Causal Artificial Intelligence for Robust Robot Reasoning under Uncertainty

Ricardo Cannizzaro

St Edmund Hall College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity Term 2025

Abstract

Autonomous robots must operate effectively in complex, uncertain, and partially observable environments. Achieving this requires the ability not only to perceive and act, but also to understand and reason about cause and effect. This thesis investigates how *causal generative AI* can provide a unified foundation for robust robot cognition under uncertainty. It introduces a conceptual framework for *Robot Causal Reasoning* that integrates structural causal models, probabilistic planning, and (deep and classical) generative world modelling to enhance robot perception, inference, decision-making, and explanation. Across six research questions, the work spans interventional and counterfactual reasoning, advancing through causal extensions to planning, manipulation, and generative modelling. The thesis contributes new methods at each level of Pearl's *Ladder of Causation*. At the interventional level, for the first time we extend online POMDP planning to reason about confounded dynamics, and introduce a causal Bayesian reasoning architecture for manipulation under uncertainty. At the counterfactual level, the thesis develops methodology for post-hoc causal robot explanations and extends causal reasoning into high-dimensional deep generative AI through counterfactual contrastive learning and a parallel-world simulation framework. Together, these advances establish a coherent causal hierarchy unifying modelling, learning, inference, and explanation. They demonstrate that causal generative models can bridge symbolic reasoning and data-driven learning, enabling robots to act, imagine, and explain in ways consistent with human causal understanding. By grounding autonomous behaviour in causal structure, this work contributes to the next generation of *trustworthy cognitive robotics*: systems that are not only capable of robust decision-making, but also of understanding and communicating the reasons for their actions.

Causal Artificial Intelligence for Robust Robot Reasoning under Uncertainty

Ricardo Cannizzaro

St Edmund Hall College

University of Oxford

Supervised by Prof. Lars Kunze & Prof. Nick Hawes

A thesis submitted for the degree of

Doctor of Philosophy

Trinity Term 2025

This thesis is dedicated to my amazing, loving family members who have always inspired, supported, encouraged, and believed in me throughout my life and my PhD journey:
my loving parents Antonella Cannizzaro and Vito Cannizzaro;
and my inspirational brother Daniel Cannizzaro.
It is only through your love and support that I have been able to complete this PhD thesis — and for this I am eternally grateful.

Ricardo Cannizzaro — 11 October 2025

'You can achieve anything you set your mind on doing'
- Cannizzaro family motto

Acknowledgements

Personal

This journey has been long, unpredictable, and full of extraordinary people who have shaped it in ways I could never have imagined.

First and foremost, I want to thank my supervisors — Lars Kunze and Nick Hawes — for their guidance, patience, and encouragement throughout this DPhil. Each of you has influenced the way I think and work in profoundly different but complementary ways. Lars, thank you for your calm advice, perspective, and infinite patience. Nick, for your clarity, sharp technical insight, and deep supervision experience. I also want to thank Robert Osazuwa Ness for your energy, pragmatism, and curiosity that made every discussion exciting and the work an absolute pleasure.

From the Defence Science and Technology Group, I am deeply grateful to Jennifer Palmer and Simon Ng, who championed my approval to undertake this PhD at the University of Oxford, which has been a truly life-changing experience. I am also grateful to Kent Rosser, Simon Crase, and Melanie Ralph for their unwavering support — not just in funding and logistics, but in believing in the long-term value of this work and in me as a researcher.

To Genevieve Lacey: thank you for making the transition from Australia to Oxford both manageable and enjoyable. I am grateful for the amazing experiences we shared together, both in Oxford and during our travels abroad.

To my amazing and wonderful friends back home: Nick, Aaron, Dean, Juliana, Jess, Camille, Luke, Wren, and Maddy Cochrane. Thank you for keeping me grounded and connected to Australia through every call, visit, online game, and laugh across time zones.

To my stellar lab-mates here in Oxford in the Cognitive Robotics Group (CRG): Rhys, Efimia, Mike, Jumana, and Ben. Thank you for the conversations that were equal parts science, nonsense, and everything in between. And to David Wisth — for first suggesting that I come to Oxford — and to Georgia, for welcoming me when I arrived. I owe you both more than you realise.

To my broader Oxford crew — Lintong, Frank, Rowan, Ethan, Ben R, Georgi, Mitch, Tobit, and Julia — thank you for making life at the Oxford Robotics Institute vibrant, supportive, and always a little chaotic. And to the extended Oxford community who became family in other ways: the Tree Artisan staff (Suelem, E, Lara, Jota, Grazia, Freya), Society Café (Ricardo and Dominic), and friends like Rosanna, Rozi, Petra, Christina, and Mona; thank you for the warmth, humour, and caffeine shared.

In Seattle, the Microsoft Research interns made my two times there unforgettable — Trenton ('stay stochastic'), Verna, Dhia, Katie, Em, Sarah, Elisa, Dan, Marijn, Rita, Ilaria, Vitica,

Jiwon, and Aditya — you turned two summer internships into some of the best chapters of my PhD. Thanks also to Hannah and the other Seattle locals for the friendship and adventures outside the lab.

To Rafa and the many other friends made at professional events, for the laughter and camaraderie through conferences and long travel days.

Looking further back, I want to acknowledge those who played a foundational role in shaping the person, scientist, engineer, and researcher I would later become.

To my parents, Antonella and Vito, and my brother Daniel: while this thesis is formally dedicated to you, I want to explicitly recognise here the depth of your influence.

Mum, from you I have learned compassion, humility, and the importance of caring for others regardless of title, rank, or status. You instilled in me a strong sense of collaboration, community, and advocacy — both for myself and for others — which has fundamentally shaped the way I lead and work with people today. I am also deeply grateful for the constant, practical support you provided throughout my life, from school and sport to everything at home, which gave me the space to focus on my studies and pursue my goals. It has always been clear to me that you are deeply proud of what I have achieved, and that you have unwavering confidence in both my abilities and my character. Your love and trust have been constant, and I carry them with me in everything I do.

Dad, from you I learned, above all else, that I can achieve anything I set my mind to. You instilled in me a deep belief in the value of education as a pathway to opportunity, stability, and upward mobility, and taught me to maintain faith in myself through challenges in study, career, and life. You were always supportive of the choices I made, encouraging me to pursue what I genuinely enjoy, and offering guidance and help wherever possible — even beyond your own field of expertise. Because of this, I have been able to build a career that I genuinely love, and have already had the privilege of many remarkable experiences — with many more still to come.

Daniel, you have always been an inspiration to me. By walking the path ahead, you showed me what is possible through hard work and conviction, and you have been a constant source of mentorship and guidance through every stage of life. You opened my eyes to the possibilities of travel, an international career, and a global perspective, and have always been generous with your time, advice, and support. You have taught me the importance of lifting others as you succeed — that a rising tide lifts all boats — and you continue to challenge me to strive to be the best version of myself.

Collectively, you created an environment grounded in curiosity, education, discipline, foresight, and perseverance. And it is from this foundation that everything in this thesis ultimately follows.

During my time at St. Bernard's College in Essendon, I was fortunate to have exceptional teachers and mentors who shaped my early intellectual development. Catherine O'Halleran and Napolean Rodezno first fostered my curiosity in mathematics, science, and independent thinking during those formative early years. Geoffrey Menon inspired a deeper appreciation for mathematics and physics, encouraging me to think beyond the curriculum and its applications.

And for teaching me to always find wonder and curiosity in my scientific explorations. David Rosel provided invaluable careers guidance at a critical stage, helping me navigate future pathways — from a lab internship at Australia's premier public-interest science organisation, CSIRO, to university open days, job opportunities, and more.

I owe much of my personal and professional development to my 10+ years with the Young Scientists of Australia (YSA) Melbourne Chapter. It was a privilege to receive mentorship and guidance from Dean Mollica and Justin Sorbello, who taught me the importance of leadership, communication, and community — lessons that have stayed with me well beyond those early years and will continue to do so throughout my career and life.

At the undergraduate level, I would like to thank Professor Zhenwei Cao for supporting my final year research project and, importantly, for encouraging me to pursue a PhD. Your confidence in me at that stage played a decisive role in setting me on this path.

This thesis is further dedicated to the memory of Alex Blain, who sadly passed away just before I began this journey. He was a strong supporter of my decision to undertake this PhD; his faith in me, and his enthusiasm and optimism, gave me the confidence not only to pursue this challenge, but to remain committed to it throughout — for which I will be forever grateful. Alex was one of the kindest and most thoughtful people I've known, and I carry his influence with me to this day.

And finally, to Maral — thank you for your support in every sense of the word: from late-night video calls, surprise food deliveries during the thesis submission, and flowers that made me feel celebrated during important milestones, to shared travel plans and constant encouragement. Thank you for flying from Seattle to Oxford to be there for my viva and to celebrate that moment with me with balloons, cards, and even more flowers; it means more to me than words can express. You have always been in my corner, believing deeply in me and in my work, and showing me a level of love and care I could not have imagined. You made the final, hardest stretches not just bearable, but genuinely meaningful and joyful. This shared accomplishment gives me great confidence for what lies ahead — in my career, and in the future we will continue to build together.

To everyone named here, and the many others who have helped along the way — thank you. This thesis is as much yours as it is mine.

Institutional

Funding

We acknowledge and thank the Australian Department of Defence and the Australian Science and Technology Group for their financial support of this thesis.

Causally Informed Planning for Robots Under Partial Observability and Unobserved Confounding

We thank Robert Osazuwa Ness for the many insightful discussions on planning and reinforcement learning under confounding, and the *practical* implementations of algorithms on real-world applications that go beyond the toy examples that are common in the causal inference literature.

A Causal Bayesian Reasoning Architecture Using Probabilistic Programming for Robot Manipulation Under Uncertainty

We thank Michael Groom for his significant efforts to conduct the large-scale simulation experiments, perform data-processing to produce statistical results, and assist with robot hardware demonstrations. Further, we thank University of Oxford Engineering Science Masters fourth year project students Oliver Orders and Jonathan Routley for their contributions towards the initial exploration of the physics-based robot task reasoning using the PyBullet physics-based simulator. We also thank Tobit Flatscher for his invaluable efforts towards robot system integration, verification, and ROS MoveIt! motion-planner debugging. Finally, we thank Arundathi Shaji Santhini and Ana Deligny for their contributions to the robot software and hardware development.

Counterfactual-Based Post-Hoc Explanations of Robot Task Execution

We thank University of Oxford Engineering Science Masters fourth year project student Lara Radojcic 4YP for their contributions to the development and evaluation of the counterfactual-based causal attribution methods.

Counterfactual Contrastive Learning for Improving Causal Consistency in Multi-Modal GenAI Models

We thank Microsoft Research researchers Robert Osazuwa Ness and Emre Kiciman for project guidance and insightful discussions on counterfactual contrasting learning methods; and fellow PhD research interns Michael Li and Yunshu Wu for their contributions to the initial dSprites proof-of-concept work.

Multiverse Mechanica: A Playable Benchmark for Learning Game Mechanics via Counterfactual Worlds

We thank Microsoft Research researcher Robert Osazuwa Ness for scientific and technical guidance, and contributions to the development of, the parallel-world counterfactual formulation of *Multiverse Mechania*; and fellow PhD research intern Yunshu Wu for their contributions to the latent diffusion counterfactual training loss methodology improvements, model fine-tuning, and the proof-of-concept mechanic learning evaluation.

Abstract

Autonomous robots must operate effectively in complex, uncertain, and partially observable environments. Achieving this requires the ability not only to perceive and act, but also to understand and reason about cause and effect. This thesis investigates how *causal generative AI* can provide a unified foundation for robust robot cognition under uncertainty. It introduces a conceptual framework for *Robot Causal Reasoning* that integrates structural causal models, probabilistic planning, and (deep and classical) generative world modelling to enhance robot perception, inference, decision-making, and explanation. Across six research questions, the work spans interventional and counterfactual reasoning, advancing through causal extensions to planning, manipulation, and generative modelling. The thesis contributes new methods at each level of Pearl's *Ladder of Causation*. At the interventional level, for the first time we extend online POMDP planning to reason about confounded dynamics, and introduce a causal Bayesian reasoning architecture for manipulation under uncertainty. At the counterfactual level, the thesis develops methodology for post-hoc causal robot explanations and extends causal reasoning into high-dimensional deep generative AI through counterfactual contrastive learning and a parallel-world simulation framework. Together, these advances establish a coherent causal hierarchy unifying modelling, learning, inference, and explanation. They demonstrate that causal generative models can bridge symbolic reasoning and data-driven learning, enabling robots to act, imagine, and explain in ways consistent with human causal understanding. By grounding autonomous behaviour in causal structure, this work contributes to the next generation of *trustworthy cognitive robotics*: systems that are not only capable of robust decision-making, but also of understanding and communicating the reasons for their actions.

Contents

List of Figures **xv**

List of Tables **xvii**

List of Abbreviations **xviii**

List of Notations **xix**

1 Introduction **1**

1.1 Overview . . . 1
1.2 Motivation & Challenges . . . 3
1.2.1 Challenges of Real-World Autonomous Mobile Robot Systems . . . 4
1.2.2 Defining Real-World Autonomous Mobile Robot Systems . . . 7
1.2.3 Motivating Examples . . . 8
1.2.4 Causal Modelling and Inference: Methods for Achieving Robot Cognition 11
1.2.5 The Ladder of Causation: A Guiding Paradigm for Robot Cognition . . . 12
1.2.6 Robot Causal Reasoning: A Conceptual Framework Towards Robust Robot Cognition . . . 13
1.3 Research Questions & Contributions . . . 13
1.4 Structure of Thesis . . . 15
1.5 Publications . . . 18
1.5.1 Mainline Work . . . 18
1.5.2 Non-Mainline Work . . . 19
1.6 Research Project Contributions . . . 20

2 Related Work **22**

2.1 Overview and Structure . . . 23
2.2 Probabilistic Decision-Making & Planning Under Uncertainty . . . 24
2.2.1 Foundations of Probabilistic Planning in Robotics . . . 24
2.2.2 Offline Planning Approaches . . . 24
2.2.3 Online Planning Approaches . . . 25
2.2.4 Handling Partial Observability & Uncertainty . . . 25
2.2.5 Limitations in the Presence of Confounders . . . 26
2.2.6 Transition. . . . 26
2.3 Causal Modelling & Inference for Robot Cognition . . . 26

2.3.1 Motivation for Causal Modelling in Robotics . . . 26
2.3.2 Structural Causal Models as Representations of Robot Knowledge . . . 27
2.3.3 Causal Discovery & Parameter Learning in Robotics . . . 27
2.3.4 Integrating Causality into Robot Planning . . . 28
2.3.5 Causal Planning versus Causal Reinforcement Learning . . . 29
2.3.6 Causal Modelling for Manipulation and Robot Component Uncertainty . 29
2.3.7 Causal Reasoning for Robust & Explainable Robot Behaviour . . . 30
2.3.8 Probabilistic Programming for Flexible Causal Modelling & Inference in Robotics . . . 31
2.3.9 Summary & Transition . . . 33
2.4 Explanations & Causal Attribution in Robotics . . . 34
2.4.1 From Interpretability to Causal Explanation . . . 34
2.4.2 Interventional vs Counterfactual Explanations . . . 35
2.4.3 SCMs for Explanation, Actual Causality, and Responsibility . . . 35
2.4.4 Explanation Methods in Robotics: From Contrastive to Counterfactual . 36
2.4.5 Natural-Language Explanations & System Integration . . . 36
2.4.6 Summary . . . 37
2.5 World Models: From Physics Simulators to Generative Models . . . 38
2.5.1 Counterfactual Contrastive Learning in GenAI . . . 38
2.5.2 Game World Models & the Notion of Mechanics . . . 39
2.5.3 A Causal Framing for Mechanics . . . 40
2.5.4 Can Mechanics Be Learned From Pixels Alone? . . . 41
2.5.5 Datasets, Testbeds, & What They Measure . . . 41
2.5.6 Evaluating Consistency in Contrastive Generation . . . 42
2.5.7 Summary . . . 42

3 Background **44**
3.1 Foundations of Probabilistic & Causal Graphical Modelling . . . 45
3.1.1 Causal Effect Estimation & Confounding . . . 45
3.1.2 Directed Acyclic Graphs (DAGs) . . . 47
3.1.3 Bayesian Networks & Inference . . . 48
3.1.4 Causal Directed Acyclic Graphs (Causal DAGs) . . . 48
3.1.5 Causal Bayesian Networks (CBNs) . . . 50
3.1.6 Structural Causal Models (SCMs) . . . 53
3.1.7 Causal Hierarchy (Pearl's Ladder) . . . 55
3.1.8 Bayesian Decision Theory . . . 57
3.1.9 Latents & Marginalisation . . . 59
3.2 Counterfactual Reasoning Tools . . . 60
3.2.1 Twin-World Algorithm: Abduction–Action–Prediction (AAP) . . . 60
3.2.2 Parallel-World and Counterfactual Graphs; Causal Consistency . . . 61
3.2.3 Estimating Counterfactual Distributions . . . 62

3.2.4 Counterfactual Effect Estimation . . . 63
3.3 Causal Attribution & Human Judgement . . . 64
3.3.1 Causal Attribution Estimation: Probabilities of Necessity, Sufficiency, and Necessity and Sufficiency . . . 64
3.3.2 Responsibility . . . 66
3.4 Decision-Making and Planning under Uncertainty . . . 67
3.4.1 Greedy Next-Best-Action Selection . . . 68
3.4.2 Markov Decision Processes (MDPs) . . . 68
3.4.3 Partially Observable MDPs (POMDPs) . . . 69
3.4.4 Challenges of Probabilistic Planning . . . 70
3.4.5 Sample-Based Planning with Monte Carlo Tree Search (MCTS) . . . 72
3.4.6 POMDPs with Unobserved Confounding (UCPOMDPs) . . . 73
3.4.7 Causal & Classical Reinforcement Learning . . . 74
3.5 Deep Generative World Models . . . 77
3.5.1 Variational Autoencoders (VAEs) . . . 78
3.5.2 Transformer Architectures . . . 79
3.5.3 Diffusion & Latent Diffusion Models . . . 79
3.5.4 Counterfactual & Contrastive Diffusion . . . 80
3.5.5 Unconditional & Conditional Sampling . . . 81
3.5.6 Image Editing & In-Fill Operations . . . 81

4 Causally Informed Planning for Robots Under Partial Observability and Unobserved Confounding 83
4.1 Introduction . . . 84
4.2 Planning in Environments with Confounded Decision-Making . . . 86
4.2.1 Effect Estimation & Confounding . . . 86
4.2.2 Sources and Implications of Unobserved Confounding in Real-World Robot Systems . . . 87
4.2.3 Limitations of Probabilistic Planning in the Presence of Unobserved Confounding . . . 94
4.2.4 Vulnerability to Confounding Bias in MCTS-Based Planners . . . 94
4.2.5 Causal Limitations of POMDP-Based Robot Planning . . . 95
4.3 Confounded GridWorld Problem . . . 97
4.4 CAR-DESPOT: A Causal Approach to POMDP Model Learning & Planning for Robots in Confounded Environments . . . 100
4.4.1 SCM Representation of POMDPs . . . 101
4.4.2 Model Parameter Learning Method . . . 104
4.4.3 CAR-DESPOT: A Causally-Informed MCTS-Based POMDP Planner . . . 107
4.4.4 Robot System Integration . . . 109
4.5 Experiments . . . 111
4.5.1 Model Parameter Learning . . . 111

4.5.2 Planner Evaluation . . . 112
4.6 Results & Discussion . . . 113
4.6.1 Analysis of learned model . . . 113
4.6.2 Analysis of planning performance . . . 115
4.7 Limitations & Future Work . . . 118
4.8 Summary . . . 120

5 A Causal Bayesian Reasoning Architecture Using Probabilistic Programming for Robot Manipulation Under Uncertainty 123
5.1 Introduction . . . 124
5.2 Causal Reasoning for Robot Manipulation . . . 125
5.3 Causal Bayesian Reasoning Architecture . . . 127
5.3.1 Robot Sequential Decision-Making Causal Model . . . 127
5.3.2 Intervention-Based Causal Inference . . . 129
5.3.3 Software Interface for Hardware Integration . . . 130
5.3.4 Operational Pipeline and Execution Flow . . . 130
5.4 Exemplar Block Stacking Task . . . 131
5.4.1 Problem Definition . . . 132
5.4.2 Exemplar Task Decision-Making Causal Model . . . 132
5.4.3 Evaluation Tasks . . . 138
5.5 High-Fidelity Gazebo Robot Simulation Evaluation . . . 140
5.5.1 Experimentation Setup . . . 140
5.5.2 Task 1: Tower Stability Prediction . . . 143
5.5.3 Task 2: Greedy Next-Best Action Selection . . . 144
5.6 Results & Discussion . . . 145
5.6.1 Task 1: Tower Stability Prediction . . . 145
5.6.2 Task 2: Greedy Next-Best Action Selection . . . 148
5.6.3 Comparison to Existing Approaches . . . 152
5.7 Real-World Robot Demonstration . . . 154
5.8 Architecture Scalability & Limitations . . . 156
5.8.1 Action Spaces . . . 156
5.8.2 Sequential Decision-Making . . . 157
5.8.3 Reward Functions and Statistical Objectives . . . 158
5.8.4 Computational Limitations and Complexity . . . 159
5.8.5 Adaptability and Domain Transfer . . . 159
5.9 Summary . . . 160

6 Counterfactual-Based Post-Hoc Explanations of Robot Task Execution 161
6.1 Introduction . . . 162
6.2 From Contrastive to Counterfactual Explanations . . . 164
6.3 Structural Causal Modelling of the Robot Task . . . 166
6.3.1 SCM Formulation for the Block Stacking Task . . . 166
6.3.2 Implementation in Pyro Probabilistic Programming . . . 168
6.3.3 Counterfactual Inference via the Twin-World Algorithm . . . 169
6.3.4 Counterfactual Attribution Metrics . . . 170
6.4 Counterfactual Explanation Methods . . . 170
6.4.1 Overview of the Explanation Pipeline . . . 171
6.4.2 Method 1: Single-Variable Intervention Most Likely to Change Outcome (SVIMLTCO) . . . 171
6.4.3 Method 2: Multi-Variable Intervention Most Likely to Change Outcome (MVIMLTCO) . . . 173
6.4.4 Method 3: Responsibility-Based Attribution . . . 175
6.4.5 Additional Model Evaluations under Placement Noise & Multi-Step Dynamics. . . . 179
6.5 Generating Natural-Language Text Explanations . . . 180
6.6 Robot Explainer System & RoboTIPS Demonstration . . . 183
6.7 Limitations & Future Work . . . 185
6.7.1 Current Limitations . . . 187
6.7.2 Future Work: Human-Participant Evaluation . . . 189
6.8 Summary . . . 190

7 Counterfactual Contrastive Learning for Improving Causal Consistency in Multi-Modal GenAI Models 192
7.1 Introduction . . . 193
7.2 Problem Statement & Causal Consistency Criterion . . . 195
7.2.1 Motivating Application Domain: dSprites Counterfactual Image Editing . 196
7.2.2 Counterfactual Image Fine-Tuning Task. . . . 197
7.3 SCM View of Text-Image Diffusion & the Parallel-World Procedure . . . 199
7.4 Method: Counterfactual Contrastive Learning for Diffusion . . . 203
7.5 Text-to-Image Latent Diffusion Architecture . . . 207
7.6 Experiments on dSprites . . . 208
7.6.1 Dataset & Pair Construction . . . 208
7.6.2 Counterfactual Image-Editing Tasks . . . 209
7.6.3 Metrics for Causal Consistency . . . 210
7.6.4 Qualitative Results . . . 211
7.7 Learning Induced Conditional Dependencies . . . 213
7.7.1 Conditional dSprites Variant . . . 213
7.7.2 Evaluation & Findings . . . 213
7.8 Limitations & Future Work . . . 214
7.9 Broader Multi-Modal Pointer (Scope Note) . . . 216
7.10 Summary . . . 216

8 Multiverse Mechanica: A Playable Benchmark for Learning Game Mechanics via Counterfactual Worlds 218
8.1 Introduction . . . 219
8.2 Background & Related Work . . . 222
8.3 Formalising & Learning a Game Mechanic . . . 223
8.3.1 Illustrating Example . . . 223
8.3.2 Formal Framework for Game Mechanics . . . 227
8.4 Multiverse Mechanica: A Playable Testbed for Learning Mechanics . . . 228
8.4.1 Game Overview . . . 228
8.4.2 Implemented Mechanics (v1.0) . . . 228
8.4.3 Data Generation . . . 229
8.4.4 Visual Design Decisions . . . 229
8.4.5 Summary. . . . 230
8.5 Game Design Decisions . . . 230
8.5.1 Bridging Game Mechanics & Causal Mechanisms with System-Based Design 230
8.5.2 Impact Frames: Defining Semantic Consistency in Dynamic Causal Model Traces via *Point of Maximum Action* Concept . . . 232
8.5.3 Summary. . . . 235
8.6 Proof-of-Concept: Learning a Mechanic with Diffusion Fine-Tuning . . . 235
8.7 Scalability & Limitations . . . 238
8.8 Summary . . . 239

9 Conclusion 241
9.1 Key Insights from the Thesis . . . 242
9.1.1 Design Trade-Offs in Causal Model Construction . . . 242
9.1.2 Causality as a Framework for Integrated Model, Data, and Inference Design in Robotics . . . 243
9.2 Summary of Contributions . . . 245
9.3 Future Work . . . 247
9.4 Thesis Impact . . . 249

References 251

Appendices

A Human-Participant Study Design 262

B Prompt Specification for Caption Generation 266
B.1 Overview . . . 266
B.2 Stage 1: Caption Generation . . . 266
B.3 Stage 2: Minimally Edited Caption Generation . . . 267
B.4 Discussion . . . 268

C Multiverse Mechanica **269**
C.1 Proof-of-Concept Dataset . . . 269
C.1.1 Generation . . . 269
C.1.2 Pre-Processing . . . 270
C.2 Mechanics Implemented in Multiverse Mechanica v1.0 . . . 271
C.2.1 Shield Mechanic . . . 273
C.2.2 Elemental Immunity Mechanic . . . 275
C.2.3 Weapon Range Mechanic . . . 279
C.2.4 Spell-Casting Mechanic . . . 283
C.2.5 Spawn Magic Projectile Spell to Perform Ranged Attack . . . 285
C.3 Specifications of Generated Video Clips . . . 294
C.3.1 Resolution & Format . . . 294
C.3.2 Frame Rate, Duration & Timing . . . 295
C.3.3 Dataset Organisation . . . 295
C.3.4 Implementation . . . 296
C.3.5 Variables generated . . . 296
C.4 Suggested Metrics for Evaluating Performance . . . 296
C.4.1 Mechanic Inference. . . . 296
C.4.2 Consistency in Contrast Generation. . . . 297
C.4.3 Summary. . . . 298
C.5 Theoretical & Implementation Details for Proof-of-Concept . . . 298
C.5.1 Background on Diffusion Models and Reverse-Sampling . . . 298
C.5.2 Background on Causal Counterfactuals in Image Generation . . . 299
C.5.3 Contrastive Training via Alignment Losses . . . 299
C.5.4 Notation . . . 299
C.5.5 Method 1: L1 – Consistency Alignment . . . 300
C.5.6 Method 2: L2 – Structure Preservation at High-Noise . . . 301
C.5.7 Abduction-Action-Prediction and Its Diffusion Emulation . . . 303
C.5.8 Diffusion-Based Emulation of AAP . . . 303
C.5.9 Additional Regularizers . . . 305
C.5.10 Loss Combination . . . 306
C.6 Software Dependencies . . . 309
C.6.1 Game Engine. . . . 309
C.6.2 Causal Modelling. . . . 310
C.6.3 Reproducibility. . . . 310
C.6.4 Graph Libraries. . . . 310
C.6.5 Graph Serialisation. . . . 310

List of Figures

1.1 Challenges for Real-World Mobile Robots . . . 4
1.2 Pearl's Ladder of Causation . . . 12
1.3 Robot Causal Reasoning Conceptual Framework . . . 14

3.1 Primitive Causal Directed Acyclic Graph (Causal DAG) Structures. . . . 49
3.2 Incremental Belief Tree Construction in Online POMDP Planning . . . 71
3.3 Illustration of the Four Phases of Monte Carlo Tree Search (MCTS) . . . 73
3.4 UCPOMDP Causal DAG. . . . 74
3.5 Diffusion Denoising Process . . . 80

4.1 Back–Door Confounding DAG . . . 87
4.2 Causal DAG of Unobserved Confounding in Robot Decision–Making . . . 91
4.3 Breaking Backdoor Confounding with DAG Interventions . . . 93
4.4 Robot Decision–Making Under Unobserved Confounding . . . 96
4.5 The *Confounded GridWorld* Toy Problem . . . 98
4.6 POMDP Causal DAG. . . . 102
4.7 Intervened UCPOMDP Causal DAG. . . . 103
4.8 *CAR-DESPOT* SCM-POMDP Planning Framework . . . 108
4.9 *CAR-DESPOT* Robot System Integration & Planning Loop . . . 110
4.10 Interventional & Observational Transition Probability Heatmaps . . . 114
4.11 Planner Performance Comparison Chart . . . 116

5.1 COBRA-PPM: A Causal Reasoning Architecture for Robot Manipulation Under Uncertainty . . . 126
5.2 Causal Decision-Making and Execution Pipeline in COBRA-PPM . . . 131
5.3 Schematic Illustration of the Exemplar Block Stacking Task . . . 133
5.4 The Dynamic Causal Bayesian Network (CBN) Robot Decision-Making Model . . . 134
5.5 Physics-Based PyBullet Simulation . . . 135
5.6 Uniformly Sampled Grid of Candidate Actions . . . 139
5.7 Task-Specific Components used in the High-Fidelity Simulation Experiments . . . 141
5.8 Illustrative Instance of the Block Stacking Task in Simulation . . . 142
5.9 Empirical Characterisation of Block Position Measurement Error . . . 146
5.10 Tower Stability Classification ROC and PR Curves . . . 147
5.11 Block Placement Error Characterisation . . . 149
5.12 Candidate Action Probability Heat Map . . . 151

5.13 Comparison of Selected Block Placement Positions for a Precarious Two–Block Tower in Gazebo Simulation . . . 151
5.14 Real-World Demonstration of COBRA-PPM . . . 154

6.1 MDP Robot Task Causal DAG. . . . 168
6.2 Three Complexity Levels of Causal DAGs of SCM Models of the Block Stacking Task . . . 168
6.3 Single-Variable Intervention Most Likely to Change Outcome (SVIMLTCO) Causal Attribution Algorithm . . . 172
6.4 Multiple-Variable Intervention Most Likely to Change Outcome (MVIMLTCO) Causal Attribution Algorithm . . . 174
6.5 *Responsibility Assignment* Causal Attribution Algorithm . . . 178
6.6 Counterfactual Robot Explainer Module Hardware Demonstration at the 2024 RoboTIPS Showcase Event . . . 186

7.1 The *dSprites* Representation Learning Dataset . . . 196
7.2 Illustration of inconsistency in generative AI image editing . . . 197
7.3 SCM Causal DAG of the *dSprites* Data Generation Process: Causal Variables, Image, Caption . . . 202
7.4 Counterfactual Image Editing Twin-World Graph . . . 204
7.5 Text-to-Image Latent Diffusion Model Architecture . . . 207
7.6 Example factual–counterfactual dSprites pair . . . 210

8.1 *Multiverse Mechanica* Consistent Clips . . . 221
8.2 Shield Mechanic *Marginalised DAG (mDAG)* . . . 225
8.3 Shield Mechanic Counterfactual Graphs . . . 226
8.4 Proof-of-Concept of Shield Mechanic Learning with a Diffusion Model Trained on Game Data . . . 237

A.1 Human-Participant Study Experimental Design . . . 264

C.1 Impact Frame Generated by Multiverse Mechanica for a Parallel-World Tuple. . . 270
C.2 Full Causal DAG of a Turn in Multiverse Mechanica . . . 272
C.3 Example Counterfactual Contrast Statement for the Shield Mechanic . . . 273
C.4 Example Counterfactual Contrast Statement for the Elemental Immunity Mechanic 276
C.5 Example Counterfactual Contrast Statement for the Weapon Range Mechanic . 280
C.6 Example Counterfactual Contrast Statement for the Summon Cloud Platform Spell Mechanic . . . 289
C.7 Example Counterfactual Contrast Statement for the Self-Transform Spell Mechanic 291
C.8 Example Counterfactual Contrast Statement for the Opponent Transform Spell Mechanic . . . 293
C.9 Example Counterfactual Contrast Statement for the Opponent Levitation Spell Mechanic . . . 295
C.10 Example Counterfactual Image Tuple Generated from Fine-Tuned Text-To-Image Diffusion Model . . . 307

List of Tables

1.1 Mapping Between Chapters, RQs, and Causal Level 17
1.2 Mapping From Chapters to *Robot Causal Reasoning* Grounded Robot Cognition Sub-Problems 18

3.1 Robot Inference Queries Across the Ladder of Causation. 56

4.1 Reactive action selection distribution for *Confounded GridWorld*. 100
4.2 Algorithm Mean Performance Comparison. 115

5.1 Characterisation of Block Position Measurement and Placement Errors 146
5.2 Tower Stability Binary Classification Results 148
5.3 Performance on the Block Stacking Task in Simulation 150
5.4 Comparison of our Method to Existing Approaches 153

6.1 SVIMLTCO-Based Probability that X or Y Independently Caused the Observed Task Outcome 173
6.2 MVIMLTCO-Based Probability that X or Y Independently or XY Jointly Caused the Observed Task Outcome 175
6.3 Responsibility-Assignment-Based Probability that X or Y Independently Caused the Observed Task Outcome 177

8.1 Level-3 Shield Mechanic Multiverse Logic Statements: Conjunctions of Conflicting Conditions 224
8.2 Evaluation Results for Diffusion Fine-Tuning on Consistent-Contrast Pairs . . . 238

C.1 Mechanic Learning Evaluation Tasks with Multiverse Mechanica 297

List of Abbreviations

CBN	Causal Bayesian network.
DAG	Directed acyclic graph.
MDP	Markov Decision Process.
POMDP	Partially Observable MDP.
SCM	Structural causal model.
SCM-UCPOMDP	SCM-based formulation of a UCPOMDP.
SSP	Stochastic shortest-path problem.
UCPOMDP	POMDP with unobserved confounding.

List of Notations

Probabilistic Graphical Models

P . . . . . A probability distribution over random variables.

G . . . . . A graph (directed or undirected).

M . . . . . A probabilistic graphical model.

$\hat{P}$ . . . . . A statistical estimate of a probability distribution.

Causal Probabilistic Graphical Models

U . . . . . A finite set of unobserved confounders.

$do(X = x)$ . . . An intervention taken on random variable X to set its value to x.

$M_{do(X=x)}$ . . . . A causal model M modified by an intervention that sets X to x.

Markov Models

S . . . . . A finite set of states.

A . . . . . A finite set of actions.

T . . . . . A state transition function.

Z . . . . . A finite set of observations.

O . . . . . An observation function that specifies the probability of making observation z after taking action a and arriving in state s'.

R . . . . . A reward function.

Planning & Decision-Making

b . . . . . A belief over possible world states, in partially observable settings.

a^* . . . . . An optimal or near-optimal action that maximises an objective function.

π . . . . . A (possibly stochastic) policy that maps a state or belief to an action.

π^* . . . . . An optimal or near-optimal policy.

1 Introduction

Contents

1.1 Overview 1
1.2 Motivation & Challenges 3
1.2.1 Challenges of Real-World Autonomous Mobile Robot Systems 4
1.2.2 Defining Real-World Autonomous Mobile Robot Systems 7
1.2.3 Motivating Examples 8
1.2.4 Causal Modelling and Inference: Methods for Achieving Robot Cognition 11
1.2.5 The Ladder of Causation: A Guiding Paradigm for Robot Cognition . 12
1.2.6 Robot Causal Reasoning: A Conceptual Framework Towards Robust Robot Cognition 13
1.3 Research Questions & Contributions 13
1.4 Structure of Thesis 15
1.5 Publications 18
1.5.1 Mainline Work 18
1.5.2 Non-Mainline Work 19
1.6 Research Project Contributions 20

1.1 Overview

Robots operating in the real world must do more than execute pre-programmed behaviours: they must *perceive*, *understand*, and *reason* about their environment in order to act effectively and safely under uncertainty. In short, robust **robot cognition** is a foundational capability for autonomy. Yet the real world is complex, dynamic, and only partially observable; actions have stochastic consequences; and causal interactions between a robot, its task, and its environment

are often intricate and uncertain. These realities challenge conventional data-driven approaches that learn correlations without explicit causal structure, limiting generalisation and impairing decision-making when conditions shift.

This thesis investigates **causal artificial intelligence (AI)** as a means to endow robots with the cognitive abilities needed for assured autonomy. The central premise is that *explicit causal modelling and inference* — when combined with probabilistic and generative representations[1] — provide the representational and computational tools required for robots to move beyond correlational pattern recognition toward *causally grounded understanding, prediction, decision-making, and explanation.* In this framing, causal models are not an end in themselves; rather, they are the *method* by which we implement reasoning in service of robot cognition.

Guided by **Pearl's Ladder of Causation**, the thesis traces a progression from interventional reasoning (Level 2) toward **counterfactual cognition** (Level 3). This progression motivates the *Robot Causal Reasoning* conceptual framework introduced in this chapter, which organises the core sub-problems that must be addressed to realise robust robot cognition — including deep and classical causal world models, inference and effect estimation, task and reward specification, formal robot semantics, planning and decision-making, and causal explanation and attribution.

The thesis develops and evaluates methods across this spectrum through several contributions. First, it integrates structural causal models with online POMDP planning to mitigate confounding in decision-making under partial observability. Second, it introduces a causal Bayesian architecture for manipulation that unifies probabilistic programming, physics-based simulation, and generative modelling for predictive reasoning and action selection. Third, it advances *counterfactual* capabilities for explanation and evaluation, aligning robot inferences with human causal intuitions about responsibility and alternative outcomes. Finally, it develops learning strategies and simulation testbeds for *causal consistency* in multi-modal generative models — enabling the learning of **Level 3 (counterfactual) representations** that support counterfactual prediction, reasoning, and explanation in aid of robot cognition.

[1]In this thesis, the term generative model is used in the probabilistic modelling sense to denote models that represent a data-generating process via a joint distribution over variables and define a procedure for generating samples from this distribution. In contrast, generative AI refers to modern deep generative models (e.g., diffusion models) that are parametrised by neural networks and trained on large datasets to learn complex data distributions, enabling the generation of high-dimensional samples such as images. The methods developed in this thesis combine both perspectives, leveraging probabilistic generative modelling to enable causal reasoning and counterfactual inference in generative AI systems.

In aggregate, the thesis argues that **robot cognition** — implemented via causal and generative modelling approaches and evaluated through the lens of the Ladder of Causation — is the pathway to safe, reliable, and explainable autonomy in open, uncertain environments. The remainder of this chapter sets out the challenges motivating this stance, introduces causal modelling and inference as the methodological foundation, formalises the guiding paradigm of the Ladder, and presents the *Robot Causal Reasoning* framework that structures the research questions and contributions pursued in the chapters that follow.

Beyond its technical contributions, this research carries both scientific and societal significance. Scientifically, it advances the understanding of how causal generative models can support meta-cognitive capabilities in robots — enabling them to reason about uncertainty, perform causal inference, and generate counterfactual explanations in complex domains. Societally, innovations in robot cognition, reasoning, and explainability have the potential to enhance the trustworthiness and safety of autonomous systems across applications such as domestic service robotics, warehouse automation, inspection and maintenance, and autonomous transport. These advancements stand to benefit diverse stakeholders — from vulnerable humans in shared environments to system designers, verifiers, and regulators — and, more broadly, help to lay the groundwork for the responsible and trustworthy development and deployment of autonomous robot systems in real-world settings.

1.2 Motivation & Challenges

Achieving safe, reliable, and assured autonomy in the real world remains one of the grand challenges of robotics. Despite decades of progress, a persistent gap exists between the performance of robot systems in controlled settings and their robustness in open, dynamic environments. Real-world operation demands that robots perceive, interpret, and act under uncertainty — conditions that expose the limits of current learning-based and rule-based approaches, particularly their inability to represent and reason about the causal structure linking perception, action, and outcome.

To realise the scientific and societal benefits outlined in the previous section, robots must acquire stronger **cognitive capabilities** — the ability to understand, predict, and influence the world through causal reasoning. Developing such robot cognition requires overcoming a set of deep, interrelated challenges that limit reliable perception, inference, and decision-making

under uncertainty. These challenges motivate the causal perspective adopted in this thesis, and highlight the need for new methods in modelling, inference, and generative representation to support assured autonomy in complex, unstructured environments.

1.2.1 Challenges of Real-World Autonomous Mobile Robot Systems

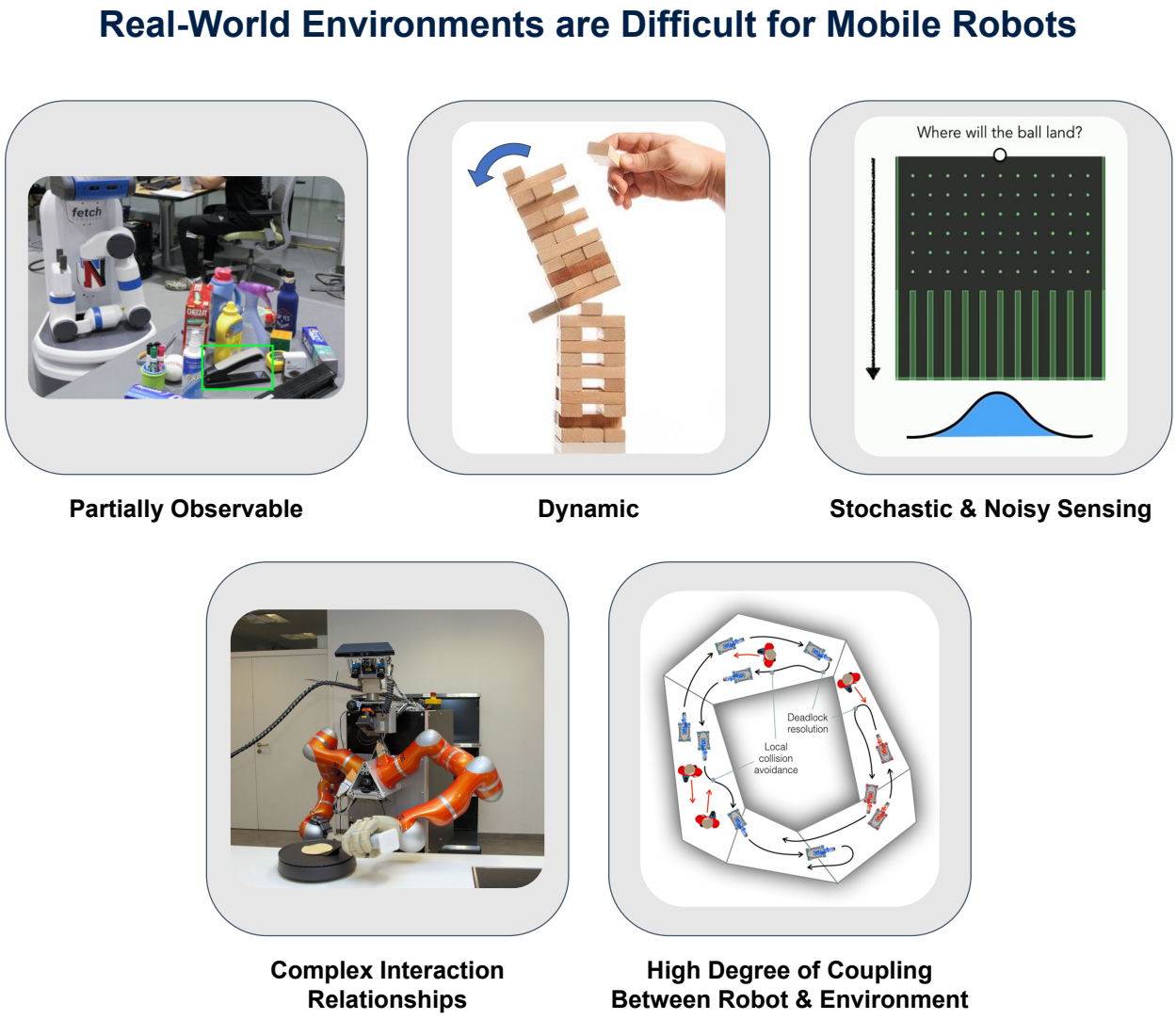


Figure 1.1: Real-world mobile robots are subject to multiple sources of uncertainty and complexity that make perception, reasoning, and decision-making particularly challenging. These include: partially observable environments [1]; dynamic and evolving environments [2]; stochastic and noisy sensing [3]; complex interaction dynamics [4]; and a high degree of coupling between robot and environment [5]. All images are adapted from the cited works.

As illustrated in Fig. 1.1, real-world autonomous robot systems operate under multiple interacting sources of uncertainty and complexity. These factors collectively constrain reliable

perception, reasoning, and decision-making, making assured autonomy especially difficult to achieve. In this thesis, we identify five key challenges that underpin these difficulties, and investigate model-based and data-driven methods to address them:

- **Partial Observability**: The robot usually does not have full observability of the states of itself and the environment, and thus it must infer hidden (i.e., *latent*) variables using indirect measurements and reason over possible world states.

- **Dynamic Nature**: The full history and reliable future state information is usually not available to the robot, requiring it to reason backward in time to infer likely causes for observed outcomes and forward to predict likely outcomes of its actions and external events.

- **Stochasticity**: Events in the real world are rarely deterministic; robot actions may fail, sensor measurements are noisy, and external processes often exhibit non-deterministic outcomes and durations.

- **Complex Interaction Relationships**: The relationships between the robot, its sensing and actuation, its task, and the environment are governed by complex, non-linear mechanisms. Faithful modelling of these requires knowledge of physics (e.g., friction, mass, inertia), electrical and mechanical systems, and sometimes application-specific domains (e.g., social norms, human physiology).

- **High Degree of Coupling Between Robot and Environment**: The decisions and actions of the robot and other entities, together with the dynamics of the world, influence the future state of both the robot and its environment. In turn, these resulting states constrain and shape the set of decisions, actions, and outcomes that remain possible for the robot in subsequent timesteps. This establishes a strong **spatial** coupling between interacting entities and a **temporal** coupling across decision horizons, both of which must be explicitly modelled to ensure coherent reasoning. Moreover, this underlying *cause-effect* coupling can give rise to **confounding**, which poses a threat to accurate statistical estimation and thereby impairs robot cognition.

Confounding in Robot Cognition. *Confounding* occurs when the *cause* and *effect* variables of a target analysis both share a *common cause* variable. This common cause is thus considered a *confounding variable* in the analysis; its presence introduces a kind of information path that allows mere correlation to propagate between the target *cause* and *effect* variables, in addition to the causal influence that flows directly from *cause* to *effect*. When quantifying the extent to which changing the *cause* variable induces a change in the *effect* variable, care needs to be taken to adjust for this confounding variable when present, to avoid conflating mere correlation with true causation. This, when performing statistical analysis in the context of robot cognition, causal inference techniques must be used to disentangle the *spurious correlation* from the causation, in order to extract only the direct and causal relationship. Although confounding is not a problem when making predictions for purely forecast-based purposes (i.e., when **not** using predictions to make decisions to take in the future), it poses a problem when predictions are used for **decision-making**. In the case of robot decision-making, predictions are used to inform the selection of a particular action, which is subsequently taken by the robot to affect the environment in aid of a desired outcome. These actions are thus interventions, which interrupt the natural flow of causation in the system. Under the presence of confounding, when an intervention is taken on the *cause* variable, the influence of the confounding variable is prevented on the cause, **but not the effect** — thus, without adjusting for this confounder, the actual outcome under the chosen action may be different due to causation not being disentangled from correlation during prediction. Consequently, there may have been another candidate action which would have yielded a better outcome, but was incorrectly estimated due to this *confounding bias* in the prediction. Making this challenge more difficult, under *unobserved confounding* — i.e., instances where the confounding variable is unobserved in data (or not even identified to exist) — traditional statistical techniques may not suffice; even causal analysis may not be able to resolve some target causal queries (i.e., *estimands*) without strong assumptions which do not always hold true in practice.

In aggregate, the high degree of coupling between the robot and the environment means that instances of confounding are more likely to occur. Furthermore, ignoring their effects can — and in practice often do — lead to robots making errors in decision-making, explanations, and causal judgements.

1.2.2 Defining Real-World Autonomous Mobile Robot Systems

In this thesis, we place a large focus on autonomous mobile robot systems in the real world, which we define as having the following properties:

- **Autonomy**: The robot must be capable of independent decision-making, beyond automation (i.e., the mechanical repetition of pre-defined behaviours), grounded in the context of its environment and prescribed tasks or goals and in response to external stimuli and observations. Consequently, robots must (under nominal conditions) operate autonomously in real-time, without reliance on low-level human control input (e.g., remote control, 'FPV' first-person-view drone control).

- **Agency**: A similar but separate dimension to autonomy, to possess agency, the robot must have the ability to exert control over its actions such that it can causally influence and change the subsequent state of itself and the environment with respect to its designated task; this presupposes that the robot is equipped with the necessary components and capabilities (e.g., sensing, actuation, and control capabilities appropriate to the task) to realise task-relevant actions, such that its decisions have a meaningful effect on future outcomes

- **Mobility**: The morphology of the robot must allow it to locomote (i.e., physically move) on its own accord. Examples include: wheeled robots, legged robots, flying drones, and submersibles. Fixed-base robots are by definition not considered mobile.

- **Robotic**: The autonomous agent is embodied in physical hardware, and thus possesses physical components for perception, cognition, and decision-making

- **System**: We use the phrase *system* to acknowledge the wide range of *physical* and *virtual* components, and their composition into a diverse range of complete robot systems; of which a robot platform may be one part.

- **Real-World**: The robot is operating in the real-world, rather than a simulation environment, thus it is subject to additional physical constraints. See below for further discussion.

Real-World Systems. For our purposes, we assume the majority of inference, reasoning, and decision-making must be done online, at deployment time, decentralised and on-board the robot to ensure the robot can *adapt* to changes in the environment in a reasonable time without dependence on external infrastructure such as communication networks. However, this does not presuppose that offline computation, such as model training and inference amortisation, is not available. These are useful and widely adopted practices to reduce the online computational burden — however, except for trivial cases, it is not feasible to pre-compute and cache decision-making to cover every possible circumstance the robot may encounter.

In a similar manner, we also assume the majority if not all sensing is done on-board the robot, without dependence on external sensing infrastructure such as global navigation satellite systems (GNSS), e.g., GPS, or precision motion capture systems.

1.2.3 Motivating Examples

Here, we provide several examples to illustrate how the challenges identified in Sec. 1.2.1 may manifest in real-world robot and AI systems in practice.

Robot Navigation. Unobserved confounding in robotics may arise from external influences that are not captured by the robot's onboard sensing and internal state estimation. Sources such as unmodelled electromagnetic fields or latent hardware faults can influence both perception and control in ways that are not directly observable. If such influences are not properly accounted for in the system model, they can lead to confounded decision-making and degraded autonomy. In Chapter 4, we discuss the *Confounded GridWorld* problem, in which a ground robot navigates a 2D grid to a goal location as part of an inspection mission, while avoiding a collision with occupied cells. In this example, we consider sensor interference from an electromagnet embedded in the partitioning wall acts as an unobserved confounder when the robot is in its proximity. This interference varies over time and is not measurable by the robot due to limited sensing capabilities, acting as a latent variable that simultaneously influences both the robot's decisions and the outcomes of those decisions. The magnet thus constitutes an **unobserved confounder** in the decision-making process — it induces a spurious correlation that cannot be resolved by standard sample-based estimators without causal adjustments. Further, left untreated, the presence of the unobserved confounder in the agent's decision-making process induces bias in the predicted transition probabilities used by sampling-based

planners. This confounding bias propagates through the planner, introducing value estimation errors that may yield sub-optimal policies — ultimately resulting in unpredictable, unsafe behaviour and task failure.

Robot Manipulation In the sequential block stacking manipulation task considered in Chapter 5, a robot incrementally builds a tower from an initial configuration and a sequence of blocks, using noisy sensor observations of the underlying latent ground truth world state and stochastic placement actions. The task is successful if the tower remains standing after the final placement and a failure if it topples at any point. A key challenge is formally representing and managing uncertainty in sensing, state estimation, and control, all of which must be accounted for to ensure reliable execution. Accurate prediction of candidate action outcomes is paramount to the success of the task; however, this requires understanding the probabilistic causal relationships governing physical object interactions, including physics-based concepts such as: collision physics, mass and inertia, friction, and gravity. Further, real-world robot dynamics are often non-linear and complex, making faithful simulation and data generation difficult.

Robot Explanations. In response to increasing concerns about the potential risks of AI and autonomous robot systems in society, there has been a growing research emphasis on *explainability* as a way to help researchers, regulators, and non-technical users better understand the internal cognitive processes and decisions of these methods and thus better manage their safe and responsible development, fine-tuning, and deployment. In particular, for circumstances in which a robot system behaves in an erroneous or non-human-intuitive manner, explanations are crucial to reconciling this violation of human expectation and repairing and rebuilding trust between human and robot system. In Chapter 6, we revisit the sequential block stacking manipulation task, but this time from the perspective of the robot generating pre- and post-hoc explanations for block stacking episodes, reasoning over its entire decision-making process including: how it made its decisions, and how the sources of sensing and manipulation uncertainty played a role in affecting both the decision-making and action outcome. To produce human-aligned explanations, the robot must reason about *actual causality*, the process by which we as humans assess how likely it was that a given cause variable was responsible for a given observed outcome to have occurred. In aid of this, the robot must find the variable most likely to have been the cause of the observed robot task outcome, considering human cognition

concepts such as responsibility, Occam's Razor, and variable semantics to address problems of causal over-determinism, multiple cause variable scenarios, and agency-based attribution bias.

Consistency in Generative AI World Models. Deep and classical generative AI robot world models allow efficient and flexible generation of novel scenarios, for use in offline model training and online reasoning. However, in addition to generating outputs — e.g., images, videos, action policies — that are visually similar to ground truth environments, generative AI world models must generate outputs that are **consistent** with human-aligned expectations of causal mechanisms that are thought to govern the interactions of depicted elements in the scene. In other words, if the generated output violates the world rules or logic, it is not faithful to the underlying causal mechanisms and, therefore, is not useful as a world model, regardless of how good it looks. Of particular relevance to robotics are deep video game world models, due to the shared requirement for world models to learn representations aligned with the visual appearance and causal behaviour of the modelled system, including: agent behaviours, physical relationships, social entity interactions, and other world logic. However, as considered in Chapters 7 and 8, for these deep world models to reliably and accurately generate outputs not only for hypothetical scenarios but also counterfactual scenarios grounded in a certain context, they must have learned the concept of *consistency* with respect to the rules of the world. Thus, to use a world model to generate novel scenarios in practical settings, e.g., for robot reasoning and policy training, we need a formalised method to ensure consistency before expending on training and deploying a model.

Having identified the limitations that arise from uncertainty, partial observability, and unobserved confounding in real-world robotic systems, we now turn to how these challenges might be addressed. The central insight underpinning this thesis is that many of these difficulties stem from a robot's inability to explicitly represent and reason about the causal relationships linking its perceptions, actions, and outcomes. When these relationships are left implicit, as in conventional data-driven learning systems, the robot is unable to distinguish correlation from causation, making it vulnerable to confounded decision-making, biased outcome prediction, and failure to generalise beyond its training environment. This distinction is critical in robotics, as decisions correspond to interventions on the environment; without causal understanding,

predictions based on correlation may not hold under action, leading to systematically biased decisions and unreliable behaviour in deployment.

To overcome these limitations, this thesis turns to the paradigm of *causal modelling and inference* — a set of methods that enable robots to build structured world models, reason about interventions, and imagine counterfactual scenarios. In the following section, we outline how causal inference provides the conceptual and computational tools required for achieving robust robot cognition.

1.2.4 Causal Modelling and Inference: Methods for Achieving Robot Cognition

The field of causal machine learning provides expressive formalisms to encode causal knowledge about the world and computational tools to exploit this knowledge for perception, planning, decision-making, and explanation. By embedding causal structure within the model itself, these approaches enable robots to move beyond purely associative pattern recognition towards understanding, predicting, and reasoning about the consequences of their actions.

Causal models confer several key advantages. First, they provide an explicit representation of cause-effect relationships, which conventional data-driven methods typically lack. This allows models to move beyond correlation and capture directional dependencies between variables. Second, causal generative models enable reasoning about latent, unobserved variables and make the assumptions of a model explicit, improving interpretability and robustness. Third, they naturally support structured representations of the spatial and temporal relationships between a robot and its environment. Finally, their probabilistic and generative nature allows uncertainty and stochasticity to be represented and manipulated within a coherent mathematical framework.

Collectively, these properties give causal models the potential to endow robots with artificial cognitive abilities — to perceive and understand their surroundings, predict the outcomes of their actions, explain observed events, and imagine causally consistent alternate worlds. In doing so, causal modelling and inference unlock pathways toward intelligent behaviours previously thought to be uniquely human.

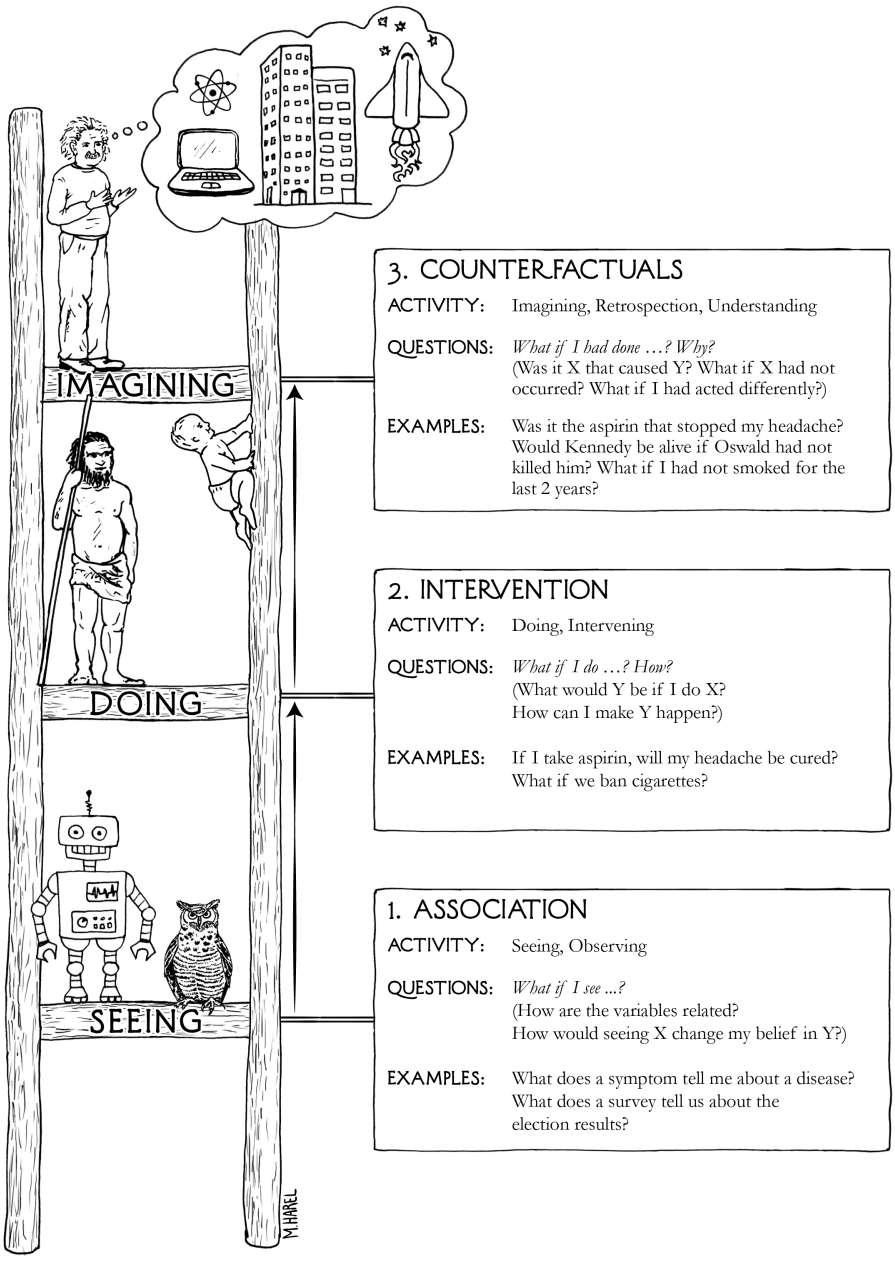


Figure 1.2: Pearl's Ladder of Causation. Adapted from Pearl and Mackenzie [6]. Each rung represents a distinct level of causal reasoning: (1) association (*seeing*), (2) intervention (*doing*), and (3) counterfactual reasoning (*imagining*).

1.2.5 The Ladder of Causation: A Guiding Paradigm for Robot Cognition

Pearl's *Ladder of Causation* [6] provides a unifying theoretical framework for understanding the levels of causal reasoning that underpin both human and artificial intelligence. As shown in Fig. 1.2, the ladder comprises three hierarchical rungs of increasing inferential complexity. Association-based reasoning concerns the recognition of statistical dependencies from

observational data; intervention-based reasoning involves understanding the consequences of deliberate actions; and counterfactual reasoning enables agents to imagine and evaluate hypothetical alternate outcomes.

The ladder thus offers a conceptual scaffold for the progressive development of robot cognition — from reactive perception and control (Level 1) to purposeful interaction with the world (Level 2), and ultimately to counterfactual imagination and reasoning about what could have been (Level 3). Achieving this progression requires integrated advances across perception, modelling, reasoning, and explanation, motivating the development of the framework introduced next.

1.2.6 Robot Causal Reasoning: A Conceptual Framework Towards Robust Robot Cognition

Building on the foundations established by the Ladder of Causation, we introduce *Robot Causal Reasoning* — a conceptual framework that identifies the key sub-problems that must be addressed to achieve robust robot cognition and reasoning under real-world uncertainty (Fig. 1.3). This framework provides a unifying structure for the thesis, outlining how progress toward generalisable, explainable, and causally grounded autonomy depends on advances across several interdependent domains of mobile robot cognition. These include the development of four core research components: causal world models, causal inference and effect estimation methods, planning and decision-making, and causal explanation and attribution; and two robot-application-specific engineering sub-problems: task and reward specification, and formal robot semantics. By bridging these research and application areas, this thesis aims to close critical capability gaps in robot autonomy — enabling safer, more reliable, and more assured operation of autonomous systems in open and uncertain environments.

