -> 1.0 : (m'=18)&(a'=2);
62      [ac] (m=19)&(a=0) -> 1.0 : (m'=19)&(a'=0);
63      [ac] (m=19)&(a=1) -> 1.0 : (m'=19)&(a'=2);
64      [ac] (m=19)&(a=2) -> 1.0 : (m'=19)&(a'=2);
65      [ac] (m=20)&(a=0) -> 0.8 : (m'=21)&(a'=0) + 0.2 : (m'=30)&(a'=0);
66      [ac] (m=20)&(a=1) -> 0.8 : (m'=21)&(a'=2) + 0.2 : (m'=30)&(a'=2);
67      [ac] (m=20)&(a=2) -> 0.8 : (m'=21)&(a'=2) + 0.2 : (m'=30)&(a'=2);
68      [ac] (m=21)&(a=0) -> 1 : (m'=22)&(a'=0);
69      [ac] (m=21)&(a=1) -> 1.0 : (m'=21)&(a'=2);
70      [ac] (m=21)&(a=2) -> 1.0 : (m'=21)&(a'=2);
71      [ac] (m=22)&(a=0) -> 1 : (m'=23)&(a'=0);
72      [ac] (m=22)&(a=1) -> 1.0 : (m'=22)&(a'=2);
73      [ac] (m=22)&(a=2) -> 1.0 : (m'=22)&(a'=2);
74      [ac] (m=23)&(a=0) -> 1 : (m'=24)&(a'=0);
75      [ac] (m=23)&(a=1) -> 1.0 : (m'=23)&(a'=2);
76      [ac] (m=23)&(a=2) -> 1.0 : (m'=23)&(a'=2);
77      [ac] (m=24)&(a=0) -> 1 : (m'=25)&(a'=0);
78      [ac] (m=24)&(a=1) -> 1.0 : (m'=24)&(a'=2);
79      [ac] (m=24)&(a=2) -> 1.0 : (m'=24)&(a'=2);
80      [ac] (m=25)&(a=0) -> 1 : (m'=26)&(a'=0);
81      [ac] (m=25)&(a=1) -> 1.0 : (m'=25)&(a'=2);
82      [ac] (m=25)&(a=2) -> 1.0 : (m'=25)&(a'=2);
83      [ac] (m=26)&(a=0) -> 1 : (m'=27)&(a'=0);
84      [ac] (m=26)&(a=1) -> 1.0 : (m'=26)&(a'=2);
85      [ac] (m=26)&(a=2) -> 1.0 : (m'=26)&(a'=2);
86      [ac] (m=27)&(a=0) -> 1 : (m'=28)&(a'=0);
87      [ac] (m=27)&(a=1) -> 1.0 : (m'=27)&(a'=2);
88      [ac] (m=27)&(a=2) -> 1.0 : (m'=27)&(a'=2);
```

```
[ac] (m=28)&(a=0) -> 1 : (m'=29)&(a'=0);
[ac] (m=28)&(a=1) -> 1 : (m'=27)&(a'=2);
[ac] (m=28)&(a=2) -> 1 : (m'=27)&(a'=2);
[ep] (m=29)&(a=0) -> (m'=29)&(a'=1);
[ac] (m=29)&(a=1) -> 1.0 : (m'=29)&(a'=1);
[ac] (m=29)&(a=2) -> 1.0 : (m'=29)&(a'=2);
[ep] (m=30)&(a=0) -> (m'=30)&(a'=1);
[ac] (m=30)&(a=1) -> 1.0 : (m'=30)&(a'=2);
[ac] (m=30)&(a=2) -> 1.0 : (m'=30)&(a'=2);
[ac] (m=31)&(a=0) -> 1.0 : (m'=31)&(a'=0);
[ac] (m=31)&(a=1) -> 1.0 : (m'=31)&(a'=2);
[ac] (m=31)&(a=2) -> 1.0 : (m'=31)&(a'=2);
[ac] (m=32)&(a=0) -> 1.0 : (m'=32)&(a'=0);
[ac] (m=32)&(a=1) -> 1.0 : (m'=32)&(a'=2);
[ac] (m=32)&(a=2) -> 1.0 : (m'=32)&(a'=2);
[ac] (m=33)&(a=0) -> 1.0 : (m'=33)&(a'=0);
[ac] (m=33)&(a=1) -> 1.0 : (m'=33)&(a'=2);
[ac] (m=33)&(a=2) -> 1.0 : (m'=33)&(a'=2);
[ac] (m=34)&(a=0) -> 1.0 : (m'=34)&(a'=0);
[ac] (m=34)&(a=1) -> 1.0 : (m'=34)&(a'=2);
[ac] (m=34)&(a=2) -> 1.0 : (m'=34)&(a'=2);
[ac] (m=35)&(a=0) -> 1.0 : (m'=35)&(a'=0);
[ac] (m=35)&(a=1) -> 1.0 : (m'=35)&(a'=2);
[ac] (m=35)&(a=2) -> 1.0 : (m'=35)&(a'=2);
[ac] (m=36)&(a=0) -> 1.0 : (m'=36)&(a'=0);
[ac] (m=36)&(a=1) -> 1.0 : (m'=36)&(a'=2);
[ac] (m=36)&(a=2) -> 1.0 : (m'=36)&(a'=2);
[ac] (m=37)&(a=0) -> 1.0 : (m'=37)&(a'=0);
[ac] (m=37)&(a=1) -> 1.0 : (m'=37)&(a'=2);
[ac] (m=37)&(a=2) -> 1.0 : (m'=37)&(a'=2);
[ac] (m=38)&(a=0) -> 1.0 : (m'=38)&(a'=0);
[ac] (m=38)&(a=1) -> 1.0 : (m'=38)&(a'=2);
[ac] (m=38)&(a=2) -> 1.0 : (m'=38)&(a'=2);
```

```
    [ac] (m=39)&(a=0) -> 1.0 : (m'=39)&(a'=0);
    [ac] (m=39)&(a=1) -> 1.0 : (m'=39)&(a'=2);
    [ac] (m=39)&(a=2) -> 1.0 : (m'=39)&(a'=2);
endmodule

label "a" = (m=0) | (m=1) | (m=2) | (m=3) | (m=4) | (m=5) | (m=6) | (m
    =7) | (m=8) | (m=29);
label "c" = (m=30);
```

Listing C.1: The PRISM model of the induced Markov chain from the probabilistic gate MDP and its optimal policy.

C.2.5 Rabinizer

We use Rabinizer 4 (Křetínský et al., 2018) to transform LTL formulae into LDBAs. The tool can be downloaded on their website [5], and we use the 'ltl2ldba' script with '-e' and '-d' options to construct the LDBA with ϵ-transitions from the input LTL formulae.

[5] https://www7.in.tum.de/ kretinsk/rabinizer4.html