1.3 Research Questions & Contributions

Primary research question:

> *How can probabilistic causal Bayesian generative AI be used to improve robot understanding and reasoning of system physicality under uncertainty and causal complexity?*

In this thesis, we investigate the following research questions in the context of real-world autonomous mobile robots operating in complex, dynamic, and partially-observable environments:

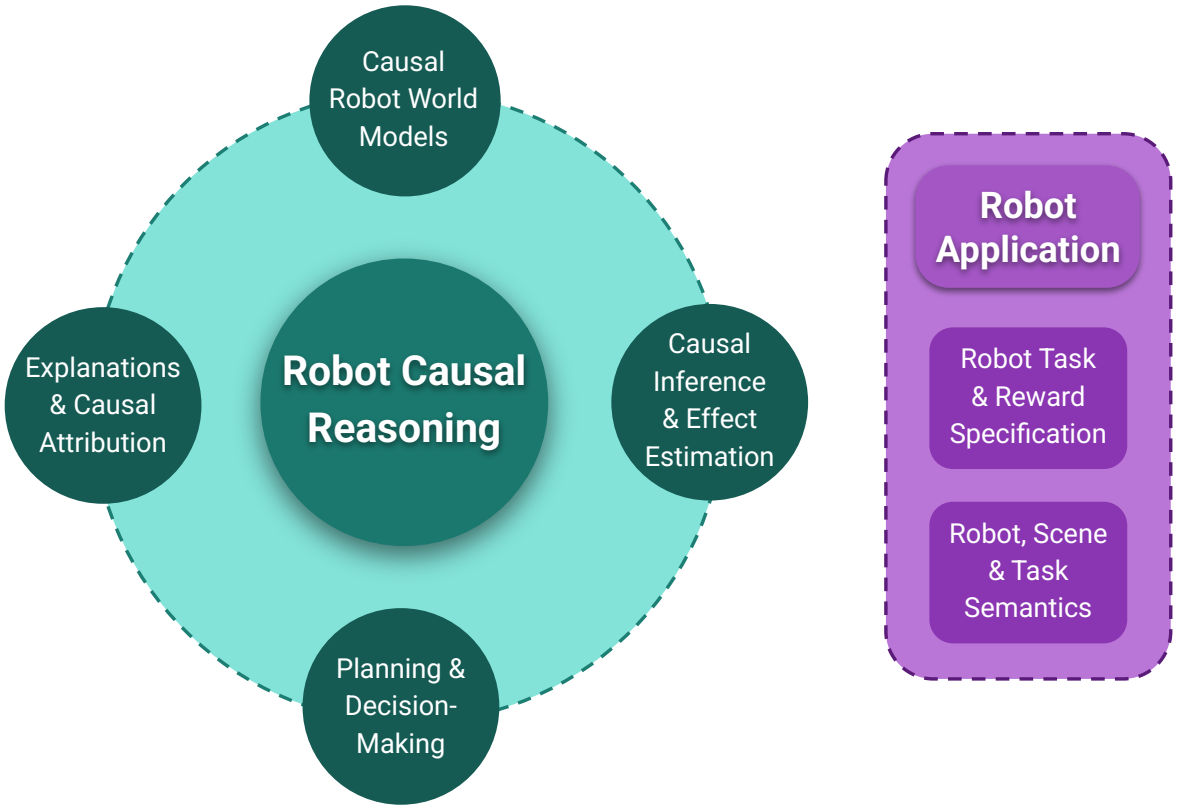


Figure 1.3: Robot Causal Reasoning — a **conceptual framework towards robust robot cognition**. The framework identifies interdependent sub-problems that must be solved to achieve **robust robot cognition and reasoning** under real-world uncertainty, guiding the contributions of this thesis. We identify four core research sub-problems and two robot-application-specific engineering sub-problems.

Q1 - Modelling: How can causal generative machine learning models be used as abstractions of uncertain and highly non-linear world dynamics and relationships?

Q2 - Learning Structure and Parametrisation: How can human-specified knowledge and data-driven learning be used to define the structure and learn the parameterisation of formal causal models?

Q3 - Confounder Bias: How can causal inference and intervention-based treatment methods be used to address confounding bias in the robot decision-making process?

Q4 - Decision Making: How can causal inference be used to extract actionable insights for robot decision-making that improve task performance and assure robust autonomy?

Q5 - Counterfactual Reasoning: How can *level 3* counterfactual knowledge and inference algorithms be used to build models that understand system physicality and generate alternative outcomes consistent with human expectations and causal intuitions?

Q6 - Counterfactual Explanations: How can counterfactual reasoning be used to generate faithful, human-aligned explanations of robot actions and outcomes, and attribute causal responsibility — considering perception, decision-making, and actuation — in support of explainable and trustworthy autonomy?

1.4 Structure of Thesis

This thesis is structured around the primary research question posed in Sec. 1.3, and is organised into chapters that progressively address the six research sub-questions (Q1-Q6). The overall narrative begins by establishing technical background and surveying related work, before moving to the core research contributions in Chapters 4-8. Each chapter makes contributions towards resolving one or more of the identified research questions:

- **Chapter 2** reviews the *related work* across robotics, causal reasoning, probabilistic programming, and explainable AI. This situates the thesis within the broader research landscape and motivates the specific research questions Q1-Q6 that are subsequently addressed.

- **Chapter 3** provides the necessary *background* on causal inference, probabilistic planning, reinforcement learning, and generative modelling. This chapter does not answer a research question directly, but instead establishes the theoretical foundations that underpin the contributions in later chapters.

- **Chapter 4** introduces *CAR-DESPOT*, a causally-informed online planner that integrates structural causal models into POMDP planning. This chapter addresses **Q2 - Structure and Parametrisation** by demonstrating how causal structure and parameterisation can be learned for use in robot planning models, and **Q3 - Confounder Bias** by showing how interventional inference mitigates confounder-induced bias in decision-making.

- **Chapter 5** presents *COBRA-PPM*, a causal Bayesian reasoning architecture that combines probabilistic programming, physics-based simulation, and causal generative models for robot manipulation. This chapter contributes to **Q1 - Modelling** by developing structured causal world models of robot-environment interactions, and to **Q4 - Decision Making** by using these models for predictive reasoning and action selection under uncertainty. It also provides the technical foundation for counterfactual reasoning and explanations explored in later chapters.

- **Chapter 6** investigates *counterfactual-based post-hoc explanations* of robot task execution. Building on the COBRA-PPM framework, it extends the causal model to a structural causal model supporting counterfactual inference. This chapter directly addresses **Q6 - Counterfactual Explanations** by proposing and evaluating causal attribution methods for generating faithful, human-aligned explanations of robot outcomes. In doing so, it also contributes to **Q1 - Modelling**, **Q2 - Structure and Parametrisation**, and **Q4 - Decision Making**.

- **Chapter 7** introduces *Counterfactual Contrastive Learning* as a method for improving causal consistency in multi-modal generative AI models. This chapter addresses **Q5 - Counterfactual Reasoning** by developing algorithms that generate alternative outcomes consistent with human expectations and causal intuitions. In doing so, it strengthens the thesis contributions toward robust counterfactual reasoning about system physicality under uncertainty.

- **Chapter 8** develops *GenAI Multiverse Counterfactuals* via *Multiverse Mechanica*, a playable benchmark for learning game mechanics through counterfactual worlds. It contributes to **Q1 - Modelling** by formalising causal generative models that instantiate parallel scenarios of the same underlying world, and to **Q5 - Counterfactual Reasoning** by specifying multiverse-style counterfactual simulation procedures and evaluation protocols for human-aligned alternative outcomes. This chapter complements Chapter 7 by providing a structured testbed for training with causal-consistency guarantees in prescribed contexts, and by introducing metrics for assessing the causal consistency of generated trajectories and outcomes, demonstrating the potential of multiverse-style counterfactual simulation for generalisable robot reasoning and causal knowledge discovery.

Chapter	Research Questions Addressed	Causal Level
4. Causally-Informed Planning	• **Q1 - Modelling** • **Q2 - Structure and Parametrisation** • **Q3 - Confounder Bias**	2. Interventional
5. Causal Reasoning for Manipulation	• **Q1 - Modelling** • **Q2 - Structure and Parametrisation** • **Q4 - Decision Making**	2. Interventional
6. Counterfactual Explanations of Robot Task Execution	• **Q1 - Modelling** • **Q4 - Decision Making** • **Q5 - Counterfactual Reasoning** • **Q6 - Counterfactual Explanations**	3. Counterfactual
7. Counterfactual Contrastive Learning	• **Q1 - Modelling** • **Q2 - Structure and Parametrisation** • **Q5 - Counterfactual Reasoning**	3. Counterfactual
8. GenAI Multiverse Counterfactuals	• **Q1 - Modelling** • **Q2 - Structure and Parametrisation** • **Q5 - Counterfactual Reasoning**	3. Counterfactual

Table 1.1: Mapping between contribution chapters, research questions (Q1-Q6), and level of causal reasoning on Pearl's Ladder of Causation (Fig. 1.2).

- **Chapter 9** concludes the thesis by synthesising the contributions of all chapters, reflecting on the research questions Q1-Q6, and identifying future research directions.

The relationship between the contribution chapters and the research questions is summarised in Table 1.1.

Taken together, these chapters provide a coherent body of work that advances the use of probabilistic causal Bayesian generative AI for modelling, inference, decision-making, counterfactual reasoning, and explanation in robotics. They collectively address the six research questions and contribute to the primary aim of improving robot understanding and reasoning of system physicality under uncertainty and causal complexity.

Chapter	Causal Inference & Effect Estimation	Task & Reward Specifications	Formal Robot Semantics	Planning & Decision-Making	Explanations & Causal Attribution	Causal World Models
4. Causally-Informed Planning	•	•	•	•	–	•
5. Causal Reasoning for Manipulation	•	○	•	•	–	•
6. Counterfactual Explanations of Robot Task Execution	•	○	•	•	•	•
7. Counterfactual Contrastive Learning	•	–	–	–	○	•
8. GenAI Multiverse Counterfactuals	•	–	–	○	○	•

Table 1.2: Mapping between contribution chapters and grounded mobile robot cognition sub-problems identified in the *Robot Causal Reasoning* conceptual framework (Fig. 1.3). Legend: • primary; ○ secondary; – not addressed.

1.5 Publications

1.5.1 Mainline Work

Research from this thesis has been published in the following works:

1. Ricardo Cannizzaro and Lars Kunze. 'CAR-DESPOT: Causally-Informed Online POMDP Planning for Robots in Confounded Environments'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Apr. 2023

2. Ricardo Cannizzaro, Jonathan Routley and Lars Kunze. 'Towards a Causal Probabilistic Framework for Prediction, Action-Selection & Explanations for Robot Block-Stacking Tasks'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2023 Workshop on Causality for Robotics*. 2023

3. Ricardo Cannizzaro, Michael Groom, Jonathan Routley, Robert Ness and Lars Kunze. 'COBRA-PPM: A Causal Bayesian Reasoning Architecture Using Probabilistic Programming for Robot Manipulation Under Uncertainty'. In: *Proceedings of the 12th European Conference on Mobile Robots (ECMR)*. Padua, Italy, Sept. 2025 †

4. Ricardo Cannizzaro, Robert Osazuwa Ness, Yunshu Wu and Lars Kunze. 'Multiverse Mechanica: A Testbed for Learning Game Mechanics via Counterfactual Worlds'. In: *The Fourteenth International Conference on Learning Representations*. 2026 *

† Winner of the Overall Best Conference Paper Award at ECMR 2025.

* Ricardo Cannizzaro, Robert Osazuwa Ness and Yunshu Wu contributed equally to this work.

1.5.2 Non-Mainline Work

In addition to the mainline thesis works above, I have also contributed to a number of non-mainline works, including the following publications:

1. Ricardo Cannizzaro, Rhys Howard, Paulina Lewinska and Lars Kunze. 'Towards Probabilistic Causal Discovery, Inference & Explanations for Autonomous Drones in Mine Surveying Tasks'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2023 Workshop on Causality for Robotics*. 2023

2. Calum Imrie, Rhys Howard, Divya Thuremella, Nawshin Mannan Proma, Tejas Pandey, Paulina Lewinska, Ricardo Cannizzaro, Richard Hawkins, Colin Paterson, Lars Kunze and Victoria Hodge. 'Aloft: Self-Adaptive Drone Controller Testbed'. In: *Proceedings - 2024 IEEE/ACM 19th Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2024* (Apr. 2024), pp. 70–76

3. Ricardo Cannizzaro, Clarissa Costen, Matthew Budd, Shu Ishida, Marc Rigter, Lars Kunze, Ioannis Havoutis, Nick Hawes and Bruno Lacerda. *Team ORIon - 2022 Team Description Paper*. 2022

4. Samuel Sze, Ricardo Cannizzaro, Matthew Munks, Kim Tien Ly, Ana Deligny, Arundathi Shanthini, Lars Kunze, Ioannis Havoutis, Nick Hawes, Victor Pozo-Fernandez, Iaroslav Okunevich and Zhi Yan. *ORIon-UTBMan 2023 Team Description Paper*. HAL-04500704. RoboCup, 2023. URL: `https://hal.science/hal-04500704`

1.6 Research Project Contributions

Mainline Work. The outcomes of this research have directly contributed to the EPSRC project Responsible AI for Long-term Trustworthy Autonomous Systems (RAILS): Integrating Responsible AI and Socio-legal Governance (EP/W011344/1) [15]. The *CAR-DESPOT* causal planning framework (Chapter 4) and the *COBRA-PPM* causal Bayesian reasoning architecture (Chapter 5) advanced RAILS objectives on responsible and trustworthy long-term autonomy, demonstrating wider impact of the thesis within a nationally funded research programme.

This research directly contributed to the EPSRC RoboTIPS project (EP/S005099/1) [16], which investigated responsible robotics for the digital economy. As part of the project's final showcase, we demonstrated a Robot Explainer Module that integrated the post-hoc counterfactual explanation methods developed in Chapter 7. This module was implemented within the framework of the Ethical Black Box — the proposal and specification of which were themselves central outcomes of RoboTIPS — making our demonstration a further key outcome of the project. In this way, the work provided a tightly aligned and tangible contribution, evidencing the practical value of causal counterfactual reasoning in responsible human-robot interaction.

Further, this research contributed to two Microsoft Research programmes. The counterfactual contrastive training in Chapter 7 informed the Societal Resilience programme [17], which pursues mission-driven research for building resilient societies. Meanwhile, the development of *Multiverse Mechanica* (Chapter 8) contributed to the AI Interaction and Learning (AIIL) group [18], supporting its agenda on interactive machine learning, evaluation, and alignment for human-AI collaboration.

Non-Mainline Work. Beyond the mainline thesis contributions, this research also contributed to the Assuring Autonomy International Programme (AAIP) (2018-2023) [19], a £12m initiative funded by Lloyd's Register Foundation and the University of York to address the global challenge of safety assurance for Robotics and Autonomous Systems. I contributed through co-authorship of the paper by Imrie et al. [12], which introduced *Aloft: Self-Adaptive Drone Controller Testbed* and demonstrated approaches for evaluating adaptive safety in autonomous aerial systems within the broader assurance agenda of the programme.

Chapter Summary

Taken together, this chapter has established the conceptual foundations of the thesis: defining the motivation and challenges of real-world autonomous robot systems, motivating causal modelling and inference as the methodological basis for achieving robot cognition, and outlining the research framework and contributions that follow. In doing so, it positions the thesis within the broader endeavour of developing robots that can reason, act, and explain in a causally coherent way under real-world uncertainty.

2
Related Work

Contents

2.1 Overview and Structure 23
2.2 Probabilistic Decision-Making & Planning Under Uncertainty . . 24
2.2.1 Foundations of Probabilistic Planning in Robotics 24
2.2.2 Offline Planning Approaches 24
2.2.3 Online Planning Approaches 25
2.2.4 Handling Partial Observability & Uncertainty 25
2.2.5 Limitations in the Presence of Confounders 26
2.2.6 Transition. 26
2.3 Causal Modelling & Inference for Robot Cognition 26
2.3.1 Motivation for Causal Modelling in Robotics 26
2.3.2 Structural Causal Models as Representations of Robot Knowledge . . 27
2.3.3 Causal Discovery & Parameter Learning in Robotics 27
2.3.4 Integrating Causality into Robot Planning 28
2.3.5 Causal Planning versus Causal Reinforcement Learning 29
2.3.6 Causal Modelling for Manipulation and Robot Component Uncertainty 29
2.3.7 Causal Reasoning for Robust & Explainable Robot Behaviour 30
2.3.8 Probabilistic Programming for Flexible Causal Modelling & Inference in Robotics 31
2.3.9 Summary & Transition 33
2.4 Explanations & Causal Attribution in Robotics 34
2.4.1 From Interpretability to Causal Explanation 34
2.4.2 Interventional vs Counterfactual Explanations 35
2.4.3 SCMs for Explanation, Actual Causality, and Responsibility 35
2.4.4 Explanation Methods in Robotics: From Contrastive to Counterfactual 36
2.4.5 Natural-Language Explanations & System Integration 36
2.4.6 Summary 37
2.5 World Models: From Physics Simulators to Generative Models . 38
2.5.1 Counterfactual Contrastive Learning in GenAI 38
2.5.2 Game World Models & the Notion of Mechanics 39
2.5.3 A Causal Framing for Mechanics 40

2.5.4 Can Mechanics Be Learned From Pixels Alone? 41
2.5.5 Datasets, Testbeds, & What They Measure 41
2.5.6 Evaluating Consistency in Contrastive Generation 42
2.5.7 Summary 42

2.1 Overview and Structure

This chapter surveys prior work directly related to the thesis research questions and methods, with an emphasis on how existing approaches enable or limit robust robot decision-making and explanation under uncertainty and confounding. Formal definitions and notation are deferred to the Background chapter (3).

Organisation. We group the literature into four domains, chosen to mirror the order of the thesis contributions:

1. **Probabilistic Decision-Making & Planning Under Uncertainty.** We review representative work on MDP/POMDP-based planning in robotics, spanning offline and online methods, and identify limitations when latent confounders are present — a gap that motivates causal extensions and connects to **Q4 - Decision Making**.

2. **Causal Modelling & Inference for Robot Cognition.** We cover applications of causal modelling and inference in robotics and ML (discovery, parameter learning, and planning), and contrast *causal planning* with *causal RL* in safety- and data-limited settings, linking to **Q1 - Modelling** and **Q5 - Counterfactual Reasoning**. This positions the causal online planner introduced later in Ch. 4.

3. **Explanations & Causal Attribution in Robotics.** We contrast interpretability with interventional and counterfactual explanations, and review causal attribution in robot tasks, addressing **Q5 - Counterfactual Reasoning** and **Q6 - Counterfactual Explanations**.

4. **World Models: From Physics Simulators to Generative Models.** We synthesise work on explicit simulators and learned/generative models (including interactive deep

genAI world models), highlighting where association-level generation violates causal consistency under interventions — motivating the need for learning causal structure. This connects to **Q1 - Modelling**.

How this positions the thesis. Together, these domains surface three cross-cutting gaps in the literature: (i) probabilistic planners that do not reason over causal structure are vulnerable to confounding; (ii) explanation methods that stop at association or intervention may misalign with human causal judgements in stochastic settings; and (iii) learned world models excel in visual fidelity yet often lack intervention-consistent dynamics. These gaps motivate the thesis's contributions in causal online planning, causal knowledge representation, and explanation for robot systems.

2.2 Probabilistic Decision-Making & Planning Under Uncertainty

2.2.1 Foundations of Probabilistic Planning in Robotics

Early approaches to robot planning typically assumed deterministic dynamics and full observability of the world, limiting their ability to cope with noise, sensor ambiguity, and actuation error. The introduction of *probabilistic* reasoning frameworks marked a turning point, enabling decision-making that explicitly models uncertainty in both perception and action. Markov Decision Processes (MDPs) and Partially Observable MDPs (POMDPs) became the canonical formalisms for sequential decision-making under uncertainty, grounding decades of research in robotics, reinforcement learning, and autonomous control. These models provide a principled means of reasoning about belief over hidden states and optimising policies that maximise long-term reward expectations. Early applications in robotics demonstrated their effectiveness for localisation, navigation, and human-robot interaction tasks [20–22], motivating the adoption of probabilistic models as the dominant paradigm for robot planning.

2.2.2 Offline Planning Approaches

Initial methods for solving MDPs and POMDPs relied on dynamic programming and value iteration to compute exhaustive policies prior to execution. This *offline* paradigm guaranteed theoretical optimality but suffered from poor scalability as the size of the state and observation

spaces increased. Point-based value iteration algorithms such as PBVI and SARSOP [21, 22] significantly improved tractability by focusing updates on sampled subsets of the belief space. These methods have been successfully applied to robot navigation, grasping, and assistive-care domains, showing that principled probabilistic reasoning can yield robust performance in partially observable settings. However, because offline solvers pre-compute policies for all reachable beliefs, they remain computationally expensive and slow to adapt to real-time changes in environment dynamics or sensory models.

2.2.3 Online Planning Approaches

To improve search speed, and thus responsiveness, later work introduced *online* planners that interleave planning and execution. These planners compute only the immediate next action at each step, conditioning on the current belief. Representative examples include POMCP [23], DESPOT [24], Adaptive Belief Trees (ABT) [25], and their variants. By employing Monte Carlo tree search and bounded policy evaluation, these approaches achieve impressive scalability and near-optimal performance in large continuous domains. In robotics, online planners have been deployed for mobile navigation, exploration, and manipulation, demonstrating the feasibility of real-time probabilistic decision-making under partial observability. Nevertheless, while these algorithms elegantly capture statistical uncertainty, they are agnostic to the *causal* structure underlying the world dynamics they model.

2.2.4 Handling Partial Observability & Uncertainty

Substantial research has addressed partial observability and uncertainty in robot perception and control. Probabilistic graphical models such as Bayesian networks, Kalman filters, and particle filters have been integrated with POMDP solvers to improve state estimation and observation modelling. These methods encode statistical correlations between variables and enable robust sensor fusion across modalities. However, they do not explicitly represent *cause–effect* relationships among system variables; rather, they capture conditional dependencies that may not generalise beyond observed data. This distinction becomes critical in domains where latent, unobserved variables influence both observations and outcomes, introducing *confounding* that purely statistical reasoning cannot resolve.

2.2.5 Limitations in the Presence of Confounders

Despite the success of probabilistic frameworks, conventional MDP and POMDP formulations treat latent variables as nuisance factors to be marginalised, rather than as potential causal influences. As a result, learned policies may exhibit *confounding bias* when unobserved factors jointly affect actions and rewards. Recent research has begun to recognise this limitation. For example, Zhang and Bareinboim [26] introduced the concept of a *Causal MDP with Unobserved Confounders*, extending standard MDP formulations with causal graphs to correct for hidden bias. Related work on confounded bandit problems and causal reinforcement learning [27, 28] demonstrates improved convergence when combining observational and interventional reasoning. However, prior to this thesis, no online POMDP planner explicitly integrated causal inference within its search process. This limitation was addressed by our work, presented in Ch. 4 and originally published in [7], which embeds causal reasoning within the probabilistic planning loop to handle decision-making under confounding.

2.2.6 Transition.

While probabilistic frameworks such as MDPs and POMDPs have proven indispensable for decision-making under uncertainty, they remain fundamentally limited by their statistical treatment of latent variables. The next section examines how causal modelling and inference have been incorporated into robotics and machine learning to address these shortcomings, paving the way for causal formulations of planning and reasoning.

2.3 Causal Modelling & Inference for Robot Cognition

2.3.1 Motivation for Causal Modelling in Robotics

Probabilistic decision-making frameworks have proven highly successful in handling uncertainty, yet they remain fundamentally limited in their ability to reason about the underlying causes of observed phenomena. In contrast, causal reasoning explicitly models how variables influence one another as *direct causal relationships*, permitting reasoning not only about what *is*, but also about what *would happen* under hypothetical interventions, and — at a deeper level — what *would have happened* under counterfactual scenarios. This distinction is central to robot cognition and autonomy, where systems must generalise beyond correlations to act robustly in novel conditions and explain their behaviour to humans.

Recent perspectives in the robotics and cognitive science literature emphasise that correlation-based models, however expressive, cannot capture the mechanistic structure required for robust and transferable behaviour [29–31]. Causal models address this limitation by explicitly representing how actions bring about changes in the world, enabling robots to anticipate the consequences of unseen interventions, diagnose failures, and interpret complex multi-modal data for explanatory reasoning. Consequently, the integration of causal reasoning into robot cognition is increasingly viewed as a necessary step toward trustworthy and generalisable autonomy.

2.3.2 Structural Causal Models as Representations of Robot Knowledge

A growing body of work represents robot world knowledge using causal graphs or *Structural Causal Models* (SCMs) [32, 33]. In these representations, each variable corresponds to a distinct aspect of the robot-environment interaction, and directed edges encode cause-effect relationships among them. SCMs differ from simpler causal Bayesian networks (CBNs) in that they include explicit structural equations, enabling both *interventional* (`do`-calculus) and *counterfactual* inference. Through this structure, robots can reason about *hypothetical* scenarios under intervention (e.g., 'What if the robot were to place the object here?') as well as about *counterfactual* alternatives relative to actual experience (e.g., 'What would have happened had the robot placed the object here, given it actually placed the object there?').

Early works demonstrated the feasibility of causal representations for task-level planning, such as causal action models for office delivery tasks [34] and multi-robot coordination [35]. Subsequent research adopted causal models to represent action-outcome relationships learned from demonstration or simulation, including table-setting [36] and block stacking tasks [37]. While these models capture useful causal structure, they often remain deterministic or lack explicit uncertainty propagation from robot components, limiting their robustness in stochastic environments.

2.3.3 Causal Discovery & Parameter Learning in Robotics

Causal discovery methods aim to infer causal structure directly from robot-environment interactions, whereas parameter learning estimates the quantitative relationships within a known structure. The latter has particular relevance for robotics, where partial causal knowledge of physical mechanisms is often available. Diehl and Ramirez-Amaro [37, 38] proposed causal

Bayesian models for manipulation that capture how parameter variations lead to success or failure in stacking tasks. Although these approaches successfully link actions to outcomes, they operate under fixed structures, are agnostic to sensor and actuator uncertainty, and rely on exhaustive offline simulations for learning.

By contrast, our work focuses on *causal parameterisation learning* under known structure, representing systems as CBNs or SCMs with probabilistic assignments. This enables robots to reason about causal dependencies while maintaining tractable belief updates within a planning framework. The first major contribution of this thesis (Ch. 4) builds directly on this principle, embedding SCM-based causal parameter learning within an online decision-making architecture to address planning under confounding. The second major contribution (Ch. 5) applies the same principle using a CBN-based formulation implemented in a probabilistic programming framework, enabling causal parameter learning for manipulation tasks.

2.3.4 Integrating Causality into Robot Planning

Integrating causal reasoning into planning offers a principled way to exploit structural knowledge of the environment while reasoning about interventions. Early causal planners relied on symbolic representations of actions and effects [34, 35], whereas more recent approaches combine causal learning with probabilistic inference for plan repair and failure prevention [38]. These models capture valuable action–outcome relationships but typically operate outside a formal Markov decision process (MDP) or partially observable MDP (POMDP) formulation, which limits their ability to reason over uncertainty and sequential decision structure. By contrast, a formal MDP-based framework provides a mathematically grounded way to represent probabilistic transitions, rewards, and observations, allowing causal dependencies to be embedded directly into the planning process. Parallel research in the causal reinforcement learning community extended standard MDP formulations to account for hidden confounders, introducing causal bandit and causal MDP problems [26, 27]. While promising, these methods typically rely on active experimentation to identify interventions and learn causal effects, making them impractical for deployment in safety-critical or physically constrained robotic systems [39]. Moreover, they do not exploit known structural relationships or quantified assignment functions that capture the mechanisms underlying these dependencies, instead re-learning them empirically from data. Consequently, they tend to be data-intensive and remain largely offline in practice.

To address this gap, our causal online planner (Ch. 4) explicitly leverages known causal structure and parametrised mechanisms within an online POMDP framework. By embedding causal reasoning inside the probabilistic search process, it enables robots to compute policies that are robust to unobserved confounders while maintaining real-time responsiveness. To the best of our knowledge, this represents the first demonstration of an **online POMDP-based robot planner that integrates causal modelling and inference** directly within its planning loop. This establishes a new class of planners capable of reasoning over both statistical uncertainty and structural causality.

2.3.5 Causal Planning versus Causal Reinforcement Learning

Causal planning and causal reinforcement learning (RL) share the goal of reasoning about cause-effect relationships but differ in how they acquire and utilise causal knowledge. Causal RL methods, such as the Confounded Bandit [27], Causal MDP with Unobserved Confounders [26], and sequence-model control frameworks [28], learn causal representations online through active experimentation. While effective in simulation, such exploration-based methods are unsuitable for safety-critical or physically constrained robotic domains, where failures are costly or irreversible [39]. Causal planning, by contrast, assumes partial structural knowledge of the system and leverages prior physical and scientific models to impose strong inductive biases. This paradigm is particularly well suited to robotics, where dynamics are governed by well-understood physical laws and uncertainty can be captured probabilistically rather than learned purely through trial and error.

2.3.6 Causal Modelling for Manipulation and Robot Component Uncertainty

In robot manipulation, causal reasoning remains comparatively underexplored. Diehl and Ramirez-Amaro [37] use a causal Bayesian network (CBN) to model block stacking, performing action selection based on outcome probabilities learned from offline simulations. While this work provides a valuable proof of concept, it does not explicitly model robot components or propagate sensor and actuator uncertainty, requiring retraining for new tasks or environmental conditions. Follow-up work on causal failure prediction and explanation [38] inherits these structural limitations. By contrast, our approach, developed in Ch. 5 and extended in Ch. 6,

performs online simulation and inference at decision time, enabling dynamic reconfiguration and explicit modelling of uncertainty through probabilistic causal mechanisms.

CausalWorld [40] introduces a benchmark environment for manipulation and transfer learning with configurable physical attributes. However, while these parameter modifications are described as interventions, they do not correspond to formal causal operations (`do`-calculus) or encode causal semantics. Instead, they act as deterministic code-level assignments without explicit modelling of underlying mechanisms. Consequently, CausalWorld lacks the causal abstraction required for interventional or counterfactual reasoning. Similarly, physics-based reasoning approaches [41] capture system dynamics but omit the causal layer necessary for probabilistic inference over interventions or confounding. Our method (Ch. 5, Ch. 6) bridges these perspectives by combining structured causal representations with formal decision-theoretic planning, enabling reasoning over physical and causal dependencies within a unified probabilistic framework.

2.3.7 Causal Reasoning for Robust & Explainable Robot Behaviour

Beyond improving decision-making, causal reasoning also underpins the explainability and accountability of autonomous systems. A growing body of work identifies causality as essential for trustworthy AI, enabling systems to provide reasons for their decisions and align with human causal intuitions [31, 42, 43]. In robotics, causal models have been employed to analyse failures, attribute responsibility, and generate explanations that mirror human causal reasoning [38, 44, 45], reflecting a broader research interest in integrating causal inference into robot cognition and behaviour. Recent advances extend this trend beyond conceptual demonstrations: Howard, Hawes and Kunze [44] show that learning causal weightings over reward metrics can yield interpretable, quantitatively validated explanations of agent interactions in real-world autonomous driving datasets, while Howard and Kunze [45] propose theoretical extensions to structural causal models that improve modularity, encapsulation, and temporal reasoning for safety-critical autonomous systems.

The second major contribution of this thesis (Ch. 5), originally published as [9], advances this line of work by embedding *interventional causal inference* within a probabilistic programming framework. This enables robots to perform hypothetical reasoning about the effects of planned interventions at the task level, providing a foundation for explainable and interpretable decision-making in manipulation and other embodied domains. Building on this, the thesis

later extends from interventional to *counterfactual* reasoning (Ch. 6), where post-hoc causal explanations are developed to account for robot behaviour in observed outcomes.

Unlike prior work focusing on autonomous vehicle interactions or theoretical model design, our approach demonstrates embodied causal reasoning through direct integration of probabilistic and causal mechanisms within the robot's decision-making process.

2.3.8 Probabilistic Programming for Flexible Causal Modelling & Inference in Robotics

Probabilistic programming languages (PPLs) provide a powerful framework for unifying model specification, simulation, and inference within a single representation. Recent advances such as Pyro [46] have extended this paradigm to support deep, hierarchical, and compositional probabilistic models, enabling flexible generative modelling over arbitrary Python-expressible functions. This flexibility makes Pyro particularly well suited to robotics, where uncertainty arises across heterogeneous sensing, actuation, and physical interaction processes.

In this thesis, Pyro serves as the causal modelling and inference *backbone* across all developed systems and experiments. It underpins the implementation of every major contribution, providing a unified foundation for defining, training, and inferring from probabilistic causal models.

Specifically, Pyro is used to implement the causal online planner (Ch. 4), where it models the dynamics of the UCPOMDP problem, including the robot–world transition function and its interventional counterpart. Within this formulation, the model can be conditioned on the previous state, surgically intervene on the robot's chosen action, and sample successor states while marginalising over unobserved confounders drawn from their natural distributions. This enables interventional transition inference to be embedded directly within the planner's probabilistic search loop. In addition, stochastic variational inference (SVI) is employed in an offline phase for *causal parameter learning* under known structure and probability distributions, allowing the estimation of categorical transition probabilities within the causal model.

Pyro is also used in the manipulation reasoning framework COBRA-PPM (Ch. 5) and the generative causal world model Multiverse Mechanica (Ch. 8), as well as for performing counterfactual inference and generating causal explanations (Ch. 6). Further, Pyro supports model training and evaluation in the counterfactual contrastive learning framework introduced later in the thesis (Ch. 7), where its flexible sampling and inference capabilities are used to model and disentangle causal generative factors in multi-modal datasets such as dSprites [47].

To the best of our knowledge, this represents the **first demonstration of Pyro as a comprehensive foundation for causal reasoning, probabilistic inference, and generative modelling in robotic and embodied AI systems**. Implementation details of the architecture are described in Sec. 5.3.

Advantages of Pyro-PPL for Causal Modelling. Pyro's tight integration with the Python ecosystem unlocks substantial advantages for robotic modelling. The language natively supports a broad collection of standard probability distributions (e.g., Gaussian, Categorical), hierarchical structures via plate notation, and custom user-defined distributions. Because the system is embedded in Python, causal models can directly incorporate existing simulation and machine learning libraries, including physics engines (e.g., PyBullet), perception modules (e.g., object detection, pose estimation), or pre-trained language and vision models for higher-level symbolic reasoning. This enables causal models to flexibly represent robot morphologies, sensing modalities, actuation mechanisms, and environment dynamics within a single executable generative program. For instance, Sec. 5.5 demonstrates how Pyro integrates with a PyBullet-based simulator to execute online causal inference through repeated forward-simulation of robot–task–world interactions.

Inference and Interventions. Representing the causal model in Pyro permits both exact and approximate inference, including sample-based methods (e.g., importance sampling [48]) and gradient-based approaches such as stochastic variational inference (SVI) [49]. Interventions are expressed through Pyro's conditioning and reparametrisation mechanisms, allowing estimation of interventional transition posteriors under the $do(\cdot)$ operator and enabling predictive queries about current or future robot states. This unified framework allows the same codebase to support interventional reasoning for decision-making, as demonstrated in Ch. 5, and to extend naturally to counterfactual inference for explanation in Ch. 6.

ROS Python Interfaces for Practical Robot System Integration. An additional advantage of using Pyro within the Python ecosystem is its natural compatibility with the *Robot Operating System* (ROS) middleware, which underpins many contemporary robot architectures. This compatibility allows causal reasoning systems developed in Pyro to be integrated directly with real or simulated robots through Python–ROS interfaces. In this thesis, we expose the

intervention-based causal inference functionality via both a pure Python API and a ROS action server API (see Sec. 5.3.3), enabling seamless deployment across simulated and hardware platforms. For example, in the COBRA-PPM implementation, the ROS interface integrates the Pyro-based inference module with the Toyota Human Support Robot (HSR) [50] in Gazebo [51], supporting real-time causal reasoning for block stacking tasks using ROS MoveIt! [52] and ArUco-based visual perception. This design demonstrates how probabilistic causal models developed in Pyro can interface transparently with operational robot systems, bridging the gap between causal inference research and practical robotic deployment.

A more detailed discussion of the implementation and advantages of Pyro for robotic causal inference is provided in Sec. 5.3.

2.3.9 Summary & Transition

In summary, causal reasoning provides a foundation for robots that can not only act under uncertainty but also reason about interventions and counterfactual alternatives to their observed experiences. While causal reinforcement learning focuses on discovering causal relationships empirically from data, causal planning exploits structured knowledge and known mechanisms to enable safe and efficient reasoning. By embedding causal inference within a formal MDP and POMDP framework, this thesis demonstrates — for the first time — how causal reasoning can be integrated directly into an online probabilistic planning architecture. This unifies probabilistic decision-making with structural causal reasoning, enabling robots to plan and act under confounding while maintaining interpretability.

Complementary advances are made in causal modelling for manipulation and component-level uncertainty, where probabilistic causal mechanisms explicitly represent sensor and actuator variability, and in the use of probabilistic programming languages (PPLs) as a unifying computational substrate for causal inference. The adoption of Pyro PPL as a modelling and inference backbone enables dynamic interventional reasoning, parameter learning, and seamless integration with real and simulated robot systems through Python–ROS interfaces. Together, these contributions establish a coherent methodological framework that bridges theoretical causal reasoning with practical robotic implementation.

Despite growing interest in causal approaches to robotics, most existing methods remain limited to predictive or interventional reasoning, lacking the counterfactual capabilities required for human-aligned explanations and responsibility attribution. The following section examines

how causal representations can be used explicitly for *explanation* and *causal attribution* in robotic systems, addressing this remaining gap.

2.4 Explanations & Causal Attribution in Robotics

Explanations in robotics must address both *why* an outcome occurred and *what would have happened otherwise*, particularly in safety-critical and human-facing domains where accountability, understanding, and trust are central. This section reviews the evolution of explanation methods for autonomous systems, contrasting interpretability-focused approaches with causal ones grounded in intervention and counterfactual reasoning. It also highlights how structural causal models (SCMs) provide a principled foundation for causal attribution, responsibility assessment, and human-aligned explanations — thereby informing the development of the counterfactual explanation framework presented in Chapter 6.

2.4.1 From Interpretability to Causal Explanation

In response to growing concerns about the safety and societal impact of AI and autonomous systems, there has been a sustained research emphasis on *explainability* as a means to enhance transparency, accountability, and public trust [53–56]. Early research in explainable AI focused on making 'black-box' neural networks more interpretable, primarily by developing tools to visualise and analyse their internal mechanisms. Such methods include attention mechanisms and saliency-based analyses that estimate the relative importance of inputs for a model's observed predictions in NLP and perception tasks [57, 58]. In robotics, contrastive explanations have also been applied to interpret the factors contributing to collision-risk predictions and autonomous vehicle behaviour [59, 60].

While these methods enhance the *intelligibility* of learned models, they remain limited to the associational level of reasoning: they highlight statistical dependencies but cannot address the causal questions of 'Why did this happen?' or 'What would have happened otherwise?'. Consequently, such methods only provide transparency and interpretability, which are at best proxies for genuine causal explanation and human attribution. To achieve explanations that are both faithful to the model's internal mechanisms and aligned with human reasoning, explanation methods must therefore engage with causal structure and reasoning.

2.4.2 Interventional vs Counterfactual Explanations

A distinction can be drawn between interventional (or *hypothetical*) explanations and counterfactual explanations. Interventional approaches examine how an outcome might change under alternative actions or parameter settings, expressed through $\texttt{do}(\cdot)$ operations that model external interventions on a system. In contrast, counterfactual explanations go further, asking: 'Given what actually happened, what would have happened if some cause variable had been different?'. This requires a structural causal model (SCM) capable of representing both factual and counterfactual worlds, using the abduction–action–prediction process to ensure consistency between them. Only counterfactual reasoning enforces the principle of *minimal change*, maintaining that all non-affected variables remain fixed while interventions alter only their causal descendants [61].

Although interventional and counterfactual inference quantities can sometimes agree, they are not guaranteed to. This is analogous to how correlation and causation may align yet remain conceptually distinct. Recent psychological work by Gerstenberg [62] demonstrates that when physical systems are sufficiently stochastic or noisy, human causal judgements align closely with counterfactual predictions, whereas hypothetical (interventional) simulations diverge. Hence, to capture how humans naturally attribute cause and responsibility, explanation methods in robotics must ascend to the counterfactual level of Pearl's Ladder of Causation [6].

2.4.3 SCMs for Explanation, Actual Causality, and Responsibility

Structural Causal Models (SCMs) offer a formal foundation for such reasoning by combining explicit causal structure with probabilistic representations of uncertainty [32]. Through their separation of exogenous and endogenous variables, SCMs enable principled *counterfactual inference* about specific instances and support reasoning about *actual causality* [63]. Two core attribution metrics introduced by Pearl [6] are the *probability of necessity* (PN) and the *probability of sufficiency* (PS), which quantify how likely a cause was necessary or sufficient for an observed outcome. Their aggregation, the *probability of necessity and sufficiency* (PNS), offers a scalar measure of causal contribution, providing a quantitative basis for assigning *responsibility*. Such counterfactual metrics have been explored in robotics only sparsely, but they form the theoretical backbone of the explanation methods developed later in this thesis.

2.4.4 Explanation Methods in Robotics: From Contrastive to Counterfactual

Causal inference has been recognised as a key enabler of trustworthy, interpretable, and fair robot autonomy [29–31, 42]. Recent work has applied causal models to diverse robotic domains, including manipulation, navigation, autonomous vehicles, and underwater systems [37–39, 55, 64, 65]. This thesis continues this broader trend by developing probabilistic and causal formulations across all core research chapters, encompassing robot manipulation, probabilistic planning, and causal explanation. Together, these contributions extend the application of structural causal models and probabilistic programming to practical robotic contexts, demonstrating how causal representations can unify perception, decision-making, and explanation within a single modelling framework. A related proposal of these ideas in the context of autonomous drones operating in subterranean mine environments was published as an extended workshop paper [11], outlining future directions for applying SCM-based causal reasoning and counterfactual explanation methods in aerial systems.

At the task level, causal Bayesian networks have been used to model and explain robot behaviour, such as in block stacking and assembly tasks [36–38, 40]. These approaches learn action–outcome probabilities and use nearest-success search strategies to identify likely causes of failures. However, since these models encode only level-2 (interventional) causal knowledge, they can produce contrastive but not counterfactual explanations, limiting their fidelity to true causal structure.

Emerging SCM-based approaches extend these models to the counterfactual level, enabling principled attribution of task outcomes to specific variables under the minimal-change assumption. This thesis contributes to this growing area by introducing a fully implemented framework for generating *post-hoc counterfactual explanations* of robot task execution, developed and validated in the block-stacking domain (Ch. 6). The framework integrates counterfactual inference (using the Twin-World algorithm) with novel explanation algorithms — including single- and multi-variable *most-likely-to-change-outcome* analyses and a responsibility-based attribution scheme — to produce human-aligned explanations of robot actions.

2.4.5 Natural-Language Explanations & System Integration

Natural-language explanation is a crucial step toward making robot reasoning accessible to non-expert users. A growing body of research explores how autonomous systems can verbalise their

decisions, actions, and situational understanding in ways that promote trust and accountability. In autonomous driving, interpretable models have been used to identify and explain risk factors contributing to collision likelihoods [59], while survey studies [54] have emphasised the need for transparency and stakeholder-specific explanations across perception, planning, and control components. Complementary human-robot interaction research has investigated *embodied question answering* as a means for robots to explain themselves in natural language, allowing users to query the causes or justifications for robot behaviour [56]. These systems often rely on symbolic or rule-based mechanisms to select and phrase explanations, improving communicative transparency but lacking an explicit causal foundation.

Recent advances have begun to couple these communicative capabilities with causal reasoning. For instance, counterfactual inference has been applied to autonomous vehicle scenarios to identify causally necessary actions that contributed to collisions [44], while related work extends structural causal models to support modular, temporally grounded attribution of agent responsibility [45]. However, these methods focus primarily on multi-agent analysis and do not generalise to embodied robot manipulation or decision-making contexts requiring fine-grained causal attributions across perception, action, and outcome variables.

By contrast, this thesis integrates causal inference, counterfactual reasoning, and natural-language explanation within a unified framework (6). Building upon the *RoboTIPS* project, the counterfactual explanation system developed here interfaces with the *Ethical Black Box* (EBB) [43] to provide post-hoc, human-interpretable accounts of robot actions grounded in quantitative causal analysis. Unlike previous work that produces symbolic or heuristic explanations, this approach verbalises results of genuine counterfactual inference derived from structural causal models. Moreover, because the architecture exposes Python–ROS interfaces, it supports deployment across both simulation and real hardware platforms, enabling the same causal reasoning and explanation mechanisms to operate seamlessly in embodied settings.

2.4.6 Summary

In summary, the evolution of explanation methods for autonomous systems has progressed from interpretability-focused techniques that visualise internal model behaviour to causal and counterfactual frameworks capable of articulating why specific actions or outcomes occurred. While early approaches improved transparency and communication, they were limited to the associational level of reasoning and could not capture the mechanistic structure required for faithful

causal attribution. Structural causal models (SCMs) address this limitation by providing a formal foundation for reasoning about *actual causality*, *responsibility*, and *counterfactual alternatives*, thereby aligning robot explanations more closely with human causal understanding.

The framework developed in this thesis (Ch. 6) integrates counterfactual reasoning, quantitative attribution, and natural-language explanation within a deployable architecture that interfaces with real and simulated robots via Python–ROS integration. This unification of causal reasoning and human-facing communication provides a foundation for robots that can not only act and decide, but also *explain why*.

2.5 World Models: From Physics Simulators to Generative Models

As robots and autonomous agents increasingly rely on simulation to anticipate the outcomes of their actions, *world models* have become central to both model-based reinforcement learning and generative AI. They provide a mechanism for internal *simulation* and *imagination* — allowing an agent to predict, plan, and reason about alternative possibilities before acting. The following sections trace this evolution from explicit physics-based simulation toward data-driven, generative, and ultimately causal world models that unify prediction, imagination, and explanation.

2.5.1 Counterfactual Contrastive Learning in GenAI

Modern multi-modal generative models, including latent diffusion systems for text-to-image synthesis [66], achieve impressive fidelity via associational objectives that align representations across different modalities or views. Standard contrastive learning, however, is agnostic to causal structure. It promotes *appearance-level* or *representational similarity* between samples that share semantic labels or textual descriptions, ensuring that visually or semantically related examples are mapped close together in latent space. Yet, such alignment concerns only the statistical co-occurrence of features, not their causal dependencies. As a result, these models learn which samples *look* similar, but not which parts of a scene are causally affected by specific changes in input. When applied to generation or editing, a targeted modification of one factor often induces collateral changes to unrelated elements, violating the principle of *causal consistency* — only descendants of an intervention should change, whereas non-descendants should remain fixed [6, 67].

Recent work has attempted to introduce causal inductive biases into diffusion and transformer-based models, for instance via structured attention or physics priors, but these approaches remain limited to correlational consistency rather than true counterfactual reasoning. This thesis extends such efforts by introducing a *counterfactual contrastive* objective that aligns training with the semantics of a structural causal model (SCM): paired generations $(v_{X=x}, v_{X=x'})$ are encouraged to differ only along descendants of X, penalising drift elsewhere. Chapter 7 develops this method for diffusion models, including SCM-grounded formulations of text-image generation and parallel-world training pairs, and evaluates adherence to causal consistency on dSprites [47] and domain variants with induced conditional dependencies. Chapter 8 then extends these ideas to deep interactive world models with formally defined mechanics and parallel-world supervision, enabling systematic tests of minimal-change behaviour under interventions.

2.5.2 Game World Models & the Notion of Mechanics

Classical physics simulators encode physical laws explicitly, ensuring mechanistic fidelity but requiring practitioners to hand-specify every relevant rule and parameter. While such simulators excel in replicating known physical systems, they struggle to generalise beyond their predefined domains, limiting their applicability to novel, stylised, or imaginative environments. In contrast, *learned world models* acquire latent representations of dynamics directly from data, offering scalability and adaptability to any relationships present in the observed domain. By conditioning on agent actions, *interactive* variants extend this paradigm further, enabling bidirectional simulation in which agents can act within and alter the learned environment. However, this flexibility comes at the cost of mechanistic interpretability: learned models may reproduce the *appearance* of valid behaviour while violating underlying causal or physical constraints, particularly under intervention.

Recent advances have explored this paradigm through the lens of **video game** world models, which provide rich, visually complex, and causally structured environments for studying interactive generation [68–72]. State-of-the-art approaches, typically leveraging deep autoregressive transformer architectures, are trained on large datasets comprising sequences of rendered frames, user inputs, and internal game-engine states. These models can generate gameplay sequences that are visually indistinguishable from the originals — an impressive feat given the cinematic fidelity of modern games. A key motivation is the procedural generation of novel, high-fidelity

gameplay experiences [73], where generated content must not only look plausible but also remain consistent with the underlying *mechanics* of the game.

Authors of game world models often claim that their systems have implicitly learned such *mechanics* — the rules governing gameplay — based on post-hoc observation of generated sequences that appear to respect those rules. For example, in their *World Models* framework, Ha and Schmidhuber [74] suggest that their model learns to simulate 'the essential aspects of the game, such as the game logic, enemy behaviour, [and] physics'. Similarly, Kim et al. [75] claim that *GameGAN* captured the collision and power-pellet mechanics of PAC-MAN, while Parker-Holder and Fruchter [69] describe the consistency of *Genie 3* as an 'emergent property'. Yet these claims are observational rather than causal: they demonstrate that a model can *appear* to replicate certain mechanics, not that it has learned them in a way that generalises across unseen contexts or supports intervention-based reasoning.

A key limitation lies in the absence of a formal definition of what it means to 'learn a mechanic'. Without such a definition, there is no principled means to assess whether a model has captured the causal dependencies that constitute the mechanic or merely learned surface-level statistical regularities. Building on design-theoretic and computational accounts [76–87], we define a *mechanic* as a modular subset of rules with preconditions and effects that update the game state and shape observable gameplay. This formalisation provides a causal basis for evaluating whether a learned world model respects or violates game mechanics under intervention, establishing the foundation for the causal analysis developed in Chapter 8.

2.5.3 A Causal Framing for Mechanics

Prior work on learned world models has typically focused on statistical prediction or appearance-level coherence, with few attempts to capture the *causal mechanisms* governing environment dynamics. In contrast, mechanics can be formalised directly as mechanisms within a *structural causal model* (SCM) [32, 33], where each rule defines a deterministic or stochastic mapping between causal variables representing game or world states. Such a framing treats a mechanic not merely as a pattern in data but as a structured process whose effects can be explicitly interrogated through intervention.

Graph-based causal representations further enable abstraction and testability. Using *marginalised DAGs* [88], analysis can focus on the subset of variables relevant to a particular mechanic, preserving the causal and interventional semantics while ignoring irrelevant factors. *Coun-*

terfactual graphs [89] extend this idea to parallel worlds, capturing equality constraints between factual and intervened instances to represent cross-world consistency. Under the *causal consistency principle* [90], variables not downstream of a given intervention must remain unchanged across these worlds, providing a concrete operational test for *minimal-change* behaviour.

This thesis adopts and extends this causal framing to evaluate the fidelity of learned world models. In *Multiverse Mechanica* (Ch. 8), mechanics are encoded as explicit SCM mechanisms within a generative world model, enabling causal testing of whether learned dynamics respect or violate these rules under intervention. This approach moves beyond correlational or descriptive analyses by providing a formal, testable link between a model's generative behaviour and its adherence to underlying causal structure.

2.5.4 Can Mechanics Be Learned From Pixels Alone?

Despite the visual fidelity of recent video and world models, empirical studies reveal that they often violate fundamental rules or physical constraints [73, 91–93]. These failures highlight that visual realism does not imply causal correctness: models can reproduce appearances while neglecting the underlying generative mechanisms that govern dynamics. Theoretically, the identifiability of causal or generative factors from purely observational data is impossible without interventions or strong structural priors [94–97]. Consequently, learning world dynamics or game mechanics from pixels alone is underdetermined. Prior approaches that rely solely on visual prediction can at best infer correlational patterns, not mechanistic rules.

This thesis adopts a causal approach that explicitly introduces intervention signals, structural constraints, and action supervision to enable reliable learning and testing of mechanics. In *Multiverse Mechanica* (Ch. 8), mechanics are not inferred post-hoc from visual coherence but represented as explicit causal mechanisms, allowing the learned model's behaviour to be validated against ground-truth interventions.

2.5.5 Datasets, Testbeds, & What They Measure

Existing benchmarks such as IntPhys and Physion probe *intuitive physics* through object permanence, collision, and stability tests [98, 99]. While valuable for assessing perceptual prediction, they do not encode formal rule structures, nor do they provide parallel-world supervision to test causal consistency. As a result, models may achieve high scores by producing perceptually plausible rollouts that nevertheless violate causal or rule-based constraints.

Multiverse Mechanica (Ch. 8) directly addresses this limitation by introducing playable generators and formally specified game mechanics, each paired with interventional control and explicit counterfactual counterparts. This enables evaluation at the *mechanics level* rather than the purely perceptual level, measuring whether models reproduce correct causal transformations under controlled interventions — an ability not captured by existing datasets.

2.5.6 Evaluating Consistency in Contrastive Generation

Evaluation of generative models remains dominated by perceptual or semantic similarity metrics. Most studies rely on human judgements of whether generated rollouts 'look right' [73], or adapt vision-language models (VLMs) to rate text-video coherence [100]. In image generation, Monteiro et al. [101] introduced *composition metrics* that test whether non-edited attributes remain constant under controlled edits — an approach conceptually related to causal consistency. However, none of these frameworks systematically evaluate *parallel-world minimal-change* behaviour, where only the causal descendants of an intervention should differ.

The *Multiverse Mechanica* framework provides the first environment for such causal evaluation. Its level-3 counterfactual data allow both human and automated (VLM-based) evaluators to be benchmarked against formal definitions of causal consistency, enabling quantitative assessment of whether generative world models respect the minimal-change criterion under $\mathrm{do}(X)$ interventions.

2.5.7 Summary

Across domains from robot cognition to generative modelling, recent research has demonstrated impressive predictive and generative capabilities yet remains largely limited to associational reasoning. Learned world models and multi-modal genAI systems frequently capture statistical regularities without respecting the underlying causal structure of the processes they model. As a result, they can reproduce visually convincing behaviour that nonetheless violates the principles of causal consistency and minimal change.

By formalising world dynamics and game mechanics as structural causal models (SCMs), this thesis provides a principled framework for identifying and testing causal mechanisms within learned generative systems. It advances both methodological and empirical frontiers: Chapter 7 introduces a counterfactual contrastive learning approach that constrains generative

training to respect causal consistency, while Chapter 8 contributes an interventionally controlled, mechanics-centric testbed for evaluating whether models uphold these causal principles in interactive environments. Together, these contributions move beyond correlational modelling toward a causally grounded foundation for explanation, simulation, and imagination in embodied and generative AI systems.

3
Background

Contents

3.1 Foundations of Probabilistic & Causal Graphical Modelling 45
3.1.1 Causal Effect Estimation & Confounding 45
3.1.2 Directed Acyclic Graphs (DAGs) 47
3.1.3 Bayesian Networks & Inference 48
3.1.4 Causal Directed Acyclic Graphs (Causal DAGs) 48
3.1.5 Causal Bayesian Networks (CBNs) 50
3.1.6 Structural Causal Models (SCMs) 53
3.1.7 Causal Hierarchy (Pearl's Ladder) 55
3.1.8 Bayesian Decision Theory 57
3.1.9 Latents & Marginalisation 59
3.2 Counterfactual Reasoning Tools 60
3.2.1 Twin-World Algorithm: Abduction–Action–Prediction (AAP) 60
3.2.2 Parallel-World and Counterfactual Graphs; Causal Consistency 61
3.2.3 Estimating Counterfactual Distributions 62
3.2.4 Counterfactual Effect Estimation 63
3.3 Causal Attribution & Human Judgement 64
3.3.1 Causal Attribution Estimation: Probabilities of Necessity, Sufficiency, and Necessity and Sufficiency 64
3.3.2 Responsibility 66
3.4 Decision-Making and Planning under Uncertainty 67
3.4.1 Greedy Next-Best-Action Selection 68
3.4.2 Markov Decision Processes (MDPs) 68
3.4.3 Partially Observable MDPs (POMDPs) 69
3.4.4 Challenges of Probabilistic Planning 70
3.4.5 Sample-Based Planning with Monte Carlo Tree Search (MCTS) 72
3.4.6 POMDPs with Unobserved Confounding (UCPOMDPs) 73
3.4.7 Causal & Classical Reinforcement Learning 74
3.5 Deep Generative World Models 77
3.5.1 Variational Autoencoders (VAEs) 78
3.5.2 Transformer Architectures 79

3.5.3 Diffusion & Latent Diffusion Models 79
3.5.4 Counterfactual & Contrastive Diffusion 80
3.5.5 Unconditional & Conditional Sampling 81
3.5.6 Image Editing & In-Fill Operations 81

As this thesis spans multiple disciplines — including causal inference, probabilistic modelling, robot decision-making, and deep generative learning — this chapter consolidates the theoretical preliminaries and mathematical notations that underpin the later contributions. Its purpose is twofold: first, to introduce the formal building blocks common to all subsequent chapters, and second, to delineate where background ends and novel contribution begins. The presentation is modular: Sec. 3.1 covers probabilistic and causal graphical models; Sec. 3.2 formalises counterfactual reasoning and estimation; Sec. 3.3 outlines causal attribution metrics used for explanation; Sec. 3.4 reviews probabilistic decision-making and planning under uncertainty; and Sec. 3.5 introduces deep generative world models used for causal imagination and data generation. Together, these sections provide a unified conceptual and mathematical foundation for the causal Bayesian and generative reasoning methods developed throughout the thesis.

3.1 Foundations of Probabilistic & Causal Graphical Modelling

Probabilistic graphical models (PGMs) and their causal extensions provide the formal language for representing, reasoning about, and manipulating uncertainty in complex systems. They enable structured modelling of how variables interact, how uncertainty propagates through a system, and how interventions or observations alter beliefs. This section introduces the key constructs of probabilistic and causal modelling that will recur throughout the thesis, starting from directed acyclic graphs and progressing to structural causal models.

3.1.1 Causal Effect Estimation & Confounding

Estimating how one variable influences another is a central problem in statistics and machine learning. In classical (non-causal) settings, *effect estimation* refers to quantifying how changes in an observed variable X are associated with changes in another variable Y, typically through regression or correlation-based analysis. Such association-based methods estimate how Y varies

with X given other covariates, but they do not determine whether changes in X *cause* changes in Y. They describe patterns of dependence, not mechanisms of influence.

From Statistical to Causal Effects. In causal inference, the goal shifts from measuring association to estimating the *causal effect* of deliberate interventions on X. Formally, causal effects are expressed using *interventional* distributions defined under the $do(\cdot)$ operator, which represents actively setting X to a chosen value while severing all incoming influences to X. For a binary treatment variable $X \in \{\text{False}, \text{True}\}$, the population-level *Average Treatment Effect (ATE)* is defined as:

$$\text{ATE} = \mathbb{E}[Y \mid do(X{=}\text{True})] - \mathbb{E}[Y \mid do(X{=}\text{False})],$$

which measures the expected change in Y when X is set to `True` rather than `False`. The *Conditional Average Treatment Effect (CATE)* further conditions this comparison on a set of covariates Z:

$$\text{CATE}(z) = \mathbb{E}[Y \mid do(X{=}\text{True}), Z{=}z] - \mathbb{E}[Y \mid do(X{=}\text{False}), Z{=}z].$$

While the ATE quantifies the overall causal effect in the population, the CATE captures how the causal effect varies across subgroups or contexts characterised by Z. These quantities are inherently *causal estimands*, as they depend on the outcomes of hypothetical interventions rather than observational conditioning alone.

Confounding and Bias. A variable Z is said to *confound* the relationship between X and Y when it influences both, opening a *back-door path* that creates a spurious statistical association even when no direct causal link exists. For example, weather may affect both the use of sprinklers (X) and the wetness of grass (Y), producing correlation without direct causation. If unaccounted for, such confounding leads to biased estimates of causal effects, since observational data cannot distinguish between genuine causal influence and correlation induced by Z.

Methods for Causal Effect Estimation. Two principal strategies are used to estimate causal effects in the presence of confounding:

1. **Causal adjustment methods.** These rely on a known or assumed causal graph to identify appropriate conditioning sets that block all back-door paths. Techniques such as the *back-door adjustment*, *front-door adjustment*, and *instrumental variable* methods enable estimation of interventional quantities $P(Y \mid do(X))$ from observational data by conditioning on relevant covariates or introducing proxy variables [32].

2. **Surgical intervention on generative models.** In probabilistic generative models such as CBNs or SCMs, causal effects can be computed by explicitly intervening in the model using the $do(\cdot)$ operator. The intervention *surgically* modifies the data-generation process by fixing X to a chosen value and removing all incoming causal influences, allowing direct computation of $P(Y \mid do(X{=}x))$ from the altered model. This mechanistic approach forms the foundation for interventional and counterfactual reasoning throughout this thesis (see Sec. 3.1.5 and Sec. 3.1.6).

Causal effect estimation thus motivates the need for structured causal models such as CBNs and SCMs, which make explicit the assumptions required to distinguish correlation from causation and provide formal machinery for identifying and estimating causal effects under uncertainty.

3.1.2 Directed Acyclic Graphs (DAGs)

A *directed acyclic graph* (DAG) $G = \langle V, E \rangle$ is a type of probabilistic graphical model (PGM) that represents a set of random variables $V = \{X_1, \ldots, X_n\}$ as nodes and directed edges E as hypothesised dependency relations among them. Edges express conditional dependence: if an edge $X_i \to X_j$ exists, X_j is said to be *conditionally dependent* on X_i given the rest of the graph. The absence of an edge implies conditional independence under the *Markov property*: each variable is independent of its non-descendants given its parents.

This graphical structure induces a *factorisation* of the joint probability distribution over all variables:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P\Big(X_i \mid pa(X_i)\Big), \tag{3.1}$$

where $pa(X_i)$ denotes the parent set of X_i in the graph. Equation 3.1 allows complex joint distributions to be represented compactly and manipulated efficiently for inference.

Graphical separation criteria such as *d-separation* provide the mechanism for identifying conditional independences directly from the graph. If a set of variables Z d-separates X and Y,

then $X \perp\!\!\!\perp Y \mid Z$ holds in every distribution consistent with the graph structure. This property forms the mathematical backbone of probabilistic reasoning, enabling inference algorithms to exploit local structure rather than operating over the full joint space.

3.1.3 Bayesian Networks & Inference

A *Bayesian network* [102] is a probabilistic model defined by a DAG together with a set of local conditional probability distributions, or *Markov kernels*, for each node:

$$P(X_i \mid pa(X_i)), \quad \forall i \in \{1, \ldots, n\}.$$

The combination of the DAG and these kernels defines a full joint distribution according to Eq. 3.1. Inference in a Bayesian network involves computing posterior beliefs over subsets of variables given evidence, typically via Bayes' rule [103, 104]:

$$P(X \mid Y) = \frac{P(Y \mid X)P(X)}{P(Y)}. \tag{3.2}$$

For high-dimensional models, exact inference is often intractable, motivating the use of approximate methods such as importance sampling [48], Markov chain Monte Carlo (MCMC) [105, 106], or variational inference [107, 108]. These methods are especially relevant to this thesis, as importance sampling underlies the estimation steps in the CAR-DESPOT planner (Ch. 4), COBRA-PPM decision-making framework (Ch. 5), and counterfactual explanation methods (Ch. 6).

3.1.4 Causal Directed Acyclic Graphs (Causal DAGs)

While directed acyclic graphs (DAGs) capture statistical dependencies, they do not by themselves specify *causal* direction. *Causal Directed Acyclic Graphs (Causal DAGs)* extend DAGs by assigning causal semantics to edges: an arrow $X_i \rightarrow X_j$ indicates that X_i is a direct cause of X_j within the model.

As illustrated in Fig. 3.1, three primitive causal DAG structures describe the fundamental causal relationships among variables: the *chain*, *common-cause* (or *common-parent*), and *collider* (or *V-structure*).

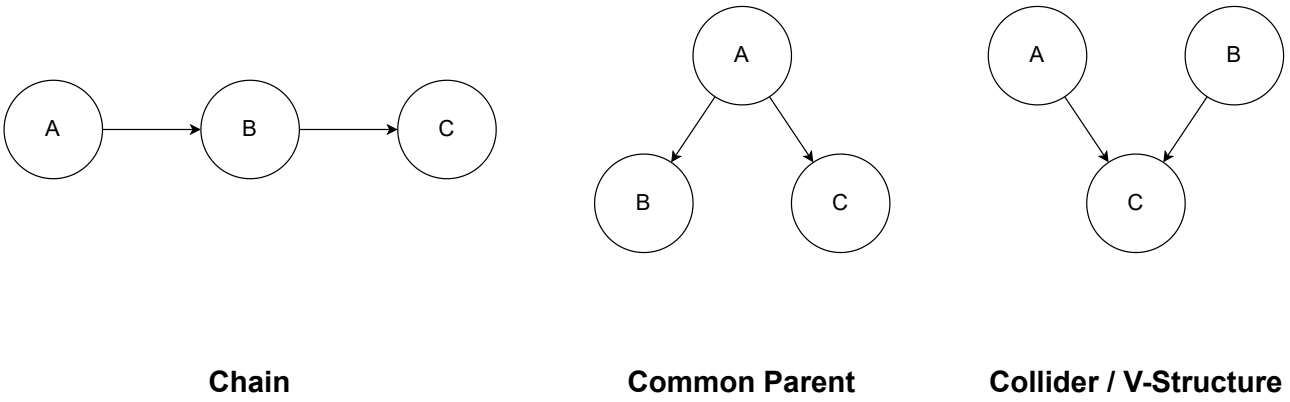


Figure 3.1: Primitive Causal Directed Acyclic Graph (Causal DAG) structures. From left to right: chain, common parent, and collider (V-structure).

Chain Structure. The *chain* structure defines variable A as a direct cause of variable B, which in turn directly causes variable C. It encodes the causal Markov kernels $P(B \mid A)$ and $P(C \mid B)$, and implies the following conditional dependencies:

- A is marginally independent of both B and C: $P(A \mid B, C) = P(A)$.
- B depends on A but is independent of C given A: $P(B \mid A, C) = P(B \mid A)$.
- C depends on B but is independent of A given B: $P(C \mid A, B) = P(C \mid B)$.

By the probability chain rule, the joint distribution factorises as

$$P(A, B, C) = P(A)\, P(B \mid A)\, P(C \mid B).$$

Common-Cause Structure. The *common-cause* structure defines variable A as a direct cause of both B and C. It encodes the causal Markov kernels $P(B \mid A)$ and $P(C \mid A)$, and implies:

- A is marginally independent of B and C: $P(A \mid B, C) = P(A)$.
- B and C are both conditionally dependent on A but independent of each other given A: $P(B \mid A, C) = P(B \mid A)$ and $P(C \mid A, B) = P(C \mid A)$.

Accordingly, the joint distribution factorises as

$$P(A, B, C) = P(A)\, P(B \mid A)\, P(C \mid A).$$

Collider Structure. The *collider* structure defines variables A and B as joint direct causes of variable C (the *collider node*). It encodes the causal Markov kernel $P(C \mid A, B)$ and implies:

- A and B are marginally independent: $P(A, B) = P(A)P(B)$.
- C is dependent on both A and B jointly: $P(C \mid A, B)$.
- Conditioning on C (or any descendant of C) induces dependence between A and B, a phenomenon known as *collider bias* or *Berkson's paradox*.

The joint distribution factorises as

$$P(A, B, C) = P(A)\, P(B)\, P(C \mid A, B).$$

These three primitives — chain, common cause, and collider — form the structural building blocks of all causal DAGs and define the basic mechanisms by which statistical dependencies emerge from underlying causal processes.

3.1.5 Causal Bayesian Networks (CBNs)

While Bayesian networks represent conditional dependencies among variables, they do not inherently encode causal direction. A *Causal Bayesian Network (CBN)* augments a causal DAG with a set of conditional probability distributions (one per node) that together define the model's generative process. Each distribution, or *causal Markov kernel*, specifies the conditional probability of a variable given its direct parents in the DAG:

$$P(X_i \mid \mathrm{Pa}(X_i)).$$

These kernels represent independent causal mechanisms in the system, each describing how a particular variable arises from its causes, This independence gives rise to the property of *causal modularity*: the mechanism governing each variable is autonomous and can, in principle, be modified without affecting others.

CBNs are particularly useful in domains such as robotics and the physical sciences, where causal relationships often correspond to underlying physical laws or designed mechanisms. In these contexts, the edges of the causal graph encode hypothesised or known physical dependencies that govern interactions in the real world — relationships that represent direct *causation*, not mere correlation. The causal Markov kernels associated with these edges capture

independent causal mechanisms that are expected to remain invariant under domain shifts or environmental changes. In physical systems, such mechanisms often correspond to well-established laws that describe how variables evolve under intervention; for example, Newton's second law ($\sum F = ma$) specifies how applied forces determine the resulting acceleration of an object, reflecting a clear direction of causation in which forces produce changes in motion, not vice versa. This relationship holds consistently across environments (e.g., on Earth and in space). This invariance property allows causal knowledge to be transported across different settings, providing robustness and interpretability beyond purely statistical models. By combining these causal kernels with probabilistic uncertainty, CBNs can represent both aleatoric (stochastic) and epistemic (modelling) uncertainty, enabling formal reasoning about systems where structure is known but behaviour is uncertain. Unlike non-causal learning methods, the assumptions in a CBN are explicitly stated and can be tested or falsified with empirical data, making them well suited for scientific and engineered systems that demand explainability and reproducibility.

Beyond representing causal dependencies, CBNs support formal reasoning about **interventions**. The intervention operator $do(\cdot)$ specifies the effect of actively setting a variable to a fixed value, thereby modifying the data-generating process itself. Mathematically, this is modelled by *surgically replacing* the causal Markov kernel for the intervened variable — e.g., setting $X_i := x'$ — while leaving all other kernels unchanged. The causal modularity property ensures that interventions are local: they affect only the mechanism governing X_i and its descendants, while the rest of the system remains intact. Pearl's *do-calculus* [32] provides a formal framework for deriving interventional distributions from such structural changes, forming the mathematical basis for counterfactual reasoning and causal inference used throughout this thesis.

3.1.5.1 Interventions

An intervention, denoted $do(X{=}x)$, removes all incoming edges to X and fixes its value to x, thereby creating a new *interventional joint distribution* over the variables in the DAG. We denote this interventional distribution by $P_{X=x}(\cdot)$, to distinguish it from the original observational distribution $P(\cdot)$. Formally, if P factorises according to the DAG as

$$P(\mathbf{V}) = \prod_{V_i \in \mathbf{V}} P(V_i \mid pa(V_i)),$$

then the intervention $do(X{=}x)$ induces a new joint distribution

$$P_{X=x}(\mathbf{V}) = \begin{cases} \prod_{V_i \in \mathbf{V} \setminus \{X\}} P(V_i \mid pa(V_i)) & \text{if } X{=}x, \\ 0 & \text{otherwise.} \end{cases}$$

Intuitively, the operation severs all causal inputs to X and sets it deterministically to x, while leaving all other mechanisms in the model unchanged.

From this modified joint, we can derive interventional conditionals such as

$$P(Y \mid do(X{=}x)) = \sum_{\mathbf{z}} P(Y \mid X{=}x, \mathbf{z})\, P(\mathbf{z}),$$

where $\mathbf{z}$ marginalises over the non-descendants of X. Such interventional distributions characterise how the world would behave if an external agent were to force X to take a particular value, rather than merely observing it.

A crucial distinction between **intervention** and **conditioning** is that an intervention on X does not provide evidence with which to update beliefs about the ancestors of X, as a conditioning operation does. Consequently, an intervention does not allow information to flow causally upstream; it only alters the target variable and its descendants in the graph. This separation between **seeing** and **doing** is central to causal inference, and to the ability of causal models to predict the effects of actions that disrupt the natural data-generating process.

Causal Structure Assumptions and Uncertainty. Throughout this thesis, the causal graph structure is assumed to be known or specified *a priori*, based on domain knowledge or system design. This assumption is common in many robotics and physical modelling settings, where the underlying mechanisms are partially understood. However, in general, the true causal structure may itself be uncertain, misspecified, or only approximately known. In such cases, inference and decision-making are conditioned on the assumed structure, and any errors in the graph may propagate to downstream predictions, interventions, and counterfactual explanations. While learning or maintaining distributions over causal structures is an important direction for future work, this thesis focuses on reasoning under a fixed, hypothesised structure, and explicitly analyses the implications of this assumption in the relevant chapters.

3.1.5.2 Dynamic CBNs

Dynamic CBNs extend this formalism to sequential settings by explicitly modelling causal structure over time. A *time-window DAG* provides the base representation, typically spanning two adjacent time steps (t and $t{+}1$), though longer windows may be used when additional temporal dependencies must be represented. Within each time window, the graph remains acyclic; temporal dependencies are captured by directed edges from variables at time t to variables at time $t{+}1$. Rolling out this local acyclic structure across multiple time steps yields a dynamic causal graph that mirrors the evolution of a stochastic system over time.

This formulation reconciles the acyclicity of causal DAGs with the apparent feedback and looping behaviour of real-world dynamical systems. What may appear as a cycle at the system level is represented instead as a directed path across successive time steps rather than as an instantaneous cycle within a single graph. Dynamic CBNs are therefore particularly useful for representing **time-series** and sequential decision processes, where causal relations unfold temporally rather than instantaneously. In this thesis, this temporal causal perspective provides a natural foundation for modelling robot–environment interactions and planning under uncertainty (see Sec. 3.4).

In this thesis, we make use of this abstraction at two levels. First, at the decision-making level, the (PO)MDP process itself is modelled as a known causal structure, providing an inductive bias over the evolution of states, actions, and observations. Second, at the transition level, we model the causal relationships between state variables, actions, and environment dynamics within each time step, where the transition function specifies how S_t and A_t determine the distribution over S_{t+1}. This separation allows us to maintain causal clarity while modelling both high-level decision processes and low-level system dynamics within a unified framework.

3.1.6 Structural Causal Models (SCMs)

A *structural causal model* (SCM) is a strictly more expressive type of causal model that augments a CBN with explicit *assignment functions* linking causes to their effects. Whereas a CBN specifies causal dependencies between variables through conditional probability distributions, an SCM further **mathematically** defines the deterministic mechanisms that generate each variable's value from its direct causes and exogenous influences. Formally, an

SCM is defined as a 4-tuple

$$\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathbf{F}, P(\mathbf{U}) \rangle,$$

where $\mathbf{U}$ is a set of exogenous (unobserved) variables, $\mathbf{V}$ a set of endogenous (observed) variables, $\mathbf{F}$ a set of deterministic structural assignments f_i such that

$$v_i := f_i(pa_i, u_i), \quad \forall v_i \in \mathbf{V},$$

and $P(\mathbf{U})$ a joint distribution over the exogenous variables. The structural functions $\mathbf{F}$ capture the deterministic mechanisms of the system, while $P(\mathbf{U})$ introduces stochastic uncertainty from factors external to the model.

This functional decomposition endows SCMs with the ability to represent and compute **counterfactuals** — queries about how outcomes would differ under alternative hypothetical interventions. By fixing the exogenous variables $\mathbf{U}$ (which encode the latent background conditions of a specific factual world) and varying the assignment functions or interventions, an SCM can simulate multiple hypothetical worlds that share the same underlying noise realisation. This capability distinguishes SCMs from CBNs, enabling reasoning not only about *what will happen* under an intervention, but also *what would have happened otherwise*. Such counterfactual reasoning forms the foundation for the methods discussed later in Sec. 3.2.

Connection to Generative AI Models. Structural causal models can be interpreted as explicit, structured generative models, in which the structural equations define a data-generating process over variables conditioned on exogenous inputs. In this view, SCMs specify how observations arise from underlying causal mechanisms, with stochasticity captured through the distribution over exogenous variables. This perspective connects naturally to modern deep generative models, such as diffusion models, which learn to approximate high-dimensional data-generating processes from data, but typically without explicit causal structure. In later chapters (Chs. 7, 8), we build on this connection by formulating generative modelling as a causal process, and show how incorporating causal structure can enforce cross-sample consistency and enable principled counterfactual reasoning in high-dimensional generative settings.

3.1.7 Causal Hierarchy (Pearl's Ladder)

As illustrated in Fig. 1.2, Pearl's *Ladder of Causation* [6] formalises three progressively expressive levels of causal reasoning:

- **1) Association**: reasoning from observations, expressed as $P(Y \mid X)$.
- **2) Intervention**: reasoning about deliberate actions, expressed as $P(Y \mid do(X{=}x))$.
- **3) Counterfactual**: reasoning about alternative realities, expressed as $P(Y_{X=x'} \mid X{=}x, Y{=}y)$.

Each higher level subsumes the lower, but not vice versa: purely observational data cannot, in general, estimate interventional or counterfactual quantities. The ladder therefore serves both as a taxonomy of reasoning capabilities and as a diagnostic for determining whether a model is genuinely causal.

Robot Inference Queries Across the Ladder of Causation. To ground the Ladder of Causation in this thesis' focus on robot cognition, Table 3.1 provides illustrative examples of robot inference queries at each level of Pearl's Ladder of Causation.

Levels of Data & Estimation. Formally, level-k statements describe events in the sample space of a level-k distribution [33]. Level-k data can be viewed as samples from such a distribution, and under i.i.d. (independent and identically distributed) sampling, the empirical distribution converges to the true sampling distribution. Level-1 (associational) data arises from passive observation of naturally occurring events. Level-2 (interventional) data is obtained when one or more variables are deliberately fixed before sampling, such as in a controlled experiment. Level-3 (counterfactual) data corresponds to paired outcomes across parallel worlds that share the same exogenous conditions but differ in the interventions applied. In real-world settings, such data cannot exist due to the *fundamental problem of causal inference* — we cannot observe multiple outcomes for the same unit across different worlds [109]. In this thesis, however, we exploit the video game setting's ability to simulate parallel instances with shared initial conditions, thereby enabling the collection of level-3 data across *virtual worlds*.

Causal Level	Example Robot Inference Queries
1. Association (Seeing)	• Based on my noisy position measurement, how likely am I to be within the magnet's interference region? • Given my current observation, what is the probability that the current block tower configuration is stable? • Given my current observation of the block tower, how confident am I in the block positions given my sensing and pose-estimation noise characteristics?
2. Intervention (Doing)	• If I steer forward instead of backward, will I still reach the goal despite magnetic interference? • What would happen if I stack the next block in this position? • If I adjust my gripper placement to correct for actuation bias, will the tower remain stable?
3. Counterfactual (Imagining)	• Why did my planned trajectory fail when sensor noise appeared negligible? • Given the tower collapsed, would it have remained stable if the top block were placed differently? • What would the resulting world-state image look like if the robot had acted differently at this step?

Table 3.1: Examples of robot inference queries at each level of Pearl's Ladder of Causation [6]. Level 1 (*seeing*) corresponds to observational inference over sensory and belief data; Level 2 (*doing*) to interventional reasoning about the effects of robot actions, as in the Confounded GridWorld (Ch. 4) and Block Stacking tasks (Ch. 5); and Level 3 (*imagining*) to counterfactual reasoning over hypothetical outcomes and generated world states, as explored in the counterfactual explanation (Ch. 6) and generative world-modelling chapters (Ch. 7, Ch. 8).

Models & their Alignment with the Hierarchy. The ladder also provides a natural mapping between model classes and the types of queries they can answer:

- A **statistical model** (e.g., a probabilistic graphical model or standard neural network) operates at level 1. It can estimate associations such as $P(Y \mid X)$ but cannot predict the effects of interventions or handle confounding.

- A **causal Bayesian network (CBN)** corresponds to level 2. When paired with a generative model, it encodes the family of interventional distributions $P(Y \mid do(X{=}x))$ over the DAG's variables, allowing the estimation of causal effects under interventions.

- A **structural causal model (SCM)** corresponds to level 3. It augments the CBN with functional mechanisms and exogenous variables, thereby encoding the family of

counterfactual distributions $P(Y_{X=x'} \mid X{=}x, Y{=}y)$ over the DAG's variables [32].

Identifiability & Data Sufficiency. Level-k data is, in general, sufficient to identify level-k models, but not models from higher levels. The *causal hierarchy theorem* states that level-k statements cannot, in general, be inferred from data below level k [33]. For instance, no amount of purely observational (level-1) data can reveal the effects of an unobserved intervention (level-2) or a counterfactual (level-3) query without additional causal assumptions. This limitation motivates the use of structured causal models such as CBNs and SCMs, which provide the additional information needed to simulate interventions and counterfactual worlds beyond what data alone can reveal.

Summary. In summary, Pearl's ladder unifies reasoning, data, and model classes into a single hierarchy: (1) associational reasoning corresponds to statistical estimation from observations; (2) interventional reasoning corresponds to causal estimation under deliberate manipulations; and (3) counterfactual reasoning corresponds to hypothetical estimation under shared exogenous conditions.

While this thesis does not perform purely observational (level 1) reasoning in isolation, observational inference remains fundamental to levels 2 and 3 due to the hierarchical nature of Pearl's ladder: interventional and counterfactual reasoning both rely on observational quantities for conditioning and estimation. The specific alignment between reasoning levels, research questions, and chapters is summarised in Table 1.1, which maps the methods developed in this thesis to their corresponding rung on the Ladder of Causation.

3.1.8 Bayesian Decision Theory

Bayesian decision theory provides a normative framework for rational decision-making under uncertainty [110–112]. It combines probabilistic inference with utility optimisation, prescribing that an agent should choose the action that maximises its expected utility given its current beliefs about the world. Formally, if $a \in \mathcal{A}$ denotes an action and θ denotes the state of the world, the optimal decision rule is

$$a^* = \arg\max_{a \in \mathcal{A}} \mathbb{E}_{P(\theta|D)}[U(a, \theta)],$$

where $U(a, \theta)$ is the utility (or reward) function and $P(\theta \mid D)$ is the posterior belief over states given observed data D. Bayes' rule provides the mechanism for updating beliefs as new evidence arrives:

$$P(\theta \mid D) = \frac{P(D \mid \theta)P(\theta)}{P(D)}.$$

This framework underlies the concept of a *rational agent* that acts to maximise expected utility rather than relying on heuristics or fixed rules. It is central to probabilistic planning and reinforcement learning, where decisions must be made under uncertainty about system dynamics and future outcomes.

In this thesis, Bayesian decision theory provides the theoretical foundation for the decision-making and optimisation processes used across multiple contributions. In Ch. 4, it forms the basis of the CAR-DESPOT planner, where Bayesian belief updates guide action selection under uncertainty. In Ch. 5, it underpins the COBRA-PPM decision-making framework, which uses probabilistic beliefs and expected-utility optimisation to select causal actions without long-horizon planning. It also informs the counterfactual explanation methods of Ch. 6, where Bayesian inference is applied to compute posterior distributions over latent causes given observed outcomes, and explanations are made under the assumption that the robot agent acts *rationally*.

Practical Inference Considerations. In practice, the Bayesian principles described above are instantiated using tractable approximate inference methods. While this thesis makes extensive use of probabilistic programming frameworks (e.g., Pyro) to implement Bayesian inference, the focus is on the underlying modelling assumptions and decision-theoretic formulation rather than low-level implementation details. In practice, we employ standard approximate inference techniques, including stochastic variational inference (SVI) with *maximum a posteriori* (MAP) point estimation, Monte Carlo methods such as importance sampling for posterior estimation, and tractable parametric modelling choices such as Dirichlet priors over categorical transition distributions and Gaussian noise models for perception and actuation uncertainty. These choices reflect a pragmatic Bayesian approach, in which full posterior inference is often computationally infeasible, and approximations are selected based on their

suitability to the task, their scalability, and their empirical performance in downstream decision-making. Where such approximations are used, their assumptions (e.g., independence, distributional form) and limitations are made explicit, and their impact on inference and control is analysed in the relevant chapters.

3.1.9 Latents & Marginalisation

In causal graphical models, some variables may be latent or unobserved and thus omitted from the model's explicit node set. Marginalising such variables from a causal DAG alters the dependency structure among the remaining observed variables, potentially introducing *induced dependencies* that do not correspond to direct causal links.

Evans [88] formalised this process through the concept of a *marginalised DAG* (*mDAG*), which represents the effects of marginalisation using *hyperedges*. Directed edges in the mDAG encode standard parent–child causal relations among observed variables, while each hyperedge connects a set of observed nodes that share a common unobserved ancestor. Formally, a hyperedge connecting nodes $\{X, Y, Z\}$ indicates the existence of at least one latent variable acting as a common cause of X, Y, and Z simultaneously. This construction preserves the causal interpretation, intervention semantics, and conditional independence properties of the original full DAG [88], thereby ensuring **causal consistency** when reasoning over partially observed systems. This property underpins the formulation used in *Multiverse Mechanica* (Ch. 8), where marginalisation and explicit reconstruction of latent structures are required to maintain consistent causal relationships across parallel worlds.

In later chapters, we extend this formulation by reinstating explicit latent parent nodes corresponding to shared exogenous factors in cases where a fully expanded DAG representation is required for counterfactual graph construction (see Sec. 3.2).

In summary, probabilistic graphical models provide the syntactic scaffolding for encoding dependencies, while causal models — e.g., CBNs and SCMs — endow this structure with semantic meaning about how the world responds to actions and hypothetical changes. The following section builds directly upon these foundations to introduce the formal machinery for counterfactual reasoning and causal estimation.

3.2 Counterfactual Reasoning Tools

Causal reasoning extends beyond predicting the consequences of interventions (Level-2 reasoning) to asking how the world *would have been different* under alternative actions or conditions (Level-3 reasoning). Such queries — *counterfactuals* — require models that explicitly represent and compare parallel worlds that differ only in certain variables while keeping everything else constant. This section introduces the core machinery for counterfactual inference used throughout this thesis: the Twin-World formulation, graphical representations of parallel worlds and their consistency constraints, and estimation strategies for counterfactual quantities. Together, these form the conceptual and computational basis for the causal effect estimation and counterfactual explanation methods developed in Chapters 6, 7, and 8.

3.2.1 Twin-World Algorithm: Abduction–Action–Prediction (AAP)

The *Twin-World* construction formalises counterfactual reasoning within a structural causal model (SCM; see Sec. 3.1.6) via a three-step procedure that constructs a hypothetical world sharing the same exogenous conditions as the factual world [32].

Often summarised as *Abduction–Action–Prediction (AAP)*, the steps are:

1. Abduction. Given factual observations $\mathbf{v}_0$ of endogenous variables (and any actions realised as factual interventions), infer the posterior of SCM model $\mathcal{M}$ over exogenous variables:

$$P(\mathbf{U} \mid \mathbf{V}{=}\mathbf{v}_0).$$

This step explains the observed world by identifying latent factors $\mathbf{U}$ consistent with $\mathbf{v}_0$.

2. Action. Construct a counterfactual model by copying the SCM, fixing the exogenous variables to the abducted posterior, and applying an intervention to a target variable:

$$do(X{=}x').$$

This yields $\mathcal{M}_{X=x'}$ in which X is clamped to x' while the remaining mechanisms are unchanged.

3. Prediction. Simulate the counterfactual world under $\mathcal{M}_{X=x'}$ to obtain the distribution of outcomes:

$$P(\mathbf{V}_{X=x'} \mid \mathbf{V}{=}\mathbf{v}_0).$$

Holding $\mathbf{U}$ fixed ensures the two worlds are comparable *all else equal*.

Algorithmically:

Algorithm 3.1 Abduction–Action–Prediction (AAP) for Counterfactual Inference

Require: SCM $\mathcal{M} = (\mathbf{U}, \mathbf{V}, \mathbf{F}, P(\mathbf{U}))$, factual $\mathbf{v}_0$, factual action x_0 (if applicable), target counterfactual x_1

1: **Abduction:** sample $\hat{\omega} \sim P(\mathbf{U} \mid \mathbf{V}{=}\mathbf{v}_0, X{=}x_0)$
2: **Action:** form $\mathcal{M}_{X=x_1}$ and set $\mathbf{U}{:=}\hat{\omega}$
3: **Prediction:** compute $\hat{\mathbf{v}}_1 = f_{\mathcal{M}_{X=x_1}}(\hat{\omega})$
4: **return** counterfactual outcome $\hat{\mathbf{v}}_1$

The Twin-World view isolates the causal effect of the intervention by ensuring the factual and counterfactual worlds share the same exogenous noise. This construction underlies our counterfactual explanations (Ch. 6), counterfactual contrastive learning method (Ch. 7) and multiverse-style generative imagination (Ch. 8).

3.2.2 Parallel-World and Counterfactual Graphs; Causal Consistency

Graphical representations make cross-world structure explicit. Given a base causal DAG G, a *parallel-world graph* duplicates the descendants of the intervened variable and indexes variables by world (e.g., $X^{(0)}$ factual, $X^{(1)}$ counterfactual), applying $do(X^{(1)}{=}x')$ in the counterfactual copy [89]. Variables that are not descendants of the intervention remain *shared* across worlds, enforcing **consistency**: if a node's parents are identical across worlds, the node itself must be identical.

Merging nodes constrained to be equal yields a *counterfactual graph* that encodes conditional independences across worlds and supports the computation of joint counterfactual distributions such as $P(Y_{X=x}, Y_{X=x'})$. We employ these graphs both analytically and operationally, as templates for supervision in simulation-based learning; see Chapters 7 and 8.

3.2.3 Estimating Counterfactual Distributions

Counterfactual quantities such as $P(Y_{X=x'} \mid X{=}x, Y{=}y)$ compare mutually exclusive outcomes and are not directly observable in the real world. Within the Twin-World AAP framework, they can be estimated by Monte Carlo sampling over abducted exogenous variables and deterministic simulation of the intervened model.

Let $\hat{\omega}^{(i)} \sim P(\mathbf{U} \mid \mathbf{V}{=}\mathbf{v}_0)$ denote samples of the exogenous variables obtained during *abduction*. Each sample defines a corresponding counterfactual realisation under the modified model $\mathcal{M}_{X=x_1}$:

$$\mathbf{v}_1^{(i)} = f_{\mathcal{M}_{X=x_1}}(\hat{\omega}^{(i)}).$$

Aggregating these samples yields the empirical counterfactual distribution

$$\hat{P}(\mathbf{V}_{X=x_1} \mid \mathbf{V}{=}\mathbf{v}_0) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbf{v}_1^{(i)}},$$

which converges *almost surely* to the true distribution $P(\mathbf{V}_{X=x_1} \mid \mathbf{V}{=}\mathbf{v}_0)$ under standard assumptions of i.i.d. sampling and model correctness.

More generally, consider an arbitrary joint counterfactual distribution P_i over a set of factual and counterfactual variables — such as $P_i = P(\mathbf{X}_{I=1}, \mathbf{X}_{I=0}, \mathbf{Z})$, where $\mathbf{X}$ denotes the target variables affected by an intervention indexed by I, and $\mathbf{Z}$ represents any conditioning variables that remain shared across worlds. Repeatedly sampling consistent factual-counterfactual contrasts $\mathcal{D}_i$ from the twin-world model yields an empirical estimate $\hat{P}_i$ obtained by averaging over the sampled joint realisations:

$$\hat{P}_i = \frac{1}{N}\sum_{n=1}^{N} \hat{P}\Big(\mathbf{X}_{I=1}^{(n)}, \mathbf{X}_{I=0}^{(n)}, \mathbf{Z}^{(n)}\Big).$$

As $N \to \infty$, the empirical estimate $\hat{P}_i$ *converges almost surely* to its population distribution P_i, providing a consistent basis for estimating multivariate counterfactual contrasts from repeated, structurally aligned simulations.

In this thesis, such Monte Carlo estimation procedures underpin the computation of empirical level-3 (counterfactual) quantities in both analytical models and virtual-world experiments (see Chapters 6, 7, and 8).

3.2.4 Counterfactual Effect Estimation

In addition to interventional causal quantities such as the ATE and CATE (Sec. 3.1.1), causal effects can also be expressed using *counterfactual* distributions defined under both factual conditioning and hypothetical interventions on the model.

A canonical example is the *effect of the treatment on the treated (ETT)*, which measures how the observed outcome for a treated subject differs from the outcome that *would have been observed* had the same subject not received the treatment. Formally,

$$ETT \;=\; \mathbb{E}\Big[Y_{X=1} - Y_{X=0} \,\Big|\, X{=}1\Big], \tag{3.3}$$

where $X{=}1$ denotes that the treatment was received ($X{=}0$ otherwise), and Y is the outcome of interest. This is a counterfactual query because it requires evaluating a hypothetical world ($Y_{X=0}$) for individuals whose factual world corresponds to $X{=}1$ — a setting that is never directly observed.

Conceptually, these three estimands occupy different granularities of causal reasoning:

- The **ATE** quantifies the causal effect at the *population-level*.
- The **CATE** captures how the causal effect varies at the *subpopulation-level*, across contexts characterised by Z.
- The **ETT** quantifies the causal effect at the *individual-level*, conditioned on having received the treatment.

Together, they bridge from interventional to counterfactual reasoning, unifying *population-level* policy evaluation with *instance-specific* causal explanation. In this thesis, ETT-like reasoning underpins the analysis of robot counterfactual explanations in Ch. 6.

Summary. The counterfactual reasoning framework introduced in this section provides the formal machinery for reasoning about alternative worlds and hypothetical outcomes. Through the Twin-World (Abduction-Action-Prediction) construction, parallel-world and counterfactual graphs, and Monte Carlo estimation of counterfactual distributions, we can compute well-defined quantities such as interventional effects, individual-level counterfactual contrasts, and the effect of treatment on the treated (ETT). Together, these tools enable models to go beyond prediction — to infer *why* an outcome occurred and *how* it would have changed under alternative conditions. This capability forms the theoretical basis for causal attribution, explanation, and decision-making under uncertainty explored in subsequent sections and chapters.

3.3 Causal Attribution & Human Judgement

Causal attribution quantifies how much specific factors or events contributed to an outcome. In the context of agents acting in an environment (human, robotic, or otherwise), such causal contributions underpin the assignment of *responsibility* or *blame*. Responsibility, however, applies only to *agents* — entities capable of intentional action and moral evaluation — whereas causal attribution more generally concerns the structural role that any variable or event plays in bringing about an effect. From a causal inference perspective, attribution is formalised through counterfactual probabilities that evaluate *necessity* and *sufficiency* relations [6, 32].

This section introduces these core attribution quantities — the **probabilities of necessity (PN)**, **sufficiency (PS)**, and **necessity and sufficiency (PNS)** — and explains how they correspond to legal and philosophical notions of causation and culpability. Together with a graded measure of *responsibility*, they provide a principled bridge between formal causal reasoning and human-like judgements of accountability.

3.3.1 Causal Attribution Estimation: Probabilities of Necessity, Sufficiency, and Necessity and Sufficiency

Probability of Necessity (PN). Given that $X{=}1$ and $Y{=}1$ were observed, PN asks whether Y would still have occurred if X had been 0:

$$PN \;=\; P\Big(Y_{X=0} = 0 \,\Big|\, X{=}1, Y{=}1\Big). \tag{3.4}$$

A high PN indicates that the presence of X was *necessary* for Y — that is, Y would not have occurred without X.

Probability of Sufficiency (PS). Given that $X{=}0$ and $Y{=}0$ were observed, PS asks whether setting $X{=}1$ would make Y occur:

$$PS \;=\; P\Big(Y_{X=1} = 1 \,\Big|\, X{=}0, Y{=}0\Big). \tag{3.5}$$

A high PS indicates that X was *sufficient* to bring about Y — that introducing X would have produced the outcome.

Probability of Necessity and Sufficiency (PNS). PNS captures cases where X is both necessary and sufficient for Y:

$$PNS \;=\; P\Big(Y_{X=1}=1,\, Y_{X=0}=0\Big). \tag{3.6}$$

In the Boolean case, this decomposes as [32]:

$$PNS \;=\; P(X{=}1, Y{=}1)\, PN \;+\; P(X{=}0, Y{=}0)\, PS. \tag{3.7}$$

PNS therefore quantifies the exclusive causal contribution of X — how often Y occurs when X is true and fails when X is false.

Interpretation & Connection to Legal Reasoning. PN and PS together define the minimal logical tests for attributing causation under uncertainty. In Western legal theory, this duality corresponds closely to two central doctrines:

- **But-For Causation** (or *factual causation*) asks whether the harm would still have occurred *but for* the defendant's action — mirroring the notion of *necessity* captured by PN.

- **Proximate Causation** concerns whether the act was sufficiently direct or potent to bring about the outcome — corresponding to *sufficiency*, as quantified by PS.

Both conditions must typically hold for an agent to be deemed *culpable*: the act must have been both a necessary and a sufficiently proximate cause of the effect. This alignment highlights how counterfactual causal reasoning provides a formal underpinning for long-standing concepts in legal and moral judgement.

In the context of autonomous systems, these same quantities offer a principled way to quantify how strongly a robot's action contributed to a given outcome. By evaluating necessity and sufficiency through simulation and intervention, we can identify whether an agent's decision was both indispensable and causally sufficient — criteria essential for assessing accountability in autonomous behaviour (see Ch. 6).

3.3.2 Responsibility

The causal attribution estimations of PN, PS, and PNS do not by themselves specify how to divide credit or blame among multiple contributing causes. Halpern [63] introduced a graded notion of *responsibility* that refines attribution by considering the minimal number of additional changes needed to alter the outcome. Formally, if X_i is identified as a cause of Y, its degree of responsibility is defined as

$$R_i \;=\; \frac{1}{N_i + 1}, \tag{3.8}$$

where N_i is the smallest number of other variables whose values must also be changed (in addition to X_i) to reverse the outcome Y.[1] A decisive cause that alone determines the outcome (N_i=0) receives full responsibility (R_i=1), whereas causes that influence the outcome only in combination with others have proportionally lower responsibility.

This graded formulation addresses two key classes of causal ambiguity that arise in multi-cause settings:

- **Overdetermination.** In *overdetermined* scenarios, an outcome has two or more independent causes, each sufficient on its own to bring about the effect (e.g., two robots simultaneously pressing a button that triggers the same event). Traditional binary notions of necessity and sufficiency fail here, since neither cause is individually necessary once the other is present. The responsibility measure resolves this by assigning each cause a fractional share of responsibility, reflecting that while each was individually sufficient, the outcome would have occurred even if one had not acted.

- **Causal interaction.** In cases of *interaction* or *joint causation*, multiple factors are individually insufficient but jointly necessary to produce the outcome (e.g., two robot subsystems that must both activate for a failure to occur). Here, responsibility reflects the joint dependence of the outcome on all contributing variables, reducing the score for any single factor while still recognising its essential role within the interaction.

[1] We use N_i here to denote the minimal number of additional changes required for counterfactual reversal, following Halpern [63]. Later chapters reuse N in different contexts (e.g., for sample counts); the meaning will be clear from context.

By quantifying the minimal causal cooperation required to alter an outcome, responsibility offers a principled way to apportion causal credit or blame among interacting causes. In the context of autonomous systems, this provides a formal basis for distinguishing between decisive and contributory actions — an essential step toward transparent, human-aligned explanations of robot behaviour (see Ch. 6).

Responsibility under Uncertainty. In the formulation above, the quantity N_i is defined with respect to a particular causal model and assignment of variables. In many practical settings, however, there is uncertainty over the underlying causal model or system state, giving rise to a distribution over possible worlds $\omega \sim p(\omega)$. Since the minimal number of changes $N_i(\omega)$ required to reverse the outcome may vary across these worlds, the corresponding responsibility $R_i(\omega) = \frac{1}{N_i(\omega)+1}$ also becomes a random variable.

In such cases, responsibility is naturally generalised as the expected responsibility under the belief distribution over possible worlds:

$$\mathbb{E}[R_i] \;=\; \mathbb{E}_{\omega\sim p(\omega)}\left[\frac{1}{N_i(\omega)+1}\right]. \tag{3.9}$$

This probabilistic interpretation yields a graded notion of responsibility in $[0, 1]$ that accounts for uncertainty in the causal model and state. This formulation is developed further and instantiated for explanation in Ch. 6.

Summary. Together, PN/PS/PNS and responsibility provide a compact, counterfactual toolkit for attributing outcomes to specific actions and events — a foundation we leverage for explanatory modules in later chapters.

3.4 Decision-Making and Planning under Uncertainty

To ensure their safe, reliable, and effective operation, autonomous agents must be capable of selecting actions that maximise long-term performance despite stochastic dynamics, partial observability, and incomplete knowledge. Decision-theoretic frameworks formalise this challenge by casting planning as an optimisation problem over probabilistic transitions and rewards. Such formulations have been successfully applied across robotic domains including mobile navigation, object manipulation, and human-robot interaction [20, 113]. This section reviews the main decision-theoretic models used throughout this thesis: greedy next-best-action selection, Markov decision processes (MDPs), and partially observable MDPs (POMDPs), followed

by value recursion, sample-based planning with Monte Carlo Tree Search (MCTS), and the limitations of conventional probabilistic planning in confounded environments. Together, these provide the foundation for the causal planning framework of Ch. 4 and the causal Bayesian manipulation architecture of Ch. 5 and Ch. 6.

3.4.1 Greedy Next-Best-Action Selection

A widely used baseline in robotics is the *greedy* or *next-best-action* policy, which selects the action that maximises immediate expected reward given the current state:

$$a^* \;=\; \arg\max_{a \in \mathcal{A}} \; \mathbb{E}[R(a \mid s)]\,. \tag{3.10}$$

Greedy selection is computationally efficient and effective when temporal coupling between actions is weak, but its short-sightedness can lead to suboptimal or cyclic behaviour. Taking actions that appear optimal in the short term may lead the agent into dead ends or oscillations that prevent progress towards globally optimal solutions requiring longer action sequences.

A concrete example arises in mobile robot navigation: a robot may greedily select the neighbouring cell that minimises Euclidean distance to the goal, only to enter a corridor that ends in a dead end. Without reasoning over longer horizons, the robot becomes trapped or oscillates between adjacent states, despite an existing viable route.

Consequently, greedy policies are most suitable for domains where sequential decisions are largely decoupled, and long-horizon reasoning offers marginal benefit. In such settings, their computational efficiency is a virtue — as in the causal Bayesian reasoning architecture for manipulation under uncertainty presented in Ch. 5. However, when actions are tightly coupled — where early choices affect the reachability of future states — myopic strategies degrade sharply in performance. This motivates more expressive decision frameworks that reason over entire state-action *trajectories*, as explored in Ch. 4.

3.4.2 Markov Decision Processes (MDPs)

When sequential dependencies matter, decision-making is modelled as a *Markov Decision Process* (MDP) [114]. An MDP formalises the interaction between an agent and its environment as a stochastic dynamical system.

Definition 3.1 (Markov Decision Process)**.** A *Markov Decision Process* (MDP) is defined as a tuple $\langle S, A, T, R \rangle$, where:

- S is the set of states representing the possible configurations of the world;
- A is the set of actions available to the agent;
- $T : S \times A \times S \to [0, 1]$ is the transition function, where $T(s, a, s') = P(s' \mid s, a)$ gives the probability of transitioning to state s' when action a is executed in state s;
- $R : S \times A \times S \to \mathbb{R}$ is the reward function, where $R(s, a, s')$ specifies the immediate reward obtained by taking action a in s and reaching s'.

The agent's objective is to find a policy $\pi : S \to A$ that maximises expected discounted cumulative reward:

$$V_\pi(s) \;=\; \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \, R(s_t, \pi(s_t)) \;\middle|\; s_0{=}s\right], \tag{3.11}$$

where $\gamma \in [0, 1)$ is a discount factor that prioritises near-term rewards. To compute the optimal policy, we apply the *Bellman optimality equation* for MDPs [115]:

$$V^*(s) \;=\; \max_{a \in A} \left\{ R(s, a) \;+\; \gamma \sum_{s'} T(s, a, s') \, V^*(s') \right\}. \tag{3.12}$$

In fully observable or deterministic environments, dynamic programming and reinforcement learning algorithms can efficiently approximate π^* from this recursive form.

3.4.3 Partially Observable MDPs (POMDPs)

Robotic systems rarely have full access to the true world state. Instead, they rely on noisy and incomplete sensory data. The *Partially Observable Markov Decision Process* (POMDP) extends the MDP to model such uncertainty [20].

Definition 3.2 (Partially Observable Markov Decision Process)**.** A *POMDP* is defined as a tuple $\langle S, A, T, Z, O, R \rangle$, where:

- S is the set of possible world states;
- A is the set of available actions;
- $T : S \times A \times S \to [0, 1]$ is the transition function, $T(s, a, s') = P(s' \mid s, a)$;
- Z is the set of observations;

- $O : S \times A \times Z \to [0, 1]$ is the observation function, where $O(s', a, z) = P(z \mid s', a)$ gives the likelihood of observing z after executing a and arriving in s';
- $R : S \times A \times S \to \mathbb{R}$ is the reward function.

Some formulations include the discount factor $\gamma \in [0, 1)$ in the tuple; here, it is treated as a tunable parameter reflecting task-specific preferences for short- versus long-term rewards.

The agent maintains a *belief* $b(s)$ — a probability distribution over S — which it updates using the Bayes filter after each action-observation pair:

$$b'(s') \;=\; \eta\, O(s', a, z) \sum_{s \in S} T(s, a, s')\, b(s), \tag{3.13}$$

where η is a normalisation constant. The belief itself becomes the state for planning, allowing the value of a policy $\pi : \mathcal{B} \to A$ to be written as

$$V_\pi(b) \;=\; \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t\, R(s_t, \pi(b_t)) \;\middle|\; b_0{=}b\right]. \tag{3.14}$$

3.4.4 Challenges of Probabilistic Planning

Despite their expressive power, exact methods for solving MDPs and POMDPs quickly become intractable in realistic domains. The scalability of POMDP planning is fundamentally limited by the *curse of dimensionality* and *curse of history* [113, 116]. The number of reachable belief states grows rapidly with planning depth D, action branching factor $|\mathcal{A}|$, and observation branching factor $|\mathcal{Z}|$, leading to exponential growth in the belief tree. Here, we use calligraphic symbols such as $\mathcal{A}$ and $\mathcal{Z}$ to refer to the abstract action and observation spaces when reasoning about their size or complexity, rather than the standard symbols such as A and Z used to refer to components of a particular model. As shown in Fig. 3.2, even for a small example with $|\mathcal{A}| = 2$, $|\mathcal{Z}| = 2$, and $D = 2$, the number of nodes increases dramatically with depth due to branching over both actions and observations. This exponential growth imposes severe memory and computational demands during online planning.

In POMDP planning, the complexity of belief tree construction poses a significant challenge. A key source of this complexity lies in the branching factor: at each belief node, the tree branches into $|\mathcal{A}|$ action nodes, and each action node branches into $|\mathcal{Z}|$ possible observations, resulting in exponential growth in the number of nodes with depth D, i.e., $\mathcal{O}(|\mathcal{A}|^D |\mathcal{Z}|^D)$.

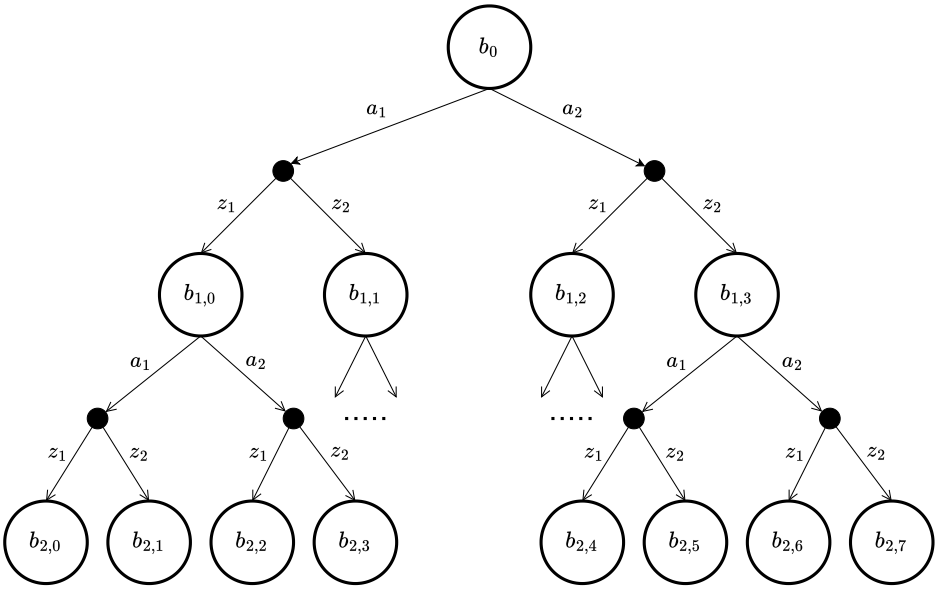


Figure 3.2: Illustration of incremental belief tree construction in online POMDP planning. This example shows a partial belief tree expanded to depth $D = 2$, with $|\mathcal{A}| = 2$ actions and $|\mathcal{Z}| = 2$ observations. Each belief node represents the agent's belief state at a given point in the search, indexed by its depth and position at that level. As the depth increases, the number of nodes grows exponentially due to the branching over actions and observations. This highlights the computational challenges associated with online belief tree expansion and motivates the need for scalable planning strategies.

Moreover, maintaining beliefs at each node incurs a significant memory cost. If beliefs are represented explicitly as probability distributions over the state space, each node stores a vector of size $|\mathcal{S}|$. To update a belief after an action-observation pair, the Bayes filter must compute a weighted sum over all current states and successor states, incurring a cost of $\mathcal{O}(|\mathcal{S}|^2)$ per node. Thus, the overall *memory complexity* for constructing a belief tree of depth D is:

$$\mathcal{O}\left(|\mathcal{A}|^D |\mathcal{Z}|^D \cdot |\mathcal{S}|^2\right) \tag{3.15}$$

This memory burden makes exact belief tree construction intractable for many real-world robotics problems, particularly those involving large state spaces or long planning horizons [116].

While practical planners often mitigate this cost through belief compression, sampling, or regularisation (as in DESPOT [117]), the exponential scaling remains a fundamental limitation of exact POMDP planning approaches [20].

This exponential complexity makes exhaustive policy computation infeasible for long-horizon or high-dimensional tasks. As a result, tractable planning requires either strong approximations (e.g., point-based updates, value function compression) or sample-based methods that simulate only a subset of reachable trajectories.

This motivates the use of approximate, sample-based techniques like Monte Carlo Tree Search (MCTS), discussed next in Sec. 3.4.5, which construct sparse approximations of the full belief tree through guided sampling rather than full enumeration.

3.4.5 Sample-Based Planning with Monte Carlo Tree Search (MCTS)

Monte Carlo Tree Search (MCTS) [118] incrementally builds a lookahead tree via stochastic sampling, balancing exploration and exploitation to approximate the optimal action policy. Each node represents a state (in MDPs) or a belief/history (in POMDPs), and each edge corresponds to a control *action*. In POMDP search trees (e.g., POMCP [23], DESPOT [24]), the structure alternates: *action edges* connect belief or history nodes to observation nodes, and *observation edges* connect observation nodes back to successor belief nodes. This alternation compactly encodes branching over both controls and sensor outcomes.

The MCTS procedure proceeds iteratively through four main phases:

- **Selection:** starting from the root, recursively select child edges that optimise an upper-confidence bound criterion (e.g., UCB1) until reaching a leaf or expandable node.
- **Expansion:** add a new child node by sampling an untried action (and, in POMDPs, branching on the resulting observation).
- **Simulation (Rollout):** from the new node, simulate a trajectory using a random or heuristic policy to estimate a cumulative reward.
- **Backpropagation:** propagate the sampled return along the visited path, updating visit counts $N(\cdot)$ and action-value estimates $\hat{Q}(\cdot)$.

By simulating many trajectories, MCTS approximates the expected return (*Q-value*) $Q(s, a)$ or $Q(b, a)$ for each action, defined as the expected cumulative reward obtained by executing action a and following the current policy thereafter. This allows the planner to identify high-value actions without exhaustively enumerating the full outcome space.

Central to the effectiveness of MCTS is the use of a *generative model* that specifies the stochastic dynamics of the environment. During rollouts, this model samples successor states and observations given the current state (or belief) and action. When the generative model faithfully encodes the underlying causal mechanisms of the environment, the resulting value estimates are asymptotically unbiased. However, if the model is *confounded* — for instance,

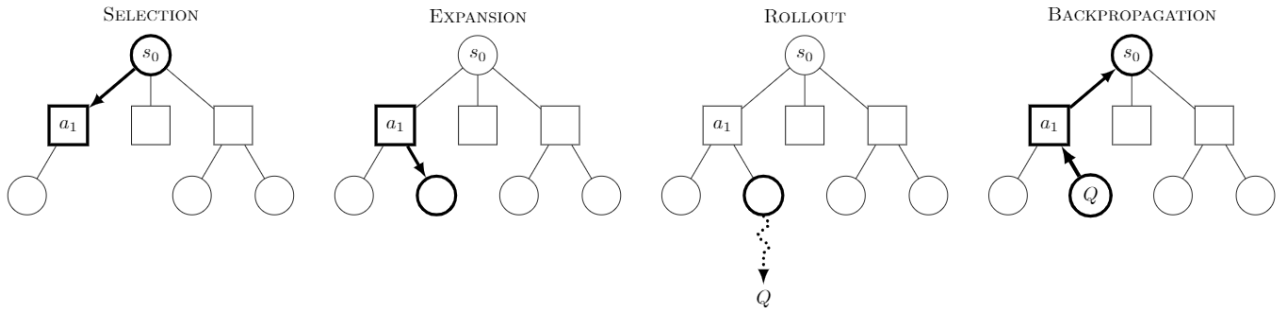


Figure 3.3: Illustration of the four phases of Monte Carlo Tree Search (MCTS): **Selection**, **Expansion**, **Simulation**, and **Backpropagation**. Each iteration expands the search tree through sampling, evaluates new nodes via rollouts, and propagates value estimates back to refine subsequent action selection. Source: adapted from [119].

when hidden variables simultaneously influence both the agent's chosen actions and their resulting observations — then the estimated transition function no longer represents genuine causal relations. This leads to systematically biased value estimates, often manifesting as over-optimistic or unsafe action choices.

These confounded dependencies are examined in depth in Ch. 4, where we introduce a causal reformulation of MCTS-based planning to restore unbiased value estimation through explicit modelling of causal structure and interventional sampling. In this thesis, MCTS serves as the computational foundation for the online planning framework underpinning the CAR-DESPOT algorithm, enabling sample-efficient, real-time decision-making under uncertainty while remaining amenable to causal adjustment.

3.4.6 POMDPs with Unobserved Confounding (UCPOMDPs)

While the classical POMDP formulation assumes that all relevant causal factors are either observable or explicitly modelled, real-world robotic systems often violate this assumption. Environmental variables that are unmeasured or only partially observable can simultaneously influence both the robot's chosen action and its resulting state transition, introducing *unobserved confounding*. This phenomenon is captured by the *Unobserved-Confounder POMDP* (UCPOMDP), which extends the standard POMDP to explicitly include latent variables U_k that affect both control and outcome dynamics:

$$A_{k+1} \leftarrow U_k \rightarrow S_{k+1}.$$

As illustrated in Fig. 3.4, this structure violates the conditional independence assumption that actions are chosen independently of the latent state given the current belief. Instead,

actions become conditionally dependent on unobserved factors, producing *reactive* rather than *deliberative* behaviour: the agent reacts reflexively to external influences rather than acting solely based on its belief state.

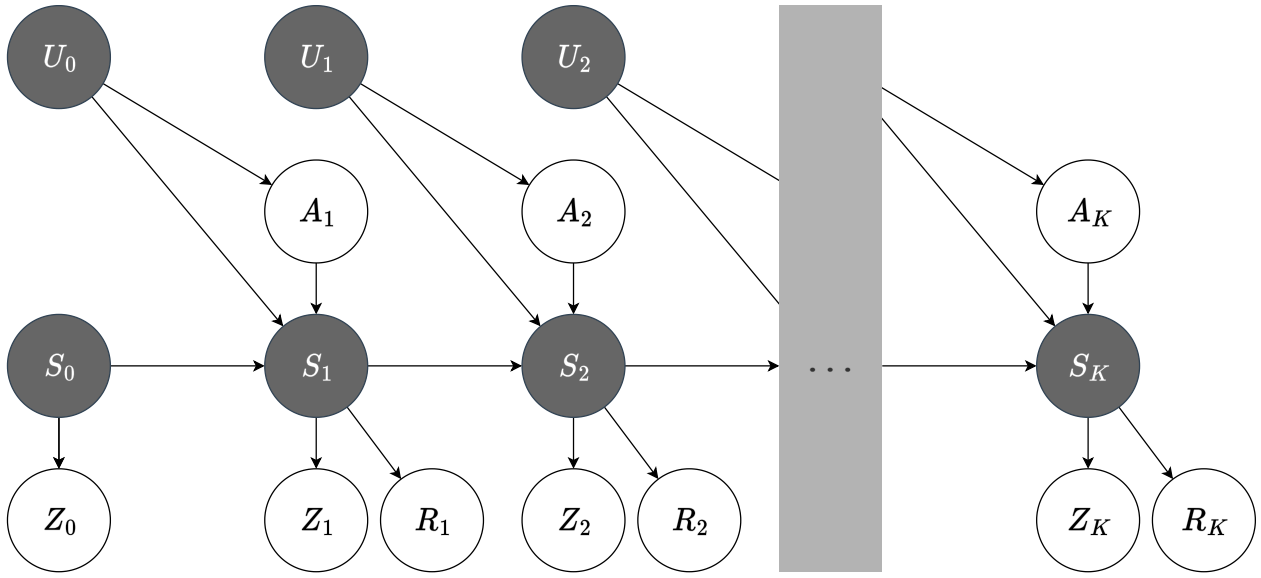


Figure 3.4: A K-step causal DAG of an Unobserved-Confounder POMDP (UCPOMDP). The agent's action A_{k+1} is conditionally dependent on the unobserved confounder U_k, which also influences the successor state S_{k+1}. This induces bias in the observational transition function $P(S' \mid A, S)$ by introducing a back-door path $A \leftarrow U \rightarrow S'$. Subscripts denote time steps.

In this setting, the observational transition model $P(S' \mid A, S)$ becomes confounded, as it entangles the true causal effect of the action ($A \rightarrow S'$) with spurious correlations mediated by U. As a result, the planner's value estimates can become systematically biased, leading to incorrect or unsafe policies. A causal planner must therefore reason over the *interventional* transition model $P(S' \mid do(A{=}a), S)$ rather than the purely observational one. This distinction is central to the causal reformulation of planning developed in Ch. 4, where the CAR-DESPOT algorithm explicitly models unobserved confounders to restore unbiased estimation of action consequences.

3.4.7 Causal & Classical Reinforcement Learning

Reinforcement Learning (RL) provides a general framework for learning optimal decision policies through repeated interaction with an environment. Formally, an agent seeks to maximise its expected cumulative reward by learning an optimal policy $\pi^*(s)$ that satisfies the Bellman equation over the true environment dynamics. In the classical case, the environment is treated as an unconfounded Markovian system, and learning proceeds from observed transitions (s_t, a_t, r_t, s_{t+1}). However, when unobserved confounders influence both actions and outcomes,

the estimated transition model and reward function may encode spurious correlations, leading to biased policy updates and suboptimal learned behaviour.

The Multi-Armed Bandit (MAB) problem. The Multi-Armed Bandit (MAB) problem [120] formalises sequential decision-making under uncertainty in its simplest form. An agent chooses from a set of K independent arms $\mathcal{A} = \{a_1, \ldots, a_K\}$, each associated with an unknown reward distribution $P(R \mid a_i)$. At each time step t, the agent selects an arm a_t and observes a reward $r_t \sim P(R \mid a_t)$. The goal is to maximise cumulative reward

$$\mathbb{E}\left[\sum_{t=1}^{T} R_t\right],$$

or equivalently, to minimise the expected *regret*

$$\mathcal{R}_T = T\,\mu^* - \sum_{t=1}^{T} \mu(a_t),$$

where $\mu^* = \max_i \mathbb{E}[R \mid a_i]$ is the optimal arm's expected reward. The MAB encapsulates the fundamental *exploration-exploitation* trade-off: whether to select the best-known arm (exploitation) or explore uncertain alternatives to improve future returns (exploration).

The Confounded Multi-Armed Bandit (MABUC) problem. In the causal reinforcement learning literature, Bareinboim, Forney and Pearl [27] introduced the *Multi-Armed Bandit problem under Unobserved Confounding (MABUC)*. In this extension of the classical MAB, a set of unobserved confounders U influence both the agent's choice of bandit arm A_t and the resulting reward R_t:

$$A_t \leftarrow U_t \rightarrow R_t.$$

Since these confounders are unobserved, their influence cannot be directly modelled or conditioned upon, violating the assumption of independent action selection. The observational reward distribution $P(R \mid A)$ becomes confounded, conflating the causal influence of the action A on reward R with the spurious correlation induced by U. This setting captures a realistic scenario in robotics, where environmental factors (e.g., terrain, lighting, or human presence) may simultaneously affect both the robot's decision process and the observed outcome. Because these confounders are not known *a priori* to the system modeller, they remain unaccounted for in classical RL updates, leading to biased value estimates.

A structural causal model (SCM) formulation [27] combines both observational and interventional quantities to avoid confounding bias by appropriately adjusting for unobserved dependencies, thereby enabling policy updates that achieve faster convergence and lower regret. *Markov Decision Processes with Unobserved Confounders (MDPUC)* extend this causal formulation to sequential decision problems with unobserved confounders [26], providing the basis for counterfactual reasoning in causal reinforcement learning.

Classical Reinforcement Learning. In the standard RL setting, value functions are estimated from experience tuples (s_t, a_t, r_t, s_{t+1}) collected during interaction with the environment:

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t,a_t) \,\middle|\, s_0{=}s\right].$$

Updates such as Q-learning and policy gradient methods assume that environment samples are i.i.d. draws from the true transition model $P(S' \mid S, A)$. This assumption fails in the presence of unobserved confounders that jointly affect both A and S', resulting in policies that optimise over biased value estimates and fail to generalise across contexts.

Causal Reinforcement Learning. *Causal reinforcement learning* generalises classical RL by explicitly modelling causal dependencies within the environment's transition dynamics [27, 121]. A causal RL agent reasons over interventional quantities $P(S' \mid do(A{=}a), S)$ rather than observational ones, thereby isolating the direct causal effect of actions on subsequent states. In addition to interventional reasoning, a causal agent may also evaluate *counterfactual quantities* — for example, estimating how an alternative action $A{=}a'$ would have altered the expected reward $R_{A=a'}$ under the same exogenous conditions. This capability enables agents to distinguish between correlation and causation, detect spurious dependencies, and reason hypothetically about unseen action outcomes.

By embedding causal structure into its world model, a causal RL agent can perform backdoor adjustment, conduct targeted interventions, or exploit instrumental variables to recover unbiased estimates of causal value functions. These techniques are particularly valuable in confounded environments, where purely data-driven approaches fail to separate environmental bias from genuine control influence.

Connection to Thesis. Causal reinforcement learning shares its underlying motivation with the causal extensions to probabilistic planning developed in this thesis. Both aim to ensure that an agent's decisions reflect genuine cause-effect relationships rather than correlations induced by latent factors. However, the focus here is distinct: whereas causal RL tackles the broader model-based learning problem — comprising both *(i)* learning transition and reward models from experience, and *(ii)* planning optimal action sequences — our work focuses exclusively on the latter. Online RL methods inherently require the agent to conduct exploratory experiments in the real world, which is often impractical or unsafe in mobile robotics domains such as autonomous driving, underwater exploration, or industrial inspection. In such settings, uncontrolled exploration may lead to irreversible failures, hardware damage, or unacceptable safety risks.

For these reasons, this thesis focuses on *causal planning* — that is, reasoning and decision-making over known or learned causal models without relying on active trial-and-error in the environment. This distinction motivates the causal extensions to online POMDP solvers (e.g., CAR-DESPOT in Ch. 4), which applies causal reasoning to achieve robust and interpretable performance in confounded robotic systems.

Summary. MDPs and POMDPs provide principled mathematical foundations for planning and decision-making under uncertainty, but their effectiveness relies on the assumption that the underlying transition and observation models are unconfounded. In real-world robotic domains, this assumption often fails: hidden variables may influence both the agent's actions and the resulting outcomes, introducing bias into learned or specified generative models. By embedding *causal semantics* into these frameworks — replacing observational transitions with interventional ones and enabling reasoning over counterfactual outcomes — we obtain planners whose value estimates reflect *causes*, not merely correlations. These causal extensions unify probabilistic planning, causal inference, and reinforcement learning, forming the foundation for the CAR-DESPOT planner (Ch. 4), COBRA-PPM decision-making framework (Ch. 5), and counterfactual explanation methods developed in Ch. 6.

3.5 Deep Generative World Models

Deep generative models complement decision-theoretic planners by learning implicit representations of how observations are generated from latent causes. Where planners simulate state trajectories, generative models synthesise observations directly from learned latent structure,

capturing both appearance and dynamics of complex worlds. This section reviews key model classes — variational autoencoders (VAEs), transformer architectures, and diffusion-based models — and shows how these models can be interpreted as implicit structural causal models (SCMs) for *counterfactual world generation*. These foundations underpin the counterfactual contrastive learning framework in Ch. 7 and the *Multiverse Mechanica* framework in Ch. 8.

3.5.1 Variational Autoencoders (VAEs)

A *variational autoencoder* (VAE) [122, 123] is an unsupervised deep generative model that learns to represent complex data distributions through a latent variable formulation. It consists of two neural networks: an **encoder** (or inference model) $q_\phi(z \mid x)$ that maps an input observation x to a latent representation z, and a **decoder** (or generative model) $p_\theta(x \mid z)$ that reconstructs x from z. Together, these networks approximate the underlying data-generating process

$$z \sim p(z), \quad x \sim p_\theta(x \mid z),$$

where the latent prior $p(z)$ is typically a unit Gaussian $\mathcal{N}(0, I)$.

During training, the encoder approximates the intractable posterior $p_\theta(z \mid x)$, while the decoder seeks to reconstruct the input from samples of z. Optimisation proceeds by maximising the *evidence lower bound* (ELBO):

$$\mathcal{L}_{\text{VAE}}(\theta, \phi; x) \;=\; \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)] \;-\; D_{\text{KL}}\Big(q_\phi(z \mid x) \,||\, p(z)\Big), \tag{3.16}$$

which combines two complementary objectives:

- A **reconstruction loss**, often implemented as a mean squared error (MSE) term, that encourages the decoder to reproduce input samples accurately from the latent representation.
- A **regularisation loss**, given by the Kullback-Leibler (KL) divergence D_{KL}, that penalises deviation of the learned posterior $q_\phi(z \mid x)$ from the unit normal prior $p(z)$, promoting smoothness and continuity in latent space.

The VAE thus learns a structured, continuous latent manifold where similar inputs map to nearby points and can be smoothly interpolated. Because the model is trained without supervision, it learns to capture the most salient factors of variation in the data distribution, enabling unsupervised discovery of semantically meaningful latent dimensions.

3.5.2 Transformer Architectures

Transformers [124] are deep neural network architectures based on the principle of *self-attention*, which allows each element in a sequence to attend to all others when computing contextualised representations. Given a sequence of token embeddings $\mathbf{X} = [x_1, \ldots, x_n]$, the transformer computes attention weights as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V,$$

where $Q = \mathbf{X}W_Q$, $K = \mathbf{X}W_K$, and $V = \mathbf{X}W_V$ are learned linear projections, and d_k is the key dimensionality. Stacking multiple layers of self-attention and feed-forward networks yields powerful context aggregation across spatial or temporal dimensions.

In diffusion models, transformers often replace or augment convolutional blocks within UNet backbones, allowing global interactions across spatial features (e.g., text-image cross-attention in latent diffusion). Transformers also form the foundation of multi-modal alignment models (e.g., CLIP [125]), which couple text and image embeddings — mechanisms leveraged later in this thesis to implement counterfactual conditioning (Ch. 7, Ch. 8).

3.5.3 Diffusion & Latent Diffusion Models

Denoising Diffusion & U-Net Architecture. Denoising diffusion probabilistic models (DDPMs) [126, 127] learn to reverse a gradual noising process that transforms clean data samples $x_0 \sim q(x_0)$ into isotropic Gaussian noise $x_T \sim \mathcal{N}(0, \mathbf{I})$ via a forward Markov chain:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\Big(x_t;\, \sqrt{\alpha_t}\, x_{t-1},\, (1-\alpha_t)\mathbf{I}\Big),$$

where $\{\alpha_t\}_{t=1}^T$ defines a variance schedule controlling the noise level at each step. The model then learns a denoising network $\epsilon_\theta(x_t, t)$ that predicts the injected noise at each timestep, enabling a reverse diffusion process that gradually reconstructs clean data from pure noise (see Fig. 3.5).

In practice, the denoising network ϵ_θ is typically implemented using a *U-Net* [128] backbone — an encoder-decoder architecture with symmetric skip connections that preserve spatial detail while capturing global context across multiple scales. The U-Net's hierarchical feature fusion enables consistent semantic reconstruction throughout the denoising trajectory, making it the standard backbone for modern diffusion and latent diffusion models.

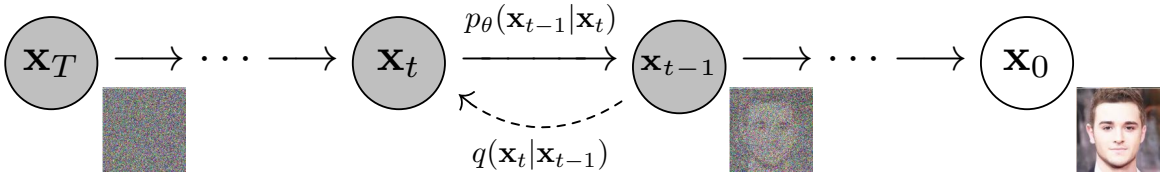


Figure 3.5: Illustration of the reverse denoising process in a diffusion model. Starting from the fully noisified image X_T, the model iteratively predicts and removes noise at each step to infer progressively cleaner latent variables X_{t-1}, ultimately reconstructing the noise-free sample X_0 after T reverse steps. Each transition is parametrised by the learned network $\epsilon_\theta(X_t, t)$, which estimates the noise component to remove. Adapted from Ho, Jain and Abbeel [126].

Reverse Diffusion as a Causal Process. At generation time, the model begins from $x_T \sim \mathcal{N}(0, I)$ and iteratively denoises toward x_0, optionally using deterministic samplers such as DDIM [129]. The initial noise u acts as an *exogenous variable*, while the denoising network parametrises the deterministic causal mechanism f_θ, producing $x = f_\theta(u, c)$. This decomposition aligns directly with the structure of an SCM $\langle \mathbf{U}, \mathbf{V}, \mathbf{F}, P(\mathbf{U}) \rangle$, enabling counterfactual world generation by holding u fixed and intervening on conditioning variables c.

Latent diffusion. Latent diffusion models [130] perform diffusion in a compressed latent space $z = E(x)$, applying the same denoising principle within a lower-dimensional manifold. The denoiser operates on z_t rather than raw pixels, reducing computational cost and improving semantic fidelity. Conditioning (e.g., text prompts) is injected via cross-attention layers that couple latent features with transformer-encoded embeddings, allowing precise control over generated content. This structure provides a unified substrate for causal interventions and counterfactual imagination, as used in Ch. 8.

3.5.4 Counterfactual & Contrastive Diffusion

From a causal standpoint, diffusion models instantiate the mapping $x = f_\theta(u, c)$, where u is exogenous noise and c represents manipulable causes or conditions. To generate counterfactuals — answering 'What would this image look like if we changed c_A to c_B while holding everything else constant?' — we apply the same *Abduction-Action-Prediction* procedure as in Sec. 3.2.1:

1. **Abduction:** Invert the diffusion process (e.g., via deterministic DDIM inversion) to infer the exogenous latent u_a corresponding to a factual image x given condition c_A.

2. **Action:** Intervene by changing the condition to c_B, holding u_a fixed.

3. **Prediction:** Generate the counterfactual $x_{\text{cf}} = f_\theta(u_a, c_B)$.

This process enforces causal consistency: only descendants of c should change between the factual and counterfactual images. Contrastive alignment losses (Ch. 7) further encourage this property by penalising deviations in shared latents between factual-counterfactual pairs.

3.5.5 Unconditional & Conditional Sampling

Diffusion models can be sampled in different modes depending on whether the generation process is guided by external conditioning information. These modes define how the model traverses the reverse diffusion trajectory from an initial random latent to a coherent image.

Unconditional Sampling. In an **unconditional latent diffusion model**, the generation process starts from an exogenous noise latent $u \sim \mathcal{N}(0, I)$ and denoises it through the learned reverse process to yield an image $x_0 = G_\theta(u)$. Since the model is trained without conditioning, samples are drawn directly from the learned data distribution $P(X)$, representing plausible but unconstrained outcomes from the training manifold.

Conditional Sampling. In contrast, a **conditional latent diffusion model** learns to sample from $P(X \mid C)$, where C represents auxiliary information such as text prompts, class labels, or scene attributes. During denoising, C is injected into the model via cross-attention layers that modulate intermediate activations with context embeddings — enabling fine-grained control over the generated content. Text-conditioned models such as Stable Diffusion [130] employ *classifier-free guidance* [131] to balance diversity and faithfulness by interpolating between conditional and unconditional denoising predictions at sampling time.

3.5.6 Image Editing & In-Fill Operations

Beyond unconditional or conditional generation from scratch, diffusion models can be applied to **image editing** and **in-filling** tasks, where only parts of an image are re-synthesised while others remain fixed. These operations constrain the diffusion process spatially or semantically, producing locally modified but globally consistent outputs.

In-Painting. **In-painting** applies a binary mask M to specify regions of an image that may be modified. During denoising, the model resamples pixels within M while freezing those outside, enforcing contextual consistency with the unmasked region:

$$x_t^{(M)} = M \odot x_t + (1-M) \odot x_{\text{fixed}},$$

where $\odot$ denotes elementwise multiplication and x_{fixed} represents the immutable region.

Limitations. While effective for simple edits, in-painting has two fundamental limitations:

1. **Locality Constraint.** The method can only modify pixels inside the mask. If a semantic change (e.g., altering an object's orientation or global lighting) requires widespread adjustments, the mask must become very large, encompassing nearly the whole image and diminishing control.

2. **Overconstraining Generation.** Conversely, small masks can over-restrict the model. For instance, changing a person's facial expression may require subtle pose adjustments of the head or shoulders; if those regions lie outside the mask, the result may appear physically inconsistent.

These constraints make in-painting insufficient for semantically rich or globally coherent edits. In this thesis, these limitations motivate the development of **counterfactual contrastive methods** (Chapters 7 and 8), which go beyond spatial masking by reasoning directly over shared exogenous noise and parallel-world interventions. By modelling these causal relationships explicitly, we achieve consistent, faithful transformations that align with the underlying world dynamics.

Summary. Deep generative world models unify probabilistic and causal reasoning by learning implicit mechanisms that map exogenous noise to observable worlds. Transformers and UNet architectures provide scalable computational substrates for learning these mechanisms, while diffusion processes instantiate them as reversible, causally interpretable mappings. By manipulating conditioning variables and shared noise, such models enable counterfactual imagination — forming the computational core of the causal contrastive learning and multiverse reasoning frameworks developed in Ch. 7 and Ch. 8.

4

Causally Informed Planning for Robots Under Partial Observability and Unobserved Confounding

Contents

4.1 Introduction 84
4.2 Planning in Environments with Confounded Decision-Making . . 86
4.2.1 Effect Estimation & Confounding 86
4.2.2 Sources and Implications of Unobserved Confounding in Real-World Robot Systems 87
4.2.3 Limitations of Probabilistic Planning in the Presence of Unobserved Confounding 94
4.2.4 Vulnerability to Confounding Bias in MCTS-Based Planners 94
4.2.5 Causal Limitations of POMDP-Based Robot Planning 95
4.3 Confounded GridWorld Problem 97
4.4 CAR-DESPOT: A Causal Approach to POMDP Model Learning & Planning for Robots in Confounded Environments 100
4.4.1 SCM Representation of POMDPs 101
4.4.2 Model Parameter Learning Method 104
4.4.3 CAR-DESPOT: A Causally-Informed MCTS-Based POMDP Planner 107
4.4.4 Robot System Integration 109
4.5 Experiments 111
4.5.1 Model Parameter Learning 111
4.5.2 Planner Evaluation 112
4.6 Results & Discussion 113
4.6.1 Analysis of learned model 113
4.6.2 Analysis of planning performance 115
4.7 Limitations & Future Work 118
4.8 Summary 120

4.1 Introduction

In this thesis, we explore how causal generative machine learning and AI can enable robots to reason about the joint dynamics that govern not only the evolution of the robot, task, and environment, but also the evolution of the *robot's own decision-making process*. To reason effectively about these often complex and interdependent dynamics, agents must consider both: 1) how their actions shape the potential future evolutions of the joint robot–environment state; and 2) how the environment itself may influence their own decision-making process. This latter capability of meta-cognition — i.e., thinking about how we think — is crucial for enabling robust and accurate prediction, action selection, and counterfactual explanation, free from *confounding bias* introduced by external factors.

The field of probabilistic planning has produced mature methods to address the problem of sequential decision-making under uncertainty. A key focus of the research has typically been on how to effectively model domain knowledge and sources of uncertainty, while maintaining efficient and computationally tractable search times [23, 117, 132]. While these approaches have proven successful in a variety of robotic applications, they typically operate on *purely statistical* model representations and lack an explicit representation of the causal structure underlying the system dynamics. Further, they typically assume that (conditioned on world observations) the agent's choice of action at plan time is completely independent of external influence — i.e., their transition probability estimates are not artificially altered by a common cause of both the agent's action choice and its outcome.

However, this assumption is frequently violated in real-world robotic deployments due to the high level of coupling between the environment and the robot's sensing, decision-making, and actuation — where unobserved confounding between action selection and outcomes is often present. As a result, standard approaches such as MDPs and POMDPs lack the causal semantics required to articulate system relationships in a truly causal manner, precluding the application of causal analysis. This prevents the use of causal analysis for action–outcome prediction during planning; instead, only correlations between actions and predicted outcomes

can be considered. This may lead to prediction errors due to confounding bias, and in turn, the generation of sub-optimal policies when better ones should otherwise be discovered [26, 27].

In this chapter, we investigate the impacts of unobserved environmental confounders that affect robot decision-making in real-world settings. We address the question of how causal generative machine learning models can be used to formally encode the bidirectional influences between the robot's decision-making process — and thus its chosen actions — and the environment (**Q1 - Modelling**). To this end, we construct an SCM-based causal model of the robot decision-making process under unobserved confounding and realise it in a PPL-based implementation. To address the additional challenge that the system dynamics may not be fully known *a priori*, we propose a method to learn a partial parameterisation of the model from data in an offline manner (**Q2 - Structure and Parametrisation**).

Using this explicitly causal model formulation, we further investigate the benefits of using *level 2* intervention-based causal treatment to address confounding bias errors. These occur when decision-making is influenced by unobserved confounders in the environment, leading to biased predictions of action outcomes (**Q3 - Confounder Bias**). To mitigate this, we apply the mathematical formalism of interventions to surgically remove conditional dependencies on the confounder in the underlying causal graph, thereby eliminating spurious influences from unobserved confounders on robot action selection. We construct an intervention-based transition function using the $do(\cdot)$ operator, allowing us to predict action outcome probabilities that are free from confounding bias.

Furthermore, in this chapter we explore how the developed causal model of confounded robot decision-making and the intervention-based causal treatment can be embedded into traditional sample-based probabilistic planning approaches. This integration prevents confounding bias in outcome probability estimates from propagating into action and state value estimation errors. These errors would otherwise lead to sub-optimal policies, and potentially incorrect, unpredictable, or unsafe robot behaviour (**Q4 - Decision Making**). To this end, we propose CAR-DESPOT, a novel causally–informed extension of the efficient anytime regularized determinized sparse partially observable tree (AR-DESPOT) online planner [24, 117], which employs an SCM-based POMDP representation and causal inference to eliminate policy bias caused by unmeasured confounders.

Finally, in support of **Q4 - Decision Making**, we experimentally evaluate our learning and planning methods on a toy robot navigation problem with an unobserved confounder, to

demonstrate the benefits of integrating causal modelling and inference with POMDP planning. Although this thesis is motivated by real-world robotics applications, we begin with a minimal toy problem that is representative of an autonomous inspection mission, serving as an initial proof of concept.

4.2 Planning in Environments with Confounded Decision-Making

4.2.1 Effect Estimation & Confounding

To realise reliable and trustworthy autonomous robots in the real world, it is critical to equip them with decision-making capabilities that are robust not only to typical sources of uncertainty such as partial observability and stochastic dynamics, but also to more subtle statistical challenges such as *unobserved confounding*. In robot planning, unobserved confounding arises when hidden variables simultaneously influence both the agent's action selection and the resulting outcomes, thereby biasing outcome predictions and degrading planning performance.

When the confounding variable is observed, standard causal adjustment techniques can be used to isolate the direct effect of an action on future outcomes, free from confounding bias. These techniques, along with the formal definition of confounding and the assumptions required for valid effect estimation, are discussed in detail in Sec. 3.1.1. Readers unfamiliar with these foundational concepts are encouraged to consult that section before proceeding.

While these methods are applicable when confounders are observable, planning becomes considerably more difficult when the confounding variable is unobserved. In such cases, additional assumptions must be enforced, such as the existence of an *instrumental variable* or a *mediator variable* in the causal graph [32, Ch. 7, 11]. These techniques rely on two key conditions. First, the relevant variable must exist in the underlying causal structure. Second, it must be observable to the agent or modeller. In practice, these conditions are often unmet in real-world robotics applications. If left unaddressed, unobserved confounding can lead to sub-optimal or unsafe behaviours, particularly in high-stakes or safety-critical scenarios.

Planning must therefore be robust to the range of uncertainty sources typically encountered in real-world environments, including stochastic action outcomes and partial or noisy sensor observations [113]. Many non-trivial aspects — such as combinatorial state-action dimensionality, the need to retain long action-observation histories, and the requirement to consider

long-term horizon rewards — make this task especially challenging for robots operating in complex and dynamic real-world environments.

In this chapter, we synthesise the problem of *confounding* in agent-based robot planning and decision-making. We begin by introducing the foundations of *effect estimation* and how unobserved confounding undermines the reliability of action-outcome predictions. We then examine how these confounding effects arise in real-world robotic systems due to sensor limitations, environmental factors, and dynamic subsystem interactions. With this foundation, we review classical approaches to *probabilistic planning under uncertainty*, and how methods such as Monte Carlo Tree Search (MCTS) fail to account for confounding bias. We identify a critical limitation of standard MDP-based planning algorithms: their inability to adjust for unobserved confounders, which leads to corrupted transition predictions. This limitation motivates the causal planning methods developed in this chapter.

4.2.2 Sources and Implications of Unobserved Confounding in Real-World Robot Systems

The limitation of sample-based planners to adjust for unobserved confounding bias has significant implications for robots operating in real-world environments. Unobserved confounding effects may arise from sources within the environment that are not directly observable to the robot but still have an influence on decision-making and action outcomes. These are latent entities or dynamics in the environment that influence both how the robot selects actions and the resulting outcomes — thus inducing a *spurious correlation* [32, Ch. 2]. This arises via an unobserved common cause, forming a *back-door path* [32, Ch. 3] in the underlying causal graph, which allows this spurious correlation to flow through.

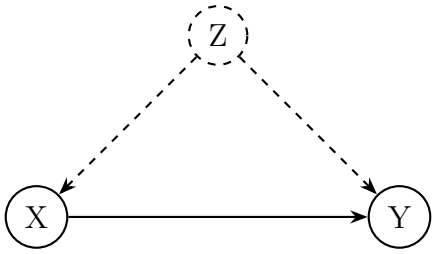


Figure 4.1: Causal DAG illustrating confounding: the back-door path $X \leftarrow Z \rightarrow Y$ (in dashed) introduces a spurious correlation between X and Y.

4.2.2.1 Confounding Arising From Sub-System Interactions

Unobserved confounding in robot decision-making often arise unintentionally due to their modular design. Robot sub-systems are typically developed to each fulfil a particular functionality based on the particular requirements of the robot, its task, and the operating environment. For mobile robots, sub-systems typically include: task-level planning (e.g., executive control), navigation and localisation, estimation and state tracking, geo-fencing and mission boundaries, and safety-critical functions like collision avoidance. Specialist robots may also require: simultaneous localisation and mapping, manipulation and grasping, human-robot interaction modules, and semantic or topological mapping.

These sub-systems often have functional and temporal dependencies on each other: they must communicate vital information (e.g., sensor data, path plans, manipulation actions) between each other; the operation of one may impact the future state of another (e.g., an activation of the collision avoidance sub-system may cause a re-plan, which in turn will change the trajectory and control signals of the robot); and must be collectively coordinated to achieve the desired emergent robot behaviour. In addition to these functional dependencies, interactions between sub-systems also involve *temporal latching*: one module may act on a snapshot of information that is only valid for a short time window, while downstream modules continue operating on that latched value until the next update arrives. This is analogous to a *D-latch* in digital systems, where an input is sampled and held stable long enough for subsequent processing, and is often necessary in robotics to maintain a temporally consistent view of the system across planning, control, and estimation modules.

Importantly, the causal DAG used for modelling robot decision-making is itself a *temporal abstraction*, in which variables such as the system state and agent action, denoted by S_k, A_{k+1}, and S_{k+1}, represent events occurring at discretised time steps. In practice, however, observations, actions, and state transitions are generated asynchronously and may correspond to different underlying timestamps. The validity of this causal abstraction therefore depends on the assumption that these quantities are temporally aligned — an assumption enforced through mechanisms such as latching, buffering, or synchronisation across sub-systems. In this sense, temporal synchronisation is not merely an implementation detail, but a modelling assumption required for the correctness of causal effect estimation.

Further, each sub-system may have its own defined safety behaviours that trigger when operational limits are exceeded — further increasing the combinatoric number of interactions that may arise. When temporal alignment assumptions are violated due to communication delays, asynchronous updates, or stale messages, the robot may act on inconsistent or misaligned state information. From a causal perspective, this introduces hidden temporal dependencies that are not explicitly represented in the DAG, creating additional pathways through which unobserved influences can affect both action selection and outcomes, and thereby contribute to confounding bias in downstream inference. Often, this large number of complex, sometimes unintended, interactions cannot all be practically anticipated and guarded against by system designers.

4.2.2.2 Robot Sub-System Conflicts

While they are designed to synergise, some robot sub-systems are inherently at odds with each other by design. For example, a path planner operating on a static 2D occupancy grid may generate a path that satisfies collision constraints, but when the robot executes the path, it encounters a previously unseen obstacle. As it approaches the obstacle, it breaches the collision avoidance threshold distance, triggering the collision avoidance module to take over control and move the robot away. Subsequently, control switches back to the path controller, which resumes the original path sequence — because the static map is not updated with the new obstacle. This causes the robot to breach the threshold again, triggering the avoidance behaviour once more, resulting in a behavioural deadlock that persists until external intervention occurs.

4.2.2.3 Sources of Environmental & Contextual Confounders in Robotics

Unobserved confounding in robotics can also arise from external influences that are not captured by the robot's onboard sensing and internal state estimation.

For example, in aerial robotics, unmeasured wind gusts can perturb flight dynamics, affecting both the chosen control inputs and the observed trajectory outcomes. In ground robots, lighting conditions can alter the performance of vision-based perception algorithms, which in turn influence the output of downstream modules such as object detectors or semantic mappers — resulting in indirect confounding of navigation decisions.

Additional sources include surface conditions (e.g., wet or uneven terrain affecting both wheel slip and obstacle classification), ambient electromagnetic interference (e.g., degrading compass or GPS accuracy, while also affecting localisation confidence), or the presence of

adversarial agents who manipulate observable features (e.g., presenting distractors or decoys) to cause the robot to misclassify goals or environmental features.

In each case, the unmeasured factor acts as a latent variable that simultaneously influences both the robot's decisions and the outcomes of those decisions — thereby inducing a spurious correlation that cannot be resolved by standard sample-based estimators without causal adjustments.

4.2.2.4 Adversarial Confounding

Building on the earlier obstacle scenario, we now consider the case where the new obstacle is an adversarial agent with knowledge of the robot's intended goal and collision avoidance behaviour, and whose objective is to prevent the robot from reaching its destination. In this scenario, the adversary could manoeuvre strategically to repeatedly activate the robot's avoidance behaviour, forcing it to retreat incrementally and effectively herding it away from the goal.

In this case, the adversary's actions influence both the robot's observations (via induced obstacle detections) and the outcomes of its actions (by preventing goal success), yet the robot does not observe the adversary's intentions directly. This constitutes a classic case of unobserved confounding: the adversary's latent strategy simultaneously influences both the robot's action selection and task success, thereby invalidating the assumptions underlying naive statistical associations.

4.2.2.5 Consequences For Robot Decision-Making

Consequently, the complex joint dynamics that arise from interactions between robot subsystems and their environments can introduce unobserved confounders into the decision-making process. This results in a *spurious correlation* between task state and action-selection variables in the causal DAG (Fig. 4.2), induced by a *back-door path* that biases the estimation of action–successor–state effects. Accurately estimating these effects is critical for predicting state-action transition probabilities, which underpins the performance of sample-based planning algorithms.

4.2.2.6 Practical Limitations of Real-World Robots

Finally, a tempting response to the challenge of unobserved confounding might be to simply eliminate the 'unobserved' part of the problem by turning every potential confounder into an

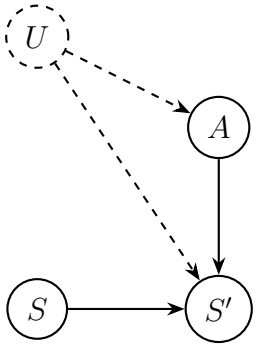


Figure 4.2: Causal DAG slice illustrating unobserved confounding in robot decision-making: the unobserved variable U influences both the selected action A and the resulting successor state S', inducing a spurious correlation along the dashed back-door path $A \leftarrow U \rightarrow S'$. This biases the effect estimation process unless causally adjusted.

observed variable by equipping robots with additional sensors to measure all relevant aspects of the environment and task state.

However, this is rarely feasible in practice. Mobile robots operate under strict size, weight, and power (SWaP) constraints, which limit the number, type, and quality of sensors that can be deployed. Adding sensors to monitor additional physical modalities — such as barometric pressure, humidity, vibration, or radiation — incurs not just hardware costs, but also increases the computational load for data processing, memory usage, and network bandwidth.

These limitations are especially pronounced in autonomous systems that must operate in real-time, in communication-denied environments, or under energy constraints (e.g., aerial drones or planetary rovers). As a result, it is not practically or economically viable to exhaustively sense all potentially relevant environmental and internal variables across all operating contexts.

This motivates the need for causal approaches that can reason about and adjust for unobserved confounding, even when such variables cannot be directly measured.

4.2.2.7 Mitigating Unobserved Confounding in Decision-Making with Generative Causal Models

In sum, complex interactions between robot sub-systems — such as safety overrides, reactive behaviours, and module-level feedback — can introduce latent dependencies that evolve over time and influence both decisions and outcomes. These effects are often difficult to anticipate or mitigate through system design alone. At the same time, real-world robots operate under strict sensing and computational constraints, making it infeasible to observe all potential confounders across diverse environments.

Together, these challenges undermine the effectiveness of standard statistical estimators, which rely on assumptions of no unobserved confounding, and limit their applicability in sampling-based planning algorithms. While classical causal adjustment strategies (e.g., back-door, front-door, or instrumental variable methods) require specific observable structures, such conditions are rarely met in robotics domains with partial observability and tight SWaP constraints.

The key insight for mitigating unobserved confounding in robot planning is that, if the robot has access to a generative causal model of its environment, it is possible to simulate interventions on action variables $\Big($i.e., sampling from $do(A = a)\Big)$ to block spurious dependencies introduced by unobserved confounders in the assumed causal graph.

As illustrated in Fig. 4.3, this breaks the back-door path $A \leftarrow U \rightarrow S'$ between the action and successor state. Without intervention, this path introduces a *spurious correlation* into the conditional probability $P(S' \mid S, A)$, confounding the true causal effect of A on S'. By performing an intervention, the influence of the unobserved confounder U is blocked, allowing unbiased estimation of the transition dynamics: $T_{\text{Intervention}}(s, a, s') = P(S' = s' \mid S = s, do(A = a))$.

This insight draws inspiration from work on confounded bandits [27], and forms the basis for our proposed planner, CAR-DESPOT, presented in Sec. 4.4, which applies this causal reasoning principle to the setting of online POMDP planning for real-world robots by performing causal sampling via simulated interventions to estimate action effects.

Model Dependence of Causal Adjustment. It is important to note that this notion of 'blocking' confounding is defined with respect to the assumed causal model. In practice, the true causal structure of a robotic system may be only partially known or approximated, and the assumed causal model may not fully reflect the underlying system dynamics. As such, intervention removes the influence of confounding pathways within the modelled causal graph, but its effectiveness depends on the fidelity of that model. If the causal structure is misspecified or incomplete, residual dependencies may persist, and the resulting estimates may still exhibit bias.

Intervention Under Structural Uncertainty. The intervention illustrated in Fig. 4.3 assumes that the underlying causal structure is correctly specified, in particular that the back-door path $A \leftarrow U \rightarrow S'$ is present and must be blocked. In practice, however, the causal structure of a robotic system may be only partially known or inferred from noisy and

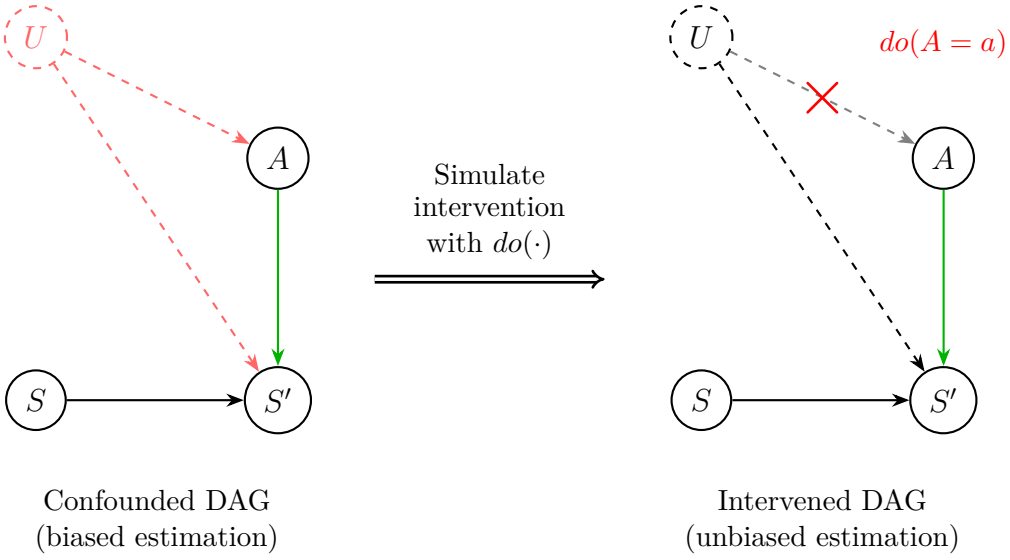


Figure 4.3: Side-by-side causal DAGs illustrating the role of intervention in breaking the *back-door path* from action A to transition outcome state S' through the unobserved confounder variable U: $A \leftarrow U \rightarrow S'$. Without causal treatment, this back-door path (shown as red dashed edges) introduces a spurious correlation into the conditional probability estimate of the direct causal effect of A on S' via the direct path $A \rightarrow S'$ (shown in green). This spurious correlation cannot be eliminated using statistical adjustment alone, as the confounder U is unobserved, and leads to *confounding bias* in the estimated transition probabilities when conditioning on the observed action a: $T(s, a, s') = P(S' = s' \mid S = s, A = a)$. Applying the intervention $do(A = a)$ blocks the influence of the unobserved confounder U on action selection, enabling unbiased estimation of the causal effect of actions on transitions: $T_{\text{Intervention}}(s, a, s') = P(S' = s' \mid S = s, do(A = a))$.

limited data, leading to uncertainty about the presence and form of such dependencies. This can be formalised as maintaining a *belief over possible causal graphs*, reflecting uncertainty in the underlying structure.

As a result, the choice of which variables to intervene on — and which dependencies to treat as confounding — is itself made under structural uncertainty. This has important implications for decision-making: incorrect assumptions about the causal structure may lead to inappropriate interventions, resulting in biased or sub-optimal estimates of action effects. More generally, this motivates approaches that incorporate prior knowledge or reason over a distribution of plausible causal structures when designing interventions. In this chapter — and throughout the thesis — we adopt the simplifying assumption that the causal structure is known, treating it as an inductive bias informed by domain knowledge, which enables tractable and interpretable causal reasoning in complex robotic systems.

4.2.3 Limitations of Probabilistic Planning in the Presence of Unobserved Confounding

While probabilistic planning techniques such as MDPs and POMDPs are powerful tools for decision-making under uncertainty, they assume that the environment dynamics can be captured accurately using observed data. These models are inherently *associational* in nature: they estimate transition and observation distributions based on statistical correlations in the data, without requiring an explicit representation of underlying causal mechanisms.

This poses a fundamental limitation when unobserved confounding is present. In such cases, statistical dependencies in the data may reflect spurious associations introduced by latent variables (see Sec. 3.1.1), rather than true causal effects. As a result, the estimated transition model $\hat{T}(s' \mid s, a)$ may produce systematically biased predictions, leading to degraded or unsafe policy performance.

This limitation is particularly problematic for sample-based planners, which rely on repeated simulations from the transition model to estimate the value of actions. If these simulations are themselves biased due to confounding, then all downstream value estimates will be corrupted. This undermines the ability of the planner to make sound decisions, even if its search and optimisation algorithms are otherwise functioning correctly.

These challenges are especially acute for sample-based planning algorithms such as Monte Carlo Tree Search (MCTS), which we examine next.

4.2.4 Vulnerability to Confounding Bias in MCTS-Based Planners

In realistic robotics domains with large state and action spaces, exact planning is computationally infeasible. Sample-based planners such as *Monte Carlo Tree Search (MCTS)* offer a practical alternative by approximating action values through simulated experience, rather than exhaustively computing them. MCTS has been widely adopted in robot planning under uncertainty due to its scalability, flexibility, and effectiveness in long-horizon decision-making tasks.

The core idea of MCTS is to construct a partial lookahead search tree, rooted at the agent's current belief or state, by iteratively sampling possible future trajectories. These simulations are performed using a generative model that encodes the environment's transition dynamics. By aggregating results from these sampled rollouts, the planner estimates the *Q-value* of actions and selects the one with the highest expected return. This enables the agent to reason over

long-term consequences while avoiding the full combinatorial explosion of the outcome space (see Sec. 3.4.5 for a detailed description of MCTS).

However, a critical limitation arises when the generative model is learned from observational data that contains unobserved confounding. Because MCTS rollouts rely on this model to simulate future transitions, any bias in transition probabilities will be systematically propagated through the tree. This can lead to incorrect value estimates and sub-optimal action selection, even when the search algorithm itself is functioning correctly. This failure mode is illustrated in Fig. 4.4, which shows how a latent confounder distorts transition predictions, corrupting value estimates during tree search and ultimately leading the agent to select unsafe or sub-optimal actions.

In the context of confounded planning, this vulnerability becomes particularly acute. Unlike model-free reinforcement learning methods, which may partially adapt to confounding through direct interaction and reward feedback, MCTS-based planning relies entirely on the generative model for decision-making. As a result, confounding in the model induces structural bias in the simulated rollouts, corrupting the very foundation of the planner's reasoning. This issue is examined in more detail in the next section.

4.2.5 Causal Limitations of POMDP-Based Robot Planning

Partially Observable Markov Decision Processes (POMDPs) provide a principled framework for planning under uncertainty in robotics, especially in domains where state information is incomplete or noisy. A formal definition and discussion of POMDP components, belief updates, and policy valuation can be found in Secs. 3.4.3.

In robotic settings, POMDPs enable agents to maintain beliefs over hidden world states and select actions that optimise expected long-term reward. However, due to the exponential complexity of belief tree construction (see Sec. 3.4.4), solving POMDPs exactly is infeasible for most real-world domains.

To address this, two main families of planning methods are commonly used: *offline* and *online* planners [24]. Offline planners (e.g., *Heuristic Search Value Iteration* (HSVI) [133], SARSOP [22]) attempt to precompute a near-optimal policy for all reachable beliefs before deployment, while online planners (e.g., POMCP [23], DESPOT [24]) build a partial policy tree during execution to find the next best action.

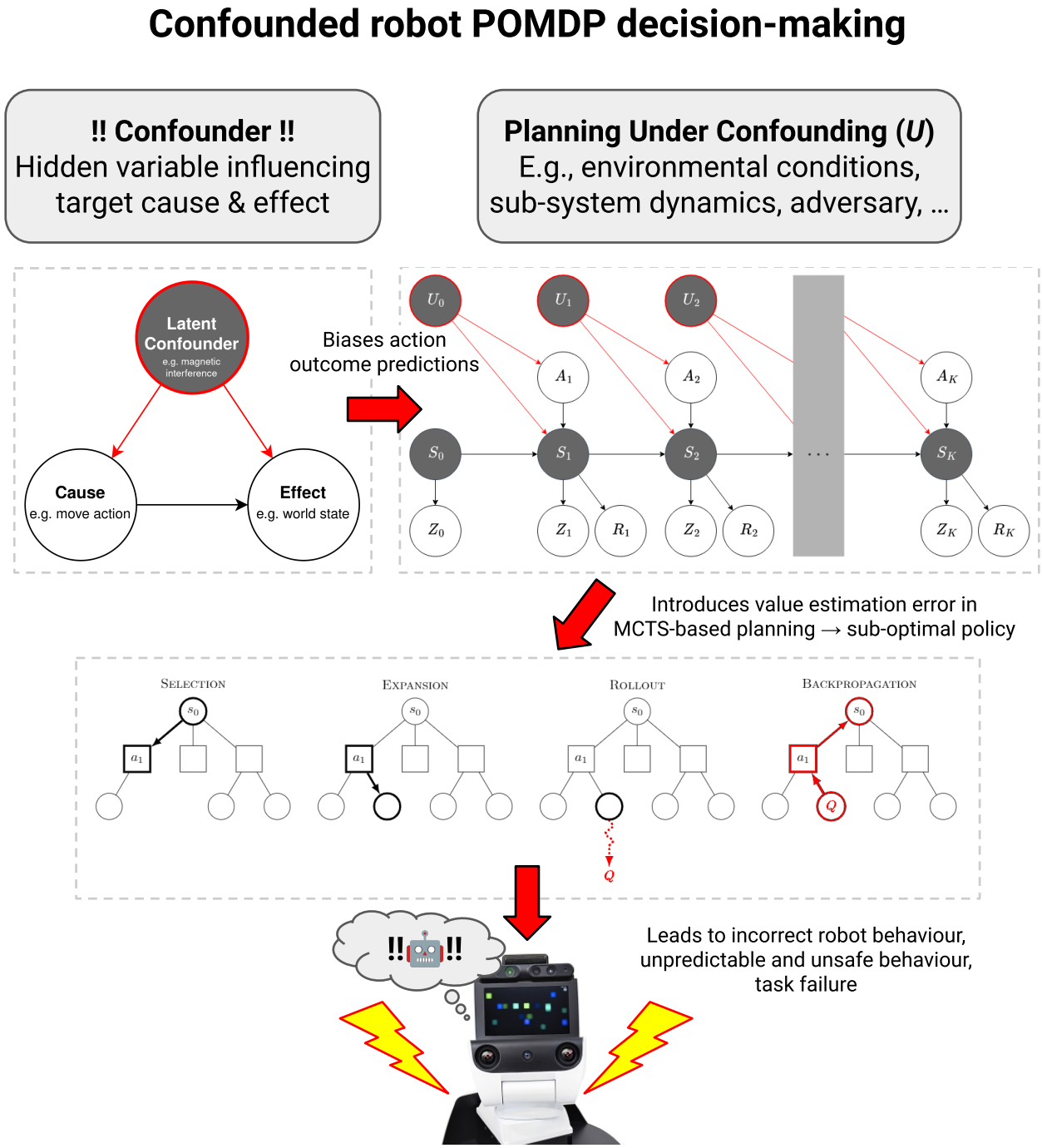


Figure 4.4: The presence of an unobserved (i.e., latent) confounder in the agent's decision-making process induces bias in the predicted transition probabilities used by sampling-based planners. This confounding bias propagates through the planner, introducing value estimation errors that may yield sub-optimal policies — ultimately resulting in unpredictable, unsafe behaviour and task failure. Standard probabilistic planning models such as MDPs and POMDPs lack explicit causal semantics, making them ill-equipped to represent and reason about such confounding influences. Consequently, a causal formulation is needed to enable principled analysis and correction of these biases. Source: created by author, includes components adapted and modified from [119].

These online methods have demonstrated strong empirical performance in domains such as navigation, object manipulation, and human-robot interaction [113]. For this reason, online POMDP planners form the foundation of our method described in Sec. 4.4.

Despite their success, a critical limitation remains: most existing POMDP planners assume an accurate, unbiased transition model, but do not provide mechanisms for causal reasoning or confounding adjustment. As discussed in Sec. 4.2.4, this leaves them vulnerable to hidden confounding, where unobserved variables bias the transition model used during planning.

Without causal semantics, POMDP planners reason over observed statistical dependencies, which may not reflect the true causal effects of actions. In the presence of unobserved confounders, these dependencies can be systematically biased, leading to flawed value estimates and sub-optimal policy choices [26].

To address this gap, in Sec. 4.4 we introduce **CAR-DESPOT**, a causal extension to the AR-DESPOT online planner, which performs interventional reasoning over a structured causal model to mitigate the effects of confounding during planning.

To illustrate these challenges, we introduce a toy example — the **Confounded GridWorld Problem** — which demonstrates the impact of unobserved confounding on robot planning and motivates key design choices in CAR-DESPOT.

4.3 Confounded GridWorld Problem

Problem Description. The *Confounded GridWorld* problem (Fig. 4.5) is a variant of the classic GridWorld environment [134], augmented with an identified unobserved confounder. In the original GridWorld, an agent navigates a 2–dimensional grid to reach a goal while avoiding collisions with occupied cells and grid boundaries. The *Frozen Lake* problem, originally part of the OpenAI Gym library [135] and now maintained in Gymnasium [136], extends GridWorld by introducing action stochasticity caused by a slippery ice-covered surface.[1]

In the adjacent causal reinforcement learning literature, Bareinboim et al.'s *Multi-Arm Bandit problem when unobserved confounders are present* (MABUC) [27] is an extension of the classical Multi-Arm Bandit problem, in which a set of unobserved confounders influence both the agent's choice of bandit arm (i.e., action) and the bandit payout (i.e., outcome).

[1] `https://gymnasium.farama.org/environments/toy_text/frozen_lake/`
Gymnasium is the official successor to OpenAI Gym and continues to maintain its environments under the Farama Foundation.

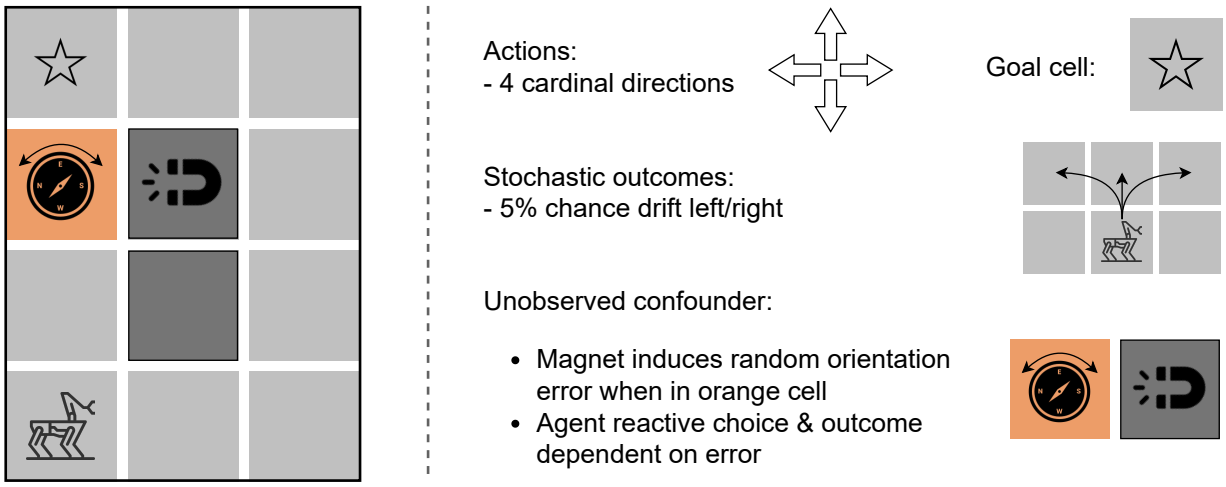


Figure 4.5: The *Confounded GridWorld* toy problem. A robot moves in the four cardinal directions in a grid world toward a goal (star), while avoiding occupied cells and borders. Variable sensor interference from an electromagnet embedded in the partitioning wall acts as an unobserved confounder when the robot is in its area of influence (orange cell). Grid cells are referenced by their (x, y) coordinates, with the origin at the bottom-left corner $(0, 0)$.

Since these confounders are unobservable, they are not directly considered in the decision-making analysis — the authors arguing the existence of which are due to them not being known *apriori* by the system modeller.

Our *Confounded GridWorld* problem combines elements from both Frozen Lake and confounded Multi-Arm Bandit problem to ground the toy problem in the autonomous robotics domain. The agent, a legged robot, is located on a $N \times M$ 2-dimensional grid and its aim is to navigate to a goal location to inspect an important asset as part of an industrial site inspection mission, while avoiding a collision with occupied cells.

State, Action Space, Observations. As per GridWorld and Frozen Lake, the robot always begins at the bottom-left grid cell $(0, 0)$ and knows the position of the goal, which is always $(0, 3)$. A column of occupied cells at $(1, 1)$ and $(1, 2)$ partitions the grid into two corridors. The state space of the problem is simply the robot's location in the $N \times M$ 2D grid: (x, y).

At each time step, the robot may move in one of the four cardinal directions: $\{$*RIGHT, UP, LEFT, DOWN*$\}$. However, due to imperfect actuation and estimation, actions are stochastic. The robot may drift one cell to the left or right (relative to the intended direction) with 5% probability each. For example, when executing the *RIGHT* action from $(0, 0)$, the robot ends up in:

- $(1, 0)$ with 90% probability,

- $(1, 1)$ with 5% probability,
- $(1, -1)$ with 5% probability.

The robot receives a noiseless observation of its location after each action. However, it never observes the orientation error induced by the electromagnet confounder, as discussed in the following section.

Unobserved Confounding. Central to this problem is the presence of an unobserved confounder, which is precluded from being explicitly considered in the analysis, as per the confounded Multi-Arm Bandit problem. A stationary electromagnet is located at cell $(1, 2)$ and induces sensor interference in the adjacent cell $(0, 2)$. When the robot occupies this cell, the magnet directly affects its orientation sensor measurements and action selection. This interference varies over time and is not measurable by the robot due to limited sensing capabilities.

The magnet thus constitutes an **unobserved confounder**, influencing both the agent's internal decision process and the outcome of its actions — a defining characteristic of unobserved confounding in decision-making. In our formulation, it is the latent variable U in the SCM-UCPOMDP model (see Fig. 3.4).

At each time step, the confounder $U_{OrientationError}$ is sampled from an independent categorical distribution over orientation offsets $\{-90°, 0°, +90°\}$ with probabilities $[0.10, 0.80, 0.10]$.

The unobserved confounder affects the robot in two distinct ways:

1. **Action Selection.** When the robot is in the magnet's influence region at $(0, 2)$, it selects actions probabilistically according to Table 4.1, conditioned on the current orientation error. These probabilities encode *reactive control behaviour*, in which the robot responds reflexively to environmental stimuli rather than deliberating over long-term outcomes (see Sec. 4.2.2). They are chosen here to illustrate the effects of confounding.

2. **State Transition.** The executed action is perturbed by the sampled orientation error. For instance, if the robot intends to move *RIGHT* (aligned with 0°), but $U_{\text{OrientationError}} = +90°$, then it instead moves *UP* (aligned with +90°). This rotation is applied before the drift dynamics.

When outside the magnet's influence, only the drift model applies.

Table 4.1: $P(A|U_{OrientationError})$: The robot's action selection distribution while under the influence of the confounder, conditioned on the orientation error.

$P(A\|U_{OrientationError})$	$-90°$	$0°$	$+90°$
RIGHT	0.05	0.45	0.05
UP	0.85	0.05	0.85
LEFT	0.05	0.45	0.05
DOWN	0.05	0.05	0.05

Reward & Terminal States. Each movement action incurs a cost of -1 to penalise inefficient paths. Reaching the goal yields a reward of $+100$ and terminates the episode. Entering an occupied cell results in a collision, a reward of -50, and episode termination. Otherwise, the scenario continues.

Application to Robotics. Although abstract, this scenario illustrates how unobserved confounders can arise in real-world robotic deployments (see Sec. 4.2.2). Sources such as unmodelled electromagnetic fields or latent hardware faults can influence both perception and control in ways that are not directly observable. If such influences are not properly accounted for in the system model, they can lead to confounded decision-making and degraded autonomy.

4.4 CAR-DESPOT: A Causal Approach to POMDP Model Learning & Planning for Robots in Confounded Environments

We present *CAR-DESPOT*, a novel causally-informed extension of the 'anytime regularized determinized sparse partially observable tree' (AR-DESPOT) planner [117], a state-of-the-art anytime online POMDP solver. CAR-DESPOT augments this framework with causal modelling and inference capabilities to eliminate errors caused by unmeasured confounder variables.

For the first time, our approach combines structural causal model (SCM) representations, interventional causal inference, probabilistic programming, and online POMDP planning to enable generalisable and extensible sequential decision-making for robots operating under stochastic, partially observable, and confounded environments.

Our method provides the robot with a *causally disentangled representation* of the transition function and an intervention-based inference mechanism. This enables it to separate the effects of its chosen action from those of unobserved confounders that influence both the

robot's decisions and the resulting outcomes. As discussed in Sec. 4.2.2.7, intervening on the action variable blocks the influence of the unobserved confounder U, enabling unbiased estimation of the transition dynamics for use in sample-based planning: $T_{\text{Intervention}}(s, a, s') = P(S' = s' \mid S = s, do(A = a))$.

This causally-informed formulation enables the robot to reason about **how** it should update its belief over transitions in the presence of partial observability and latent structure (e.g., *How likely am I to move forwards safely without collision given my current uncertainty?*), and **how** it should act to achieve its long-term planning goals (e.g., *Which trajectory maximises my chance of safe navigation?*).

Our method consists of two main components:

1. Learning the parameterisation of a structural causal model that represents the transition dynamics; and

2. Using this model, together with interventional inference, to improve POMDP planning performance in the presence of unobserved confounding.

We begin by describing our adopted causal representation of the POMDP model.

4.4.1 SCM Representation of POMDPs

We adopt a structural causal model (SCM) formulation of POMDPs, following the SCM-based MDP formulation in [26], to enable interventional and counterfactual reasoning. This section introduces the SCM representation of POMDPs and outlines how it enables reasoning under unobserved confounding, focusing on the causal DAG structure, endogenous and exogenous variables, and the semantics of interventional inference.

POMDP Causal DAG. We begin by modelling the decision-making process of a POMDP as a causal DAG, shown in Fig. 4.6. In this representation, the underlying world state S_k is unobserved (grey nodes), while the agent's actions A_k, observations Z_k, and immediate rewards R_k are observable (white nodes), consistent with standard POMDP assumptions.

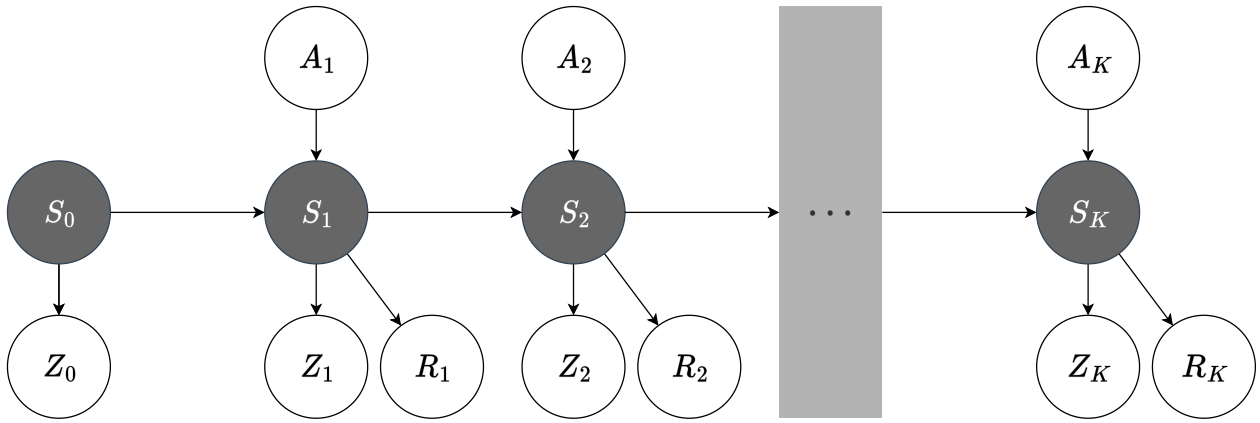


Figure 4.6: A K-step POMDP represented as a causal DAG. Subscripts denote time. Observed variables are shown as white nodes; unobserved as grey.

SCM Formulation of UCPOMDPs (SCM-UCPOMDPs). An SCM-UCPOMDP augments a UCPOMDP (Sec. 3.4.6) with an explicit SCM structure, enabling causal reasoning through structural assignments and interventions, and supporting *level 2* interventional and *level 3* counterfactual inference. As described in Sec. 3.1.6, an SCM is a 4-tuple $M = \langle U, V, F, P(U) \rangle$, where:

- U is the set of exogenous variables;
- $P(U)$ is the probability distribution over exogenous noise terms;
- V is the set of endogenous variables;
- F defines how each $v_i \in V$ is deterministically computed from its parents and u_i: $v_i := f_i(pa_i, u_i)$.

To model a UCPOMDP as an SCM, we decompose the model's transition, observation, and reward functions into: *exogenous variables* and their distributions $P(U)$; and *endogenous variables* and deterministic assignment functions F.

The agent's choice of action $A_{k+1} \in A$ and the independent unobserved confounder U_k are also included in the causal DAG and decomposed into their endogenous and exogenous terms. The SCM-UCPOMDP thus has the following structure:

$$
\begin{aligned}
n_{u_{k+1}} &\sim P(N_{u_{k+1}}) \\
n_{a_{k+1}} &\sim P(N_{a_{k+1}}) \\
n_{s_{k+1}} &\sim P(N_{s_{k+1}}) \\
n_{z_{k+1}} &\sim P(N_{z_{k+1}})
\end{aligned}
\qquad
\begin{aligned}
u_{k+1} &:= f_u(n_{u_{k+1}}) \\
a_{k+1} &:= f_a(n_{a_{k+1}}, u_{k+1}) \\
s_{k+1} &:= f_s(n_{s_{k+1}}, s_k, a_{k+1}, u_{k+1}) \\
z_{k+1} &:= f_z(n_{z_{k+1}}, s_k, a_{k+1}, s_{k+1}) \\
r_{k+1} &:= f_r(s_k, a_{k+1}, s_{k+1})
\end{aligned}
\tag{4.2}
$$

Removing Confounder Bias Using the *do*-operator. To obtain an unbiased estimate of the true transition function, we apply the *do*-operator to perform a simulated intervention on the action: $P(S' \mid do(A = a), S)$. The choice of intervention target (i.e., intervening on A) is determined by the assumed causal graph and is explicitly specified by the system designer to block identified confounding paths (e.g., $A \leftarrow U \rightarrow S'$). This severs the back-door path by removing the influence of U on A, retaining only the direct causal effect $A \rightarrow S'$. We illustrate this in Fig. 4.7.

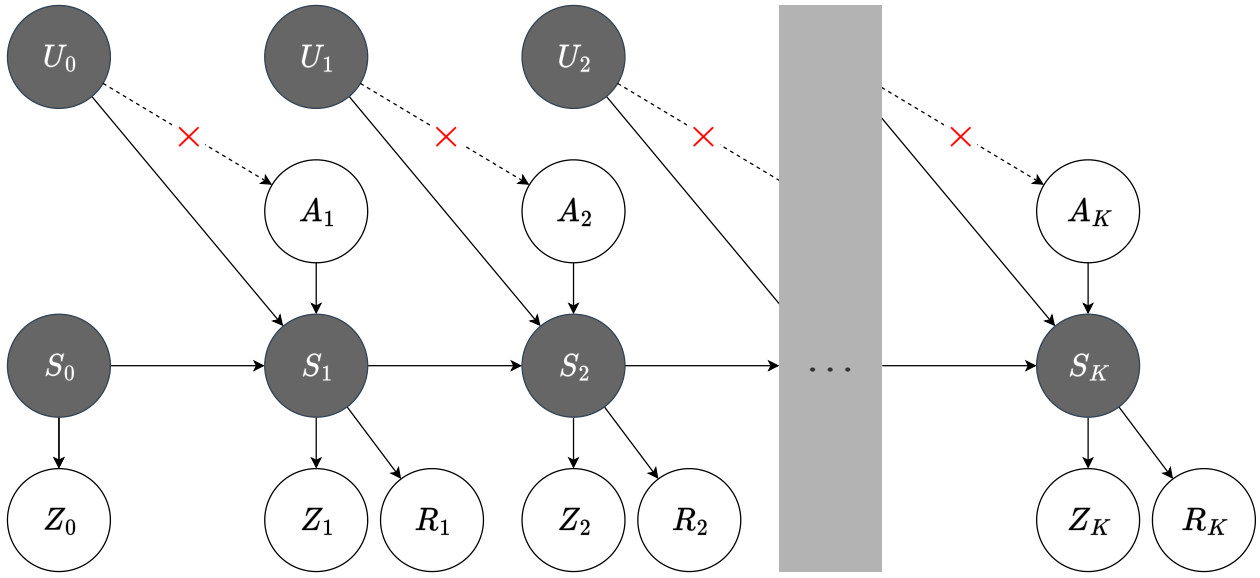


Figure 4.7: A K-step UCPOMDP causal DAG under simulated intervention. Applying $do(A_k = a_k)$ removes the influence of the confounder U_k on the agent's action, yielding the interventional distribution $P(S' \mid do(A = a), S)$. This isolates the causal influence $A_{k+1} \rightarrow S'_{k+1}$ by severing the back-door path $A \leftarrow U \rightarrow S'$.

Finally, although our focus is on *level 2* interventional inference, this SCM formulation provides a foundation for the future application of *level 3* counterfactual queries in POMDP planning under unobserved confounding, as explored in [27].

4.4.2 Model Parameter Learning Method

Our goal is to learn the parameterisation of an SCM-UCPOMDP model from a ground truth system in a privileged setting prior to planning. This reflects real-world robotics scenarios in which the full system dynamics are not known *a priori*, but must instead be constructed using a hybrid approach that combines domain knowledge with offline data collection before robot deployment.

We define a ground truth SCM-UCPOMDP model M that serves as a proxy for the real-world system. During learning, we aim to obtain a model $\hat{M}$ that approximates the transition probabilities from data generated by observing a robot operating under the dynamics of M. We assume sufficient domain knowledge exists to identify the structure of causal relationships between action, confounder, and state variables. Accordingly, we fix the causal structure and seek to learn the associated conditional probability distributions.

Transition Function Dimensionality Reduction Using Domain Knowledge. To avoid learning a large $|S||A||S|$-sized transition matrix, we exploit the Markov property of POMDPs and decompose the transition function into an additive composite function:

$$f_{s'} = f_{s'_s}(s) + f_{s'_{\Delta s}}(\Delta s), \tag{4.3}$$

where $f_{s'_s}$ captures the influence of the current state, and $f_{s'_{\Delta s}}$ encodes the relative transition outcome. This decomposition allows us to encode task-specific inertial dynamics using domain knowledge and reduces the parameter dimensionality.

In the Confounded GridWorld problem, the robot may move to any of its 8 neighbouring cells, due to intended actions, confounder influence, and potential slip. Hence, the relative transition distribution for a given (s, a) pair can be compactly represented as a 3×3 probability matrix. This reduces the full transition model from size $|S||A||S|$ to a compact $|S||A| \times 9$ matrix (see Fig. 4.10).

Consequently, since we have this partition of initial and relative transition terms, we further assume that the transition function outcome uncertainty (marginalised over the unobserved confounder variable U) arises only from the relative transition term. As such, we model the uncertainty solely as the relative transition probability distribution $P(\Delta S|A, U)$.

We further assume that uncertainty in the transition function — marginalised over the unobserved confounder U — arises entirely from the relative transition term. As such, and

since we have this partition of initial and relative transition terms, we model the transition distribution as $P(\Delta S \mid A, U)$.

We further exploit the fact that in the Confounded GridWorld problem, the agent is not always under the influence of the confounder. We identify two distinct modes in the transition function: 1) when the agent is under the influence of the confounder; and, 2) when it is not. Thus, we partition the transition function into two components accordingly:

$$P(S'|S, A, U) = \begin{cases} P_{UC}(S'|\Delta S, A, U) & \text{for } s \in S_{UC} \\ P_0(S'|\Delta S, A) & \text{otherwise,} \end{cases} \tag{4.4}$$

and learn the relative transition distributions $P_{UC}(\Delta S \mid A, U)$ and $P_0(\Delta S \mid A)$. Here, we use UC to denote the case for which the agent is 'under confounding', and 0 when not.

For a grid of size $W \times L = 3 \times 4$, we have $|S| = 12$ and $|A| = 4$. Thus, the full model size of $12 \times 4 \times 12 = 576$ entries is reduced to $2 \times 4 \times 9 = 72$ — yielding an $8\times$ reduction. This substantially improves the efficiency of our SVI-based parameter learning (Sec. 4.4.2: Parameter Learning Method.).

In the Confounded GridWorld problem, P_{UC} corresponds to the transition model that applies when the agent is located in the orange (confounded) cell (see Fig. 4.5), while P_0 applies elsewhere.

Access to Privileged Information During Training. We assume a two-phase data-collection and robot deployment pipeline, typical in industrial robotics: (1) an initial site survey with humans in the loop for data collection, followed by (2) autonomous robot deployment.

In this setting, we assume the robot-and-human site survey team is equipped with sensors to observe all ground truth model variables, including the confounder variable U that is unobserved during autonomous deployment. Given these measurements of confounder U, we seek to learn a parameterisation for $P(U)$, the independent distribution governing the probabilities of the confounding variable.

This approach allows us to leverage privileged training data to learn a deployable causal model that supports interventional reasoning at planning time, even when key variables such as U are no longer observable.

Importantly, since the confounder U is observed during training, it does not bias the learning process. Its influence is explicitly conditioned on during estimation of the marginal transition model $P(S' \mid S, A)$, and thus is controlled for.

Limitations of Observability Assumptions. The assumption of full observability during data collection represents an idealised setting. In practice, even during site surveys, full observability of all relevant variables may not be achievable due to sensor noise, limited sensing modalities, or environmental uncertainty. As a result, variables such as the confounder U may be only partially observed or indirectly inferred, introducing additional uncertainty into the learned model. This can degrade the accuracy of causal effect estimation and limit the effectiveness of subsequent intervention-based planning.

Sensitivity to Data Quality. The effectiveness of this approach further depends on the quality of the offline dataset used for model learning. If the collected data is noisy, biased, or contains measurement errors in any variables used to estimate the transition model (e.g., variables contained within S, A, U, or S'), then the learned model $\hat{M}$ may misrepresent both the system dynamics and the causal relationships encoded in the SCM. In particular, incorrect estimation of $P(U)$ or the conditional transition distributions can lead to biased effect estimates, which propagate to the planner and degrade decision-making performance at deployment time. This highlights the sensitivity of the approach to the quality of offline data used for model learning.

Practical Identification of Latent and Extraneous Variables. In practice, identifying relevant latent or extraneous variables is a non-trivial process that often requires real-world deployment. During initial system operation, unexpected behaviours or performance degradations may reveal the presence of previously unmodelled influences affecting the system dynamics or observations. These may include latent state variables, disturbances influencing the transition function, or confounders that affect both action selection and state evolution.

This motivates an iterative workflow in which deployment informs the identification of relevant variables, which in turn guides the design of sensing and data collection strategies for subsequent site surveys. As a result, the set of variables included in the causal model is refined over time, reflecting both domain knowledge and empirical observations from the operating environment.

This process is critical for ensuring that the assumed causal structure captures the key dependencies required for valid effect estimation, and aligns with an iterative cycle of model

refinement and falsification, where discrepancies between predicted and observed behaviour drive revision of the assumed causal relationships.

Assumed Model Components. We assume the sensor model $P(Z \mid S')$ is known, as is a deterministic reward function designed by system engineers. Hence, we do not learn parameterisations for $P(Z \mid S')$ or $P(R \mid S)$. In fact, since the reward function is deterministic, it is not represented as a probability distribution (see Eq. 4.2).

Parameter Learning Method. We begin the learning process by sampling a dataset D of N observations from the ground truth model M, implemented as a probabilistic program using the Pyro PPL [46]. Each observation is a record of the tuple $\langle \mathrm{UC}, U, A, \Delta S \rangle$, where UC is a Boolean value indicating whether the agent was under the influence of the confounder:

$$UC(S) = \begin{cases} \text{True} & \text{if } (S_{col}, S_{row}) = (1, 2), \\ \text{False} & \text{otherwise.} \end{cases} \tag{4.5}$$

We define a learning model $\hat{M}$ with the same causal structure as M, but with unparametrised distributions $P(U)$, $P_{UC}(\Delta S \mid A, U)$, and $P_0(\Delta S \mid A)$. These are modelled as categorical distributions.

We learn the parameterisation of these three distributions by fitting $\hat{M}$ to the dataset D through stochastic variational inference (SVI) [49, 137] in Pyro PPL.

We use the Adam optimiser [138] and Pyro's *TraceEnum_ELBO*, which supports efficient enumeration over discrete latent variables during learning.

4.4.3 CAR-DESPOT: A Causally-Informed MCTS-Based POMDP Planner

We now describe **CAR-DESPOT**, a causally informed extension of AR-DESPOT that unites the online planning efficiency of AR-DESPOT with the causal expressivity of structural causal models (SCMs), thereby eliminating policy errors arising from confounding bias. Since SCM-UCPOMDPs subsume standard POMDPs, they retain the latter's properties and can be used as drop-in replacements within the DESPOT framework. Accordingly, CAR-DESPOT formulates models as SCM-UCPOMDPs to exploit the existing POMDP search machinery while enabling interventional reasoning.

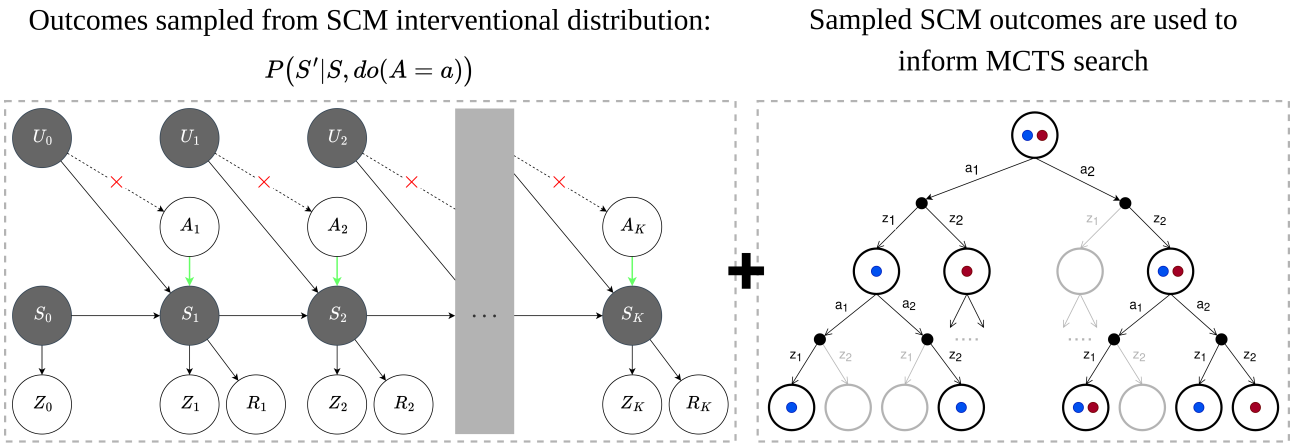


Figure 4.8: *CAR-DESPOT* Planning Framework: a causally informed MCTS-based POMDP planner that pairs the causal expressivity of structural causal models (left) with sample-based online search to achieve efficient planning performance (right).

Core Change: Interventional Rollouts. A fundamental modification in CAR-DESPOT is how the transition function is sampled during exploration rollouts. Whereas AR-DESPOT samples from the *observational* transition distribution $P(S' \mid S, A)$ — which is biased in the presence of unobserved confounders — CAR-DESPOT samples from the *interventional* distribution

$$P\Big(S' \mid S,\, do(A = a)\Big),$$

which severs the back-door path from the confounder to the action and removes spurious dependencies. This change improves planning in two complementary ways: (i) it prevents the incremental DESPOT construction from being steered by confounded transition estimates towards less promising regions of the tree; and (ii) it yields more accurate value estimates for discovered action-observation outcomes, reducing the chance of settling on sub-optimal policies.

The same interventional treatment is applied when computing the observation function where appropriate. In models where observations depend only on the successor state (as in our toy domain), $P(Z \mid S') \equiv P\Big(Z \mid S',\, do(A = a)\Big)$, so no change is required.

Causal Generative Backend. SCM-UCPOMDPs are implemented as causal generative probabilistic programs in *Pyro* PPL. CAR-DESPOT queries interventional transition posteriors via importance sampling inference [48], with results memoised for reuse during online search. This provides the planner with unbiased transition probabilities at rollout time without altering the DESPOT tree-search logic itself.

Key Modifications Relative to AR-DESPOT.

1. **Model Class:** POMDP models are expressed as SCM-UCPOMDPs to support interventional queries.

2. **Rollout Dynamics:** replace $P(S' \mid S, A)$ with $P\Big(S' \mid S, do(A = a)\Big)$ when sampling state transitions in rollouts and belief updates.

3. **Inference Pathway:** compute interventional probabilities via a causal inference backend (importance sampling over the SCM), with caching/memoisation for efficiency.

Implementation. CAR-DESPOT extends the authors' `C++` AR-DESPOT implementation[2] to interface with the SCM-based causal inference backend. Evaluations follow the planning and assessment pipeline of Somani et al. [24], with interventional transition probabilities supplied to the rollout and belief-update routines at plan time.

4.4.4 Robot System Integration

To demonstrate how the proposed causal planner operates as part of a complete robotic decision-making pipeline, we integrate **CAR-DESPOT** into a modular robot architecture comprising perception, causal inference, and control subsystems (Fig. 4.9). The integration preserves the standard *sense–plan–act* cycle while inserting causal reasoning into the planning stage.

System Overview. At the front end, CAR-DESPOT retains the standard anytime online POMDP interface of AR-DESPOT, exposing a high-level planning API to task-level executive control modules (e.g., mission planners or supervisory controllers). The backend replaces the conventional generative model with a bespoke *causal inference engine* that provides:

- Abstractions of SCM-based causal world and robot models;
- Inference routines for observational and interventional queries; and
- A generative model sampling interface for the Monte Carlo Tree Search (MCTS) planner.

[2]AR-DESPOT: `https://github.com/AdaCompNUS/despot`

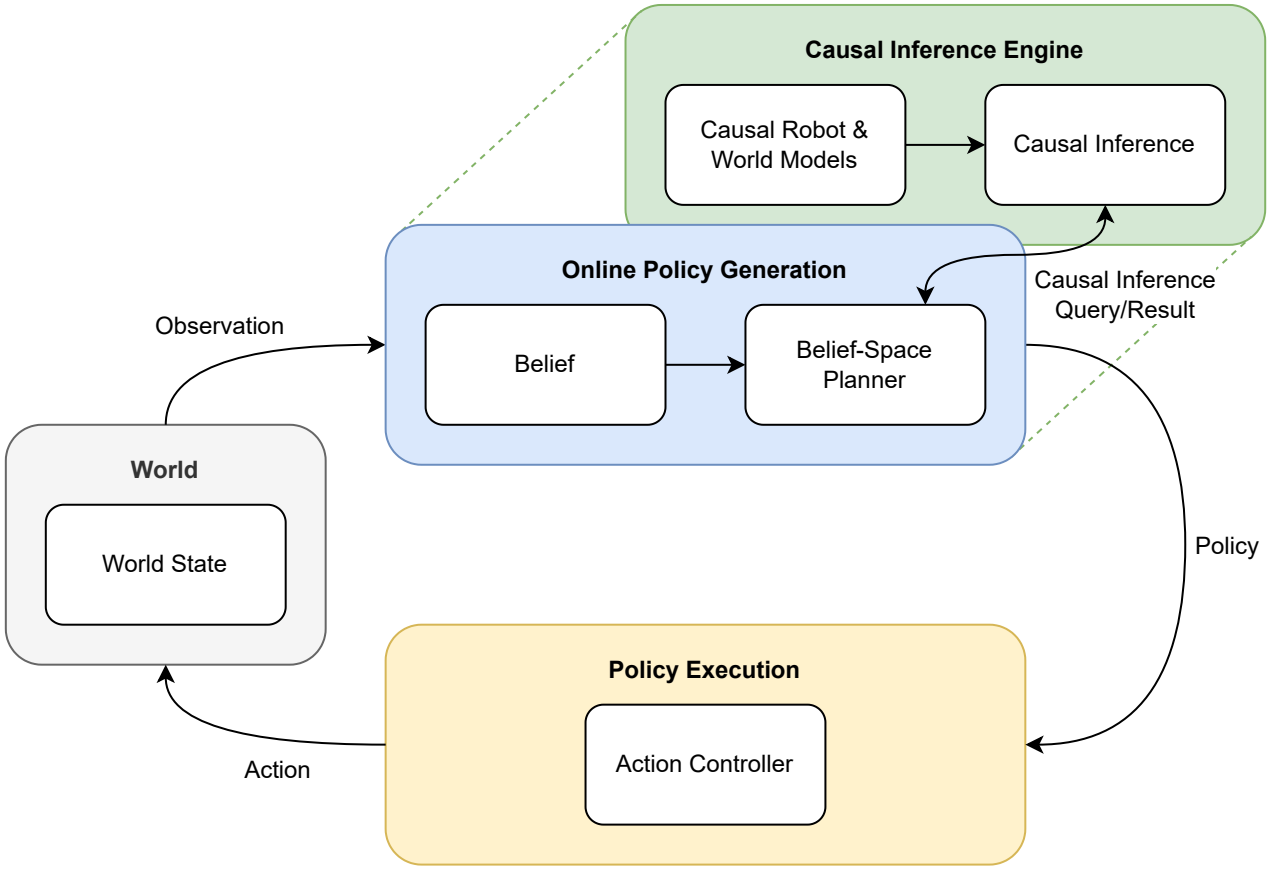


Figure 4.9: System architecture showing how *CAR-DESPOT* integrates causal inference with online robot planning and execution. The planner interfaces with high-level executive control modules and a causal inference back-end that provides SCM-based generative models and interventional sampling capabilities.

Planning Loop. At runtime, the robot acquires a partial observation of the world state through its onboard sensors and fuses this information to update its internal belief over latent variables. From this belief, CAR-DESPOT performs interventional rollouts over the SCM-UCPOMDP model to generate an approximate long-horizon policy. Only the first action of this policy is executed in the real world by the relevant control module (e.g., base motion, navigation, manipulation, or human-robot interaction). After action execution, a new observation is received, the belief is updated, and the next online planning cycle begins. This iterative loop continues until the task goal or a termination condition is reached.

Causal Reasoning in the Loop. During each planning iteration, the causal inference engine is queried to compute interventional transition probabilities $P(S' \mid S, do(A = a))$ and associated observation likelihoods. These values are provided to the MCTS rollout process, ensuring that all simulated trajectories are free from confounding bias. The causal

backend operates synchronously, caching frequently queried distributions to reduce inference latency during online execution.

Functional Integration. This architecture allows CAR-DESPOT to be deployed within existing robot software stacks with minimal modification: only the transition and observation sampling routines are replaced with interventional queries, while the rest of the task-planning interface, belief update mechanism, and controller integration remain identical to AR-DESPOT. Consequently, CAR-DESPOT can serve as a drop-in replacement for standard planners in robotics middleware — such as the Robot Operating System (ROS) — enabling causal decision-making without disrupting existing control infrastructure.

4.5 Experiments

4.5.1 Model Parameter Learning

We generate a data set of size $N = 800{,}000$ by sampling from the ground truth model M and use this as input to stochastic variational inference (SVI) training. We use a learning rate $lr = 0.01$ and 5001 training steps to obtain the learned model $\hat{M}$, including the parameterisation of: $P_{UC}(\Delta Position, A, U_{OrientationError})$, $P_0(\Delta Position, A)$, and $P(U_{OrientationError})$. We learn 106 latent model parameters, using uniform Dirichlet priors for categorical distribution parameters in our learning model, defined over transition distributions conditioned on the action variable (i.e., $P(S' \mid S, A, U)$), and the *AutoDelta* automatic guide for the SVI guide. This formulation aligns with the downstream use of the model in sample-based planning, where transitions are generated by querying the conditional model $T(S, A, [U], S')$ for a given state-action pair, allowing each transition distribution to be modelled independently using a categorical distribution with a Dirichlet prior.

We report the final ELBO loss, as well as the Kullback-Leibler (KL) divergence between the learned full transition probability distribution $\hat{P}(S'|S, A, U)$ and the ground truth.

Stochastic variational inference approximates the intractable posterior over model parameters using a variational distribution $q(\theta)$, which is optimised by maximising the evidence lower bound (ELBO), a tractable surrogate for the log marginal likelihood. This optimisation can be interpreted as minimising the KL divergence between the variational distribution and the true posterior.

In this work, we use Pyro's *AutoDelta* guide, which corresponds to a deterministic (point-estimate) approximation of the posterior, yielding a *maximum a posteriori* (MAP) estimate of the model parameters. Thus, optimisation of the ELBO in this setting corresponds to directly learning a single set of parameters that best explains the observed data under the assumed model structure. This reflects a practical Bayesian approach, in which full posterior distributions are not explicitly propagated, but instead approximated by point estimates to maintain computational tractability in downstream planning.

4.5.2 Planner Evaluation

We evaluate CAR-DESPOT on the Confounded GridWorld toy problem and compare it with a baseline method OAR-DESPOT, a version of AR-DESPOT modified to be compatible with SCM-UCPOMDP model representation but using observational transition quantities as done in the original algorithm. To assess the impact of our proposed learning method on planning performance, we evaluate each method in 2 cases, in which: 1) the learned model $\hat{M}$ is used for planning but the ground truth model M is used for policy execution as a proxy to the real-world; and, 2) the ground truth model is used for both planning and policy execution.

The performance is assessed according to the average total discounted reward over 100 independent online planner executions. We define the maximum scenario length, maximum search depth, and policy simulation depth as 15 steps. A search budget of 150 seconds is given to the planners for each time step. This was chosen empirically to allow the planners to sample a sufficient number of trials during the incremental DESPOT creation to sample the low-probability transition outcomes leading to failure in the toy problem. The default discount factor $\gamma = 0.95$ and target gap ratio $\epsilon = 0.95$ is used, with regularisation pruning constant $\alpha = 0.01$. The number of scenarios is 500, a sufficient amount to achieve performance in similar problems in [117]. To produce results that we expect to generalise across other similar problems, the domain-independent default initial upper and lower bounds are used [24]. To avoid penalising the first step of online planning, all required transition inference queries are computed and memoized prior to planning. Observation function inference queries are computed as-needed during belief updates but cached for future re-use. 5000 particles are used in Importance Sampling for both inference types; this was chosen empirically to accurately infer probabilities for low-probability outcomes. All planner evaluations are performed on a

desktop PC with Ubuntu 20.04 operating system, using CPU-only, on an AMD Ryzen 9 5900x 12-core processor with 128 GB of RAM.

4.6 Results & Discussion

4.6.1 Analysis of learned model

We begin by presenting and analysing the partially-learned SCM-UCPOMDP model of the Confounded GridWorld problem. SVI training yields an ELBO-loss of 3.300M (4.125 per data point) and produces interventional relative distribution probabilities each with an absolute error of less than 0.01 from ground truth values. Further, the probabilities for $U_{OrientationError}$, $[0.10, 0.80, 0.10]$, are accurately recovered during training: $[0.1004, 0.7997, 0.0999]$. The KL divergence of the learned full transition probability distribution $\hat{P}(S'|S, A, U)$ from ground truth distribution $P(S'|S, A, U)$ is $D_{KL} = 0.0021$, indicating a very close fit. Given the large dataset size (N = 800,000), this low divergence and small absolute errors (< 0.01) provide strong evidence that the learned model closely approximates the ground truth distribution, and are very unlikely to be due to estimation error from finite sampling of the underlying distribution.

Fig 4.10 shows the learned parameters produce calculated **interventional** transition probabilities and inferred **observational** relative transition probabilities that are extremely close to the ground truth probabilities, when under the influence of confounding. Observational transition probabilities are inferred using Importance Sampling. These relative transition probabilities are provided as exemplar inference results; the full (i.e., non-relative) transition inference results used for planning are computed immediately prior to plan time.

All non-zero probability outcomes are captured by the learned model, even very low probability events (e.g., $p < 0.001$), demonstrating that both the drift and magnet interference transitions dynamics are recovered accurately by the learning method. The non-confounded observation and interventional transition probabilities are also recovered accurately. Due to space constraints these results are not shown as they are a subset of the confounded transition dynamics.

Critically, the learned model is able to capture the difference between the observational and interventional transition dynamics when under confounding. Comparing the probabilities shown in Fig. 4.10, the learned model correctly captures the probability mass shift due to confounding bias in the observational transition dynamics. This is most evident for the *UP* action (corresponding to `MOVE_UP` in Fig. 4.10), where the interventional distribution places

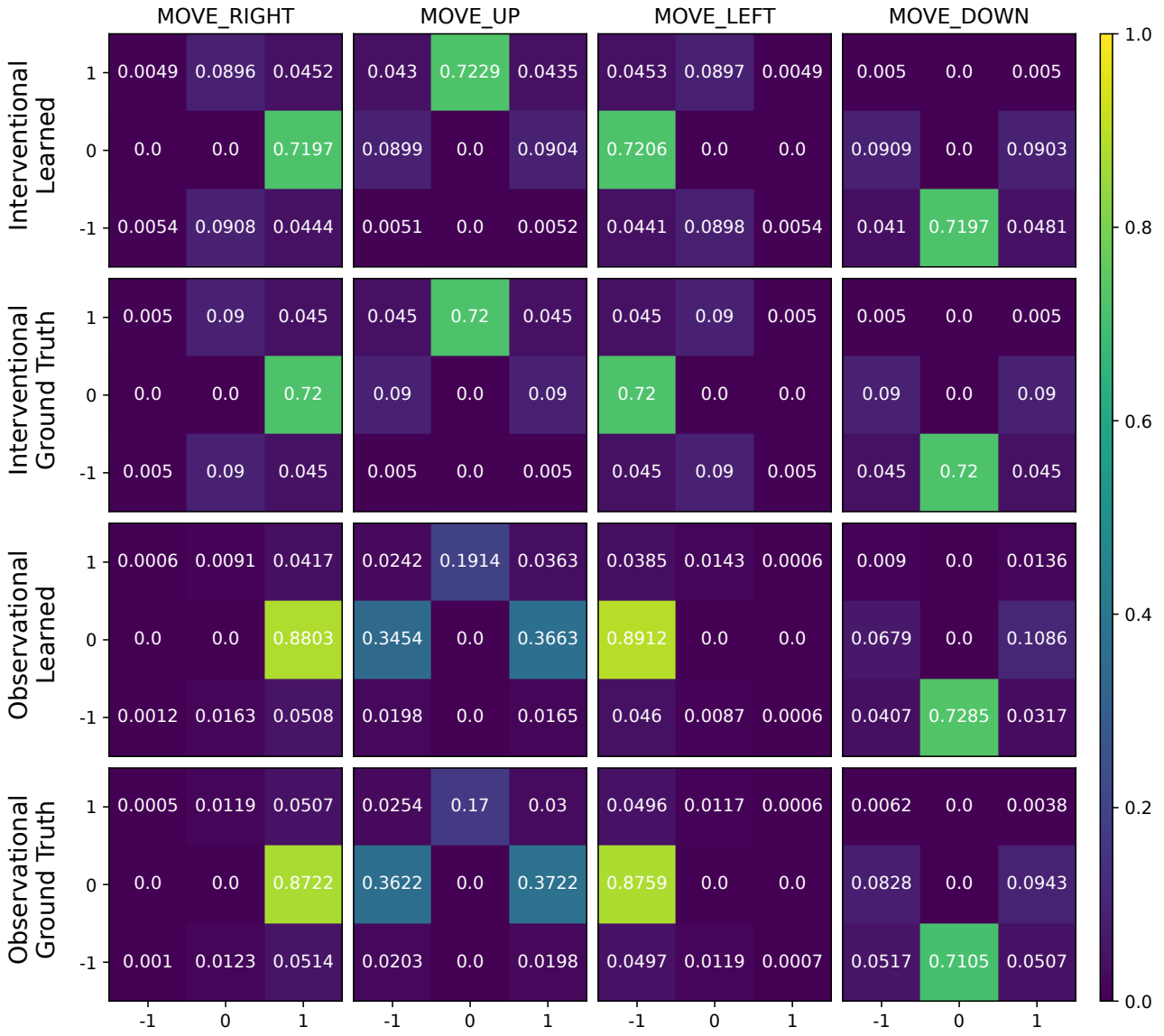


Figure 4.10: Interventional and observational transition probabilities of ground truth and learned models, under the influence of confounding, as a function of action choice. Results are computed from a dataset of size $N = 800{,}000$. Formulating actions as causal interventions removes confounding bias errors present in the observational transition predictions. The learned interventional transition probabilities match ground truth within an absolute error of 0.01, indicating a close fit that is very unlikely to have arisen by chance.

Table 4.2: Mean total discounted reward over $N = 100$ evaluation runs for each planner-model combination. Values are reported as mean ± standard error. Higher values indicate better task performance, reflecting both higher task-success rates and more efficient trajectories (i.e., fewer steps and reduced discounting). Lower values arise from increased failure rates and suboptimal path selection due to confounding bias.

	Learned Model	Ground Truth Model
OAR-DESPOT	0.74 ± 7.73	10.81 ± 8.53
CAR-DESPOT	21.93 ± 9.21	15.54 ± 9.34

the majority of probability mass on the intended outcome $(0, +1)$, while the observational distribution redistributes this mass to lateral outcomes $(-1, 0)$ and $(+1, 0)$. This effect is clearly visible in the heatmaps in Fig. 4.10, where the observational rows show substantial lateral probability mass compared to the interventional case. This causes the observational distribution to significantly underestimate the interventional probability: 0.1914 instead of 0.7229.

This discrepancy arises because the observational estimate reflects a mixture over the unobserved confounder U, which simultaneously influences both the executed action and the resulting state transition. In particular, when the confounder induces orientation error, the intended *UP* action is more likely to result in lateral transitions, shifting probability mass away from the $(0, +1)$ outcome. As a result, the observational distribution conflates the effect of the action with the influence of the confounder, leading to an underestimation of the true causal effect captured by the interventional distribution.

We expect that when the learned model is used for planning with OAR-DESPOT, the *UP* action will be undervalued when under the effects of confounding. We expect this when using the ground truth model also.

4.6.2 Analysis of planning performance

We present and discuss the planner evaluation results for the Confounded GridWorld problem. The distribution of policy performance over 100 runs of each planner and model combination is given in Fig 4.11. We report the total discounted reward of policies executed on the ground truth model. These performance distributions are summarised in Table 4.2. The reported mean values therefore reflect both how often trajectories result in task success versus failure, and how efficient the successful trajectories are (i.e., how quickly the goal is reached with less discounting), as observed in Fig. 4.11. Reporting mean ± standard error assumes an approximately Gaussian sampling distribution of the mean. Although the underlying reward

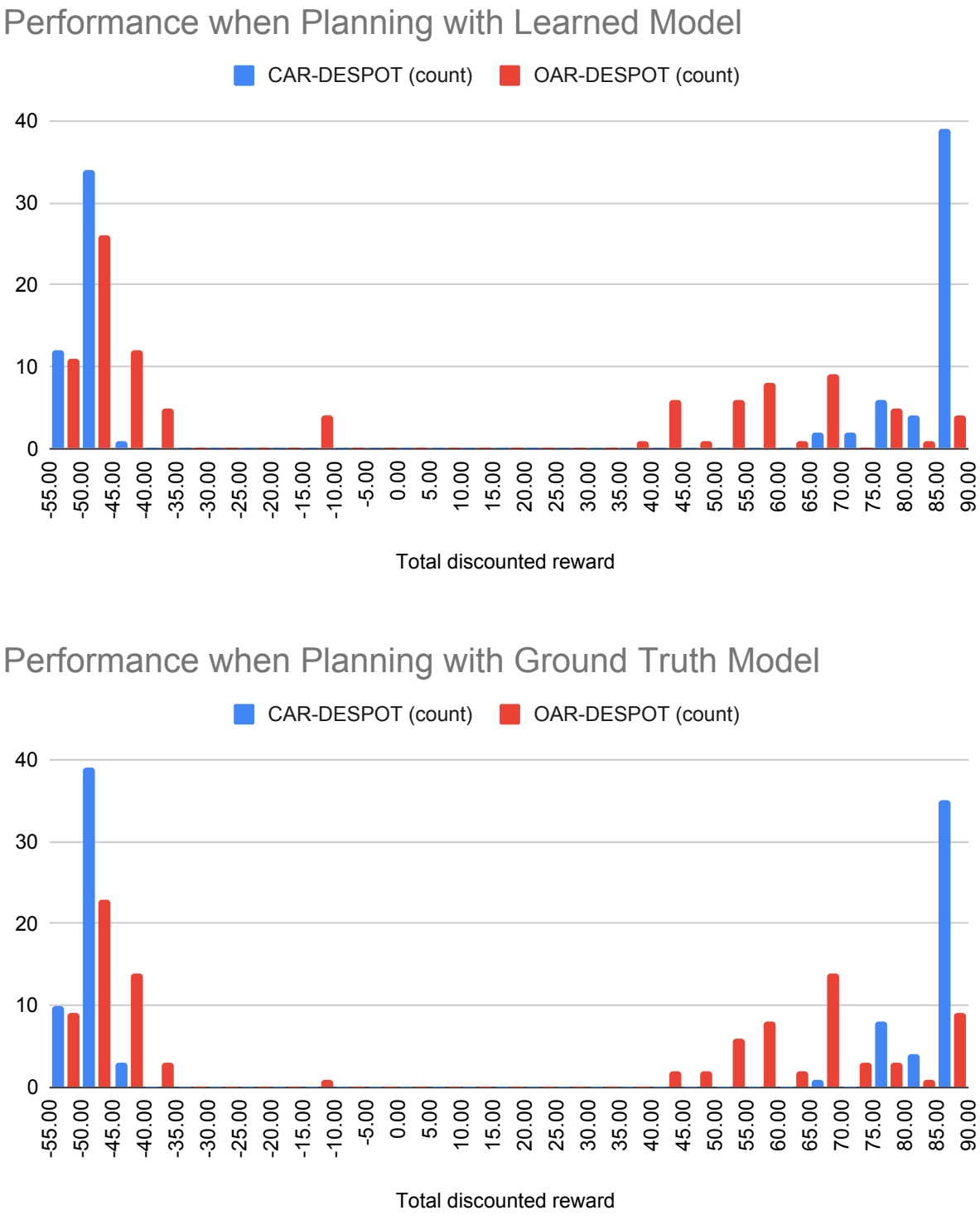


Figure 4.11: Distribution of CAR-DESPOT and baseline planner performance, with learned and ground truth models, according to total discounted reward over $N = 100$ runs. The bimodal structure reflects task failure (low reward) and task success (high reward). CAR-DESPOT shifts probability mass towards the high-reward regime and produces more concentrated, higher-valued success outcomes, indicating both increased task-success frequency and more efficient trajectories (i.e., fewer steps and less discounting). This reflects improved action evaluation resulting from correcting confounding-induced bias.

distributions are non-Gaussian (Fig. 4.11), this approximation is justified by the Central Limit Theorem under repeated independent runs ($N = 100$), and is consistent with prior evaluation methodology [24, 117].

The performance distributions of all planner-model combinations are largely bimodal, with rewards either clustered around the lower end of the reward range or upper end, corresponding to task failure by colliding with a wall or task success by reaching the goal respectively. Notably, for OAR-DESPOT there is also a small third peak around -10 corresponding to task failure by means of exhausting the maximum allowable number of steps. This occurred in 4 runs for learned model and 1 run for ground truth, in which the OAR-DESPOT-generated policies caused oscillatory behaviour that prevented reaching the goal within the action budget. This did not occur in any runs for CAR-DESPOT. Overall, differences in planner performance are reflected as shifts in probability mass between the failure and success modes, as well as changes in the location and concentration of the success peak. Thus, improvements in decision-making manifest both as increased task-success frequency and as higher-quality (shorter, less-discounted) successful trajectories.

For the **learned model**, the task-success peak is more concentrated for CAR-DESPOT than OAR-DESPOT and has a higher mode. The reason for this is that since there is no confounding bias in the inferred interventional transition dynamics when the robot is at $(0, 2)$ and thus under the influence of the confounder, CAR-DESPOT correctly estimates the value of taking actions to move along the left-hand side of the grid, which is shorter and therefore results in a higher score due to fewer movement penalties and cumulative rewards being discounted over a shorter time horizon. Thus, in all 100 runs CAR-DESPOT chooses to take the left-hand path. There is also another benefit of taking the shorter path: the cumulative probability of accidentally drifting into a wall resulting in task failure is lower due to the fewer steps taken. Thus, more runs achieve task success: 53 for CAR-DESPOT and 42 for OAR-DESPOT; this further increases the mean performance. Additionally, 39 runs achieve the maximum score. The lower tail end of the task-success peak reflects a small number of runs which exhibited minor oscillatory behaviour before reaching the goal. The task-failure peak is again more concentrated but with a lower mode. These reflect runs in which the outcomes of the correctly chosen actions by chance resulted in unintended outcomes, due either to the drift dynamics or the confounding variable influencing the outcome, leading to task failure. Since the CAR-DESPOT policies choose the left-hand path, failures occur earlier in the run and thus are discounted less.

Conversely, OAR-DESPOT underestimates the value of taking the left-hand path due to the confounding bias in the inferred transition probabilities for taking the *UP* action at $(0, 2)$ when under the influence of the confounder. Because of this, it has a strong preference to take the right-hand path to the goal, which is free from confounding but has a longer action cost and higher cumulative chance of failure due to drift. This yields a lower total discounted score. Interestingly, in 4 of the runs OAR-DESPOT takes the left-hand path and achieves the maximum possible score. This is likely due the incremental DESPOT construction missing either: 1) the failure outcomes at the confounder influence location; or, 2) the success outcome for taking the right-hand path. This demonstrates that CAR-DESPOT produces systematically better policies by correcting confounding-induced bias in action evaluation, resulting in a favourable redistribution of trajectory outcomes towards higher-reward regimes.

Similar results are observed for the **ground truth model**. The CAR-DESPOT task-success peak is again narrower and concentrated around a higher value than that of OAR-DESPOT, while the task-failure peak is narrower and concentrated around a lower value, for the same reasons identified previously. The number of task-success runs using the ground truth model with CAR-DESPOT is slightly lower compared to the learned model case: 48 compared to 53. This slight difference in frequency-based counts is expected due to the stochastic nature of the problem — even executing an optimal policy can occasionally and randomly result in task failure. However, we expect that these task-success counts would converge in the limit of performing more evaluation runs. Notwithstanding this, these results demonstrate that, when using the ground truth model, planning with CAR-DESPOT produces a better overall performance than OAR-DESPOT. Further, the similar results obtained from use of learned and ground truth models for planning further demonstrates the ability of our learning method to learn the relative transition dynamics accurately.

4.7 Limitations & Future Work

Model Representation Assumptions. One limitation of our model learning method is the assumption that transition dynamics can be decomposed into a relative transition function and a known assignment function of the current state and relative state change. While this is suitable for the robot 2D position state representation in our toy problem, it may not be suitable for other

state representations. In these cases, the full transition function of size $|S| \times |A| \times |S|$ will need to be learned, which will require a larger data set and an increased number of training steps.

Scalability and Data Requirements. We have evaluated our proposed method on a simple toy POMDP model with a relatively small state size (12), action space (4), and observation space (30), corresponding to a small discrete grid-world domain. Despite this, the search time required to compute a good-performing policy online at each step is found to be 150s, which is prohibitively high for a mobile robot operating in real-time. This limitation is not unique to our approach, but is a fundamental challenge of POMDP planning more broadly. As discussed in Sec. 3.4.4, the scalability of POMDP planners is constrained by the *curse of dimensionality* and *curse of history*, which lead to exponential growth in the belief tree with planning depth, action branching factor, and observation branching factor. Consequently, scaling CAR-DESPOT to larger grid sizes or more complex domains will inherit these challenges, requiring approximations or more efficient planning strategies.

In addition, the approach requires a substantial amount of data to accurately estimate transition distributions. In this work, a dataset of size $N = 800{,}000$ is sufficient to recover the transition dynamics of a small grid-world with $|S| = 12$ and $|A| = 4$. Scaling to larger domains would significantly increase the number of transition parameters to be learned, and thus the data requirements and training time. Similarly, the effectiveness of the approach depends on the availability of informative priors (e.g., Dirichlet priors over transition distributions conditioned on actions), which in this domain encode simple locality and symmetry assumptions. In more complex robotic systems, specifying such priors may be more challenging, and weaker or misspecified priors may lead to slower learning or degraded performance.

Structural Assumptions. A further limitation is the assumption of a known or well-specified causal structure. In this work, the structure is treated as an inductive bias that enables tractable causal inference and planning. However, in more complex real-world robotic systems, the causal graph may itself be uncertain or only partially specified. Extending the approach to handle structural uncertainty — for example, by maintaining a belief over possible causal graphs — would increase modelling fidelity but introduce additional computational and inference challenges.

Transferability to Larger and Real-World Systems. While the current results demonstrate the benefits of causal modelling in a controlled grid-world setting, transferring this approach to larger or real-world systems presents additional challenges. These include scaling to higher-dimensional state and observation spaces, handling continuous dynamics, and learning under limited or noisy data. Furthermore, the combined requirements of data, prior specification, and online planning computation may become prohibitive without additional structure or approximation. These challenges are not limited to robotics, but also arise in other domains involving sequential decision-making under uncertainty, such as healthcare and medical decision-making, where system dynamics are complex, partially observed, and often only weakly specified. Addressing these challenges will require more data-efficient learning methods, scalable inference mechanisms, and approaches for learning or adapting causal structure and priors in more complex domains.

Future Work: Counterfactual Policy Evaluation. Finally, analysis in [26] has shown that for MDP-based problems with unobserved confounders, optimising policies based on the interventional transition function is not always guaranteed to produce optimal policies. Rather, policy optimisation using counterfactual inference produces a dominant policy. In this approach, the agent's reactive action selection is taken into account, which provides information about the state of the hidden confounder, and is used to compute the individual-level *effect of the treatment on the treated* (ETT) [32]. In this context, the ETT is a counterfactual quantity that contrasts — everything else held equal — the expected reward the agent would receive if they took the action they naturally wanted, with the reward they would have received instead if they had taken another candidate action that goes against their intuition. We expect that this improvement will translate to our work in POMDP planning. Consequently, we expect that the performance of CAR-DESPOT can be further improved by using the counterfactual quantity to evaluate actions. This is a promising avenue for future work.

4.8 Summary

This chapter addressed the challenge of *confounded decision-making* in robot planning under uncertainty, investigating how causal inference can mitigate bias introduced by unobserved environmental influences. It directly contributes to **Q1 - Modelling**, **Q2 - Structure and Parametrisation**, and **Q4 - Decision Making**, which together ask how causal generative

models can be used to encode and reason about the interdependence between a robots actions, environment, and internal decision processes, and how such models can support bias-free planning and action selection.

We first formalised the problem of unobserved confounding in robotic decision-making, illustrating how hidden environmental factors can induce spurious correlations between actions and outcomes, leading to systematic bias in standard probabilistic planners. To capture these effects explicitly, we proposed the **SCM-UCPOMDP** formulation, which extends the traditional POMDP model with structural causal semantics, enabling interventional reasoning via the $do(\cdot)$ operator. We then introduced a method to learn partial parametrisations of this model from data in a privileged training phase, addressing **Q2 - Structure and Parametrisation**, and demonstrated accurate recovery of ground-truth transition dynamics using stochastic variational inference within a probabilistic programming framework.

Building on this foundation, we developed **CAR-DESPOT**, a novel causally informed extension of the AR-DESPOT online POMDP planner — the first of its kind. CAR-DESPOT applies interventional inference during Monte Carlo Tree Search rollouts to eliminate confounding bias from transition and observation probabilities, thereby producing more reliable value estimates and policies. This directly addressed **Q4 - Decision Making** by demonstrating how causal inference can be embedded within online planning to improve robustness and policy quality under partial observability and unobserved confounding.

Through experiments on the *Confounded GridWorld* benchmark, we showed that CAR-DESPOT consistently outperforms the observational baseline (OAR-DESPOT), achieving higher total discounted reward and more reliable task completion. The results confirmed that causal treatment of transition dynamics using interventional sampling yields measurable improvements in planning performance, even when operating with learned rather than ground-truth models.

The methods developed here form a foundation for future extensions involving counterfactual reasoning. A natural next step is therefore to incorporate *counterfactual* reasoning for policy evaluation and improvement — e.g., leveraging the effect of treatment on the treated (ETT).

Overall, this chapter establishes the first step in the thesis technical contributions: integrating causal reasoning with sample-based POMDP planning to enable bias-free decision-making under uncertainty. It lays the conceptual and algorithmic groundwork for subsequent

chapters, which extend these ideas from online causal planning to causal world models for manipulation (Chapter 5) and to counterfactual explanation, reasoning, and generative simulation (Chapters 6–8).

The research presented in this chapter has been published in an IEEE venue as [7].

5

A Causal Bayesian Reasoning Architecture Using Probabilistic Programming for Robot Manipulation Under Uncertainty

Contents

5.1 Introduction 124
5.2 Causal Reasoning for Robot Manipulation 125
5.3 Causal Bayesian Reasoning Architecture 127
5.3.1 Robot Sequential Decision-Making Causal Model 127
5.3.2 Intervention-Based Causal Inference 129
5.3.3 Software Interface for Hardware Integration 130
5.3.4 Operational Pipeline and Execution Flow 130
5.4 Exemplar Block Stacking Task 131
5.4.1 Problem Definition 132
5.4.2 Exemplar Task Decision-Making Causal Model 132
5.4.3 Evaluation Tasks 138
5.5 High-Fidelity Gazebo Robot Simulation Evaluation 140
5.5.1 Experimentation Setup 140
5.5.2 Task 1: Tower Stability Prediction 143
5.5.3 Task 2: Greedy Next-Best Action Selection 144
5.6 Results & Discussion 145
5.6.1 Task 1: Tower Stability Prediction 145
5.6.2 Task 2: Greedy Next-Best Action Selection 148
5.6.3 Comparison to Existing Approaches 152
5.7 Real-World Robot Demonstration 154
5.8 Architecture Scalability & Limitations 156
5.8.1 Action Spaces 156
5.8.2 Sequential Decision-Making 157
5.8.3 Reward Functions and Statistical Objectives 158
5.8.4 Computational Limitations and Complexity 159
5.8.5 Adaptability and Domain Transfer 159

5.9 Summary 160

5.1 Introduction

In this thesis, we consider how causal generative machine learning and AI can be used for robots to reason about the *physicality* of systems under uncertainty. To reason effectively about physicality, agents must operate over abstractions that reflect the underlying *causal structure* of interactions, rather than relying on *statistical associations* between semantic representations — such as the correlations between language and vision learned by large-scale deep learning models like transformers or latent diffusion models. Such causal abstractions are essential for supporting robust inference, intervention, and generalisation in dynamic, unstructured environments.

Many existing world model learning approaches — such as model-based reinforcement learning (e.g., PETS [139], DreamerV3 [140]), latent dynamics models (e.g., PlaNet [141]), and planning-based agents (e.g., MuZero [142]) — learn internal predictive models to support decision-making and control. While effective for simulation and reward-driven planning, these systems typically operate in latent spaces and lack explicit representations of causal structure or physical laws.

In contrast, reasoning about physicality — how objects interact, constrain, and influence one another — requires structured, causal models that support counterfactual reasoning, compositionality, and generalisation under intervention. Such capabilities are critical in robotics, where agents must understand not only what will happen, but *why* and *how* outcomes unfold in the physical world.

In this chapter, we synthesise key sources of uncertainty that affect sensing, decision-making, and acting in mobile robot manipulation tasks to address the question of how causal generative machine learning models can serve as abstractions of world dynamics (**Q1 - Modelling**). To this end, we construct a generalised decision-making causal model and realise it in a PPL-based implementation — creating a flexible, extensible causal Bayesian reasoning architecture for robot manipulation under uncertainty that allows for the combination of Python-based sub-components such as custom task definitions, physics simulators, and deep learning models. We further ground this architecture in the block stacking exemplar task to demonstrate how the modelling can be combined with level-two interventional causal inference to address the

question of how actionable insights can be extracted from the causal world model to improve robot prediction and decision-making, which in turn enhances task performance and safety (**Q4 - Decision Making**).

5.2 Causal Reasoning for Robot Manipulation

Robot manipulation is central to applications such as warehouse logistics and domestic service robotics, and remains a long-standing focus of research [143–145]. Despite substantial progress, purely data-driven approaches often fail in novel scenarios not seen during training, which can lead to unsafe execution. Real-world robots also face multiple sources of uncertainty [113] — such as partial and noisy observations and stochastic actions — which further challenge generalisation.

Model-based methods offer a principled alternative by incorporating knowledge of system dynamics, enabling reasoning in unforeseen circumstances. In manipulation tasks, this requires understanding the probabilistic causal relationships governing object interactions, including physics-based concepts such as: collision physics, mass and inertia, friction, and gravity. However, real-world robot dynamics are often non-linear and complex, making faithful simulation and data generation difficult.

While deep learning has shown promise for learning manipulation skills, such methods typically operate at the level of statistical association, thus lack mechanisms for causal reasoning (i.e., interventions). They do not model symbolic or causal structure — both essential for building explainable and trustworthy autonomous systems. Probabilistic programming languages (PPLs) enable the implementation of generative models as programs, offering a flexible and principled foundation for causal reasoning in robotics. Their structured and modular design supports generalisable model construction and reasoning under uncertainty.

In this chapter, we formulate *COBRA-PPM*: a novel causal Bayesian reasoning architecture using probabilistic programming for robot manipulation under uncertainty. For the first time, our approach combines causal Bayesian network (CBN) modelling and inference, probabilistic programming, and online physics simulation to support generalisable, extensible robot decision-making in uncertain real-world environments. Crucially, our architecture explicitly accounts for perception and actuation noise, delivering robust performance across sequential placements

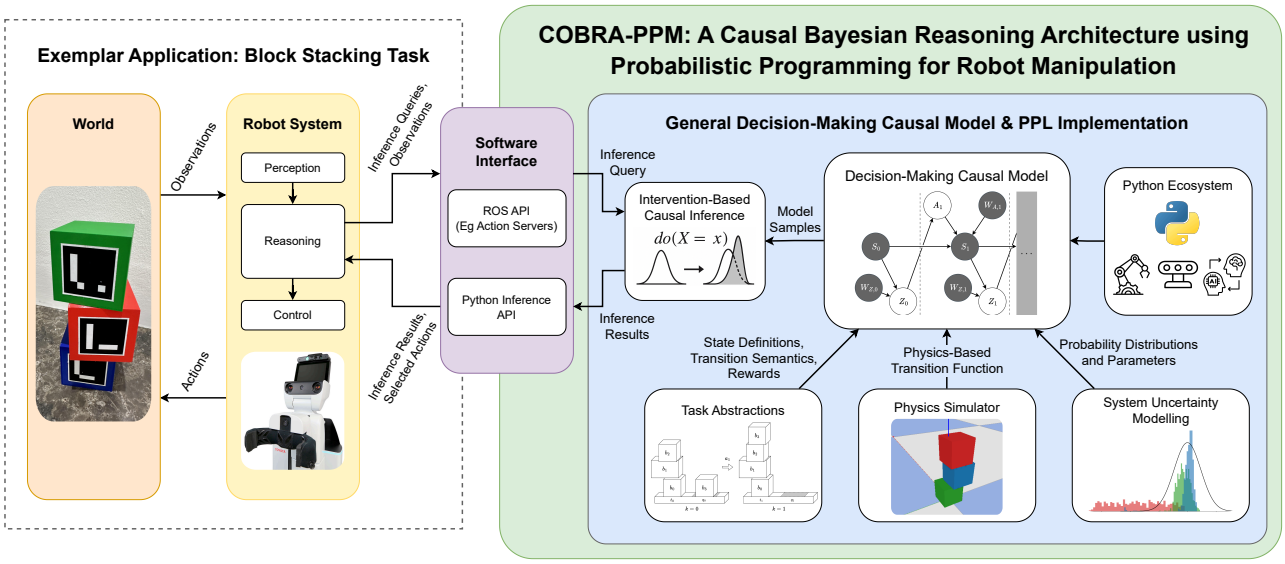


Figure 5.1: COBRA-PPM: our proposed causal reasoning architecture for robot manipulation under uncertainty, combining **causal Bayesian networks** and **probabilistic programming**, and integrated with a full robotic system for an exemplar block-stacking task. The figure illustrates both **generalisable, task-agnostic components** (green box) and **application-specific components**. The generalisable components include a Pyro-based causal model with causal inference methods, integration with the Python ecosystem, and Python/ROS interfaces enabling seamless deployment across diverse robot platforms. The application-specific components demonstrate how the architecture is instantiated for the block-stacking task, including the world environment, robot hardware, and the **flexible composition of CBN subcomponents**: task abstractions, the PyBullet physics simulator, and robot system modelling modules.

despite cumulative uncertainty. We validate the proposed architecture in high-fidelity simulation and on a real domestic robot, demonstrating its predictive accuracy and robustness in block stacking tasks under uncertainty.

The main contributions of this chapter are:

- The formulation of COBRA-PPM, a novel causal reasoning architecture using probabilistic programming for robot manipulation under uncertainty (**Sec. 5.3**);
- The design of software interfaces (Python and ROS) that enable practical integration of the architecture into real and simulated robot systems (**Sec. 5.3.3**);
- A real-world hardware demonstration of our architecture using a domestic service robot (**Sec. 5.7**).

5.3 Causal Bayesian Reasoning Architecture

We present *COBRA-PPM* (Fig. 5.1), a novel architecture for robot manipulation under uncertainty. For the first time, it combines CBN modelling, probabilistic programming, and online physics simulation to support generalisable and extensible decision-making across a wide range of robot morphologies, sensing modalities, and environments.

The architecture bridges low-level perception and control with high-level knowledge and reasoning, enabling the robot to pose questions about **what** it should believe given incomplete knowledge of the world (e.g., *Should I expect the observed block tower to be stable?*), and **how** it should act to achieve its goals (e.g., *Where should I place this block?*).

To illustrate its practical application, Fig. 5.1 shows the integration of the architecture into a mobile robot system for the exemplar block stacking task (Sec. 5.4–5.5).

5.3.1 Robot Sequential Decision-Making Causal Model

At the core of COBRA-PPM is a dynamic CBN that models the robot-world system across discrete time steps. This formulation captures the probabilistic dynamics governing the evolution of system state during robot interaction and enables both prediction and intervention-based inference.

We model the robot sequential decision-making process as a dynamic CBN due to its generality and flexibility. CBNs have been successfully applied to a wide range of agent-based decision-making, planning, and bandit problems, across both fully and partially observable domains and under stochastic or deterministic transitions [7, 27]. The CBN representation is also inherently compatible with many existing planning and reinforcement learning solvers.

Crucially, the model offers a **unified representation** for both sampling and inference. A single model definition supports generative data sampling (e.g., simulation rollouts) and probabilistic inference (e.g., conditioning on observations), enhancing modularity and reuse across tasks and domains.

5.3.1.1 Discrete Time POMDP Model Abstraction

We implement the CBN using a discrete-time partially observable Markov decision process (POMDP) [146], a standard formalism for decision-making under uncertainty [7, 147, 148]. A POMDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, T, \mathcal{Z}, O \rangle$, where $\mathcal{S}$ is the set of world states, $\mathcal{A}$ is the

set of agent actions, $T(s, a, s')$ defines transition probabilities, $\mathcal{Z}$ is the set of observations, and $O(s', a, z)$ defines observation likelihoods.

The dynamic CBN uses time-indexed variables (e.g., S_t, A_t, Z_t) and models observation noise (W_{Z_t}) and actuation stochasticity (W_{A_t}), as shown in Fig. 5.4. In fully observable domains, it reduces to a Markov decision process (MDP) by omitting Z_t and making A_{t+1} directly dependent on S_t.

Although we use a POMDP abstraction in our exemplar task, the model can represent any Python-computable probabilistic program. Inference variables must be defined using Pyro *sample* statements, but this imposes no practical constraint. Pyro supports a large collection of common distributions (e.g., Gaussian, Categorical), hierarchical structures via plate notation, and also allows users to define custom distributions as needed. This enables the architecture to flexibly support a wide range of robot morphologies, actuation and sensing modalities, and domain-specific stochastic processes.

5.3.1.2 Unlocking the Power of the Python Ecosystem

We implement the robot decision-making model as a generative probabilistic program in Pyro, a probabilistic programming language (PPL) built on Python. This enables integration with a range of scientific and robotics packages, allowing the construction of causal models using existing libraries.

In our exemplar task, the model leverages the PyBullet simulator during inference-time sampling to evaluate candidate actions and task outcomes. Beyond physics, Python provides access to scientific libraries for tasks such as signal modelling (e.g., for lidar, image, or wireless signal propagation), object classification, human pose estimation, and cognitive models for human-robot interaction (e.g., theory of mind or intent estimation). These components can be embedded directly into the causal model to support reasoning over perceptual, social, and contextual variables.

The architecture also supports pre-trained models via Python-based ML libraries. These include large language models (LLMs) for inferring symbolic goals from instructions, and multimodal models that produce latent variables (e.g., scene or intent embeddings) to condition simulation rollouts. Such components enhance causal reasoning in tasks requiring language, perception, or contextual understanding.

5.3.2 Intervention-Based Causal Inference

A second key component of our architecture is intervention-based causal inference, used to answer prediction and action-selection queries. Representing the model in Pyro enables both exact and approximate inference methods, including sample-based techniques such as importance sampling [48] and gradient-based approaches like stochastic variational inference (SVI) [49]. Pyro also permits flexible composition of conditioning statements with interventions via the $do(\cdot)$ operator, allowing estimation of interventional transition posteriors conditioned on robot observations. These posteriors support predictive queries over current and future task states for decision-making.

5.3.2.1 Operational Interpretation of Interventions.

Interventions are applied to the action variable A_k and correspond directly to candidate robot actions drawn from the action space $\mathcal{A}$ (see Sec. 5.4.1). That is, evaluating $do(A_k = a)$ corresponds to evaluating the causal effect of executing a specific action $a \in \mathcal{A}$ within the model. The intervention values a are selected from this action space based on the task formulation — for example, by evaluating candidate block placements defined over a discretised (x, y) grid (see Sec. 5.4.3.2).

The choice of intervention target (i.e., intervening on A_k) is determined by the structure of the causal graph, specifically the POMDP-based formulation of the dynamic CBN (see Sec. 5.4.2.1), in which actions are modelled as externally set variables under the robot's control. In contrast to the use of interventions in Sec. 4.4.1 to remove confounding bias, here the *do*-operator is used to represent deliberate action selection — enabling evaluation of the causal effect of candidate actions on future states. While in principle action choice could be represented via conditioning, treating actions as interventions aligns with the standard formulation of decision-making in causal models and ensures compatibility with counterfactual reasoning at higher levels of the causal hierarchy (see Ch. 6). Interventions are therefore not arbitrary manipulations of internal variables, but are grounded in the robot's actionable decision space — linking the abstract notion of intervention to physically executable actions and enabling direct use in robot decision-making.

Without intervention, the model represents an observational regime in which actions follow a default policy (e.g., uniform sampling over $\mathcal{A}$). In our experiments, however, the model is always

queried under intervention. For prediction tasks, this corresponds to a no-op intervention, while for decision-making tasks, interventions are applied over candidate actions to evaluate and select the greedy next-best action.

5.3.3 Software Interface for Hardware Integration

To maximise the benefit to the robotics research community, our reasoning architecture includes software interfaces that expose the intervention-based causal inference functionality through both a pure Python API and a ROS action server API (Fig. 5.1). In Section 5.4, we demonstrate how the ROS interface integrates into a mobile robot system for the exemplar block stacking task.

5.3.4 Operational Pipeline and Execution Flow

While the preceding sections describe the structural components of COBRA-PPM, it is also important to clarify how these components interact during online robot operation. In particular, practical deployment requires careful handling of temporal consistency, perception–action coupling, and repeated interaction with a dynamic environment.

Figure 5.2 illustrates the end-to-end operational pipeline used in our implementation.

At each decision step, the system begins by acquiring observations from onboard sensors and estimating the current state of the environment. Crucially, this state estimate is *latched* prior to inference, ensuring that all downstream reasoning operates on a fixed and internally consistent representation of the world. This avoids inconsistencies that can arise if perception continues to update while inference or planning is underway.

Given the latched state S_k, the CBN is queried under intervention on the action variable A_k to evaluate candidate actions. This involves performing probabilistic inference to estimate the distribution over resulting states S_{k+1} under each intervention $do(A_k = a)$, enabling comparison of expected outcomes.

The selected action is then executed via the robot control stack (e.g., motion planning and low-level control). Execution results in changes to the environment, which may include object motion, contact dynamics, or task-relevant events such as block instability or collapse.

Following execution, the system re-acquires observations and updates its state estimate, effectively polling the environment at each time step. This repeated perception–inference–action loop enables the system to adapt online to changes in the environment, including unexpected

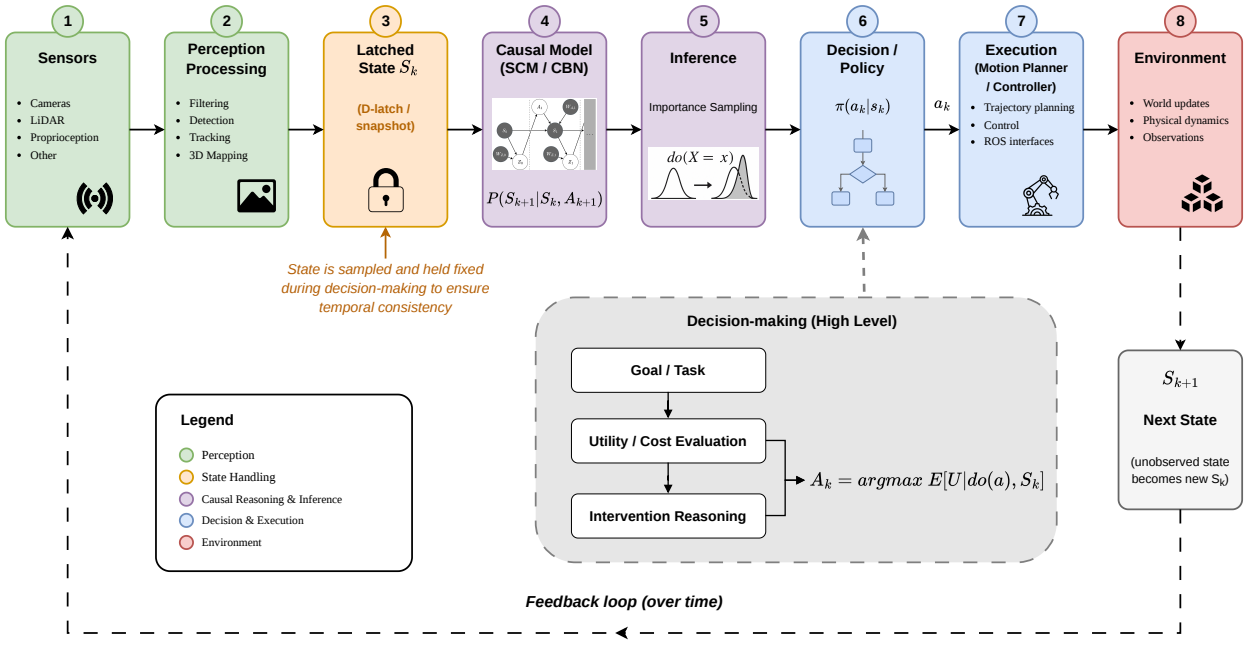


Figure 5.2: Causal decision-making and execution pipeline in COBRA-PPM. The diagram shows the flow of information during online operation, from sensing through to action execution and environment feedback. Perceptual inputs are processed to form a state estimate, which is then latched to produce a temporally consistent snapshot prior to inference and decision-making. A causal Bayesian network (CBN) defines the transition dynamics and supports intervention-based inference over future states. Actions are selected based on expected utility under intervention and executed through the robot motion planning and control stack. The environment evolves in response to the executed action, and new observations are acquired, forming a closed-loop process over time.

disturbances or deviations from predicted outcomes. This is particularly important in scenarios where the environment may change due to external disturbances (e.g., when a block to be placed is moved between decision steps, or when the configuration of the tower shifts due to gravity or human intervention), requiring the system to re-estimate the state and make decisions independently at each step.

Importantly, this operational pipeline bridges the abstract causal model with real-world robotic execution, ensuring that intervention-based reasoning remains grounded in physically observable system dynamics.

5.4 Exemplar Block Stacking Task

To demonstrate the practical application of our reasoning architecture, we apply it to an exemplar manipulation task: sequential block stacking (Fig. 5.14, 5.3). The robot incrementally builds a tower from an initial configuration and a queue of blocks, using noisy sensor observations and stochastic placement actions. The task is successful if the tower remains

standing after the final placement and a failure if it topples at any point. A key challenge is managing uncertainty in sensing, state estimation, and control, all of which must be accounted for to ensure reliable execution.

As our focus is on a generalisable formulation for probabilistic prediction and action selection — rather than on planning methods and search — we restrict the task to single-column towers and frame it as a sequence of independent next-best action selection problems. This allows us to demonstrate the reasoning capabilities of the model without introducing additional planning complexity. Nevertheless, the model can be extended with a (PO)MDP planner to support trajectory-level optimisation when required [7].

5.4.1 Problem Definition

Formally, the robot's task is to construct a stable block tower by sequentially placing one block at each discrete time step $k = 1, \ldots, K$, where K is the total number of additional blocks to be placed. The task state at time k, denoted s_k, consists of the current tower configuration t_k and the queue of remaining blocks q_k: $s_k = t_k \cup q_k$.

We formulate the greedy next-best placement problem as selecting an approximately optimal action $\hat{a}_k$ from the set of candidates $\mathcal{A}$. Each action $a = (x, y)$ corresponds to placing the next block at a position on top of the current tower. The goal is to select the action that maximises the probability that the resulting state s_k is stable, conditioned on the previous (latent) state s_{k-1} and its noisy observation z_{k-1}:

$$\hat{a}_k = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \; P\left(\text{IsStable}(s_k) \mid z_{k-1},\ a\right). \tag{5.1}$$

The variables and decisions required for this task are defined by the joint design of the state representation, action space, and causal model; we refer the reader to Sec. 5.4.2.1, Sec. 5.4.2.5, and Sec. 5.4.3.2 for detailed specification of these components.

5.4.2 Exemplar Task Decision-Making Causal Model

5.4.2.1 Exemplar Task Causal DAG

Following the decision-making causal model formulation of our architecture (Section 5.3), we model the K-action block stacking task as a dynamic CBN. A dynamic causal DAG for a two-action ($K = 2$) block stacking task is shown in Fig. 5.4.

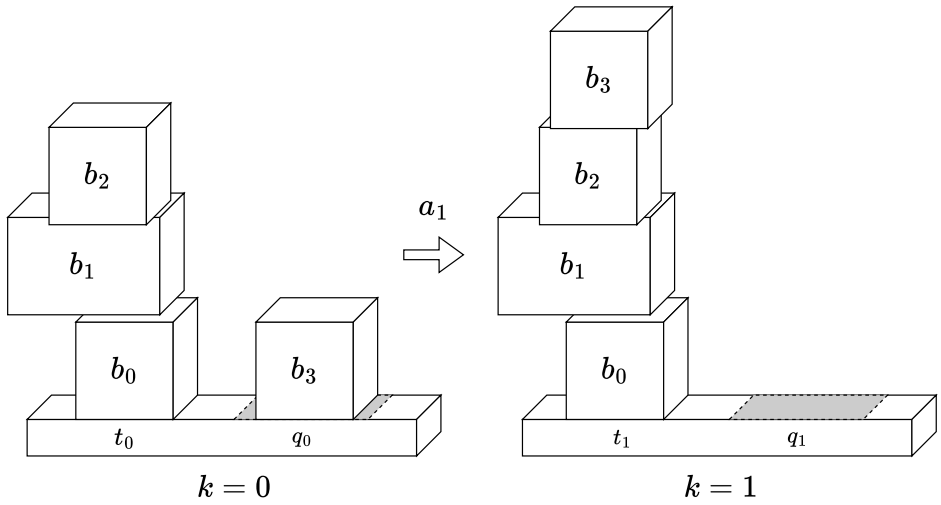


Figure 5.3: Schematic illustration of the exemplar block stacking task. At each time step k, the robot selects action A_k to place block b from queue q_{k-1} onto previous tower t_{k-1}, resulting in the new state (t_k, q_k). The goal is to incrementally construct a stable tower through sequential placements.

To ground the model in our task requirements, our exemplar causal model captures two main sources of uncertainty in the robot–task system: observation noise $W_{Z,k}$ and actuation noise $W_{A,k}$ at each time step k, which perturb the robot's environment observation Z_k and execution of action A_k, respectively.

The ground-truth state S_k is not directly observable by the robot; instead, it must rely on its noisy environment observation Z_k, corrupted by $W_{Z,k}$. Similarly, the execution of a chosen action A_k does not deterministically result in the intended block placement. Rather, it is stochastically perturbed by the actuation noise variable $W_{A,k}$, which captures cumulative error from perception, state estimation, and control imperfections in the mobile robot system.

This causal DAG formulation enables principled reasoning under uncertainty by explicitly representing the stochastic dependencies that govern how noisy observations and perturbed actions influence task outcomes over time.

5.4.2.2 Task State Representation

The task state at time k, denoted s_k, consists of the current tower configuration t_k and the queue of remaining blocks q_k: $s_k = t_k \cup q_k$. An example showing the initial state s_0 and resulting state s_1 after action a_1 is illustrated in Fig. 5.3. The tower state t_k is represented as an ordered list of block references: $t_k = [b_{n_i,k}]$ for $i = 0, \ldots, I_k$, where n_i is the unique identifier of the block at position i in the tower, and $I_k + 1$ is the number of blocks in the tower at time k. Similarly, the queue state is: $q_k = [b_{n_j,k}]$ for $j = 0, \ldots, J_k$, where n_j is the unique identifier of the block at

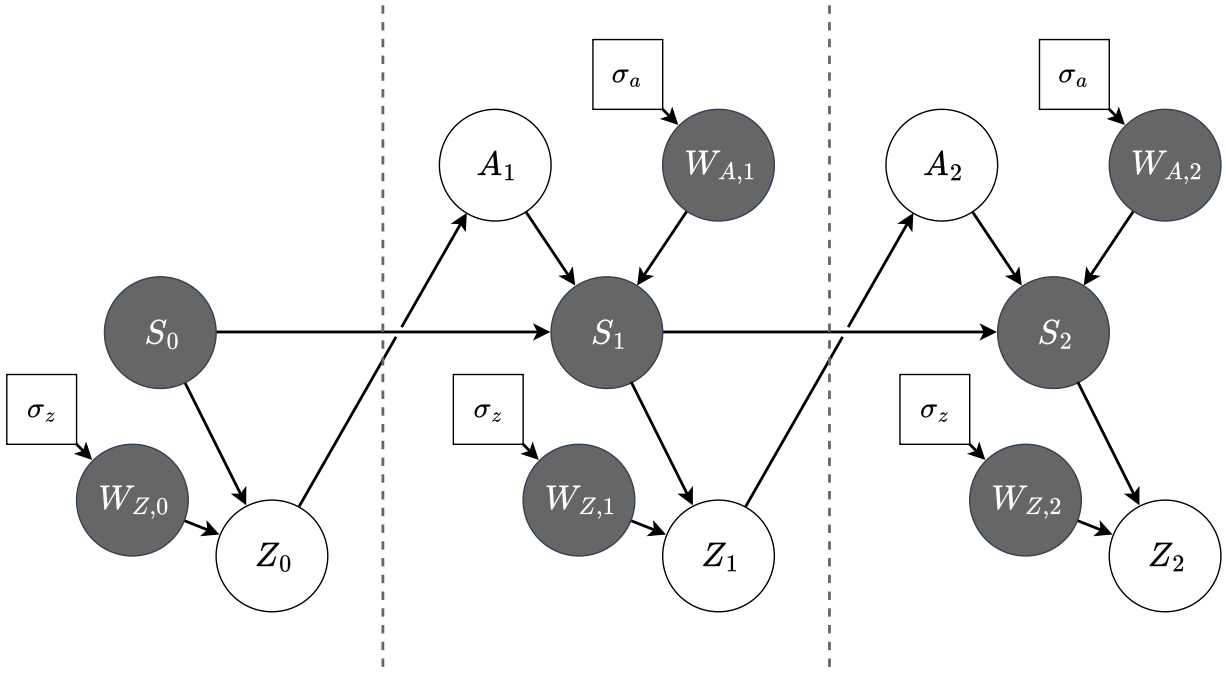


Figure 5.4: A dynamic causal Bayesian network (CBN) used as the decision-making model within our reasoning architecture (Fig. 5.1), illustrated for a two-action block-stacking task (Sec. 5.4). The CBN encodes a POMDP structure tailored to the task, capturing domain-specific causal dependencies and probabilistic uncertainty. At each time step k, the robot selects a stochastic action A_k based on a noisy observation Z_{k-1} of the previous state. The action is executed with noise $W_{A,k}$, leading to a transition from the latent state S_{k-1} to S_k, followed by a new noisy observation Z_k subject to $W_{Z,k}$. Observed variables are shown as white nodes, unobserved variables as grey nodes, and fixed model parameters as squares.

position j in the queue, and $J_k + 1$ is the number of remaining blocks in the queue. Each block state $b_{n,k}$ encodes the physical attributes of block n at time k, including its 6-DOF pose (position and orientation), 3D dimensions, and mass. We note that additional block attributes — such as inertia, coefficient of friction, or coefficient of restitution — can be trivially incorporated into the block state vector $b_{n,k}$ if required by the physics simulation or task reasoning.

State Modelling Assumptions. We note that the robot state is absent from our task state abstraction. This is because we assume that the task–world causal system is independent of the robot state; that is, it is conditionally dependent only on the task state and robot action. To permit this, we assume that the noise and error variables W_Z and W_A — and thus the action-choice and state update variables — are conditionally independent of the robot state. This is reflected in the causal DAG structure shown in Fig. 5.4. While these assumptions are almost surely violated in the physical robot–block–tower system we are modelling, we assume that the resulting errors are minor and do not significantly affect the overall accuracy of the robot–

task–world model. To further mitigate the impact of this modelling error on the implemented system's performance, we restrict the robot's viewpoint at decision time to a designated 'neutral' position in front of the task workspace. This position was empirically selected to ensure that the blocks remain within the robot's sensor field of view, and that the associated measurement and actuation errors are largely independent of robot state.

5.4.2.3 Action Representation

Each action $a \in \mathcal{A}$ is defined as a 2D placement coordinate $a = (x, y)$, specifying where the next block is to be placed on top of the current tower. Similar to the state representation, the action definition can be arbitrarily expanded with additional elements to support alternative action parameterisations and task extensions if required for reasoning — for example, incorporating block placement orientations, unordered block selection, or non-single-column block towers.

5.4.2.4 Transition Function Representation

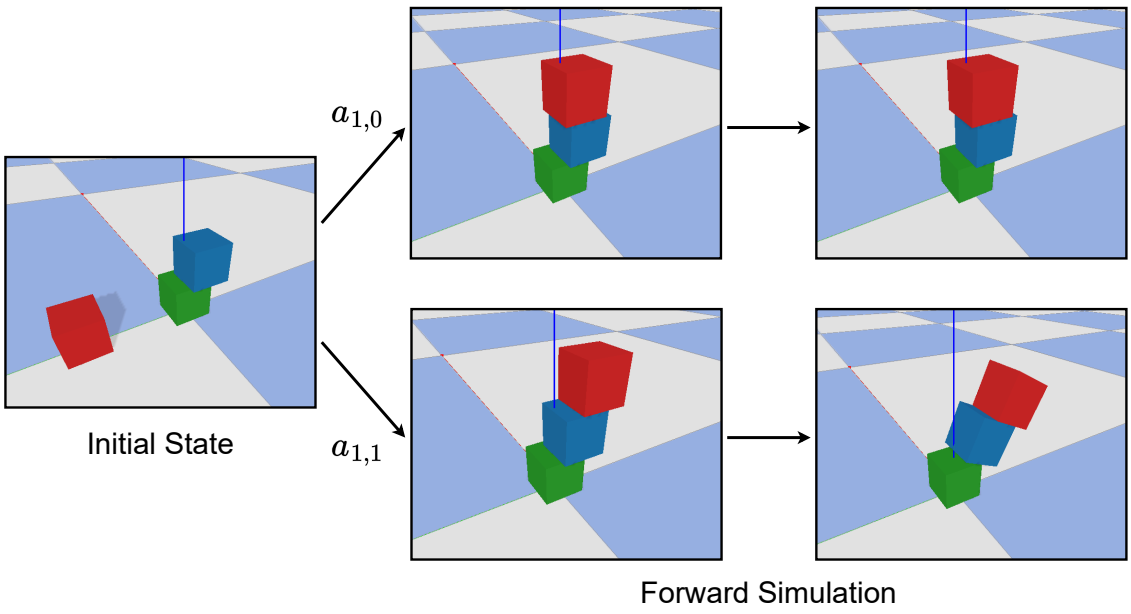


Figure 5.5: Physics-based PyBullet simulation used in the architecture decision-making causal model to predict successor state tower stability probability.

We model the state transition function T following the standard POMDP formulation [146], adopting a causal perspective by treating the agent's action as an intervention using the $do(\cdot)$ operator. This allows A_k to be interpreted as a direct manipulation of the system, marginalising

over execution noise $W_{A,k}$:

$$T(S_{k-1}, A_k, S_k) = P\Big(S_k \mid S_{k-1}, do(A_k = a_k)\Big). \tag{5.2}$$

To generate the successor task state s_k from the transition distribution T, we use the PyBullet 3D physics simulator [149] to sample the next tower state by simulating the 6-DOF rigid body dynamics of the block placement, conditioned on the agent's noisy observation Z_{k-1} of the previous state. The remaining components of the task state, such as the block queue, are updated deterministically based on the known action context.

The PyBullet simulator is used to forward-simulate the evolution of the physical system starting from the moment the block is placed onto the tower (i.e., after it is released by the robot), until either a predefined testing period has elapsed or the system detects that the block and/or tower has fallen, whichever occurs first. Fig. 5.5 illustrates two such forward simulations. Upon termination of the forward simulation, the final state of the tower in PyBullet is assigned as the sampled successor state s_k.

To assess whether the resulting tower configuration remains stable, we apply a stability check that compares the final vertical (z-axis) position of the top block to its position at the time of release by the robot gripper. If the displacement exceeds a small threshold, the tower is considered unstable during the simulation period.

This approach illustrates how physics-based subcomponents can be seamlessly incorporated into the broader architecture to support more complex forms of physical reasoning — a key feature of our modular causal modelling framework.

5.4.2.5 Noise & Error Modelling

We model both observation and manipulation errors as zero-mean, isotropic Gaussian noise in 3D. Specifically, the observation noise W_Z and the action execution noise W_A are modelled as independent random vectors sampled from $\mathcal{N}(0, \sigma_Z^2 I)$ and $\mathcal{N}(0, \sigma_A^2 I)$, respectively. Here, σ_Z and σ_A represent the standard deviations of uncertainty in perception and control, respectively, arising from sensor noise, state estimation error, and actuation imprecision. All noise components are assumed to be independent across the x-, y-, and z-axes and temporally independent over time steps.

The observation Z_t is defined as a noisy measurement of the latent world state S_t, via the additive noise model:

$$Z_t \coloneqq S_t + W_{Z,t} \quad . \tag{5.3}$$

The state transition incorporates manipulation noise during execution of action A_t as:

$$S_t \coloneqq f(S_{t-1}, A_t, W_{A,t}), \tag{5.4}$$

where $f(\cdot)$ denotes the deterministic transition function implemented by the physics-based simulator.

These noise variables W_Z and W_A correspond to the *exogenous* sources of uncertainty in the causal DAG introduced in Section 5.4.2.1 (see Fig. 5.4), meaning they capture all randomness arising from factors outside the system being explicitly modelled.

Distribution Choice Motivation. The noise model parameters σ_Z and σ_A are estimated empirically from collected data. For observation noise, we compute deviations between measured block poses and their corresponding ground truth simulator states; for action noise, we compute deviations between intended and realised block placements in simulation. We estimate axis-wise standard deviations and use their average as a point estimate for a shared isotropic variance parameter (see Table 5.1).

We adopt a zero-mean Gaussian noise model based on standard assumptions commonly used in robotics and probabilistic modelling: errors are expected to be approximately symmetric (i.e., no systematic bias in direction), unimodal with a peak around zero error, and exhibit a gradual decay towards the tails. This provides a reasonable first-order approximation of uncertainty and serves as a pragmatic baseline model. Model complexity is only increased if required by poor empirical fit or degradation in downstream predictive performance.

For simplicity, we further assume isotropic noise, using a single variance parameter shared across the x-, y-, and z-axes. This assumes that uncertainty is approximately similar across spatial dimensions, which is a reasonable approximation in our controlled setting. This assumption is coupled with a fixed-viewpoint data collection setup, in which the camera pose remains constant and the world coordinate frame is aligned with the image plane and depth axes. Under changing viewpoints, this alignment would no longer hold, and a more general observation model would be required to account for projection effects.

However, as shown in Fig. 5.9, the isotropic assumption does not strictly hold in practice: in particular, the depth-aligned axis exhibits higher variance due to the inherent ambiguity of monocular depth estimation from image observations.

More expressive noise models — such as anisotropic Gaussians with axis-specific or full covariance structure, non-zero mean models, or non-Gaussian distributions (e.g., skewed or heavy-tailed) — could better capture these effects. Extending the model to mobile or multi-view robotic settings would additionally require incorporating the robot's 3D pose (position and orientation) into the causal model, introducing further complexity in modelling, training, sampling, and inference. However, these extensions would increase model complexity and parameter estimation requirements, with limited impact on downstream performance in our experiments. We therefore retain the isotropic Gaussian assumption as a deliberate trade-off between fidelity and tractability. Further analysis of axis-dependent error characteristics and their impact on model performance is discussed in Sec. 5.6.

5.4.3 Evaluation Tasks

We evaluate our architecture on two tasks: (1) tower stability prediction, and (2) greedy next-best action selection.

5.4.3.1 Task 1: Tower Stability Prediction

We frame tower stability prediction as a binary classification problem: estimating the probability that a perceived tower configuration remains stable in the successor state. We use our causal model to estimate the query $\Phi_{stable,k}$, the probability that the unobserved true tower state S_{k+1} remains stable, conditioned on the robot's noisy observation z_k: $\Phi_{stable,k} = P(\text{IsStable}(S_{k+1}) = \text{True} \mid z_k)$. We apply a threshold value $\tau_{stable,Z}$ to convert the probability into a binary decision.

The threshold $\tau_{stable,Z}$ acts as a tuning parameter representing tolerance to uncertainty from perception noise. Systems with higher measurement error may require a more conservative threshold to reduce false positives. In safety-critical tasks such as tower construction or fault detection, minimising false positives is crucial. The cost of misclassifying an unstable tower may far exceed that of a missed opportunity, e.g., a block falling onto a nearby human.

5.4.3.2 Task 2: Greedy Next-Best Action Selection

We pose greedy next-best action selection as an optimisation problem: finding the action that maximises the predicted probability of the tower remaining stable after block placement.

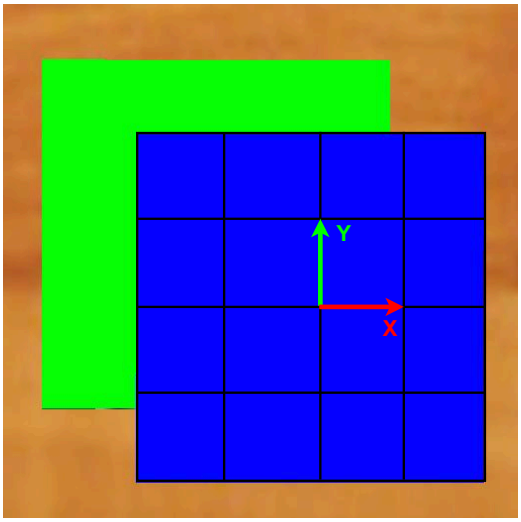


Figure 5.6: A top-down view of a block tower in Gazebo illustrating how the candidate action positions (x, y) are uniformly sampled from a grid spanning the top of the current top block (blue), in the local coordinate frame located at the face centre. A 4×4 grid is shown here for illustrative purposes.

We begin by generating candidate actions $a = (x, y)$ via uniform sampling from an $L \times W$ grid over the top face of the current block. Candidate placement locations are expressed in the local coordinate frame defined by the centre of the top face of the current top block, aligned with the top block's orientation as shown in Fig. 5.6. The height of the next block is always chosen such that it is placed on the top surface of the previous block, and thus is omitted from the action representation.

For each candidate, we compute the intervention-based inference query $\Phi_{stable,k,a}$ using our CBN model, estimating the probability that the tower remains stable in the successor state S_k, conditioned on the robot's observation z_{k-1} and the intervention $do(A_k = a_k)$.

We define the filtered action set $\mathcal{A}_\tau$ as those candidates whose predicted stability probability exceeds the minimum threshold $\tau_{stable,A}$. The most stable action is then selected:

$$a_k^* = \underset{a_k \in \mathcal{A}_\tau}{\operatorname{argmax}} \; P\Big(\text{IsStable}(S_k) = \text{True} \mid do(A_k = a_k),\; z_{k-1}\Big). \tag{5.5}$$

We define the stable set $\mathcal{A}_{stable}$ as actions from $\mathcal{A}_\tau$ that are within a probability margin $\tau_{cluster}$ of $p_{a_k^*}$:

$$\mathcal{A}_{stable} = \{a \in \mathcal{A}_\tau : |p_{a^*} - p_a| \leq \tau_{cluster}\}. \tag{5.6}$$

We assume the stable set $\mathcal{A}_{stable}$ forms a convex hull approximating the region of stable placements. We compute the geometric mean of the (x, y) positions of actions in this set and select the centroid as the next-best action a_k. The post-placement stability threshold $\tau_{stable,A}$ defines the minimum acceptable probability for action success. The cluster threshold $\tau_{cluster}$ controls how close an action must be to a_k^* to be included in the geometric mean. Like $\tau_{stable,Z}$, both should be tuned based on the application's risk profile.

Risk-Sensitive Decision Criteria. In this work, action selection is based on maximising the predicted probability of success, which corresponds to optimising the expected outcome under a binary stability variable. This is appropriate for the block stacking task considered here, where failures have relatively low cost and the objective is to maximise overall success rate.

However, in higher-risk applications — such as those involving potential harm to humans, damage to equipment, or safety-critical system failures — expectation-based decision rules may be insufficient. In such settings, risk-sensitive criteria that explicitly account for worst-case or tail outcomes may be more appropriate.

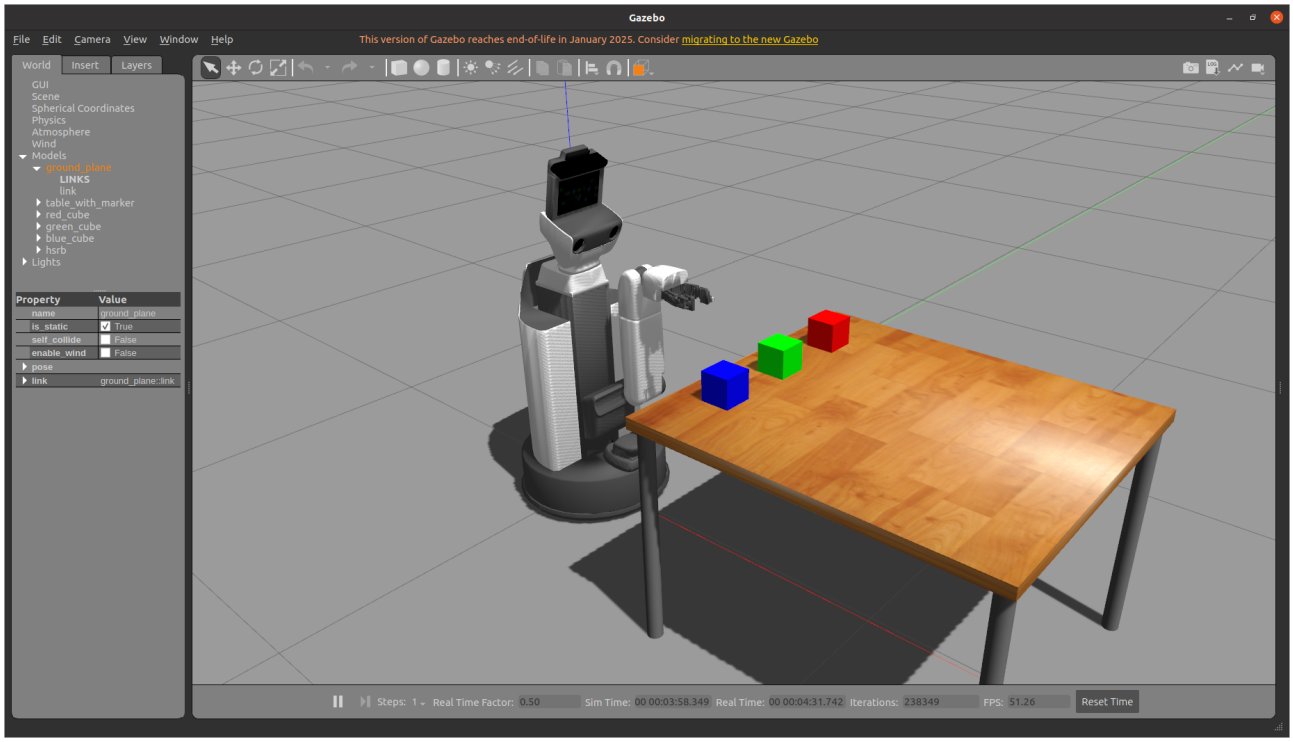
Our architecture supports alternative decision rules operating on the inferred posteriors, such as conditional value-at-risk (CVaR) [150], which focuses on the expected outcome in the worst-case tail of the distribution. Such approaches enable more conservative decision-making by prioritising worst-case robustness over average-case performance, reflecting asymmetric outcome costs in which negative events incur substantially higher penalties than the rewards associated with positive outcomes.

5.5 High-Fidelity Gazebo Robot Simulation Evaluation

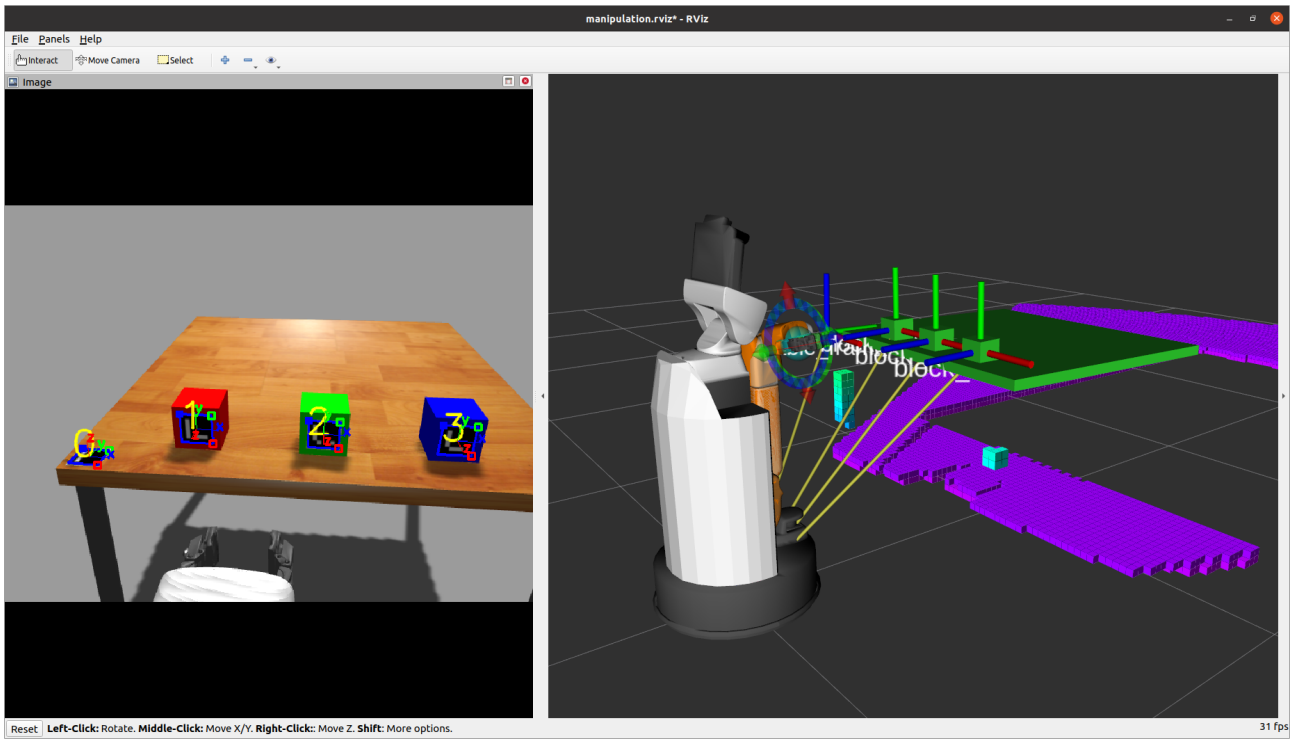
5.5.1 Experimentation Setup

5.5.1.1 Robot Simulation Environment

As shown in Figs. 5.7 and 5.8, we evaluate our architecture on the exemplar block stacking task using a simulated Toyota Human Support Robot (HSR) [50] in Gazebo [51]. The environment uses three 7.5 cm cube blocks, matching those used in the real-world demonstration (Fig. 5.14). To simulate realistic block position estimation, ArUco markers [151] are attached to each block, and 6-DOF pose observations are generated from simulated RGB-D

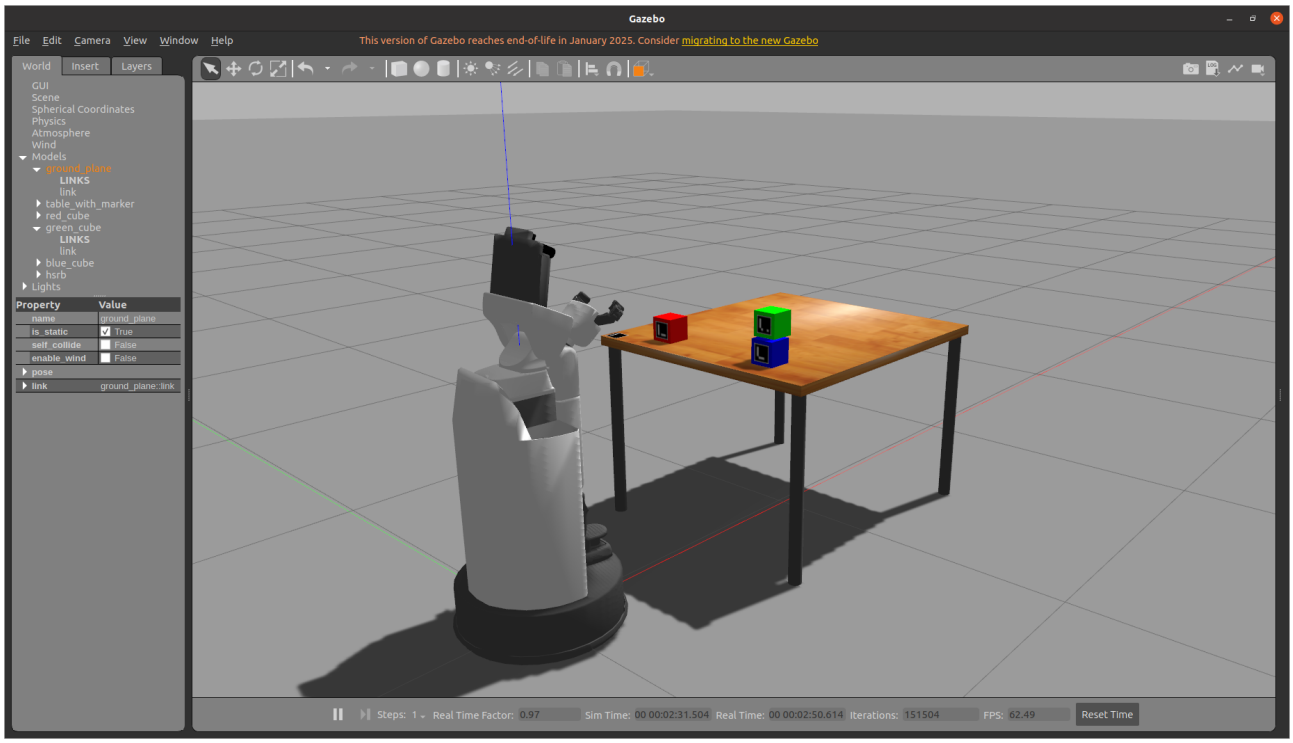


(a)

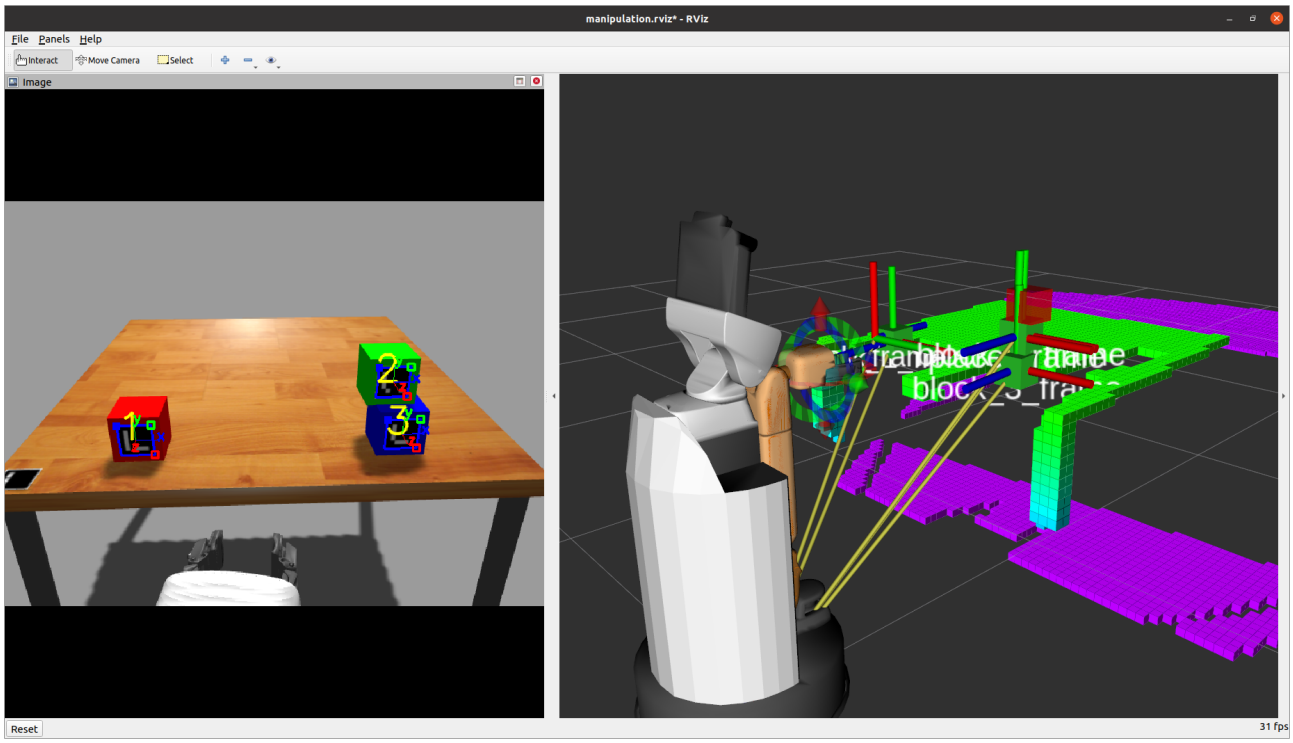


(b)

Figure 5.7: An illustration of the task-specific components used in the high-fidelity simulation experiments: (a) the physics-based Gazebo robot simulation environment, and (b) RViz visualisation of robot image-based block pose estimation (LHS) and 3D scene reconstruction (RHS). We use a Toyota Human Support Robot (HSR) simulation model and three 7.5 cm cube blocks. To simulate realistic robot hardware, we use ArUco markers for perception and ROS MoveIt! for motion plans to execute pick-and-place actions.



(a)



(b)

Figure 5.8: An illustrative problem instance of the block stacking task in simulation where the robot must place the red block onto the existing two-block tower to form a stable tower, as viewed from: (a) the Gazebo simulation environment, and (b) the RViz visualisation of block pose estimation (LHS) and 3D scene reconstruction (RHS). The target block placement position, as selected by our architecture, is indicated by the red cube in the 3D scene reconstruction window.

data using the OpenCV ArUco module. Pick-and-place actions are executed via motion plans generated by ROS MoveIt! [52].

5.5.1.2 Causal Modelling & Inference

We implement the robot–task–world system as a causal generative model expressed as a probabilistic program in Pyro, written in Python. The model invokes the PyBullet simulation API during execution to set up and run physics-based simulations online and access the resulting world state. Each model sample involves forward-simulating the tower dynamics in PyBullet for approximately 2 seconds of simulation time (500 frames at 240 fps), terminating early if the tower falls to reduce computational cost. To support deployment in both simulation and real-world settings, we integrate the tower stability and next-best action selection methods (see Section 5.3) into the robot system via a ROS Action Server API, as illustrated in Fig. 5.1.

5.5.2 Task 1: Tower Stability Prediction

5.5.2.1 Block Pose Estimation Error Characterisation

To characterise the error in the robot's block position estimation, we generate a dataset of 250 randomly sampled 3-block tower configurations in Gazebo simulation. For each configuration, we record the 3D positions of all blocks as observed by the robot, along with their ground-truth positions in the simulation environment. This yields a total of $N = 750$ block pose samples (3 blocks per configuration), from which position errors are computed. The estimation error is quantified by computing the standard deviation of the position error along each axis: $\sigma_{Z,x}$, $\sigma_{Z,y}$, and $\sigma_{Z,z}$. Empirically, we find that using the average of these values provides a sufficiently accurate approximation for a shared standard deviation σ_Z, applied across all axes in the zero-mean, isotropic 3D Gaussian noise model (see Sec. 5.4.2.5).

5.5.2.2 Tower Stability Classification Accuracy

We generate a dataset of 1000 randomly sampled 3-block tower configurations in simulation to evaluate our model's classification accuracy. Ground-truth stability labels are assigned by forward-simulating each tower configuration in the Gazebo physics engine and observing whether the tower remains standing. We use the Importance Sampling inference method in Pyro, drawing 50 samples per query, to estimate tower stability based on our exemplar task decision-making causal model (see Sec. 5.4.2) and the previously estimated value of σ_Z. Binary

classifications are produced by thresholding the predicted stability probability, and evaluated across a range of threshold values. To assess performance, we compute standard classification metrics — F_1 score, AUC, precision, and recall — across a range of thresholds.

5.5.3 Task 2: Greedy Next-Best Action Selection

5.5.3.1 Block Placement Error Characterisation

To characterise block placement error using the robot's manipulation subsystem, we generate a dataset of 25 randomly sampled 2-block tower configurations in Gazebo. Each defines the initial tower state for a placement trial. For each configuration, the robot performs 10 independent attempts to place an additional block at a predefined target position, resulting in 250 placement experiments. After each attempt, the placement error (i.e., deviation from the intended location) is recorded. Placement error is quantified by computing the standard deviation along each axis: $\sigma_{A,x}$, $\sigma_{A,y}$, and $\sigma_{A,z}$. We find that the average of these values provides a sufficiently accurate approximation for a shared standard deviation σ_A, applied across all axes in the zero-mean, isotropic 3D Gaussian model representing manipulation uncertainty.

5.5.3.2 Greedy Next-Best Action Selection Performance

To evaluate system performance on the greedy next-best action selection task in simulation, we generate a test dataset of 50 randomly sampled initial 2-block tower configurations. For each configuration, we use our action selection method and causal model to choose a placement position for a third block (i.e., a single-action task with $K = 1$). Candidate actions are sampled from a uniform 5×5 grid over the top face of the current block. Inference is done using Pyro's Importance Sampling with 50 samples per query. Each selected action is executed in 10 independent trials, yielding 500 total experiments. These repetitions are used to compute empirical success rates. To mitigate against extraneous effects caused by failures associated with robot sub-systems that are out-of-scope for this evaluation (e.g., robot base control, sample-based motion planning), we implement a simulation experiment supervisor program to detect failures outside of the scope of this evaluation and trigger a reset of the simulated experiment episode upon failure detection.

Threshold Selection. The threshold values $\tau_{stable,Z}$, $\tau_{stable,A}$, and $\tau_{cluster}$ are selected empirically based on model performance across validation data. For tower stability prediction, $\tau_{stable,Z}$ is chosen using standard classification analysis (see Fig. 5.10), balancing precision and recall according to the task's safety requirements. For action selection, $\tau_{stable,A}$ is set to a high value (0.8) to enforce conservative, high-confidence placements, reflecting the asymmetric cost of failure in manipulation tasks. The clustering threshold $\tau_{cluster}$ is chosen to admit a sufficiently large set of near-optimal actions, enabling robust centroid-based selection while avoiding overly aggressive filtering.

These thresholds are therefore not derived from a formal analytical procedure, but are tuned to reflect task-specific risk preferences and empirical performance characteristics. In principle, more formal approaches — such as decision-theoretic optimisation (e.g., expected utility or risk-sensitive criteria such as CVaR) — could be used to derive thresholds systematically. However, in this proof-of-concept setting, empirical selection provides a simple and effective mechanism for calibrating system behaviour.

5.5.3.3 Naïve Baseline Action-Selection Method

We compare our method against a baseline that follows a naïve, heuristic policy: always placing the next block at the centre of the current top block. This simple strategy can often produce stable towers under ideal conditions. However, it lacks any understanding of block physics, system dynamics, or task uncertainty, and is not expected to make robust placement decisions under realistic noise and variability.

5.6 Results & Discussion

We present experimental results for both evaluation tasks, each followed by focused discussion. Broader architectural considerations — including scalability and limitations — are addressed separately in Section 5.8.

5.6.1 Task 1: Tower Stability Prediction

5.6.1.1 Block Pose Estimation Error Characterisation

Fig. 5.9 shows the distribution of block position **measurement** errors along the X-, Y-, and Z-axes, along with the fitted probability density function (PDF) of a zero-mean, isotropic 3D

Gaussian model. Our sensor noise model adopts this fitted distribution, using the average standard deviation σ_Z estimated across axes.

Summarised statistics for the measurement and placement errors are provided in Table 5.1.

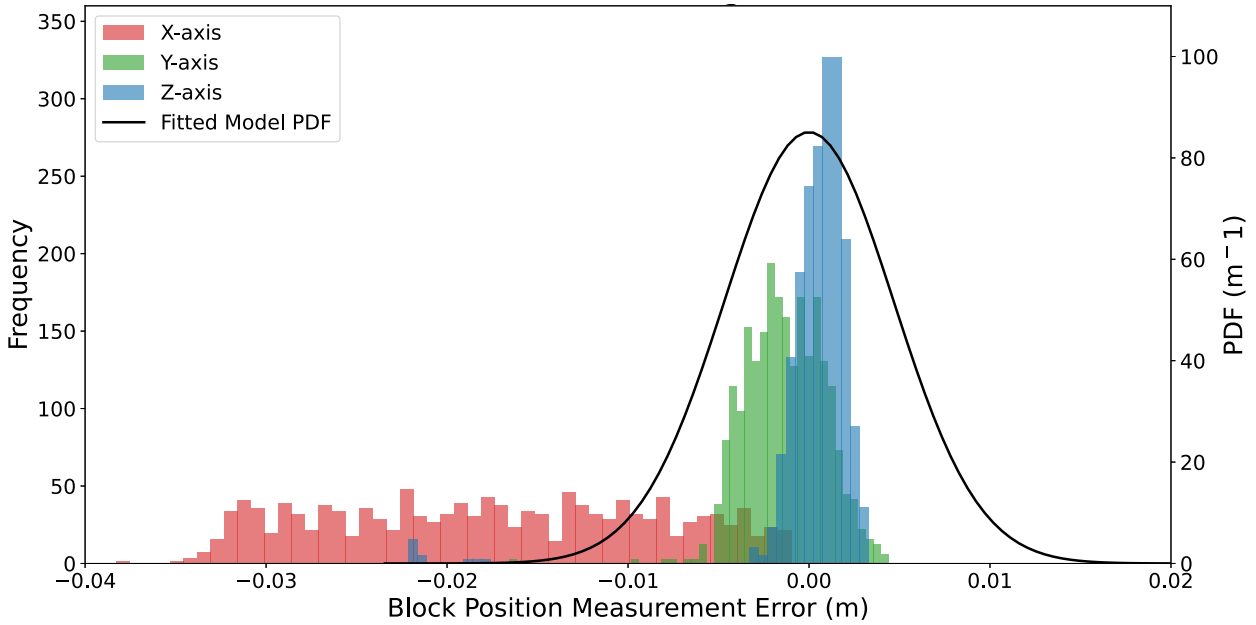


Figure 5.9: Empirical characterisation of block position measurement error. Histograms show the distribution of measurement errors along the X-, Y-, and Z-axes, collected from the simulation dataset (N = 750 samples). The overlaid curve shows the probability density function (PDF) of a zero-mean, isotropic 3D Gaussian model, fitted using the average standard deviation $\sigma_Z = 0.469$ cm estimated across axes. Notably, the X-axis error — approximately aligned with the camera's depth axis — exhibits a non-zero mean and a larger variance than the Y- and Z-axes. Although not fully consistent with the sensor model's assumptions, this deliberate simplification introduces a noticeable approximation error, particularly along the depth-aligned axis. However, this mismatch does not materially affect downstream classification performance and is therefore acceptable within the scope of this model.

Table 5.1: Characterisation of block position measurement and placement errors. Values report standard deviation (in cm) of position error along each axis. Measurement error statistics are computed from N = 750 samples, and placement error statistics from N = 250 samples. The final column reports the mean across axes, denoted σ_*, and used as the model parameter σ_Z or σ_A depending on the error type.

Error Type	**X-axis**	**Y-axis**	**Z-axis**	**Avg. (σ_*)**
Measurement	0.906	0.216	0.284	0.469
Placement	1.790	2.770	0.146	1.570

The data reveal that our initial assumption — that the random error term W_Z is independent of the robot's state and that the standard deviation is shared across axes — does not strictly

hold. In particular, measurement error along the X-axis — approximately aligned with the camera's depth axis — exhibits a non-zero mean and a substantially larger standard deviation than those along the Y- and Z-axes. Despite this modelling error, we do not observe any significant degradation in overall model performance. We therefore retain the simplifying assumption of zero-mean, state-independent noise. The error distributions along each axis ($\sigma_{Z,x}$, $\sigma_{Z,y}$, $\sigma_{Z,z}$) confirm that the measurement noise is approximately Gaussian. We take the average of these values as a sufficiently accurate approximation for a shared standard deviation, setting $\sigma_Z = 0.469$ cm across all axes in our 3D Gaussian noise model.

5.6.1.2 Tower Stability Classification Accuracy

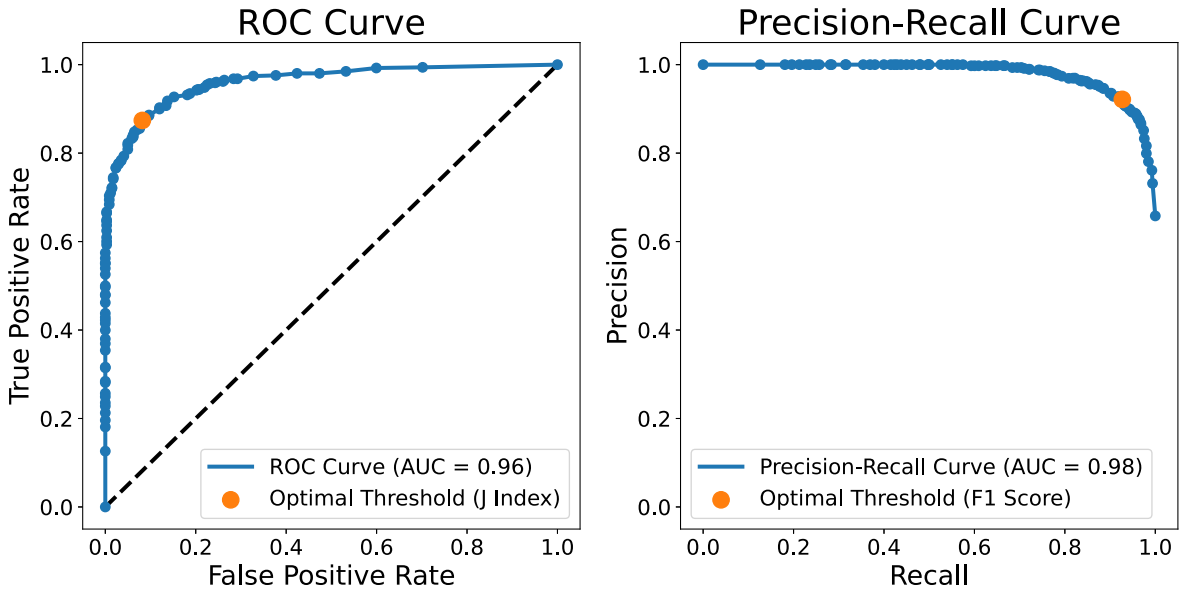


Figure 5.10: Receiver operating characteristic (ROC) and precision-recall (PR) curves for tower stability classification based on model predictions (N = 1000 tower configurations). The curves demonstrate strong overall binary classification performance, with the optimal classification threshold selected using Youden's Index ($\tau_{stable,Z} = 0.4$), corresponding to a precision of 0.955 and recall of 0.868.

The model's binary classification performance on the test set is visualised in Fig. 5.10 and summarised in Table 5.2.

The optimal binary classification threshold was identified as $\tau_{stable,Z} = 0.40$ (see Sec. 5.4.3.1), based on Youden's Index [152] from the ROC curve, and corresponds to a precision of 0.955 and recall of 0.868. Motivated by the safety-critical considerations discussed in Section 5.4.3.1, this threshold was selected for its lower false positive rate compared to the F_1-optimal threshold from the precision-recall curve.

This threshold is selected retrospectively based on evaluation over the test dataset using ROC analysis. As such, its value reflects the data distribution and noise characteristics of the current system, and may not directly transfer to new settings without recalibration. In practice, the choice of $\tau_{stable,Z}$ should be treated as a system-level tuning parameter, adapted to the sensing, actuation, and risk profile of the deployment environment.

These values are close to ideal classification performance and demonstrate the strong predictive ability of our model.

Table 5.2: Tower stability binary classification results evaluated on a dataset of randomly sampled tower configurations (N = 1000), generated independently of parameter estimation, with ground-truth labels obtained via forward simulation in the Gazebo physics engine (see Sec. 5.5.2.2). The model achieves near-ideal classification performance.

Classification Metric	Prediction Accuracy	F1 Score	Precision	Recall	AUC Score
Score (↑)	**88.6%**	0.909	0.955	0.868	**0.961**

5.6.2 Task 2: Greedy Next-Best Action Selection

5.6.2.1 Block Placement Error Characterisation

Fig. 5.11 shows the distribution of block **placement** errors along the X-, Y-, and Z-axes, along with the fitted probability density function (PDF) of a zero-mean, isotropic 3D Gaussian model.

As previously mentioned, summarised statistics for both measurement and placement errors are provided in Table 5.1.

As with measurement error, placement error along the X- and Y-axes exhibits substantially larger standard deviations than along the Z-axis. The largest error occurs along the Y-axis, though this directional bias is not explicitly modelled in our isotropic noise formulation. Despite this anisotropy, the simplifying assumption of zero-mean, isotropic noise does not significantly degrade downstream performance. We therefore retain this assumption and take the mean of the axis-specific values to define a shared standard deviation of $\sigma_A = 1.57$ cm, representing manipulation noise in our Gaussian error model.

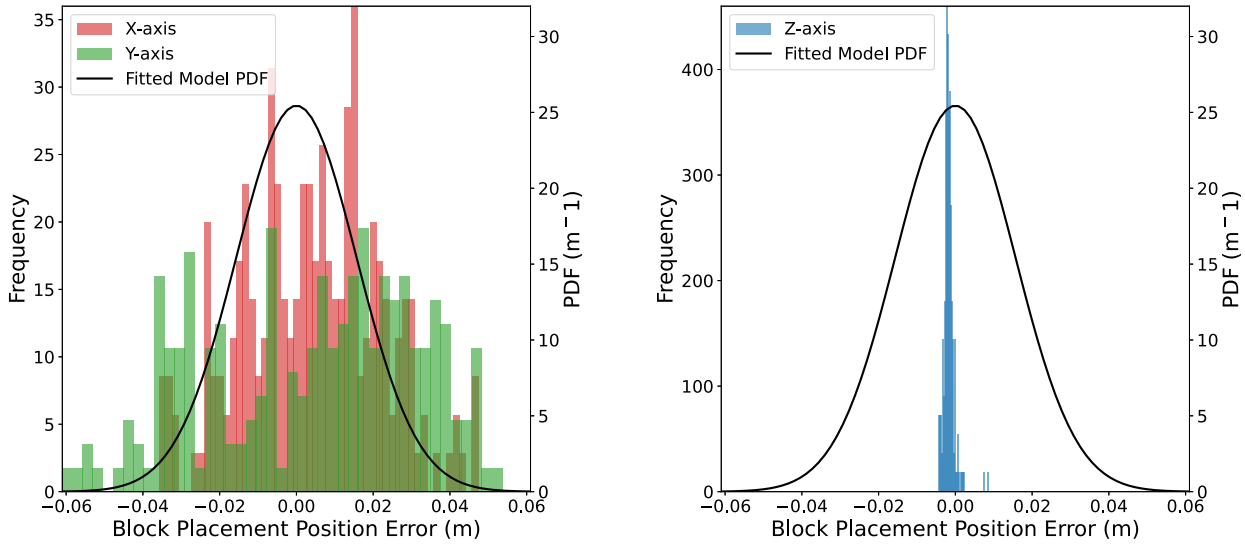


Figure 5.11: Block placement error characterisation. Histograms show block placement position errors on the X- and Y-axes (left) and the Z-axis (right), collected from the training dataset (N = 250 placement trials). The overlaid curves show the fitted probability density functions (PDFs) of a zero-mean, isotropic 3D Gaussian model, using the average standard deviation $\sigma_A = 1.57$ cm estimated across axes. Notably, the Z-axis error exhibits a highly concentrated distribution with significantly lower variance than the X- and Y-axes, resulting in a pronounced mismatch with the fitted isotropic Gaussian model. This reflects the fact that vertical placement is constrained by contact dynamics with the supporting surface in the simulator, which enforces small, tightly bounded height errors, whereas lateral (XY) placement is not directly constrained and therefore exhibits greater variability. The isotropic model therefore overestimates uncertainty along the Z-axis, yielding a conservative approximation of placement noise. Despite this mismatch, the isotropic Gaussian assumption is retained as a simplifying trade-off, with limited impact on downstream action selection performance. A slight negative bias in the Z-axis error distribution is also observed, likely arising from a small positive vertical offset applied during motion planning, where the block is released slightly above the target surface to avoid unintended collisions.

5.6.2.2 Greedy Next-Best Action Selection Performance

Results of the next-best action selection evaluation using the simulated robot are shown in Table 5.3. Our causal reasoning architecture achieves a success rate of **94.2%**, compared to **74.4%** for the baseline, resulting in a **19.8 percentage point** improvement under realistic system noise. In the idealised setting with no manipulation error, our method achieves **100%** success versus **70.0%** for the baseline, corresponding to a **30 percentage point** improvement.

To assess statistical significance, we model task success as a Bernoulli process and perform Bayesian inference using a Beta-Binomial model with a uniform prior. The posterior 95% credible interval for the baseline method success probability is $[0.704, 0.780]$, while that of our method is $[0.918, 0.959]$ under realistic noise. In the no-noise setting, the corresponding intervals are $[0.572, 0.820]$ for the baseline and $[0.931, 0.999]$ for our method. The intervals

Table 5.3: Performance on the block stacking task in simulation (realistic noise: $N = 500$ trials; no-noise: $N = 50$ trials). Our architecture significantly outperforms the naïve baseline, achieving a **19.8 percentage point** improvement in success rate under realistic noise, and **100%** success without manipulation error (**30 percentage point** increase), highlighting the benefit of reasoning about physical stability and action uncertainty. Using a Beta-Binomial model with a uniform prior, posterior analysis yields well-separated credible intervals and $P(p_{\text{ours}} > p_{\text{baseline}} \mid \text{data}) \approx 1.0$ under realistic noise and > 0.999 in the no-noise setting, with posterior comparisons estimated via Monte Carlo sampling (10^6 samples), indicating near-certain superiority of the proposed method.

Action-Selection Method	**Task Successes**	**Task Failures**	**Success Rate**
Baseline	372	128	74.4%
COBRA-PPM **(Ours)**	471	29	**94.2%**
Baseline (No Manipulation Noise)	35	15	70.0%
COBRA-PPM (**Ours**, No Manipulation Noise)	50	0	**100%**

are well separated, indicating a clear and statistically robust improvement. Posterior sampling using 10^6 Monte Carlo samples yields $P(p_{\text{ours}} > p_{\text{baseline}} \mid \text{data}) \approx 1.0$ under realistic noise and > 0.999 in the no-noise setting, providing strong evidence that the observed performance gains are not attributable to sampling variability.

Fig. 5.12 compares predicted stability probabilities for candidate placement positions during decision-making. The figure illustrates how our architecture accounts for both perception and manipulation uncertainty under low and high noise conditions, highlighting its robustness to real-world variability.

5.6.2.3 Qualitative Analysis of Results

Comparison to Naïve Baseline. The improved performance of our architecture over the baseline arises from its ability to explicitly model rigid-body physics and account for uncertainty in perception and manipulation. By integrating the PyBullet 3D physics simulator into the causal model, our method enables realistic forward simulations of candidate block placements. This allows the robot to anticipate stability outcomes under noisy conditions during online decision-making. In contrast, the baseline uses a simple heuristic that ignores both system dynamics and uncertainty.

As a result, our architecture generalises more effectively across varied initial tower configurations. It remains robust even in challenging cases — for example, when intermediate blocks are placed off-centre, as illustrated in Fig. 5.13. This robustness is further demonstrated by

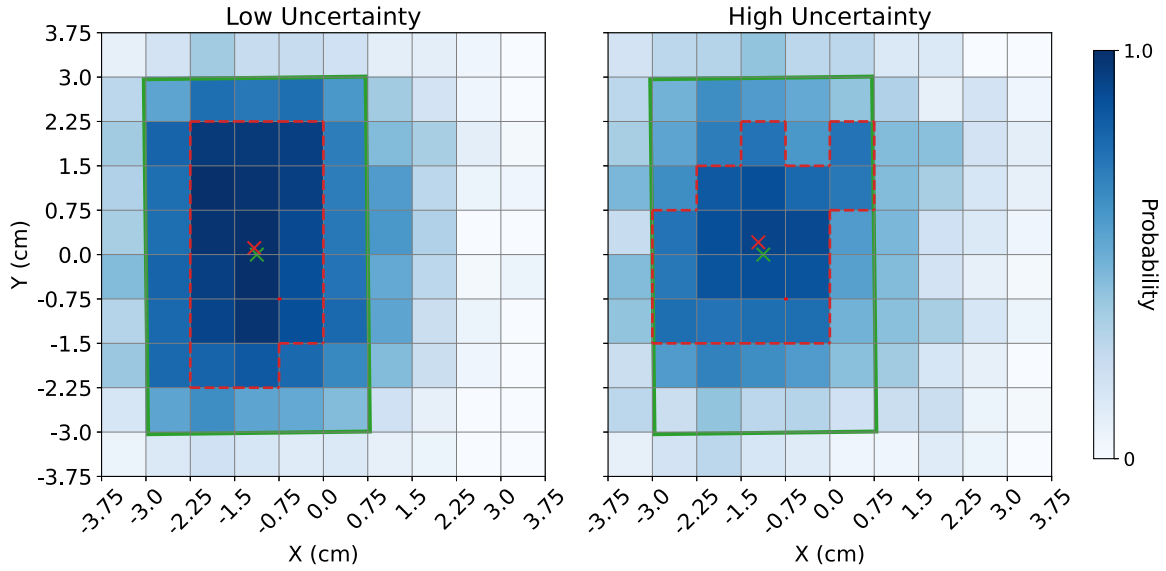


Figure 5.12: Predicted tower stability probabilities for candidate block placements under low (left) and high (right) system uncertainty, given the same initial tower. Predicted stable sets and centroids (robot's actions) are shown in red; ground-truth stable regions and centroids in green. Under higher uncertainty, the predicted stable region shrinks and centralises due to edge risk and isotropic 3D Gaussian noise on the robot's belief.

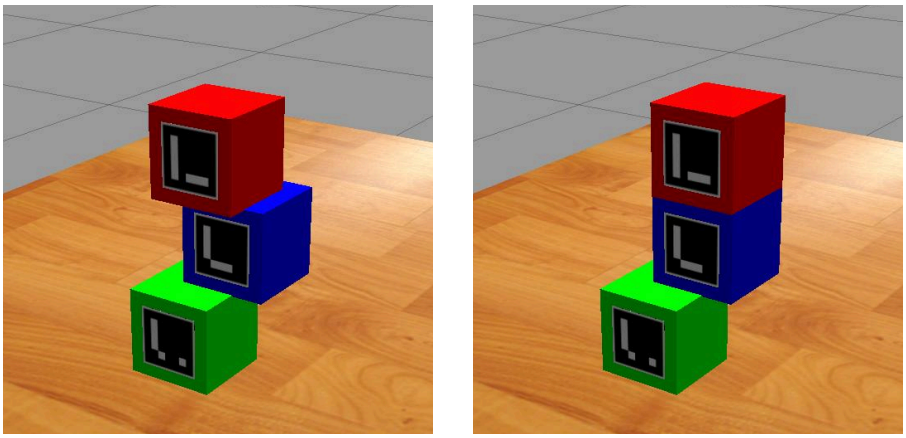


Figure 5.13: Comparison of selected block placement positions for a precarious two-block tower in Gazebo simulation. Our architecture (left) selects a stabilising action by explicitly modelling rigid-body physics and accounting for uncertainty, compensating for the off-centre blue block. The baseline method (right), which ignores dynamics and uncertainty, places the block in a way that destabilises the tower. This illustrates how our approach generalises better to challenging configurations by reasoning over physical interaction outcomes.

the **100% task success rate** achieved in the idealised setting without manipulation error, underscoring the value of precise, stability-aware action selection.

In contrast, the baseline method's lack of physics-based reasoning leads to higher failure rates, especially on more precarious tower states sampled from the test dataset. These results strongly suggest that the baseline's reduced performance in both the standard and noise-free

conditions stems from its inability to reason about the physical consequences of its actions.

Manipulation Errors. While our method is designed to account for manipulation errors during decision-making, its decreased performance in the standard case (compared to the no–manipulation–error setting) suggests that some failures still occur. These are likely attributable to two factors: (1) rare manipulation error tail events, and (2) model misspecification.

In certain instances, if a sampled manipulation error significantly exceeds the standard deviation of the modelled distribution (i.e., a tail event), the actual task success probability may diverge substantially from the *expected* value used by the decision function. The estimated standard deviation of manipulation error is 1.570 cm, which is approximately 45% of the block half-width. This implies that only 19% of placements are expected to fall within the 0.75 cm target cell, and 53% within its immediate 8-cell neighbourhood. Given this relatively high level of uncertainty relative to the block size, and the small margin of error for stable placement in the stacking task, our method may select an optimal action that still results in task failure due to random variation.

A second contributing factor is manipulation model misspecification. We assume that manipulation errors along the x-, y-, and z-axes are independent, identically distributed Gaussian random variables with a shared scale parameter σ_A. While this simplification reduces data requirements and inference cost, it assumes the error distribution is invariant to robot base pose, arm configuration, and placement location — factors that may plausibly affect placement accuracy.

Further investigation is needed to identify a minimal set of system variables that improve model fidelity without introducing excessive training data or inference computational demands. Without detailed characterisation of the robot hardware and full-body motion execution, it is difficult to determine whether task failures are primarily caused by rare error realisations or by inaccuracies in the noise model. A more detailed investigation of this distinction is beyond the scope of this chapter and is left for future study.

5.6.3 Comparison to Existing Approaches

In robot manipulation, causal reasoning remains underexplored. While early work has made meaningful progress in bridging these fields, this area of research is still in its early stages [40]. In this section, we compare the capabilities of our causal reasoning architecture with existing

approaches. As seen in Table 5.4, our approach is the only one that supports all capabilities needed to formulate and solve robot manipulation tasks in a mathematically rigorous and causally grounded way: a CBN-formulation implemented in a PPL, integration with a physics simulator, explicit modelling of both action and sensor uncertainty, and high configurability and extensibility for modelling and inference.

Table 5.4: Comparison of our method to related work across key architectural dimensions. Our architecture is the only one that supports all key capabilities for causal, probabilistic reasoning in robot manipulation, including physical simulation, uncertainty modelling, and flexible inference.

Approach	Modelling			Robotics Suitability			Implementation & Flexibility		
	CBN-based	SCM-ready	Causal decision-making	Physics sim	Accounts for action uncertainty	Accounts for sensor uncertainty	Extensible causal model	PPL-based	Configurable online inference
Beetz et al. (2012) [41]	✗	✗	✗	✓	✓	✓	✗	✗	✓
CausalWorld (2020) [40]	✓[1]	✗[2]	✗	✓	✗[3]	✗[3]	✗	✗	✗
Diehl & Ramirez-Amaro (2022) [37]	✓	✓	✗	✓	✓	✗	✗	✗	✗
Diehl et al. (2023) [38]	✓	✓	✗	✓	✓	✗	✗	✗	✗
COBRA-PPM (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓

Diehl & Ramirez-Amaro [37] use a CBN to model block stacking and perform action selection based on outcome probabilities learned from offline simulations. However, their model lacks configurability, does not account for sensor or actuator uncertainty, and must be retrained for new tasks. Follow-up work [38] addresses failure prediction but inherits the same limitations. In contrast, our method performs online simulation at inference time, allowing for dynamic reconfiguration and explicit modelling of uncertainty.

CausalWorld [40] introduces a simulation benchmark for transfer learning in manipulation tasks involving physical attributes, but it does not provide a principled causal reasoning framework for decision-making. Similarly, physics-based reasoning approaches [41] simulate dynamics but lack the causal semantics required for probabilistic causal inference, which are central to our approach.

Recent advances in PPLs such as Pyro [46] enable modular, generative causal modelling over complex distributions. Building on this, our architecture implements a Pyro-based CBN that

[1] Authors provide a list of high-level variables in the simulation environment exposed for interventions but no causal DAG is specified. Further, there is no mechanism for conditioning statements, so it cannot be used for Bayesian inference.

[2] Although authors state that changes to the exposed variables can be *considered* do-interventions on the underlying SCM, there is actually no underlying formal SCM model formulation. As such, the two primary benefits of SCMs — abduction of exogenous variables and thus counterfactuals — are not possible.

[3] To the best that we can understand from the paper and online documentation, neither action nor sensor stochasticity is included. A random seed is mentioned in the experimentation description but this is only used by random sampling for RL policy learning algorithms.

can be naturally extended to a structural causal model (SCM) [32], enabling counterfactual inference [7].

Counterfactual reasoning aligns closely with human causal judgement [62] and is increasingly recognised as essential for explainable and responsible robot systems [31, 43, 153]. Our architecture provides the foundation for such capabilities in future causal explanation and behaviour analysis.

These comparisons highlight the architectural contributions of our approach and its potential as a unifying framework for probabilistic causal reasoning in robotic systems, with broader implications revisited in the discussion on scalability and limitations in Section 5.8.

5.7 Real-World Robot Demonstration

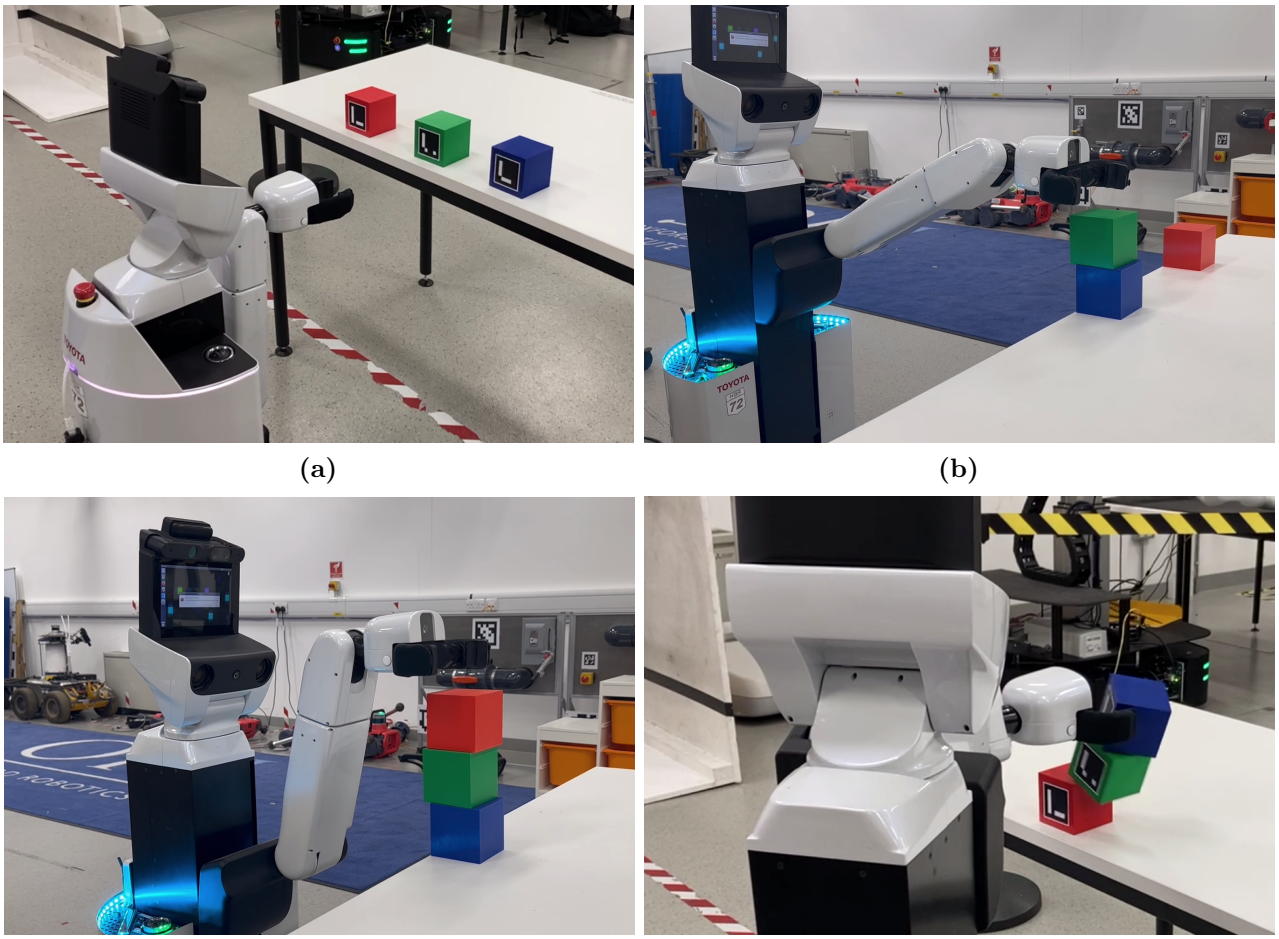


Figure 5.14: Real-world demonstration of the two-action block stacking task used to evaluate our causal reasoning architecture, showing the robot executing on physical hardware, progressing from: (a) the initial state, (b) an intermediate stable-tower state after the first action, to two possible outcomes: (c) a stable tower indicating success, or (d) an unstable tower indicating failure.

To support the applicability of our approach beyond simulation, we deploy our architecture on a real-world Toyota HSR performing the exemplar block stacking task (Fig. 5.14). The physical setup closely mirrors the simulation environment, with one key variation: while simulation trials begin with a pre-constructed two-block tower, real-world trials start from a single block and sequentially place two blocks, to demonstrate robustness to cumulative action-outcome uncertainty in a qualitative setting. No parameters were re-tuned. The same perception and manipulation pipeline was used in both settings, and the physical blocks matched the simulated dimensions.

The system successfully completed 4 out of 5 trials (**80% success rate**), demonstrating strong sim-to-real generalisation and suitability for real-world deployment without retraining or parameter adjustment.

We note that the number of real-world trials ($N = 5$) is small and therefore does not support statistically meaningful estimation of performance metrics. Instead, this experiment is intended as a qualitative demonstration of sim-to-real transfer, verifying that the proposed causal reasoning architecture and noise-aware modelling assumptions remain valid when deployed on physical hardware. Conducting large-scale real-world trials is constrained by practical considerations such as execution time; however, the observed success across multiple sequential placements provides evidence that the system operates reliably under real-world sensing and actuation uncertainty. Notably, each trial involves multiple sequential actions, such that failures compound over time, making successful completion indicative of consistent performance across multiple decision steps.

While initial configurations differ slightly, the core task remains unchanged: reasoning under uncertainty about tower stability and action consequences. One possible source of performance variation is that, in simulation, the robot places a third block onto randomly generated two-block towers, which may be less stable than the deliberately constructed intermediate towers used in real-world trials. Nonetheless, because our model explicitly accounts for perception and actuation noise, it achieves robust performance across sequential placements despite the compounding uncertainty introduced at each step.

5.8 Architecture Scalability & Limitations

Our architecture is designed to be modular and extensible, enabling its application to a wide range of manipulation scenarios. In this section, we explore its scalability along three axes: action complexity, sequential reasoning, and reward and uncertainty modelling. We also discuss limitations and considerations when scaling to more complex tasks.

5.8.1 Action Spaces

The proposed dynamic CBN-based decision-making causal model can be extended to support richer action representations. In principle, additional action variables — such as diverse block geometries, frictional properties, or non-canonical placement orientations — can be incorporated by expanding the causal graph. Pyro's plate notation and conditional independence enable efficient factorisation of the joint distribution, improving scalability over flat representations. Our model also supports multi-tower or interacting structure scenarios by modularising local subgraphs and leveraging a physics-based simulator — such as PyBullet, as demonstrated in our exemplar task — to capture the relevant physical interactions.

Scope of Environment Modelling. In practice, the causal model need only explicitly represent those aspects of the environment that are causally relevant to the task and the queries being evaluated. This reflects the fact that the state space, action space, and causal model are jointly designed to support decision-making, rather than to capture all available sensory information. Although a robot operating in the real world may rely on high-dimensional, multi-modal sensor data and complex processing pipelines — such as RGB-D perception, mapping, localisation, and motion planning — these can be distilled into a compact set of variables sufficient for high-level reasoning. For the block stacking task, this corresponds to modelling the geometric configuration of the tower and a low-dimensional parametrisation of the action space, where each action is defined by a candidate block placement (x, y). In our implementation, perception and control are handled by external systems (e.g., ArUco marker-based pose estimation and motion planning via ROS MoveIt!), while the causal decision-making model operates on this distilled representation. We assume, and empirically validate, that all candidate actions lie within the robot's feasible manipulation workspace, allowing the model to focus on selecting among valid placements rather than reasoning about reachability. Other

environmental factors — such as detailed contact dynamics or unmodelled physical effects — are either abstracted into stochastic noise variables or handled implicitly by the underlying physics simulator. This selective modelling reflects a trade-off between fidelity and tractability, ensuring that the model captures the dominant causal mechanisms required for decision-making without incurring unnecessary complexity.

5.8.2 Sequential Decision-Making

While the present work focuses on greedy, single-step reasoning, the framework is readily extensible to sequential settings such as MDPs and POMDPs. In these cases, the CBN-based probabilistic reasoning layer can inform higher-level control modules by modelling stochastic transitions under interventions. Our formulation is compatible with CAR-DESPOT (see Ch. 4; originally introduced in [7]), a causal-aware POMDP planner that complements our approach. This enables hybrid architectures in which CBNs provide semantically grounded transition models, while planners coordinate long-horizon action sequences. Such a composition retains the interpretability and structure-awareness of our framework, while enabling principled reasoning over policies and belief updates under partial observability.

Causal Structure vs Adaptation. A key distinction in sequential settings is between the fixed causal structure and the adaptive components of the system. The causal graph (i.e., the dynamic CBN) defines the structural relationships between states, actions, and observations, and remains invariant across time steps and episodes, reflecting the assumption that the underlying causal mechanisms governing the environment do not change. This structure encodes the data-generating process and physical interactions relevant to the task. Adaptation, by contrast, occurs through inference and decision-making within this fixed structure. At each time step, the robot updates its beliefs based on observations and evaluates interventions over candidate actions to select the next action. In extended sequential settings (e.g., POMDP planning), this corresponds to belief updates and policy optimisation operating over the same underlying causal model. This separation enables the model to generalise across different task instances and trajectories, while allowing behaviour to adapt online to observed outcomes and uncertainty without modifying the underlying causal structure. This also enables the system to respond to changes in the environment across time steps. For example, if the configuration of the tower or the next block is perturbed between actions, the updated observation at the next

time step is incorporated into the inference process, and subsequent decisions are conditioned on this new state. Crucially, each decision step operates on a temporally consistent snapshot of the system state, analogous to a *D-latch* in digital systems, where observations are sampled and held fixed for the duration of inference and decision-making before being updated at the next time step. This ensures that causal queries are evaluated on a well-defined state, avoiding inconsistencies arising from asynchronous updates across sensing and control modules. Thus, adaptation occurs through online sensing and re-evaluation at each decision step, rather than through modification of the causal structure itself. In this sense, the model operates in a receding-horizon manner, where each action is selected based on the current observed state at the time of decision.

5.8.3 Reward Functions and Statistical Objectives

The architecture supports flexible objectives beyond expectations over Gaussian rewards. Risk-aware metrics such as CVaR, percentile thresholds, or multi-objective criteria can be incorporated without changes to the underlying causal graph. For example, a practitioner may wish to optimise for both task success and robustness to disturbances such as surface vibrations or wind. Probabilistic programming enables sampling-based estimation under arbitrary distributions, allowing the use of non-Gaussian, asymmetric, or heavy-tailed beliefs. This flexibility supports a wide range of task definitions, robot morphologies, sensing modalities, and deployment environments.

In this work, we adopt a Gaussian noise model as a deliberate modelling trade-off (see Sec. 5.4.2.5). While the empirical error distributions exhibit some deviations from a Gaussian distribution, particularly along individual axes, the Gaussian assumption provides a compact and tractable parametrisation that integrates naturally with the causal model and supports efficient sampling-based inference. Importantly, this approximation is sufficient to capture the dominant uncertainty characteristics relevant to decision-making in this task and, as demonstrated in Sec. 5.6, does not significantly degrade downstream performance. More expressive non-Gaussian models (e.g., mixture distributions or skewed noise models) could be incorporated within the same probabilistic programming framework; however, these were not explored here in order to maintain interpretability and to focus on evaluating the causal reasoning architecture rather than distributional modelling choices. While this work focuses on a simple success-based objective for clarity and evaluation, the above formulation provides a flexible

foundation that practitioners can extend to more complex, risk-sensitive, or multi-objective task definitions as required.

5.8.4 Computational Limitations and Complexity

A key limitation lies in the computational cost of inference when scaling to large causal graphs with many variables and dependencies. While exact inference is intractable in such settings, our current implementation uses importance sampling to approximate posterior distributions over latent variables. More scalable methods, such as stochastic variational inference (SVI), could be adopted in future work — for example, using Pyro's plate notation to enable parallelism across samples.

Headless PyBullet simulation runs faster than real-time in our task and is not a major bottleneck. However, simulation cost may increase in more complex scenes, especially those involving many rigid bodies or advanced dynamics such as soft-body deformation or fluid interactions. Techniques such as simulation caching, surrogate models, or amortised inference offer promising paths to reduce computational load.

5.8.5 Adaptability and Domain Transfer

Although causal models are interpretable and structured, deploying them in new environments may require domain-specific assumptions or structure learning, introducing some overhead compared to fully data-driven approaches. However, our framework is explicitly designed for modularity, reuse, and abstraction: the causal graph, inference engine, and decision-making components are loosely coupled and can be adapted independently. Further, causal models are well suited to domain transfer, as they aim to capture fundamental causal mechanisms that govern system behaviour and are expected to remain invariant across environments. This composability supports flexible transfer across domains by reusing learned causal templates, swapping simulators, or integrating new inference targets — without redesigning the overall architecture.

Overall, while scalability introduces trade-offs in model complexity, inference cost, and task-specific assumptions, our architecture remains a structured and extensible foundation for causal reasoning in robot manipulation. Its modular design, probabilistic semantics, and integration with physical simulation position it well for scaling to more complex tasks and deployment environments.

5.9 Summary

In this chapter, we introduced COBRA-PPM, a novel causal Bayesian reasoning architecture that combines a decision-making causal model, probabilistic programming, and data-driven components to support robust manipulation under uncertainty. To address **Q1 - Modelling** and **Q4 - Decision Making**, we developed a structured probabilistic model of physical robot-world interactions. This enables predictive reasoning and greedy action selection in manipulation tasks, where reasoning about physical object dynamics and action-outcome uncertainty is critical for scene understanding and task success.

We evaluated the architecture on an exemplar block stacking task, integrated into a complete autonomous robot system in which sensing, inference, and actuation are performed onboard. Our empirical validation in a high-fidelity simulation environment demonstrated accurate prediction of manipulation outcomes and robust decision-making under typical perception and actuation uncertainties in mobile robot systems. We further showed strong sim-to-real generalisation by deploying the architecture on a real-world domestic service robot, without retraining or parameter tuning. We also examined the scalability of our architecture along three dimensions — action complexity, sequential reasoning, and reward and uncertainty modelling — and discussed potential limitations associated with scaling to more complex domains.

This work contributes a novel generalised causal probabilistic reasoning architecture for manipulation and opens new opportunities for causal and counterfactual reasoning in trustworthy autonomous systems.

Causal generative decision-making models are central to how we address **Q1 - Modelling**, **Q4 - Decision Making**, **Q5 - Counterfactual Reasoning**, and **Q6 - Counterfactual Explanations** in the chapters that follow. As such, this chapter lays the foundation for our exploration of counterfactual explanations for robot behaviour in Chapter 6, and counterfactual reasoning about the physicality of systems in Chapter 7.

The research in this chapter has been published in two IEEE venues. The decision-making causal model and physics-based integration were first introduced in [8], while the complete causal reasoning architecture, system integration, simulation validation, and real-world demonstration were presented in [9].

6

Counterfactual-Based Post-Hoc Explanations of Robot Task Execution

Contents

6.1 Introduction 162
6.2 From Contrastive to Counterfactual Explanations 164
6.3 Structural Causal Modelling of the Robot Task 166
6.3.1 SCM Formulation for the Block Stacking Task 166
6.3.2 Implementation in Pyro Probabilistic Programming 168
6.3.3 Counterfactual Inference via the Twin-World Algorithm 169
6.3.4 Counterfactual Attribution Metrics 170
6.4 Counterfactual Explanation Methods 170
6.4.1 Overview of the Explanation Pipeline 171
6.4.2 Method 1: Single-Variable Intervention Most Likely to Change Outcome (SVIMLTCO) 171
6.4.3 Method 2: Multi-Variable Intervention Most Likely to Change Outcome (MVIMLTCO) 173
6.4.4 Method 3: Responsibility-Based Attribution 175
6.4.5 Additional Model Evaluations under Placement Noise & Multi-Step Dynamics. 179
6.5 Generating Natural-Language Text Explanations 180
6.6 Robot Explainer System & RoboTIPS Demonstration 183
6.7 Limitations & Future Work 185
6.7.1 Current Limitations 187
6.7.2 Future Work: Human-Participant Evaluation 189
6.8 Summary 190

6.1 Introduction

Explanations are central to responsible autonomy: they connect what a system did to *why it did so* and what would have happened under relevant alternatives. In this chapter, we investigate post-hoc explanations of robot task execution that are grounded in counterfactual reasoning. Building on the causal world model developed in Chapter 5, we move from using causal structure to support prediction and decision-making to using it to generate faithful, human-aligned accounts of behaviour and outcomes. In doing so, the chapter directly addresses **Q5 - Counterfactual Reasoning** and **Q6 - Counterfactual Explanations**, and contributes to **Q1 - Modelling** by refining the causal representation needed for counterfactual inference.

A growing body of work employs contrastive explanations that compare hypothetical outcomes under alternative inputs generated from a learned model. While useful, such approaches typically operate at the interventional level and need not respect the structural mechanisms of the world. As a result, they can produce associations that do not track actual causal responsibility and may deviate from human judgements. Our thesis-wide stance is that explanations for autonomous systems should respect the system's causal structure and support queries of the form 'What would have happened, had X been different — everything else equal?'. This requires *structural causal models* (SCMs) and counterfactual inference.

The chapter therefore extends the earlier CBN-based formulation to an SCM that separates exogenous uncertainty from endogenous deterministic assignments, enabling level-3 counterfactual queries. Using this SCM, we develop a small suite of counterfactual attribution methods that answer questions such as 'Why did the tower fall?' or 'Why did this task episode succeed?' in a block-stacking task. The methods combine twin-world counterfactual simulation with attribution quantities drawn from actual causality, including the probability of necessity (PN), the probability of sufficiency (PS), and related responsibility measures. We then translate these quantities into concise natural-language explanations intended for non-expert users.

This chapter has two primary objectives. First, we aim for *faithfulness*: explanations should be consistent with the assumed causal structure and assignment functions. Second, we aim for *human alignment*: explanations should reflect core dimensions of human causal judgement, such as minimal-change counterfactuals and sensible allocation of responsibility across multiple contributing factors. These aims support **Q6 - Counterfactual Explanations** and complement the decision-making benefits established earlier for **Q4 - Decision Making**.

The work reported here comprises:

1. the SCM extension of the block-stacking world model and a Pyro-PPL-based implementation that supports arbitrary compositions of conditioning and interventions;

2. three counterfactual attribution methods that consider single-variable changes, multi-variable changes, and principled allocation of responsibility when causes interact or over-determine outcomes;

3. a light-weight natural-language generation layer that verbalises causal attributions; and

4. an integrated 'robot explainer' prototype, demonstrated as part of the RoboTIPS showcase, that answers sense-think-act queries with counterfactual justifications.

Empirically, we present qualitative proof-of-concept results and a hardware demonstration that illustrate how counterfactual explanations can align with human intuitions in simple physical tasks. A proposed human-participant study intended to measure alignment against human judgements did not proceed within the thesis timeline; its design and hypotheses are retained and left as future work (Sec. 6.7).

In summary, this chapter contributes a counterfactual explanation pipeline for robot manipulation grounded in SCMs. It shows how to move from causal reasoning for action to causal reasoning for *explanation*, and how to operationalise actual-causality quantities in a form suitable for end users. The remainder of the chapter proceeds as follows. Sec. 6.2 motivates the shift from hypothetical contrastive analyses to counterfactual explanations and situates the approach within the thesis narrative. Sec. 6.3 presents the SCM formulation and counterfactual inference machinery. Sec. 6.4 details the explanation methods and their comparative properties, and provides qualitative results that demonstrate a proof-of-concept of the developed methods. Sec. 6.5 describes the natural-language generation layer. Sec. 6.6 documents the robot explainer system and the RoboTIPS hardware demonstration. Sec. 6.7 discusses limitations and outlines the future human-participant study. Finally, Sec. 6.8 summarises the contributions and links forward to the learning-centric developments in Chapter 7.

6.2 From Contrastive to Counterfactual Explanations

Understanding *why* a robot acted in a particular way or *why* a task outcome occurred is fundamental for trust, safety, and accountability in autonomous systems. However, as reviewed in Sec. 2.4, most explanation methods in robotics still rely on *contrastive* reasoning: they ask what would happen if some input feature or parameter were changed, and compare the resulting model output to the observed one. These contrastive approaches are typically implemented through hypothetical simulations of learned models or statistical sensitivity analyses over input-output pairs. Although they can highlight which variables are influential, they do not ensure that such relationships are causally valid. Without an explicit causal model, these methods operate at the *interventional* level of reasoning (see Sec. 2.4) and may yield dependencies arising from spurious correlations rather than genuine cause-effect relations.

For example, when analysing a robot's block stacking task, a contrastive method might identify the x-position of a block as the most influential factor for tower stability, because varying that parameter changes the predicted outcome. Yet without encoding the physical causal structure of the world, the method cannot determine whether that change propagates through the task's actual mechanics or through statistical regularities captured by the model. Such explanations risk violating causal consistency: they may ascribe responsibility to variables that correlate with the outcome but are not its true cause. Similar limitations were observed in prior CBN-based robot explanation frameworks [37, 38], which remained confined to level-2 causal reasoning.

To produce explanations that align with human causal reasoning, we must move beyond contrastive and hypothetical analysis to the third tier of Pearl's Ladder of Causality: *counterfactual reasoning*. Counterfactual reasoning asks questions of the form 'What would have happened if this variable had been different, everything else held equal?'. This principle of minimal change, or *closest possible world semantics* [61], ensures that all factors not causally downstream of the intervention remain fixed. By comparing the factual world to its minimally modified counterfactual counterpart, we isolate the true effect of a specific cause on an outcome, respecting the system's internal mechanisms.

As discussed in Sec. 2.4, psychological studies such as Gerstenberg [62] show that humans intuitively reason in this counterfactual manner. When people assess responsibility for an event, they mentally simulate alternative worlds consistent with what actually happened, differing

only in the variables of interest. If the outcome changes under this minimal modification, the variable is judged causally responsible. Consequently, explanation methods that emulate such reasoning are more likely to produce intuitively satisfying and trustworthy accounts.

Structural causal models (SCMs) provide the formal machinery required to realise counterfactual reasoning in autonomous systems. As introduced in the literature review (Sec. 2.4), SCMs explicitly represent both the causal dependencies between variables and the stochastic exogenous factors that introduce uncertainty [32, 63]. This separation between endogenous and exogenous components enables paired factual and counterfactual worlds that share the same exogenous noise but differ in targeted interventions. Through the twin-world algorithm, we can compute the posterior distribution over counterfactual outcomes given factual observations, thereby answering principled queries of the type

$$P(Y_{X=x'} \mid X = x, Y = y),$$

which reads as: 'given that X was observed to be x and the outcome Y was y, what would Y have been if X had instead been x'?' This corresponds to level-3 causal knowledge and cannot be achieved by contrastive or purely interventional approaches.

In Chapter 5, we established a causal world model of the robot block stacking task using a causal Bayesian network (CBN) to support prediction and decision-making under uncertainty. That formulation permitted association- and intervention-level inference but not counterfactual reasoning. Here, we extend that model to a structural causal model, following the trajectory outlined in the literature (Sec. 2.4) from interventional to counterfactual explanations. This enables explicit separation of exogenous noise sources and allows simulation of both factual and counterfactual task executions. This extension marks the conceptual and methodological bridge between the earlier chapters focused on causal reasoning for control and the present chapter focused on causal reasoning for *explanation*.

The next section (Sec. 6.3) introduces the structural causal model used in this work, explains its implementation in the Pyro probabilistic programming language, and describes how it supports counterfactual inference and causal attribution within the block stacking task.

6.3 Structural Causal Modelling of the Robot Task

To operationalise counterfactual reasoning in the robot block stacking domain, we extend the causal Bayesian network (CBN) introduced in Chapter 5 into a *structural causal model* (SCM). This extension builds directly on the conceptual motivations outlined in Sec. 6.2 and the related literature on SCMs for causal attribution and responsibility (see Sec. 2.4). The SCM provides the mathematical framework required to represent both the deterministic structure and the stochastic variability of the robot task, enabling the computation of principled counterfactual inferences.

In the CBN formulation used previously, all random variables were jointly distributed and connected by directed edges indicating conditional dependencies. While this permitted inference at the associational and interventional levels, it did not distinguish between the deterministic mechanisms that generate variable values and the exogenous randomness that injects uncertainty into those mechanisms. The SCM augments this formulation by introducing an explicit split between *endogenous* and *exogenous* variables. Each endogenous variable v_i is assigned a deterministic function f_i that maps its parents pa_i and its corresponding exogenous variable u_i to its value:

$$v_i = f_i(pa_i, u_i).$$

The set of all f_i defines the causal mechanisms of the system, while the exogenous variables U capture the latent noise sources and are jointly distributed according to $P(U)$. This structural decomposition makes it possible to reason about specific realised events and to construct parallel *factual* and *counterfactual* worlds that share the same exogenous noise but differ in the variables targeted by an intervention.

6.3.1 SCM Formulation for the Block Stacking Task

We model the robot block stacking task as a Markov decision process (MDP) expressed in SCM form, extending the graphical model from Fig. 5.4 in Chapter 5. In this chapter, our primary objective is to develop and assess *explanation methods* rather than to address the computational complexity of planning and inference under partial observability. Accordingly, we adopt the MDP formulation to focus analysis on the *faithfulness* and *human-alignment* properties of the proposed counterfactual attribution methods, rather than on the algorithmic challenges specific

to POMDP belief updates and state estimation. The causal explanation mechanisms developed here are model-agnostic and are expected to generalise to partially observable domains, where they would operate over inferred latent states rather than fully observed ones.

At each discrete time-step $k \in [1, K]$, the agent samples an exogenous variable N_{A_k} representing random influences on its action selection (for instance, exploration noise or policy stochasticity). Given this sample, the action A_k is determined deterministically by an assignment function f_{A_k}:

$$A_k = f_{A_k}(N_{A_k}).$$

The environment then samples an exogenous variable N_{S_k} capturing stochastic disturbances in the transition dynamics (e.g., perception or actuation noise), and deterministically computes the next state:

$$S_k = f_{S_k}(S_{k-1}, A_k, N_{S_k}).$$

This formulation embeds the agent's decision process within a causal structure that explicitly separates endogenous decision variables, deterministic update functions, and the latent exogenous factors driving uncertainty.

A simplified schematic of the resulting SCM is shown in Fig. 6.1, and a hierarchy of increasingly complex task models is provided in Fig. 6.2. The first model captures deterministic single-step stacking, the second introduces stochastic actuation noise, and the third extends to multi-step sequences with uncertainty in both decision and actuation. Each model preserves the same structural pattern while adding realism through additional exogenous variables.

Extended SCM Variants: Placement Noise & Multi-Step Actions. Beyond the base single-step model, we instantiated two extended SCMs to probe realism and scalability. First, a *placement noise* model augments the action with an exogenous term N (2D zero-mean Gaussian; st.dev. configurable, e.g., 0.03 m), applied additively to the selected (X, Y) before simulation (Fig. 6.2b). In this setting, abduction infers a posterior over N consistent with the factual episode, enabling minimally changed counterfactuals that hold the estimated noise fixed when interrogating alternative placements. Counterfactual tests for noise as a cause use an intervention $do(N = 0)$ to assess whether eliminating placement noise would have changed the outcome. Second, a *multi-step* variant models sequences of placements with time-indexed

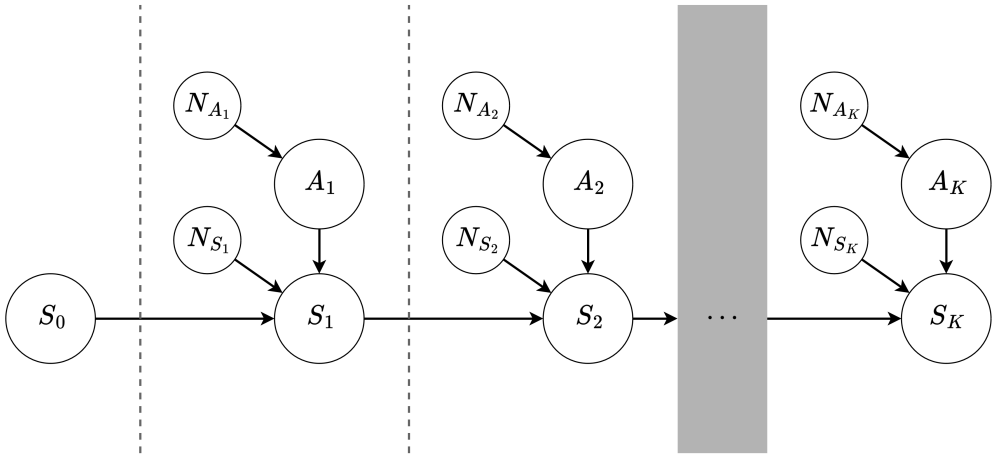
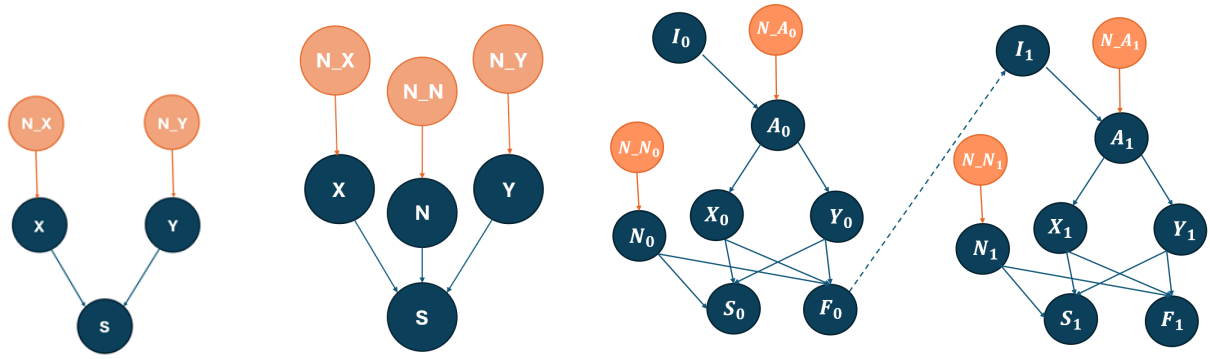


Figure 6.1: The MDP-based robot task formulation, representing the SCM's underlying causal DAG.



(a) Deterministic action **(b)** Stochastic action **(c)** Multiple stochastic actions

Figure 6.2: Causal DAGs for three SCM models for the block stacking task of increasing complexity: a) a single-action model with no action uncertainty; b) a single-action model with action uncertainty; and, c) a two-action model with action uncertainty. Source: adapted from Radojcic 2024 [154].

copies of the core SCM (Fig. 6.2c), and introduces a proxy state metric for explanation, tracking the tower's planar centre-of-mass distance to the ideal origin via variables I and F. This supports action-level explanations of the form 'Why action A instead of B?' by comparing the probability that $F_A < F_B$ under the twin-world construction.

Full evaluations of these variants are omitted for brevity; a qualitative synthesis is provided in Sec. 6.4.5.

6.3.2 Implementation in Pyro Probabilistic Programming

To implement these models, we developed an `SCM` class within our *RobotCausalReasoning* Python library. The class represents an SCM as the 5-tuple $\mathcal{M}_G = \langle G, U, V, F, P(U) \rangle$ [27], where G is the underlying directed acyclic graph, U the set of exogenous variables, V the

set of endogenous variables, F the set of deterministic assignment functions, and $P(U)$ the joint distribution over exogenous noise. The implementation leverages the Pyro probabilistic programming language to support:

- sampling traces consistent with the partial topological order of G, and
- computing the joint log-probability of observed traces for inference estimation.

Within the Pyro framework, inference is implemented using importance sampling and variational conditioning. Two base query types are supported:

- **Associational inference:** computing $P(V \mid E)$ for arbitrary evidence E, and
- **Interventional inference:** computing $P(V \mid \mathrm{do}(X = x))$ for any intervention target X.

These two capabilities form the computational primitives for counterfactual queries, which require both conditioning and intervention. By combining Pyro's `condition` and `do` operators, we can compose arbitrary counterfactual queries of the form $P(V_{X=x'} \mid V_{obs})$.

6.3.3 Counterfactual Inference via the Twin-World Algorithm

Counterfactual inference in this work follows the *Twin-World Algorithm* described by Pearl [32]. The algorithm proceeds in three conceptual steps:

1. **Abduction:** condition the factual model on observed evidence to infer a posterior over the exogenous variables, $P(U \mid V_{obs})$. This identifies which latent noise values are most consistent with the observed outcome.
2. **Action:** create a twin copy of the model, M', replacing its exogenous priors with the abducted posterior $U' = U_{abducted}$, and intervene on target variables using $\texttt{do}(X = x')$.
3. **Prediction:** simulate the counterfactual world to estimate the posterior over outcomes, $P(V_{X=x'} \mid V_{obs})$.

This sequence ensures that factual and counterfactual worlds share the same underlying exogenous noise, thereby satisfying the minimal-change principle and preserving causal coherence. In our implementation, these steps correspond to two nested inference calls using the SCM class methods for abduction and intervention. The resulting counterfactual distributions provide the foundation for computing actual-causality quantities and generating robot explanations.

6.3.4 Counterfactual Attribution Metrics

Once counterfactual queries can be computed, we can quantify causal responsibility using the attribution metrics described in Sec. 2.4. Following Pearl [6], the key quantities are:

- **Probability of Necessity (PN):** $P(Y_{X=0} = 0 \mid X = 1, Y = 1)$ — how likely it was that the cause $X = 1$ was necessary for the outcome $Y = 1$.
- **Probability of Sufficiency (PS):** $P(Y_{X=1} = 1 \mid X = 0, Y = 0)$ — how likely it was that $X = 1$ would have been sufficient to produce $Y = 1$.
- **Probability of Necessity and Sufficiency (PNS):** $P(Y_{X=1} = 1, Y_{X=0} = 0)$ — a combined measure capturing both necessity and sufficiency.

These quantities extend beyond purely statistical correlations by isolating the contribution of a variable to an observed outcome under minimal change. In our *RobotCausalReasoning* library, we implemented PN, PS, and PNS using Monte Carlo estimation over samples drawn from paired factual and counterfactual models generated by the twin-world algorithm. This provides a practical and extensible mechanism for quantifying actual causality in robot manipulation tasks.

The next section (Sec. 6.4) builds on this modelling foundation to describe how these counterfactual quantities are combined into algorithms for generating *post-hoc* explanations of robot task outcomes, including both single- and multi-variable causal analyses and a responsibility-based attribution scheme.

6.4 Counterfactual Explanation Methods

Having established the structural causal model and inference framework in Sec. 6.3, we now present the suite of counterfactual explanation methods developed to analyse robot task outcomes. These methods use the SCM to generate *post-hoc* explanations that quantify and communicate how specific causes contributed to an observed outcome. Each method performs counterfactual inference using the twin-world algorithm (Sec. 6.3.3) and applies quantitative attribution metrics (Sec. 6.3.4) to identify and rank influential variables. Together, they provide multiple complementary perspectives on causal responsibility: identifying minimal changes that would alter an outcome, combinations of variables most likely to have caused a change, and the proportional degree of responsibility borne by each factor.

6.4.1 Overview of the Explanation Pipeline

The explanation process follows the following general workflow. Given a factual episode (S_0, $A_0, S_1, \ldots, S_K$) from the robot block stacking task, the pipeline proceeds as follows:

1. **Abduction:** condition the SCM on the factual trajectory to infer the posterior distribution over exogenous variables $P(U \mid V_{obs})$;
2. **Counterfactual Simulation:** construct a counterfactual twin model and evaluate outcome distributions under one or more hypothetical interventions $do(X = x')$;
3. **Attribution:** compute counterfactual attribution scores (PN, PS, PNS, or derived metrics) that quantify each variable's contribution to the observed outcome; and
4. **Explanation Generation:** translate high-scoring causal attributions into interpretable outputs, which may be visual, tabular, or linguistic.

This pipeline is intentionally modular: the inference and attribution stages are model-agnostic and can be coupled with different front-end interfaces for explanation delivery, as described later in Sec. 6.5. By separating causal reasoning from explanation presentation, the framework supports reuse across diverse robotic contexts and modalities.

6.4.2 Method 1: Single-Variable Intervention Most Likely to Change Outcome (SVIMLTCO)

The first method identifies the individual variable whose intervention would most likely change the observed outcome. For each candidate cause variable X_i, we compute the probability that intervening on X_i with an alternative value x_i' would change the outcome Y, conditioned on the factual observation $(X_i = x_i, Y = y)$:

$$P(Y_{X_i=x_i'} \neq y \mid X_i = x_i, Y = y).$$

This score estimates how sensitive the outcome is to that variable under minimal change, directly reflecting the counterfactual criterion of necessity. Variables are then ranked according to this probability, yielding a list of the most influential single-variable causes.

The algorithm shown in Fig. 6.3 iterates over each variable, performs paired factual and counterfactual inference using the twin-world SCM, and estimates the probability of outcome

change by Monte Carlo sampling. Because each variable is evaluated independently, this method provides clear and interpretable attributions, but may underestimate interactions where multiple causes jointly determine the outcome.

Algorithm 1 Calculate the most probable cause of an observation in an SCM

```
procedure PROB_CAUSE(scm, observation, values)
    probabilities ← []
    if observation is possible then
        Abduction on N_X and N_Y: {X : o[X], Y : o[Y], S : o[S]}
        prob_x, prob_y, prob_xy ← [], [], []
        for i in values do
            if i ≠ observation["X"] then
                prob_X ←prob(¬o[S], do(X = i) |N_X, N_Y = abducted)
                prob_x.append(prob_X)
            else
                prob_x.append(0.0)
            end if
            if i ≠ observation[Y] then
                prob_Y ←prob(¬o[S], do(Y = i) |N_X, N_Y = abducted)
                prob_y.append(prob_Y)
            else
                prob_y.append(0.0)
            end if
        end for
        probabilities.append(max(prob_x))
        probabilities.append(max(prob_y))
    else ▷ If the observation is not possible, all 0 probabilities will be returned
        probabilities.append(0.0)
        probabilities.append(0.0)
    end if
    return probabilities
end procedure
```

Figure 6.3: An algorithm for finding the single-variable intervention most likely to change outcome (SVIMLTCO). Source: adapted from [154].

Qualitative Results & Limitations. Sweeping over the finite action set yields simple, scenario-specific probabilistic explanations (Table 6.1). Consider $X = -0.04$, $Y = -0.02$ with an observed unstable outcome ($S =$ False; second row): the method attributes full responsibility to X (1.0) and none to Y (0.0). This matches intuition: holding $X = -0.04$ fixed and varying Y across its range never stabilises the tower, so X is solely implicated. However, SVIMLTCO fails under *overdetermination* (row 1), where either X or Y alone suffices to cause failure; it returns 0.0 for both, contradicting human judgements that would assign both as independent causes. It also under-represents *interaction*: when both X and Y jointly lie in the stability

Action parameters		Is X a cause of S?		Is Y a cause of S?	
X	Y	S = True	S = False	S = True	S = False
-0.04	-0.04	–	0.0	–	0.0
-0.04	-0.02	–	1.0	–	0.0
-0.04	0.0	–	1.0	–	0.0
-0.02	-0.04	–	1.0	–	1.0
-0.02	-0.02	1.0	–	1.0	–
-0.02	0.0	1.0	–	1.0	–

Table 6.1: The probability that the parameterisation of X or Y, independently, was the cause of the observed task outcome (stable or unstable tower), based on the *single-variable intervention most likely to change outcome (SVIMLTCO)* algorithm. We sweep over uniformly sampled values of X and Y, both in the range $[-0.04, +0.04]$ metres in increments of 0.2. Due to the symmetry of the block stacking dynamics, we omit action parametrisations that would yield the same result. Source: adapted from [154].

region (rows 5-6 for successes), human attribution treats both as necessary, but a single-variable probe cannot express joint necessity. These limitations motivate a multi-variable extension.

6.4.3 Method 2: Multi-Variable Intervention Most Likely to Change Outcome (MVIMLTCO)

To address cases where causes interact or exhibit redundancy, the second method extends the single-variable formulation to search over combinations of variables. For a candidate subset of variables $X_S = \{X_{i_1}, X_{i_2}, \ldots, X_{i_m}\}$, the counterfactual probability of outcome change is computed as:

$$P(Y_{X_S=x'_S} \neq y \mid X_S = x_S, Y = y),$$

where x'_S represents an alternative joint assignment to the selected variables. The search is performed using a greedy heuristic that incrementally adds variables that maximise the expected probability of change, balancing tractability and interpretability while preserving minimal change at the subset level. When a single- and a multi-variable explanation are tied, we apply Occam's razor and prefer the simpler (single-variable) explanation.

The algorithm shown in Fig. 6.4 considers both single- and multi-variable interventions and implements the tie-breaking rule above.

Qualitative Results & Limitations. MVIMLTCO correctly captures *overdetermination*: in row 1, XY is identified as a joint cause when both parameters push the placement outside the stable region. It also reveals a structural quirk: because there is usually at least one alternative

Algorithm 2 Calculate the most probable cause of an observation in an SCM

```
procedure PROB_CAUSE(scm, observation, values)
    probabilities ← []
    if observation is possible then
        Abduction on N_X and N_Y: {X : o[X], Y : o[Y], S : o[S]}
        prob_x, prob_y, prob_xy ← [], [], []
        for i in values do
            if i ≠ observation["X"] then
                prob_X ←prob(¬o[S], do(X = i) |N_X, N_Y = abducted)
                prob_x.append(prob_X)
            else
                prob_x.append(0.0)
            end if
            if i ≠ observation[Y] then
                prob_Y ←prob(¬o[S], do(Y = i) |N_X, N_Y = abducted)
                prob_y.append(prob_Y)
            else
                prob_y.append(0.0)
            end if
        end for
        for i in values do                          ▷ This will refer to the X positions
            for j in values do                      ▷ This will refer to the Y positions
                if i ≠ observation[X] AND j ≠ observation[Y] then
                    prob_XY ←prob(¬o[S], do(X = i, Y = j) |N_X, N_Y = abd. )
                    prob_xy.append(prob_XY)
                else
                    prob_xy.append(0.0)
                end if
            end for
        end for
        probabilities.append(max(prob_x))
        probabilities.append(max(prob_y))
        probabilities.append(max(prob_xy))
    else  ▷ If the observation is not possible, all 0 probabilities will be returned
        probabilities.append(0.0)
        probabilities.append(0.0)
        probabilities.append(0.0)
    end if
    return probabilities
end procedure
```

Figure 6.4: An algorithm for finding the value from the single- and multiple-variable sets with the highest probability of being the cause of the observation. Source: adapted from [154].

(X', Y') pairing that flips the outcome, XY tends to appear as a valid joint explanation in many rows. This is formally correct but not always human-aligned, since people often prefer simpler attributions. We therefore prefer a single-variable explanation when exactly one of $\{X, Y\}$ ties with XY (rows 2-4). When all three explanations $\{X, Y, XY\}$ tie (rows 5-6), the

Action parameters		Identified cause(s) of S	
X	Y	S = True	S = False
-0.04	-0.04	–	XY
-0.04	-0.02	–	**X**, XY
-0.04	0.0	–	**X**, XY
-0.02	-0.04	–	**Y**, XY
-0.02	-0.02	X, Y, XY	–
-0.02	-0.02	X, Y, XY	–

Table 6.2: Table showing the probability that the parameterisation of X or Y independently, or X and Y jointly (XY), was the cause of the observed task outcome (stable or unstable tower), based on the *multiple-variable intervention most likely to change outcome (MVIMLTCO)* algorithm. In cases where only one single-variable explanation and the multi-variable explanations are equally probable, the single-variable explanation is chosen (bold) as the simplest one, in line with the principle of Occam's razor. When all three explanations are equally probable (rows 5-6), it is not clear with this method how to quantify the causal contributions of each variable to the outcome, either independently (X, Y) or jointly (XY). Source: adapted from [154].

method cannot disambiguate their relative contributions in a principled way. These ties expose an identifiability gap that motivates a responsibility-based assignment.

6.4.4 Method 3: Responsibility-Based Attribution

The third method attempts to quantify the causal contributions of each variable to the outcome, either independently (X, Y) or jointly (XY). We adopt the responsibility-based method developed by Halpern [63], which states that if a variable X_i is identified as a cause, its responsibility is defined as:

$$R_i = \frac{1}{N_i + 1}, \tag{6.1}$$

where N_i is the number of variables that would need to have occurred differently for the outcome to have changed as a result of altering the analysed cause variable.

Responsibility under Uncertainty. As discussed in Sec. 3.3.2, the quantity N_i is defined with respect to a particular causal model and assignment of variables. Under uncertainty in the model or system state, N_i may vary across possible worlds $\omega \sim p(\omega)$, and responsibility becomes a random variable $R_i(\omega) = \frac{1}{N_i(\omega)+1}$. In this case, a natural generalisation is the expected responsibility:

$$\mathbb{E}[R_i] \;=\; \mathbb{E}_{\omega \sim p(\omega)} \left[\frac{1}{N_i(\omega) + 1} \right]. \tag{6.2}$$

In the primary model considered in this thesis (Fig. 6.2a), however, the structural causal model is deterministic, given the selected action. As a result, there is a single realised world and N_i is uniquely defined, so the expectation reduces to the deterministic form $R_i = \frac{1}{N_i+1}$. In the stochastic SCM variants, uncertainty is incorporated through abduction and probabilistic causal attribution (e.g., PN/PS), but responsibility is computed using this deterministic formulation conditioned on the selected explanation, rather than as an expectation over possible worlds. This corresponds to a point-estimate approximation of responsibility, where uncertainty is resolved prior to attribution rather than propagated through to the responsibility metric itself. Extensions to stochastic models, including placement noise and multi-step dynamics, are discussed in Sec. 6.4.5 and analysed in detail in Radojcic [154].

The Firing Squad Scenario. To illustrate, we recall the firing squad scenario from Pearl [32], in which a court orders the execution of a prisoner by firing squad. The Captain gives the order for two soldiers to shoot; both soldier A and soldier B fire, and the prisoner dies. If we analyse soldier A in isolation, had A not fired, B would still have fired and the prisoner would have died. Likewise, if soldier B had not fired, A would still have fired and the prisoner would have died. Under naive analysis, neither would appear to be a cause. However, this contradicts our human notion of causal judgement: both soldiers are clearly responsible for the death.

Applying Eq. 6.1 for soldier A, we have $N_i = 1$ because one other variable (soldier B not firing) would have had to change for the outcome to depend on A's action. The same applies symmetrically to soldier B. Thus, each soldier's responsibility is:

$$R_i = \frac{1}{N_i + 1} = \frac{1}{1 + 1} = \frac{1}{2},$$

which matches our intuition: each is equally responsible for the outcome, and together they sum to a total responsibility of one. This metric scales naturally with the number of co-occurring causes; for example, if a firing squad had ten soldiers firing simultaneously instead of two, each would have a responsibility of $R = \frac{1}{10}$.

We can also partition the range of responsibility values into a more human-intuitive classification:

- $R = 0$: not a cause of the observed outcome, no responsibility;

Action parameters		Responsibility			
X	Y	$S = True$		$S = False$	
		X	Y	X	Y
-0.04	-0.04	–	–	$\frac{1}{2}$	$\frac{1}{2}$
-0.04	-0.02	–	–	1	0
-0.04	0.0	–	–	1	0
-0.02	-0.04	–	–	0	1
-0.02	-0.02	1	1	–	–
-0.02	0.0	1	1	–	–

Table 6.3: Table showing the probability that each action parametrisation variable caused the observed outcome, based on the *Responsibility Assignment* causal attribution algorithm. We sweep over uniformly sampled values of X and Y, both in the range $[-0.04, +0.04]$ metres in increments of 0.2. Due to the symmetry of the block stacking dynamics, we omit action parametrisations that would yield the same result. Source: adapted from [154].

- $0 < R < 1$: a sufficient cause of the observed outcome, sharing responsibility with one or more other variables;
- $R = 1$: sole cause of the observed outcome, full responsibility.

Application to the block stacking task. The same logic applies to the robot block stacking task: there may be times when an action parametrisation variable is totally, partially, or not at all responsible for the observed task outcome. We use Eq. 6.1 to quantify this in a mathematically principled way. The algorithm shown in Fig. 6.5 — the *Responsibility Assignment* causal attribution algorithm — allocates responsibility to the observed values of potential cause variables, given the probabilities computed from the MVIMLTCO algorithm.

Qualitative Results & Limitations. The computed responsibilities show strong alignment with human causal attributions. In cases where X and Y were jointly identified as necessary causes (row 1), each receives $R = \frac{1}{2}$, reflecting equal shared responsibility. When either, but not both, were independently identified as causes (rows 2-4), the respective cause variable is given $R = 1$, representing sole responsibility for the outcome. In cases where all three explanations are equally probable (rows 5-6), both X and Y independently receive $R = 1$. Here, the observed outcome is a stable tower, so either variable alone could be changed to move the placement outside the 2D region of stability. This differs from the overdetermined case in row 1, where **both** variables must change to alter the outcome from unstable to stable.

Algorithm 3 Calculate Responsibility

```
procedure RESPONSIBILITY_OUT(probabilities)
    labels ← ["X", "Y", "XY"]
    variables ← ["X", "Y"]
    max_value ← max(probabilities)
    max_causes ← [labels[index] for variable that corresponds to max_value]
    responsibility ← {}
    if len(max_causes) > 1 then ▷ 2+ causes identified: only accept individual
        for i in variables do
            if i in max_causes then
                responsibility[i] ← 1
            else
                responsibility[i] ← 0
            end if
        end for
    else
        cause ← max_causes[0]
        if len(cause) > 1 then          ▷ Combined variable identified as cause
            for i in variables do
                if i in cause then
                    responsibility[i] ← 0.5
                else
                    responsibility[i] ← 0
                end if
            end for
        else                            ▷ Individual variable identified as cause
            for i in variables do
                if i in max_causes then
                    responsibility[i] ← 1
                else
                    responsibility[i] ← 0
                end if
            end for
        end if
    end if
    return responsibility
end procedure
```

Figure 6.5: The *Responsibility Assignment* Causal Attribution Algorithm. An algorithm for calculating how much responsibility should be ascribed to each potential cause variable for the observed outcome having had occurred, using the probabilities generated from the MVIMLTCO algorithm. Source: adapted from [154].

Overall, the responsibility metric provides a more human-aligned account of causal attribution than the SVIMLTCO and MVIMLTCO methods, reflecting graded responsibility rather than binary causation. However, two limitations remain: (i) responsibility values depend on the discretisation of the action space and on the set of interventions evaluated; and (ii) in richer task domains, normative factors such as intent, agency, or role expectations may influence perceived

responsibility beyond purely physical causation. These considerations motivate further work, including the human-participant evaluation discussed in Sec. 6.7.

6.4.5 Additional Model Evaluations under Placement Noise & Multi-Step Dynamics.

We also applied the pipeline to the *placement noise* and *multi-step* SCM variants described in Sec. 6.3.1. Detailed quantitative tables are omitted here for brevity, but full experimental outcomes are reported in Radojcic [154].

To summarise findings, the qualitative trends are consistent with the base model but exhibit informative differences:

- **Noise-aware abduction.** Conditioning on factual failures at near-centre placements produced abducted posteriors over block placement noise N that shifted toward larger, sign-consistent offsets (e.g., positive shifts when the observed failure would require overshoot), aligning with the intuition that unusually high noise can explain otherwise stable choices.

- **Previously impossible cases become rare but non-zero.** With block placement noise $N \sim \mathcal{N}(0, \sigma^2)$, positions outside the stability region occasionally succeed, and near-centre placements occasionally fail. As a result, single-variable PS/PN values that were degenerate in the noiseless model became small but non-zero, and varied systematically with σ and proximity to the stability region.

- **Sensitivity of single-variable PN/PS.** Single-variable PN showed heightened sensitivity to discretisation and noise variance: as σ decreases (e.g., $0.03 \rightarrow 0.01$), PN and PS recover the noiseless pattern (centre more sufficient for success; extreme placements more sufficient for failure), whereas higher σ flattens and spreads the scores.

- **Responsibility with noise as a candidate cause.** The responsibility method extends to include block placement noise N as an explanatory variable, and joint causes such as XN and YN. For factual failures near the stability region, $do(N = 0)$ often flips the outcome, attributing substantial responsibility to N. When both placement and noise contribute, joint causes receive split responsibility (e.g., 1/2 each). In rare tri-cause situations (X, Y, and N all necessary), we adopt an Occam-style policy that prefers

the simplest sufficient explanation (e.g., XY) unless high-noise regimes warrant explicit tri-part attribution.

- **Multi-step action selection and explanations.** For sequential stacking, action selection was framed hypothetically via $P(S_k = \text{True} \mid do(A_k = a), S_{k-1} = \text{True})$; explanations of preference used a counterfactual comparison of tower centre-of-mass distances, reporting $P(F_A < F_B \mid A = a, do(A = b))$, where F_A and F_B denote the distance of the tower's centre of mass from the ideal centre under actions A and B, respectively. This yields user-facing justifications of the form 'A is preferable to B because it is more likely to keep the tower closer to centre', even when noise-induced drift accumulates across steps.

Taken together, these observations support the main conclusions from the base model: (i) single- and multi-variable *most-likely-to-change* methods are informative but brittle in overdetermination and interaction cases; (ii) responsibility assignment remains the most robust and human-aligned across added stochasticity and temporal depth; and (iii) noise-aware abduction is essential to avoid misattributing failures near the stability boundary to poor placement rather than to plausible hardware-induced offsets.

The next section (Sec. 6.5) describes how these quantitative attribution results are translated into natural-language explanations for end users, forming the communicative layer of the counterfactual explanation framework.

6.5 Generating Natural-Language Text Explanations

Given our goal of creating a human-interpretable explanation system, we also require a method to convert the numeric responsibility values derived in Sec. 6.3.4 into natural-language text explanations. This conversion provides the communicative layer of the counterfactual explanation framework, allowing users to ask and receive answers to questions such as 'Why did the robot fail its task?' or 'Why did it succeed?'. The approach builds upon prior work by Diehl and Ramirez-Amaro [37], extending their style of causal template generation to counterfactual explanations derived from responsibility-based attributions.

In the Case of Observed Failure Outcomes. The question of 'Why did it fail?' is answerable by constructing a templated text explanation using the variable with the highest responsibility value. Where two variables are tied, the text is adjusted to indicate that they are jointly responsible ($0 < R < 1$). Some examples are given below from Table 6.3, with the identified cause variable(s) highlighted in bold:

- $R = 0$: 'The **chosen x-axis block position** did not cause the block to fail.' (row 4)
- $0 < R < 1$: 'The **chosen x-axis and y-axis block positions** were both responsible for the task failure. Each contributed to the failure, but either one would have been sufficient for the failure to have occurred.' (row 1)
- $R = 1$: 'The task failed solely because of the **chosen x-axis block position**.' (rows 2–3)

These examples illustrate how the explanation templates directly verbalise the responsibility assignments reported in Table 6.3. Each phrasing reflects the degree of causal contribution: null ($R = 0$), shared ($0 < R < 1$), or sole ($R = 1$) responsibility. The approach provides a clear mapping between quantitative counterfactual reasoning and the linguistic structure of human explanations, ensuring that phrasing corresponds to underlying causal semantics.

In the Case of Observed Success Outcomes. The question of 'Why did it succeed?' is answerable by constructing a similar templated text explanation using the variable(s) with the highest responsibility value, with corresponding adjustments when two variables are jointly or independently responsible. Due to the physics of the block stacking task, both the x- and y-position variables must lie within the 2D region of stability for the tower to remain upright. As seen in Table 6.3, the only valid counterfactual queries for observed success cases correspond to configurations where both cause variables are independently responsible ($R = 1$). We therefore generate explanations of the following form (omitting the other two cases here for brevity):

- $R = 1$: 'The **chosen x-axis and y-axis block positions** both caused the task success. Both were needed for this outcome.' (rows 5–6)

This phrasing distinguishes between *independent sufficiency* in failure cases (where either variable alone could cause instability) and *joint necessity* in success cases (where both must

be correct for stability). Such contrastive phrasing mirrors established human reasoning patterns, in which causes of failure are often singular or sufficient, whereas causes of success are often conjunctive or necessary.

System Integration. To support deployment in embodied systems, the natural-language explanation component is implemented as a modular interface within the robot's autonomy architecture. Responsibility values computed in Sec. 6.3.4 are passed as structured data to the explanation module, which fills the corresponding text templates and returns natural-language responses to user queries. This module is integrated with the *Ethical Black Box* (EBB) [43], allowing post-hoc retrieval of causal explanations for real or simulated episodes. Because the causal inference and explanation stages are decoupled, the same framework can be extended to visual or spoken explanations with minimal modification.

Qualitative Results & Limitations. The generated text explanations demonstrate strong alignment with both the quantitative results of Table 6.3 and human causal intuitions. They clearly differentiate between single, shared, and null causation, producing statements that are faithful to the counterfactual logic of the underlying SCM. The templated structure ensures interpretability and consistency across diverse robot actions, while still remaining grounded in mathematically derived causal attributions.

However, several limitations remain. First, the templates currently discretise responsibility into categorical forms ($R = 0$, $0 < R < 1$, $R = 1$), which may oversimplify partial or uncertain responsibility in more complex domains. Second, the expressiveness of the generated text is limited by the fixed template design and domain-specific vocabulary, potentially restricting generalisation to higher-dimensional tasks. Finally, the current phrasing is purely descriptive and does not yet incorporate *normative or ethical context* (e.g., intent, justification, or moral responsibility), which are crucial for fully human-aligned explanation. These extensions are discussed further in Sec. 6.7, which explores future directions for integrating causal reasoning, human feedback, and linguistic expressiveness.

6.6 Robot Explainer System & RoboTIPS Demonstration

Building upon the responsibility-based counterfactual explanation framework presented in Sec. 6.4.4 and the natural-language generation pipeline described in Sec. 6.5, we implemented a full *Robot Explainer System* for embodied demonstration and evaluation. This system integrates causal reasoning, counterfactual explanation, and human-facing communication within a unified architecture, providing a practical embodiment of the methods developed in this chapter.

The explainer system employed the *placement noise* SCM variant described in Sec. 6.3.1, allowing the generated explanations to account for stochastic discrepancies between intended and actual block placements. This choice reflects the physical realities of the robot hardware, where small actuation errors or slippage can alter the block's final position. By incorporating placement noise as an explicit causal variable, the system can correctly attribute certain failures to execution uncertainty rather than poor decision-making, thus producing more faithful and human-aligned explanations during real-world operation.

System Overview & Integration. We developed a human-robot natural-language explanation system for a human-support robot, based on the responsibility assignment method described in Sec. 6.4.4. The explainer system was integrated with an *Ethical Black Box* (EBB) data-recording framework developed under the *RoboTIPS: Developing Responsible Robotics for the Digital Economy* project [155]. This integration allowed causal explanations to be recorded alongside system logs for post-hoc analysis, enabling reconstruction of both the robot's factual and counterfactual reasoning during task execution. By coupling the EBB with the counterfactual inference pipeline, we established an interpretable and auditable sense-think-act explanation mechanism.

Sense-Think-Act Explanation Pipeline. The explainer system operates through a graphical user interface (GUI) that provides user-level control and observability across the robot's entire sense-think-act cycle. Through the GUI, a user can:

1. **Make a decision:** trigger the robot to select where to place a specified block onto the existing tower, based on its observation of the world state and internal causal world model;

2. **Explain its decision-making:** request a text explanation of the robot's decision-making process before execution;

3. **Execute its action:** command the robot to perform the chosen manipulation, physically attempting the block placement in the real world;

4. **Abort safely:** interrupt or cancel an ongoing action, ensuring user control and physical safety during manipulation;

5. **Describe the episode:** after each action attempt, generate a textual summary of the robot's observations, decisions, and actions during the episode; and

6. **Explain the outcome:** query the system to answer the question 'Why did this task succeed or fail?', generating a counterfactual explanation based on the responsibility-based causal analysis.

This sequence forms a closed-loop explanation workflow in which the robot not only acts in the physical world but can also justify and communicate its reasoning and outcomes to a human observer. The explainer system thereby links the robot's internal decision structure to an interpretable, user-facing narrative of both intention and outcome.

For each episode, the EBB records the robot's state and action trajectories, which are subsequently processed by the counterfactual inference pipeline described in Sec. 6.4.1. The explanation subsystem applies the responsibility-based attribution method to identify the most influential causal variables and generate a corresponding natural-language explanation. In cases of failure, the system identifies the variable(s) most responsible for the undesirable outcome; in cases of success, it verbalises the set of necessary causal factors that jointly produced the desired result. This process mirrors the templated explanation structures presented in Sec. 6.5, thereby linking counterfactual reasoning to natural-language communication.

To ensure temporal consistency between sensing, decision-making, and explanation generation, the system operates on latched snapshots of the robot state and observations at each decision point. This behaviour is analogous to a *D-latch*, in which the system samples and holds a consistent view of the relevant variables long enough for causal inference and explanation to be performed. In practice, this means that both the forward decision-making query and the subsequent counterfactual explanation are grounded in the same fixed state representation,

even as the physical environment may continue to evolve. This avoids inconsistencies that could otherwise arise from asynchronous sensing or delayed execution, and ensures that the generated explanations remain causally well-defined with respect to the decision context.

Live Demonstration & Deployment. The integrated explainer system was demonstrated as part of the June 2024 *RoboTIPS Showcase Event* (Fig. 6.6). The demonstration featured a Toyota Human Support Robot (HSR) performing an exemplar manipulation task with real-time counterfactual explanation generation. The GUI displayed the robot's planned and executed actions, counterfactual simulation results, and the corresponding textual explanations. This event provided a public-facing demonstration of the explainer system's ability to communicate causal reasoning and task accountability through natural-language interaction.

Qualitative Results & Impact. The demonstration validated that responsibility-based counterfactual explanations can be effectively embedded in an embodied robotic system and communicated to human observers in real time. The system successfully generated intelligible, context-specific explanations that aligned with human causal intuitions for both successful and failed task outcomes. By linking quantitative causal reasoning to natural-language communication, the demonstration provided a practical validation of the framework's potential to enhance transparency, accountability, and trust in autonomous systems. This experiment also evidenced a direct contribution to the aims of the *RoboTIPS* project, showing how the Ethical Black Box can be coupled with causal explanation methods to support responsible robotics in real-world deployments.

6.7 Limitations & Future Work

The counterfactual explanation framework presented in this chapter demonstrates that structural causal models (SCMs) can be used to generate faithful, interpretable, and human-aligned explanations of robot decision-making. However, several limitations remain that motivate future extensions and empirical validation. We group these into two areas: current methodological limitations, and planned future work toward human-participant evaluation.

Figure 6.6: The counterfactual-based robot explanation module demonstrated at the June 2024 RoboTIPS Showcase Event. Top: the explainer module GUI, with controls and visualisations for decision-making, action execution, and explanation generation. Bottom: real-world demonstration of the explainer system using the Toyota HSR performing a manipulation task.

6.7.1 Current Limitations

While the framework establishes a strong proof of concept, several aspects constrain its generality, interpretability, and scalability across broader robotic domains.

Model Scope & Generalisation. The block stacking domain provides a controlled environment for evaluating causal explanation methods but captures only a narrow subset of real-world robotic complexity. Our SCM formulation assumes full state observability (an MDP abstraction), which omits the partial observability, sensor uncertainty, and dynamic occlusions present in real manipulation scenarios. Although this simplification allowed us to focus on the explanation mechanisms themselves, future work should extend the approach to POMDP-based causal representations, validating whether the responsibility-based metrics generalise under uncertainty.

Despite the successful initial qualitative and quantitative proof-of-concept, the method we used to aggregate counterfactual quantities and identify the most likely cause of the observed robot task outcome is not expected to generalise well to other domains. The choice of method relies on strong assumptions of human behaviour in multi-domain systems (e.g., cyber-physical systems), particularly how we, as researchers, *believe* humans tend to perform causal attribution. We do not currently have empirical evidence to support this hypothesis. Thus, additional exploration is required to determine how to combine counterfactual quantities (e.g., probabilities of necessity and sufficiency) and scenario semantics (e.g., decisions, perceptions, actions, outcomes) to develop explanation methods that exhibit closer alignment with human causal attribution dimensions, including degrees of normality or similarity when compared across people or agents (consensus), stimuli (distinctiveness), and temporal occurrences (consistency) [156].

Although counterfactual explanations have been shown to align well with human causal judgements in simple physical systems [62], it remains an open research question how to assess and rank the relative contributions of multiple causal variables to an outcome of interest in a way that reflects human causal attribution biases. In more complex mixed physical-social systems involving human or artificial agents, human observers use different types of attribution depending on the context, and these attributions often exhibit multiple cognitive biases. Consequently, humans may assign different weightings to each causal variable depending on the situation and their preconceptions of the roles they expect agents to play [156]. Understanding

and modelling these attributional factors remains a key challenge for developing more human-aligned counterfactual explanation frameworks.

Attribution Biases & Causal Completeness. The responsibility metric provides a clear quantitative interpretation of causal contribution, but its accuracy and robustness depend on the completeness and correctness of the underlying SCM. If relevant variables are omitted, or if latent confounders are not modelled, the explanation may misattribute responsibility, over-attribute effects to observed variables, or fail to capture joint causal interactions. In such cases, the explanations remain internally consistent with respect to the assumed model, but may not reflect the true causal mechanisms of the physical system. This highlights an important assumption of the framework: explanations are only as valid as the causal model on which they are based. In practice, this implies that model misspecification propagates directly into both decision-making and explanation generation, potentially leading to systematically biased or incomplete explanations. However, the structured nature of SCMs provides a degree of robustness compared to purely data-driven approaches, as assumptions about causal structure are explicit and can be inspected, refined, and incrementally extended. Future work should explore methods for handling model uncertainty — for example, by maintaining distributions over causal structures, incorporating latent variable models, or performing sensitivity analysis to assess how explanations change under alternative structural assumptions.

Expressivity of Natural-Language Generation. The natural-language explanations developed in Sec. 6.5 rely on a templated structure that maps discrete responsibility values to predefined sentence patterns. This design is intentional: it ensures that the generated explanations remain faithful to the underlying causal reasoning process, providing deterministic, transparent, and reproducible mappings from causal quantities to language. However, this approach limits linguistic variability and nuance, especially in multi-cause scenarios or when uncertainty needs to be expressed. Two methods could be used to increase diversity in the generated text. First, the system could randomly select from a larger library of suitable text templates that encode semantically equivalent but stylistically varied phrasings. This would preserve interpretability while improving readability and user engagement. Second, an alternative is to provide the identified causal variables and symbolic explanation structure

as input to a large language model (LLM), prompting it to generate one or more natural-language renderings of the same underlying explanation (see Sec. 7.6.1). In this case, the LLM is not responsible for performing the causal reasoning itself, but only for automating the linguistic realisation of the explanation. This separation avoids the known limitations of LLMs in counterfactual reasoning, as discussed by Kcman et al. [157], while still leveraging their strengths in natural-language generation. More expressive language models, when grounded in the same causal reasoning framework, could ultimately enable explanations that capture richer relations such as intent, justification, or trade-offs in decision-making.

Embodied Deployment Constraints. Although the RoboTIPS demonstration (Sec. 6.6) validated the approach on a physical robot, the current prototype is limited by real-time computational constraints and manual triggering of the explanation cycle. Integrating continuous causal monitoring and on-demand explanation generation will require further optimisation and parallelisation of the twin-world inference pipeline, as well as automated logging of sensor and actuator data streams in the Ethical Black Box architecture.

6.7.2 Future Work: Human-Participant Evaluation

A key direction for future work is the empirical evaluation of counterfactual explanations with human participants, to assess their impact on interpretability, causal alignment, and trust in robot autonomy. While prior work suggests that counterfactual explanations align well with human causal reasoning, the framework presented in this chapter has not yet been validated through user studies.

A detailed study design was developed, including experimental protocols, task domains, and evaluation criteria for comparing explanation types. However, due to time constraints, this study was not conducted as part of this thesis. The design is therefore presented as illustrative future work and is included in Appendix A for completeness.

In summary, such a study would enable systematic evaluation of whether counterfactual explanations improve human understanding of robot behaviour and support more calibrated trust in autonomous systems.

6.8 Summary

This chapter directly addressed **Q5 - Counterfactual Reasoning** and **Q6 - Counterfactual Explanations**, and contributed to **Q1 - Modelling** by extending the causal world model introduced in Chapter 5 to support post-hoc counterfactual explanation of robot task execution. Our central goal was to develop methods that enable a robot to answer questions such as 'Why did this task succeed or fail?' in a manner that is both faithful to its underlying causal model and aligned with human causal judgements.

To achieve this, we reformulated the block stacking domain as a structural causal model (SCM) expressed within a Markov decision process (MDP) abstraction (Sec. 6.3.1). This formulation explicitly separated endogenous variables, deterministic update functions, and exogenous noise sources, thereby enabling counterfactual reasoning at Pearl's level 3. On this foundation, we developed a modular explanation pipeline (Sec. 6.4.1) and introduced three complementary causal attribution methods: (1) the **Single-Variable Intervention Most Likely To Change Outcome (SVIMLTCO)**, (2) the **Multi-Variable Intervention Most Likely To Change Outcome (MVIMLTCO)**, and (3) the **Responsibility Assignment** method based on Halpern's notion of actual causality. Together, these provided a hierarchy of explanation fidelity, from minimal single-variable sensitivity analysis to graded measures of shared causal responsibility.

Qualitative analysis of these methods (Secs. 6.4.2–6.4.4) showed that the responsibility-based approach best captured human-like causal reasoning. It successfully resolved cases of causal overdetermination and interaction, which challenged the simpler single- and multi-variable methods. This supported the chapter's first objective of *faithfulness* —that explanations should remain consistent with the system's structural causal mechanisms — and its second objective of *human alignment* —that explanations should reflect intuitive judgements about necessity, sufficiency, and shared responsibility.

Building upon these quantitative measures, we developed a natural-language generation layer (Sec. 6.5) that translated responsibility scores into concise text templates. This allowed the robot to express its counterfactual reasoning in a form interpretable to non-expert users, providing a communicative bridge between causal inference and user-facing explanation. The integration of this reasoning and communication capability was demonstrated in the *Robot*

Explainer System developed for the RoboTIPS project (Sec. 6.6), which embodied the full sense-think-act explanation cycle. In a live demonstration using the Toyota HSR robot, the system generated explanations of observed task outcomes grounded in its causal model, contributing to the RoboTIPS initiative on responsible robotics and ethical accountability through the Ethical Black Box framework.

The chapter's methodological and system-level contributions can be summarised as follows:

1. An SCM-based causal world-model extension of the block-stacking domain that supports conditioning and interventions for counterfactual reasoning;

2. Three complementary causal attribution algorithms for identifying and quantifying causal responsibility;

3. A lightweight natural-language generation layer for transforming quantitative causal measures into interpretable explanations; and

4. A functional robot explainer prototype demonstrating counterfactual sense-think-act explanations within the RoboTIPS framework.

In combination, these contributions advance the field of explainable robotics by grounding explanations in explicit structural causal models rather than heuristic or symbolic reasoning. They provide an operational framework that links causal inference, probabilistic reasoning, and natural-language communication in embodied systems. While the chapter demonstrated a proof-of-concept qualitatively, the proposed human-participant study (Sec. 6.7.2) outlines the next step toward empirical validation of human alignment and trust calibration.

By connecting causal reasoning for decision-making to causal reasoning for explanation, this chapter establishes the methodological foundation for the next stage of the thesis. The following chapter (7) extends these principles to learning-based world models, exploring how counterfactual and contrastive reasoning can be integrated into deep generative architectures to achieve causal alignment in data-driven settings.

7

Counterfactual Contrastive Learning for Improving Causal Consistency in Multi-Modal GenAI Models

Contents

7.1 Introduction 193
7.2 Problem Statement & Causal Consistency Criterion 195
7.2.1 Motivating Application Domain: dSprites Counterfactual Image Editing 196
7.2.2 Counterfactual Image Fine-Tuning Task. 197
7.3 SCM View of Text-Image Diffusion & the Parallel-World Procedure 199
7.4 Method: Counterfactual Contrastive Learning for Diffusion 203
7.5 Text-to-Image Latent Diffusion Architecture 207
7.6 Experiments on dSprites 208
7.6.1 Dataset & Pair Construction 208
7.6.2 Counterfactual Image-Editing Tasks 209
7.6.3 Metrics for Causal Consistency 210
7.6.4 Qualitative Results 211
7.7 Learning Induced Conditional Dependencies 213
7.7.1 Conditional dSprites Variant 213
7.7.2 Evaluation & Findings 213
7.8 Limitations & Future Work 214
7.9 Broader Multi-Modal Pointer (Scope Note) 216
7.10 Summary 216

7.1 Introduction

In this chapter, we consider how formal structural causal modelling and counterfactual reasoning methods can be used to improve *causal consistency* in multi-modal **deep** generative AI (genAI) models, enabling generated representations of physical and interactive scenarios to better align with human-centred principles of causality. As a culmination of previous thesis work, this chapter brings together the formal methods, representations, and insights developed in Chapters 4–6, addressing **Q1 - Modelling**, **Q2 - Structure and Parametrisation**, and **Q5 - Counterfactual Reasoning**.

Recent advances in deep multi-modal generative AI have yielded powerful, general-purpose models capable of generating and editing coherent, diverse, and high-fidelity outputs across modalities such as text, image, and video. Latent diffusion architectures, such as DALL-E [66] and Stable Diffusion [130], effectively align image features with text semantics via CLIP-based encodings and attention mechanisms, enabling remarkable text-to-image synthesis and variation. However, despite being *generative models*, these systems typically lack a structured representation of causal relationships between elements in a scene. They operate primarily at the level of statistical association — what Judea Pearl terms *level-one* knowledge in the ladder of causation [6]. As a result, when tasked with modifying part of a scene (for example, changing one object for another), they may unintentionally alter unrelated elements, violating the principle of *causal consistency* — only variables causally downstream of an intervention should change (Sec. 3.2.2).

Consider a scenario in which a robot uses a text-conditioned image generator to reason about possible outcomes of its block-stacking task (Ch. 5), visualising how different final tower configurations might appear. The robot first prompts the model to generate an image of a red block being placed at a candidate location on the block tower. It then modifies the prompt to change only the top block's colour to blue. Suppose that when regenerating the image, the model correctly changes the block's colour but also alters the block's dimensions and the scene's lighting, despite these features being unrelated to the requested change. This constitutes a violation of causal consistency, as the model has changed elements of the scene that are not causally downstream of the specified intervention.

Without an internal model of the physical or causal dependencies that govern the elements within the generated world, deep generative models cannot explicitly reason about how changes

should propagate. While level-1 associational representation and reasoning enable the model to learn *what should change* due to a change in prompt, only level-3 counterfactual reasoning can teach the model *what should stay the same*. Consequently, this limitation prevents deep generative systems from maintaining faithfulness to human intuitions of cause and effect.

This issue is particularly consequential for robotics, where generative and reasoning models must maintain causal coherence when reasoning about physical outcomes. The proposed *counterfactual contrastive learning* method is highly relevant to robotics, as it addresses the critical need for robots to reason effectively across multi-modal sensor data and take appropriate actions. Multi-modal models provide robots with the capability to integrate visual, proprioceptive, and action-based information, allowing them to operate in complex environments. However, for tasks requiring credit assignment and root cause attribution (e.g., robot explanations or counterfactual model refinement), robots must possess counterfactual reasoning abilities that rely on strong mechanistic causal models. Such models cannot be derived entirely from observational or experimental data alone. While multi-modal embeddings can capture partial causal structure through natural language semantics or temporal relations (e.g., Granger causality [158]), they often fall short in generating reliable counterfactual predictions. In particular, generated counterfactuals may violate the laws of causal consistency within the robot's environment, leading to implausible or physically inconsistent predictions of action outcomes.

To address these limitations, and contributing directly to **Q1 - Modelling**, **Q2 - Structure and Parametrisation**, and **Q5 - Counterfactual Reasoning**, this chapter investigates how structured causal modelling and counterfactual reasoning can be integrated into the training of deep generative AI models. We adopt Pearl's *principle of causal consistency* as a foundational constraint for generation, aiming to improve alignment between generated outputs and human expectations about cause and effect in both physical and semantic domains.

We propose a novel *counterfactual contrastive learning* method that guides latent diffusion model training to promote text-to-image generation behaviour aligned with four human-centred principles:

1. **Controllability**: the generated image should faithfully reflect semantically meaningful changes in the input prompt;

2. **Causal Structure**: generated content should adhere to coherent, human-aligned causal rules and relationships;

3. **Diversity**: the model should support multiple distinct yet causally valid outputs, reflecting natural variability among plausible counterfactual outcomes; and

4. **Causal consistency**: counterfactual outputs should reflect only the effects of the specified intervention — preserving all other elements of the scene.

The approach formalises text-to-image generation as a structural causal model (SCM), described in Sec. 7.3, and introduces a training-time *parallel-world* procedure (Sec. 7.4) in which factual and counterfactual image pairs share exogenous noise, ensuring minimal-change behaviour under interventions. Architectural extensions (Sec. 7.5) integrate causal-variable embeddings alongside text and image latents, providing a unified substrate for causally grounded conditioning during diffusion.

To support evaluation, we focus on the *dSprites* disentanglement benchmark [47] and a conditional variant that induces dependencies between latent factors (Sec. 7.6–7.7). These datasets enable controlled testing of whether the proposed method improves causal consistency under known factorisations and interventional structure. Broader interactive and multi-modal evaluation is developed formally in *Multiverse Mechanica* (Ch. 8), which provides playable generators and explicit mechanics for causal testing.

A potential future human participant study is also discussed (Sec. 7.8), to assess whether improved causal consistency in deep generative models corresponds to closer alignment with human causal expectations.

The remainder of this chapter is organised as follows. Sec. 7.2 formalises the problem and defines the formal criterion for causal consistency. Sec. 7.3 presents the structural causal model view of text-image diffusion and its parallel-world formulation. Sec. 7.4 and Sec. 7.5 describe the proposed training method and architectural modifications. Empirical studies on dSprites and its conditional variant are reported in Secs. 7.6–7.7, followed by discussion of broader multi-modal implications (Sec. 7.9) and limitations (Sec. 7.8). The chapter concludes with a summary of contributions and findings in Sec. 7.10.

7.2 Problem Statement & Causal Consistency Criterion

Multi-modal **deep** generative AI models aim to produce coherent, controllable outputs across multiple data modalities, such as text, image, and video. However, their generation processes

are typically learned from observational data without explicit causal supervision. This limitation causes them to behave associationally: they reproduce statistically co-occurring features rather than reasoning about the causal dependencies that determine how a particular change should (or should not) propagate through a scene. As discussed in Sec. 7.3, such models can be viewed as implicit structural causal models (SCMs), where the latent noise serves as an exogenous variable and the network implements a deterministic mapping from causes to effects. Within this causal framing, the fundamental problem addressed in this chapter can be expressed as a violation of the **principle of causal consistency**.

7.2.1 Motivating Application Domain: dSprites Counterfactual Image Editing

We consider the task of counterfactual image editing in a controlled, factorised domain using the *dSprites* dataset [47]. The dataset comprises 64×64 binary images of a single white object on a black background, generated from a set of underlying factors including shape, scale, rotation, and spatial position. Example images are shown in Fig. 7.1.

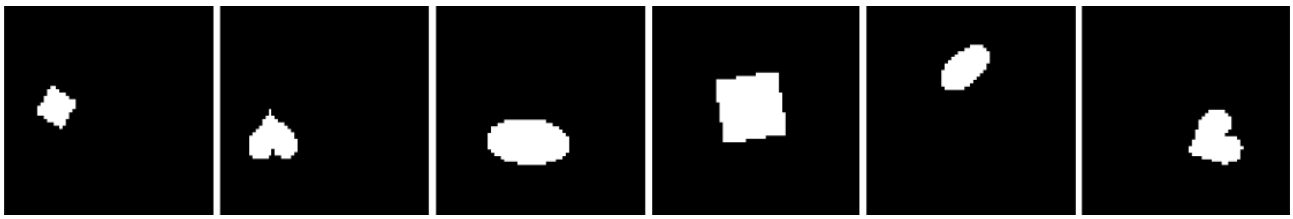


Figure 7.1: Example images from the *dSprites* representation learning dataset.

A key property of dSprites is that its generative factors are constructed via a complete combinatorial sweep over attribute values. As a result, the underlying causal variables are **statistically independent** in the data-generating process. This makes dSprites a suitable testbed for causal modelling, as the ground-truth generative factors are known, disentangled, and free of confounding.

In this chapter, we treat these generative factors as causal variables and study how interventions on individual factors (e.g., changing shape while holding all other factors fixed) affect the generated image. This setting provides a controlled environment for analysing whether generative models produce counterfactual edits that are both faithful to the intervention and consistent with the underlying causal structure.

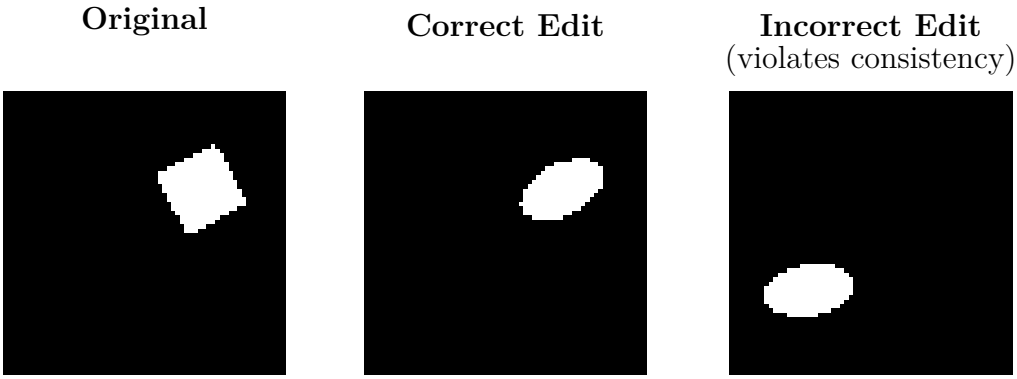


Figure 7.2: Illustration of generative AI image editing. A correct edit changes the object shape (square → ellipse) while modifying only the specified factor (`shape`) and any factors causally downstream of it, leaving all other factors unchanged. The incorrect edit additionally alters rotation and position, introducing unintended changes beyond the specified edit.

However, generative models do not inherently enforce these structural constraints. In practice, counterfactual edits may introduce unintended changes to attributes that should remain invariant.

Fig. 7.2 illustrates this issue in generative AI image editing. Given an original generation, the goal is to apply a targeted edit that modifies a specified factor and any factors causally downstream of it, while leaving all other factors unchanged. While the intended edit changes only the object shape, the incorrect generation also alters rotation and spatial position, introducing changes beyond the specified edit.

This structured, factorised representation of the dSprites data enables a direct correspondence between data-generating factors and nodes in a structural causal model (SCM).

7.2.2 Counterfactual Image Fine-Tuning Task.

We formalise counterfactual image editing as a causal generation problem. Given a factual image and an intervention on one semantic factor, the objective is to generate a corresponding counterfactual image that reflects the intervention while preserving all causally unaffected aspects of the scene.

Let $\mathbf{C}$ denote the set of causal variables and let X denote the generated image. Given a factual assignment $\mathbf{c}$ and an intervention $do(C_k = c_k')$, the goal is to generate a counterfactual image $\tilde{X}$ such that:

1. the intervention on C_k is faithfully realised in $\tilde{X}$, and

2. all variables not causally downstream of C_k remain unchanged.

In the dSprites setting, interventions on causal variables are expressed through corresponding modifications to the textual description.

In this chapter, we instantiate this formulation using a text-to-image latent diffusion model conditioned on a textual description T, where $X = f_\theta(U, T)$, with U denoting exogenous noise. Given a modified prompt T' encoding an intervention (e.g., 'change the square to an ellipse'), the model generates $\tilde{X} = f_\theta(U, T')$.

This formulation corresponds directly to Pearl's consistency rule, which requires that factual and counterfactual worlds coincide on variables not affected by the intervention.

Definition (Causal Consistency). Let $\mathbf{C}$ denote the set of causal variables represented in the image. We partition $\mathbf{C}$ into three subsets: the intervened variable C_k, its causal descendants $D(C_k)$, and the unaffected set $U(C_k) = \mathbf{C} \setminus \Big(\{C_k\} \cup D(C_k)\Big)$. A generated counterfactual image $\tilde{X}$ is *causally consistent* with respect to an intervention $C_k \leftarrow c'_k$ if the following conditions hold:

$$\text{(i) } X_{C_k} \neq \tilde{X}_{C_k}, \quad \text{(ii) } X_{D(C_k)} \neq \tilde{X}_{D(C_k)}, \quad \text{and} \quad \text{(iii) } X_{U(C_k)} = \tilde{X}_{U(C_k)}.$$

In words: only the intervened variable and its descendants may change, while all non-descendants remain invariant between factual and counterfactual generations.

Standard contrastive learning objectives in diffusion models do not enforce these cross-world constraints. They encourage semantic alignment between paired text-image samples, but are blind to whether differences between generated images correspond to causally relevant or irrelevant factors. As a result, edits to one feature often induce uncontrolled collateral changes elsewhere in the image.

More fundamentally, controllable image editing requires a model to learn not only *what should change* in response to a prompt intervention, but also *what should remain invariant*. In causal terms, this corresponds exactly to preserving all variables that are not descendants of the intervention. The principle of causal consistency therefore provides a natural optimisation objective: it formalises the requirement that only causally affected components of the scene should vary, while all others remain unchanged.

Our aim, therefore, is to introduce an inductive bias during training that penalises such collateral changes, ensuring that non-descendant features remain stable while descendant features vary appropriately.

In this sense, the counterfactual image fine-tuning problem can be viewed as learning a mapping:

$$(T, T', U) \;\mapsto\; (X, \tilde{X}),$$

that satisfies the causal consistency property defined above. The key challenge is that the model must learn not only *which* features to modify when the prompt changes (*level-1 association*) but also *which* features to preserve (*level-3 counterfactual invariance*). Achieving this requires structured training data that expose the model to both factual and counterfactual variations of the same underlying world state.

Building on the conceptual principles outlined in Sec. 7.1, we seek to train deep genAI models that satisfy these desiderata for controllability, causal structure, diversity, and causal consistency.

The next section (Sec. 7.3) formalises this reasoning within a structural causal model of text-image diffusion and introduces the parallel-world formulation used to operationalise the counterfactual consistency criterion during training.

7.3 SCM View of Text-Image Diffusion & the Parallel-World Procedure

To operationalise causal consistency in deep generative models, we must first establish a causal interpretation of the text-image diffusion process. Diffusion models can be expressed naturally as structural causal models (SCMs) that map exogenous noise to observed data through deterministic mechanisms conditioned on textual input. This causal perspective allows us to define interventions on semantic factors in the text and to generate counterfactual samples by holding the exogenous variables fixed while altering the conditioning inputs.

Diffusion as a Structural Causal Model. Let U denote the exogenous noise latent drawn from $P(U) = \mathcal{N}(0, I)$, and let T represent a text embedding describing the intended scene. A text-conditioned latent diffusion model defines a mapping

$$X = f_\theta(U, T),$$

where X is the generated image and f_θ is the learned denoising network parametrised by θ.

We interpret this mapping through the lens of a structural causal model (SCM) by separating the stochastic and deterministic components of the generation process. The randomness

of the model is entirely captured by the exogenous variable U, which specifies the initial noise realisation (and, equivalently, the full reverse diffusion trajectory). Conditioned on a fixed U and prompt T, the reverse denoising process implemented by the U-Net is deterministic, defining a structural mechanism that maps causes to effects.

Under this interpretation, U plays the role of an exogenous variable capturing background factors (e.g., lighting, texture, viewpoint), while T acts as a manipulable cause that specifies high-level semantic constraints. The learned mapping f_θ therefore corresponds to the structural equations of an SCM, where the generated image X is a deterministic function of its parents (U, T). This perspective allows interventions on T to be interpreted causally, and enables counterfactual reasoning by holding U fixed while modifying T.

Intervention & Counterfactual Generation. In the SCM framework, an intervention corresponds to substituting the input variable T with a modified value T', yielding a new outcome $\tilde{X} = f_\theta(U, T')$. By fixing U while changing T, we create a *counterfactual world* that reflects what the same underlying latent causes would produce under the new prompt. This directly instantiates Pearl's *Abduction-Action-Prediction* (AAP) procedure (Sec. 3.2.1):

1. **Abduction:** infer or sample U that explains a factual image X given prompt T;
2. **Action:** intervene by modifying T to T';
3. **Prediction:** generate the counterfactual image $\tilde{X} = f_\theta(U, T')$.

This provides a clear causal semantics for diffusion-based generation and enables the notion of *parallel worlds* — a set of images that differ only due to explicitly specified interventions.

Parallel-World Formulation. To embed causal consistency into model training, we extend this idea to a *parallel-world* procedure. The model jointly processes two worlds — one factual (T, X) and one counterfactual $(T', \tilde{X})$ — that share the same exogenous noise U. The shared U enforces a common latent basis across worlds, while differences between (T, T') represent causal interventions. The model's objective is to learn a mapping $(T, T', U) \mapsto (X, \tilde{X})$ that satisfies the causal consistency criterion of Sec. 7.2: only the descendants of the intervened factors may differ between X and $\tilde{X}$.

Under this shared-noise formulation, the factual and counterfactual generations correspond to two reverse diffusion trajectories initialised from the same latent noise realisation U, instantiated via a shared random seed. Here, U captures the full stochastic realisation of the diffusion process, including both the initial noise and any step-wise noise terms. Since the denoising process is deterministic given (U, T), the factual and counterfactual trajectories follow a common underlying stochastic realisation and remain closely aligned throughout the reverse process, deviating only where required by the intervention encoded in T'. Intuitively, this behaviour is analogous to a *Brownian bridge* in the sense that the factual and counterfactual generations are constrained to evolve from a shared stochastic realisation while remaining closely coupled throughout the reverse diffusion process. However, the counterfactual L_2 loss used in this chapter is in fact stronger than this analogy suggests: it explicitly penalises differences between the predicted noise terms at each denoising step, thereby encouraging the two denoising trajectories to coincide as closely as possible in a piecewise sense. This makes the Chapter 7 training objective overconstraining, because factual and counterfactual trajectories should not remain identical everywhere — they must ultimately diverge in ways that reflect the effect of the intervention. This observation motivates the refined loss formulation introduced in Chapter 8, where trajectory alignment is enforced more selectively — preserving global semantic structure while allowing intervention-specific differences to emerge during later stages of the reverse diffusion process.

When only two worlds are considered — a factual and a counterfactual instance — this reduces to the *twin-world* case of the parallel-world formulation. In the context of image generation or editing, the factual and counterfactual correspond to the original and modified images respectively. This two-world configuration provides a minimal yet sufficient structure for enforcing counterfactual alignment in diffusion training.

Causal Structure of the Data Generation Process. Before introducing the parallel-world formulation, we first outline the structural causal model that governs the generation of the dSprites data itself. Figure 7.3 shows the *causal DAG underpinning the SCM formulation* of the probabilistic data generation process used to sample the causal variables of a dSprites sprite and to generate its corresponding image and caption. The variables are partitioned into exogenous variables (denoted with N subscripts) encapsulating all system randomness, and endogenous variables, each defined as a deterministic function of their parent variables in the

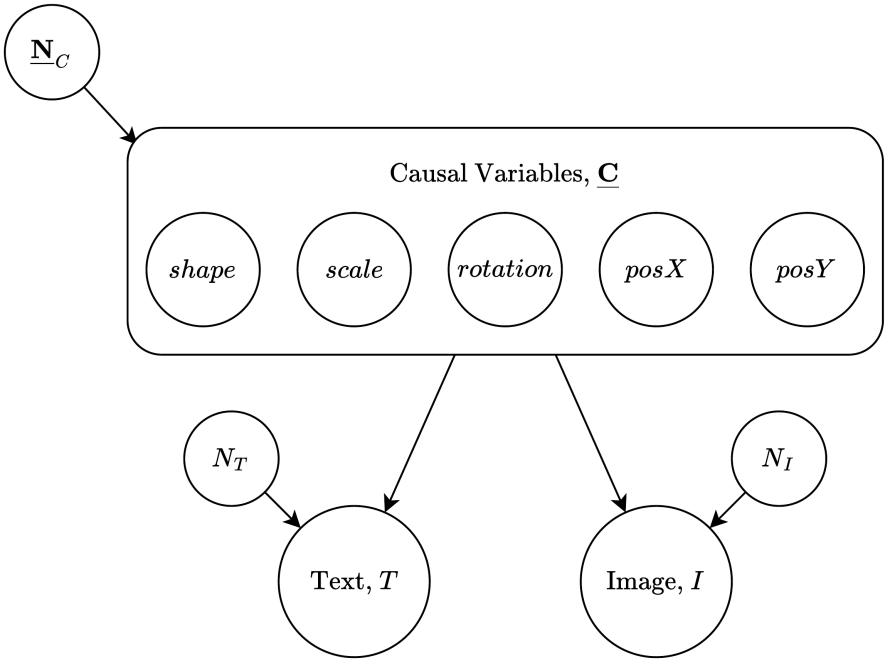


Figure 7.3: The causal DAG underlying the SCM formulation of the probabilistic data-generation process for a dSprites sprite, including (i) causal variables, (ii) image, and (iii) text caption. The variables $\underline{\mathbf{C}} = \{\text{shape}, \text{scale}, \text{rotation}, \text{posX}, \text{posY}\}$ are shown grouped for visual clarity, but each represents an independent causal variable with its own directed edges to both the text T and image I outputs. The bounding box does not denote a single node or joint variable, but a conceptual grouping of parent variables. Exogenous variables (denoted with N subscripts) capture system randomness, while endogenous variables are defined as deterministic functions of their parents. This representation aligns the dSprites data-generation process with the structure of a structural causal model (SCM).

graph. This formulation enables the data generation process to conform to the structure of a structural causal model (SCM), thereby supporting Pearlean counterfactual queries.

The structure of this SCM causal DAG has been designed to map directly to components of the latent diffusion text-to-image process. It formalises how humans might mentally generate images and textual descriptions based on underlying semantic content. First, the set of causal variables $\underline{\mathbf{C}}$ representing image semantics is assigned values through a deterministic mapping

$$\underline{\mathbf{c}} \coloneqq f_C(\underline{\mathbf{n}}_C),$$

where f_C transforms the exogenous noise $\underline{\mathbf{N}}_C$ into sampled causal variable values. Given these semantic variable assignments, the text caption and image are then generated independently as deterministic functions of the causal variables and their respective exogenous noise terms.

These components of the *forward* generation process correspond directly to the *denoising* process used in text-to-image latent diffusion models (Fig. 7.5). The CLIP text encoder [125] infers a latent representation of the causal variables expressed in the caption text, $q(\underline{\mathbf{C}} \mid T)$, while

the image VAE encoder [159] infers a latent representation of the causal variables depicted in the image, $q(\underline{\mathbf{C}} \mid X)$. These latent representations are combined and passed to the denoising U-Net to infer the latent-space reconstruction, after which the VAE decoder generates the final image.

Our hypothesis is that if the latent diffusion architecture is modified to operate explicitly with the exogenous terms defined in this SCM causal DAG (Fig. 7.3), then the entire text-to-image generation process becomes deterministic given the exogenous variables. Consequently, its sampling and inference behaviour would mirror that of an SCM. If this holds, the modified latent diffusion model would generate fine-tuned images that are: (1) faithful to the model's internal mechanisms, (2) faithful to level-3 counterfactual reasoning, and (3) aligned with human causal attribution.

Graphical Representation. Building on this generative causal foundation, Figure 7.4 illustrates the corresponding two-world (factual and counterfactual) formulation used during counterfactual contrastive training. Each world corresponds to an instantiation of the causal graph over variables $\{U, T, X\}$, linked by equality constraints on U and by the intervention $T' = \text{do}(T \leftarrow T')$. The causal paths from T and T' to their respective images capture how prompt-level interventions propagate to observable differences. Under the principle of causal consistency, nodes not descended from T — that is, those influenced only by U — must remain unchanged across worlds. This constraint forms the basis for the contrastive loss introduced in Sec. 7.4, which penalises non-descendant drift between parallel generations.

Summary. Viewing text-to-image diffusion through the lens of structural causal models enables a principled definition of counterfactual generation and minimal-change behaviour. The parallel-world formulation operationalises this view, establishing the structural foundation for the counterfactual contrastive learning objective described next. Sec. 7.4 formalises this procedure into a training objective that jointly optimises over factual-counterfactual pairs, enforcing causal consistency by construction.

7.4 Method: Counterfactual Contrastive Learning for Diffusion

Having established the causal interpretation of text-image diffusion, we now describe how causal consistency can be operationalised during model training. The goal is to ensure that the

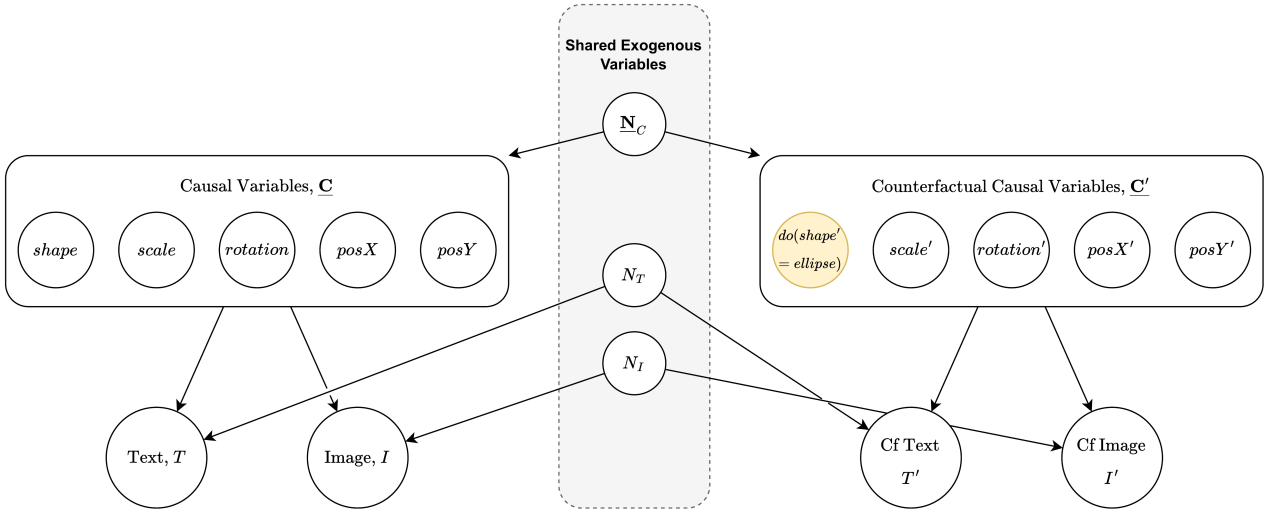


Figure 7.4: Counterfactual image editing twin-world graph. The causal DAG illustrates the twin-world procedure used to compute a Pearlean counterfactual query, exemplified by: 'What would the image have looked like if the shape were an ellipse?'. First, the exogenous noise variables are abducted by performing inference under the factual model (left). Second, an intervention is applied to the counterfactual model (right), setting the shape to an ellipse. Third, the abducted exogenous variables are shared from the factual to the counterfactual world and combined with the intervention to simulate the resulting outcome.

model learns to generate counterfactual image pairs $(X, \tilde{X})$ that differ only in features causally downstream of the specified intervention. We achieve this by pairing factual and counterfactual worlds that share the same exogenous noise latent U, and by introducing a contrastive learning objective that encourages minimal-change behaviour under interventions.

Parallel-World Data Pairing. Each training example comprises a pair of text prompts and corresponding images (T, X) and $(T', \tilde{X})$, describing a factual and a counterfactual scene, respectively. The model samples a single exogenous latent $U \sim \mathcal{N}(0, I)$ and generates the paired images via

$$X = f_\theta(U, T), \qquad \tilde{X} = f_\theta(U, T').$$

Sharing U across both generations ensures that the factual and counterfactual images correspond to the same underlying world instance, differing only in the causal consequences of the prompt-level intervention from T to T'. This design operationalises the *parallel-world* training procedure described in Sec. 7.3 and constitutes the empirical basis for the counterfactual contrastive loss.

Causal Alignment Objective. The learning objective enforces two complementary forms of consistency between the paired worlds:

1. **Descendant Variation:** features causally downstream of the intervened variable should differ appropriately between X and $\tilde{X}$, reflecting the intended intervention;

2. **Non-Descendant Invariance:** all other features (those independent of the intervention) should remain unchanged.

We operationalise the concept of *counterfactual consistency* by implementing a novel training loss for a text-to-image latent diffusion model that explicitly enforces shared exogenous structure between the factual and counterfactual worlds. Although demonstrated here with the dSprites disentanglement benchmark, the principle generalises to any deep generative architecture trained with a contrastive loss, providing a mechanism to embed causal constraints directly within the learning process. We employ a modified *twin-world* process that reverses the flow of steps to ensure both worlds share the same exogenous noise terms at every diffusion step. In the standard twin-world formulation, the procedure follows the sequence *Factual* $\rightarrow$ *Abduction* $\rightarrow$ *Action* $\rightarrow$ *Counterfactual Prediction*, where the counterfactual world is generated after performing an intervention on the factual model. In contrast, our modified version performs *bidirectional abduction*: exogenous terms are abducted independently from both the factual and counterfactual images and then aligned to enforce equality, corresponding to the reversed sequence *Factual Image* $\rightarrow$ *Abduct* $\rightarrow$ *Shared Exogenous Terms* $\leftarrow$ *Abduct* $\leftarrow$ *Counterfactual Image*. This ensures that both worlds remain anchored to a common exogenous base, providing a stable shared reference for comparing their denoising trajectories. Intuitively, for a fixed random seed U, the predicted noise at each denoising step for the factual and counterfactual generations should be almost identical, except where causal descendants of the intervention differ.

To achieve this, we impose an additional mean-squared error (MSE) term penalising the difference between the predicted noise for the factual and counterfactual image pairs, in addition to the standard diffusion loss that matches predicted and ground-truth noise. Let ϵ denote the sampled ground-truth noise, and t the diffusion timestep sampled from the noise scheduler β_t, which controls the variance at each denoising step. Let $\hat{\epsilon}_\theta(X_t, t, T)$ and $\hat{\epsilon}_\theta(\tilde{X}_t, t, T')$ denote

the network's predicted noise for the factual and counterfactual prompts, respectively. The combined training objective can then be written as:

$$\mathcal{L} = \mathbb{E}_{t,U,T,T'}\Big[\underbrace{\|\epsilon - \hat{\epsilon}_\theta(X_t, t, T)\|_2^2}_{\text{Standard diffusion loss}} + \underbrace{\|\hat{\epsilon}_\theta(X_t, t, T) - \hat{\epsilon}_\theta(\tilde{X}_t, t, T')\|_2^2}_{\text{Counterfactual consistency loss}}\Big].$$

By the triangle inequality, the discrepancy between the predicted noise for the factual and counterfactual generations is upper-bounded by the sum of their respective deviations from the ground-truth noise:

$$\|\hat{\epsilon}_\theta(X_t, t, T) - \hat{\epsilon}_\theta(\tilde{X}_t, t, T')\|_2^2 \;\leq\; \|\epsilon - \hat{\epsilon}_\theta(X_t, t, T)\|_2^2 + \|\epsilon - \hat{\epsilon}_\theta(\tilde{X}_t, t, T')\|_2^2.$$

Hence, by minimising the addition of these two standard diffusion losses — collectively forming the *counterfactual contrastive loss (CCL)* training objective,

$$\mathcal{L}_{\text{CCL}} = \|\epsilon - \hat{\epsilon}_\theta(X_t, t, T)\|_2^2 + \|\epsilon - \hat{\epsilon}_\theta(\tilde{X}_t, t, T')\|_2^2,$$

the model is indirectly constrained to minimise the cross-world discrepancy $\|\hat{\epsilon}_\theta(X_t, t, T) - \hat{\epsilon}_\theta(\tilde{X}_t, t, T')\|_2^2$.

We adopt an L_2 penalty rather than an L_1 formulation, as it more strongly penalises deviations at each denoising step, enforcing tight alignment of the predicted noise across the factual and counterfactual trajectories. In contrast, an L_1 loss is more tolerant to local discrepancies and encourages agreement in a distributional sense, which can permit step-wise deviations that accumulate over the diffusion process.

In this way, enforcing step-wise alignment of the denoising trajectories enables the model to learn to preserve non-descendant structure across parallel worlds while faithfully representing the intended causal change, thereby operationalising the principle of counterfactual consistency in practice.

Summary. In summary, causal consistency is enforced during diffusion model training through the counterfactual contrastive loss, which pairs factual and counterfactual generations sharing the same exogenous noise. By optimising $\mathcal{L}_{\text{CCL}}$, the model experiences matched stochastic conditions across both worlds, forcing it to express the intervention through semantically appropriate, causally grounded changes while preserving all other features. Over time, this process encourages the emergence of a structured latent geometry in which causal factors become disentangled and *level-3 counterfactual invariance relationships* are learned. Consequently, interventions on a

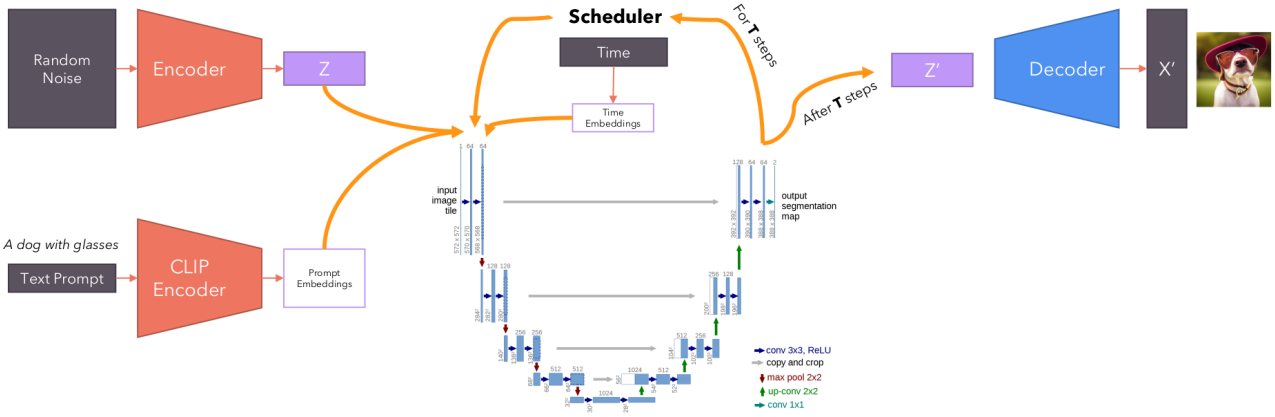


Figure 7.5: The architecture of a text-to-image latent diffusion model. Annotations describe the iterative *denoising* process used during image generation. Source: adapted from `https://github.com/hkproj/pytorch-stable-diffusion`.

single prompt dimension produce targeted modifications while maintaining invariance elsewhere. The next section (Sec. 7.5) details the architectural modifications and conditioning mechanisms required to implement this objective within a text-conditioned latent diffusion framework.

7.5 Text-to-Image Latent Diffusion Architecture

Latent Diffusion Backbone. The counterfactual contrastive training method builds upon the standard text-to-image latent diffusion architecture [130], shown in Fig. 7.5. The model comprises: (i) a **variational autoencoder (VAE)** that encodes input images into a spatial latent space and decodes denoised latents back into pixel space; (ii) a **CLIP-based text encoder** [125] that maps the conditioning prompt into a semantic embedding; and (iii) a **U-Net denoiser** that iteratively predicts noise at each timestep t during the reverse diffusion process, conditioned on both the image and text latents. During generation, the U-Net receives a noisy latent X_t sampled from the forward process and predicts $\hat{\epsilon}_\theta(X_t, t, T)$, from which the next denoised latent is obtained. The noise scheduler β_t defines the stepwise variance across timesteps.

Training Integration. While no architectural modifications were introduced, the proposed method integrates directly into this pipeline through the training loss. The counterfactual contrastive loss $\mathcal{L}_{\text{CCL}}$ is applied to the predicted noise outputs of the U-Net for paired factual and counterfactual prompts (T, T'), sharing the same random seed and hence the same exogenous noise trajectory. This enables a *parallel-world* training regime in which both branches are processed through identical network weights and conditioning pathways, differing only by the

prompt intervention. The VAE and text encoder are trained jointly within this framework to ensure consistent representations across both worlds.

Interpretation. In effect, the latent diffusion backbone provides a natural substrate for representing the structural causal model defined in Sec. 7.3. The shared exogenous latent U corresponds to the diffusion noise seed, the prompt embeddings serve as interventions on causal variables, and the denoising U-Net instantiates the deterministic causal mechanism f_θ. Within this view, the counterfactual contrastive training objective imposes a level-3 constraint on the learned mapping, encouraging invariant causal structure to emerge across parallel denoising trajectories.

The next section (Sec. 7.6) demonstrates this mechanism empirically using the *dSprites* disentanglement dataset [47], a controlled environment in which causal factors and interventions can be explicitly specified and measured.

7.6 Experiments on dSprites

7.6.1 Dataset & Pair Construction

The experiments use the dSprites dataset [47], introduced in Sec. 7.2.1. Each image is indexed by six latent factors: colour, shape, scale, rotation, x-position, and y-position. Since colour is fixed (white), five factors vary combinatorially, yielding approximately 737 k images.

Factual–Counterfactual Pairing. To enable training on parallel worlds with minimal changes, we construct the *Counterfactual dSprites dataset* by sampling structured pairs of examples from the original dSprites dataset. Each pair $(X, \tilde{X})$ and its corresponding text prompts (T, T') differ in exactly one labelled factor C_k, while all other factors are held fixed. For instance, a *shape* intervention takes the form

$$[\text{“square”}, s, r, x, y] \;\rightarrow\; [\text{“ellipse”}, s, r, x, y].$$

A valid counterfactual pairing in which only the shape changes is therefore:

$$[\text{“square”}, 0.9, 0.0, 0.5, 0.5] \;\rightarrow\; [\text{“ellipse”}, 0.9, 0.0, 0.5, 0.5].$$

Here, no non-descendants of the intervened variable are altered, so the counterfactual world satisfies the principle of causal consistency.

Automatic Natural Language Caption Generation Natural-language captions for each factual–counterfactual pair are generated using GPT-4 via a controlled few-shot prompting scheme conditioned on the symbolic latent variables and a structured description of the dSprites domain.

The prompt is programmatically constructed from the latent variable values by converting each attribute (shape, scale, orientation, position) into a structured textual description using predefined templates. This description is embedded within a fixed prompt that specifies (i) the semantic meaning of each latent factor, (ii) the mapping from symbolic attributes to visual properties, and (iii) a set of few-shot examples demonstrating valid caption variations. The model is then prompted to generate a single-sentence caption describing the image without explicitly referencing the underlying symbolic values. See Appendix B for the full prompt template used for caption generation.

This yields controlled linguistic variation while preserving a consistent and semantically grounded mapping between latent factors and textual descriptions. While this introduces some variability due to language generation, the captions remain tightly constrained by the symbolic inputs and prompt structure, limiting the impact of caption noise on training.

These paired samples enable systematic training and evaluation of whether $U(C_k)$ (non-descendant set) is preserved and $D(C_k)$ (descendants) are correctly updated, as defined in Sec. 7.2.

Example Counterfactual Pair An example of a factual–counterfactual pair is shown in Fig. 7.6, illustrating a controlled intervention on the `shape` variable while all other attributes are held fixed. This example highlights the structural property exploited during training: both images share the same underlying exogenous factors, differing only in the causal descendants of the intervention.

7.6.2 Counterfactual Image-Editing Tasks

We evaluate the model under two types of edit regimes that differ in how completely the causal factors are specified in the text prompt:

1. **Fully-Specified Prompts** — all latent factors are explicitly described, with only one factor changing between T and T'. This regime requires no inference over exogenous terms and represents the easier setting.

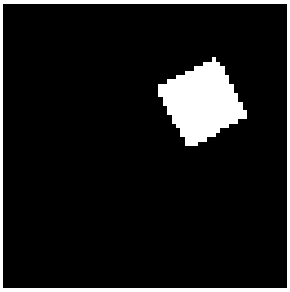
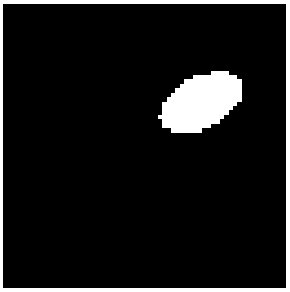
	Factual	**Counterfactual**
Image		
Caption	*A medium-sized square, with a slight rotation to the right, positioned near the right edge and towards the top of the image.*	*A medium-sized ellipse, with a slight rotation to the right, positioned near the right edge and towards the top of the image.*
Attributes		
Shape	square	ellipse
Scale	0.80	0.80
Orientation	0.16	0.16
Position X	0.94	0.94
Position Y	0.10	0.10

Figure 7.6: Example factual–counterfactual pair illustrating an intervention on the `shape` variable. The counterfactual is formed by applying $do(\texttt{shape} = \texttt{ellipse})$ while holding all other attributes fixed. This demonstrates the causal consistency principle used throughout this chapter: only the intervened variable changes, while non-descendant attributes remain invariant.

2. **Partially-Specified Prompts** — some latent factors are intentionally omitted from the factual prompt T, allowing them to be sampled from the model's internal distribution. An intervention is then applied to one specified factor in T', and the omitted (unspecified) factors are expected to remain invariant across worlds. This setting is more challenging, as it depends on accurate inference of the underlying exogenous variables.

These two regimes jointly test the model's controllability, minimality, and invariance under progressively less constrained conditions.

7.6.3 Metrics for Causal Consistency

We propose the following operational metrics aligned with Sec. 7.2. They are introduced to standardise future quantitative evaluation; in this chapter, they are applied qualitatively to assess generated factual–counterfactual pairs. Where relevant, we note how each metric could be computed exactly on dSprites using ground-truth labels.

1) Edit Success (Target Factor). Did the intervened factor C_k reach its requested value under T'? On dSprites, this can be computed exactly using labels:

$$\text{EditSuccess}(C_k) \;=\; \mathbb{1}\Big[\hat{c}_k(\tilde{X}) = c'_k\Big].$$

Use in this chapter: assessed qualitatively.

2) Non-Descendant Preservation (NDP). For each non-affected factor $c \in U(C_k)$, measure invariance between worlds. Specifically, we compute the fraction of non-affected factors $c \in U(C_k)$ that are preserved (i.e., consistent) across worlds:

$$\text{Preserve}(c) \;=\; \mathbb{1}\Big[\hat{c}(X) = \hat{c}(\tilde{X})\Big], \qquad \text{NDP} \;=\; \frac{1}{|U(C_k)|} \sum_{c \in U(C_k)} \text{Preserve}(c).$$

This captures the core 'what should stay the same' requirement. *Use in this chapter:* assessed qualitatively.

3) Diversity Under Constraints (Stochastic Models Only). In stochastic deep generative models, limited observability or weak constraints in the factual world may leave parts of the exogenous noise U underdetermined during abduction. When an intervention T' reveals new causal pathways or previously unobserved aspects of the scene, multiple valid counterfactuals may be consistent with the same factual sample. A causally faithful model should therefore express diversity only within the causal descendants $D(C_k)$ affected by the intervention, while maintaining invariance in non-descendants $U(C_k)$. Operationally, this can be evaluated by sampling from the posterior over plausible U (rather than arbitrary noise) and measuring variance across the resulting counterfactual generations, restricted to $D(C_k)$. *Applicability:* since the dSprites attributes are fully observed and the rendering process is deterministic given these attributes, this metric is not applicable here and is therefore not analysed in this chapter.

7.6.4 Qualitative Results

Due to pending approval for the release of experimental results from Microsoft Research, the visual comparisons and qualitative examples are not included in this submission. These materials will be incorporated in the post-submission revisions of the thesis, following completion of the internal approval process. The following summary therefore provides a high-level qualitative synthesis of observed trends during evaluation, without reproducing side-by-side grids or numerical tables.

Preliminary assessments indicate that the proposed training objective produces targeted edits that respect the requested intervention while leaving unrelated factors visually unchanged. The most consistent improvements were observed for *shape* and *rotation* interventions, where baseline models frequently exhibited positional drift or small-scale artefacts. Under counterfactual contrastive training, the generated counterfactuals appeared to maintain spatial alignment and preserve non-descendant features more faithfully, in accordance with the causal consistency criterion defined in Sec. 7.2. These early qualitative findings suggest that introducing shared exogenous structure during training promotes more stable and causally coherent generative behaviour, though formal quantitative evaluation remains an important direction for post-submission work (see Sec. 7.8).

7.6.4.1 Error Analysis

Qualitative inspection of generated factual–counterfactual pairs revealed several recurring failure modes that highlight current architectural and training limitations:

- **Geometric Artefacts:** occasional incomplete or irregular shapes, including edge speckling and partial occlusions. These effects likely arise from the counterfactual contrastive loss over-constraining the counterfactual denoising trajectory to track the factual trajectory too closely. Because the loss is applied across all diffusion timesteps rather than concentrated at the mid-range steps — where semantic content is most stably represented — it may restrict the model's capacity for valid causal variation and prompt controllability (see discussion in Ch. 8). Limited VAE encoder-decoder capacity and spatial resolution further compound these artefacts.

- **Text–Image Misbinding:** inconsistent alignment between prompt semantics and generated attributes, particularly in partially specified prompts. This behaviour is likely due to the CLIP and VAE components not being trained under the same counterfactual constraints as the diffusion denoiser, causing slight mismatches between textual and visual representations across worlds.

Overall, these qualitative observations suggest that the proposed counterfactual contrastive loss effectively promotes causal invariance, yet residual inconsistencies stem from both the uniform application of the loss across diffusion steps and its limited scope within the architecture.

Extending the loss to the text and image encoders — and restricting its temporal application to semantically stable denoising phases — is expected to yield further improvements in causal alignment and prompt consistency (see Sec. 7.8).

7.7 Learning Induced Conditional Dependencies

7.7.1 Conditional dSprites Variant

To test whether counterfactual contrastive training enables models to learn level-2 conditional dependencies from data, we construct a modified version of the *Counterfactual dSprites* dataset that retains its structured factual-counterfactual pairing property while introducing an additional dependency between two causal factors: **shape** and x**-position**. Specifically, we filter the original dataset to remove combinations that violate the imposed dependency, defining a conditional distribution

$$p(x \mid \text{shape})$$

with disjoint supports: square sprites appear only on the left-hand side of the frame, and ellipses only on the right-hand side. All other latent factors (*scale*, *rotation*, *y-position*) remain sampled as in the original dataset. This conditional variant therefore preserves the single-factor intervention structure of the Counterfactual dSprites pairs, while making *x-position* a causal descendant of *shape*, enabling controlled evaluation of whether the model learns to respect induced conditional structure during counterfactual generation.

7.7.2 Evaluation & Findings

We evaluate whether the trained model captures this induced dependency by intervening on *shape* in the counterfactual prompt T', while holding all other factors fixed. Under the induced conditional, a valid counterfactual should change both the shape and the corresponding x-position, while leaving rotation, scale, and y-position unchanged.

Qualitative inspection confirms that the counterfactual contrastive training encourages this behaviour: factual images containing a square on the left generate counterfactuals showing an ellipse on the right, and vice versa, consistent with the induced $p(x \mid \text{shape})$. The resulting images preserve all other non-descendant features, demonstrating adherence to the causal consistency criterion. This behaviour indicates that the model successfully learns both *what should change*

(descendant variables following the induced conditional) and *what should stay the same* (non-descendant variables), aligning with the level-1 / level-3 mapping described in the Introduction.

7.8 Limitations & Future Work

While the proposed counterfactual contrastive learning framework demonstrates qualitatively improved causal consistency, several limitations remain, motivating concrete directions for future work:

- **Dependence on Labelled Causal Factors and Known Causal Structure.** The method is strongly dependent on access to labelled causal factors and knowledge of the underlying causal structure in the current formulation, as these are used to construct interventional training pairs and to determine which attributes should remain invariant. Without these, it is not possible to directly specify which factors should change or remain unchanged under an edit, making both training and evaluation less well-defined. Errors in the labelled factors or assumed causal structure would propagate into the training signal, potentially enforcing incorrect invariances or allowing unintended changes, and thus degrading the consistency of generated edits. Future work is required to assess how generation quality and causal consistency degrade as error in the counterfactual supervision increases, for example due to measurement noise or misspecification of the underlying causal structure.

- **Scope of Counterfactual Supervision in Latent Diffusion Training.** The current training objective is applied only to the diffusion denoiser (UNet). In standard latent diffusion training, the VAE is first trained to learn a latent representation of the data distribution, after which its weights are frozen during UNet training. As a result, the learned latent space is not explicitly constrained to align with the causal structure assumed in the counterfactual formulation. Extending the counterfactual contrastive loss to the VAE (and text encoder) would allow the latent representation itself to reflect the same causal constraints, potentially reducing prompt misbindings and improving geometric fidelity (see Sec. 7.6.4.1).

- **Diffusion Timestep Application of the Loss.** The present implementation applies the loss uniformly across all diffusion timesteps, which overconstrains the model during

early noisy states and late near-deterministic reconstruction phases, where semantic structure is either not yet formed or already fixed (see Sec. 7.6.4.1). This can suppress valid variations and reduce controllability for subtle edits. Restricting the loss to semantically stable mid-range timesteps (as implemented in Ch. 8) could alleviate these effects and better target the stages where high-level attributes are represented.

- **Absence of Observable Causal Factors in Real-World Data.** The current evaluation is conducted on a controlled synthetic dataset with explicitly defined factors. In real-world image domains, these factors are not directly observed and must instead be inferred through learned representations (e.g., via a trained encoder). This introduces an additional layer of uncertainty: even if the causal structure is correctly specified, the inferred factors may not align with semantically meaningful or disentangled variables. As a result, the counterfactual supervision signal may be ill-posed, leading to edits that do not exhibit consistent or minimal-change behaviour. Future work is required to jointly address causal factor discovery and counterfactual training in realistic settings.

- **Generalisation to Unseen Visual Concepts.** Generalisation to visual concepts previously unseen in the training data, including novel shapes such as a triangle and visual style changes such as photorealistic or animated renderings, depends on the underlying latent representation. In practice, this would require adapting or retraining the encoder (e.g., VAE) to capture the expanded distribution of latent states before consistent editing behaviour can be expected.

- **Scaling to Multi-Object and Physically Grounded Scenes.** Extending the framework beyond single-object synthetic scenes introduces additional challenges, including occlusions, multiple interacting objects, and indirect effects of edits. In such settings, determining which attributes are causally affected by a change becomes less straightforward, and the assumption of clearly separable factors may no longer hold.

- **Proposed Quantitative Evaluation.** Future work will implement the metrics proposed in Sec. 7.6.3 to provide formal quantitative assessment of causal consistency. These include *Edit Success*, *Non-Descendant Preservation*, and *Minimality*. Aggregate results across single-factor interventions will enable standardised benchmarking and reproducibility in future studies.

- **Evaluation Bottlenecks.** For natural images, automatic measurement of non-descendant preservation remains an open problem. Future approaches may employ hybrid evaluation protocols combining human judgements with vision-language models (VLMs) to assess adherence to causal consistency at the semantic level.

- **Human Study.** A proposed human participant study would test whether improved causal consistency in deep generative models corresponds to closer alignment with human causal expectations and intuitions about minimal-change editing behaviour. This study is not implemented in the present work but remains a potential direction for future investigation.

7.9 Broader Multi-Modal Pointer (Scope Note)

While demonstrated on images, the training-time counterfactual objective is modality-agnostic. In multi-modal settings (e.g., image, audio, proprioception, language), interventions can be posed on structured causal variables, with *parallel-world* supervision enforcing invariance across non-descendant modalities and factors. This connects naturally to robotics contexts where counterfactual edits may involve action commands or symbolic goals, while sensor modalities should remain stable unless causally downstream (see Ch. 8 for interactive settings).

7.10 Summary

This chapter introduced a counterfactual contrastive loss objective for training deep latent diffusion models that embeds the *principle of causal consistency* into the generative process. By sharing exogenous noise across *parallel worlds* and penalising drift in non-descendant features, the proposed counterfactual contrastive learning method enforces minimal-change behaviour under prompt-level interventions. The resulting model learns both *what should change* — the intervened variable and its causal descendants — and *what should stay the same* — the non-descendants — thereby improving controllability and faithfulness to human causal intuition.

Empirical evaluation on the newly constructed *Counterfactual dSprites* dataset demonstrated qualitatively improved causal consistency in fine-tuned generations. A conditional variant further confirmed that the method can respect induced level-2 conditional dependencies, learning coherent descendant relationships between causal factors while preserving independent

variables. Together, these findings support the hypothesis that counterfactual contrastive supervision introduces a useful causal inductive bias for deep generative learning.

Although this chapter focuses on visual generation, the counterfactual objective is inherently modality-agnostic. Its principles extend naturally to multi-modal domains — such as audio, proprioception, or language — where interventions may act on structured causal variables while parallel-world supervision enforces invariance across non-descendant modalities. This generality is particularly relevant for robotics, where causal consistency across heterogeneous sensory and symbolic representations is essential for grounded reasoning about physical and interactive outcomes.

In the broader thesis context, this work directly addresses **Q1 - Modelling**, **Q2 - Structure and Parametrisation**, and **Q5 - Counterfactual Reasoning**, showing how explicit causal structure and counterfactual reasoning can be integrated into deep generative AI. The next chapter, *Multiverse Mechanica* (Ch. 8), extends these ideas to interactive, multi-agent worlds, embedding structural causal models within dynamic game environments to evaluate causal alignment and counterfactual reasoning in embodied settings.

8

Multiverse Mechanica: A Playable Benchmark for Learning Game Mechanics via Counterfactual Worlds

Contents

8.1 Introduction 219
8.2 Background & Related Work 222
8.3 Formalising & Learning a Game Mechanic 223
8.3.1 Illustrating Example 223
8.3.2 Formal Framework for Game Mechanics 227
8.4 Multiverse Mechanica: A Playable Testbed for Learning Mechanics 228
8.4.1 Game Overview 228
8.4.2 Implemented Mechanics (v1.0) 228
8.4.3 Data Generation 229
8.4.4 Visual Design Decisions 229
8.4.5 Summary. 230
8.5 Game Design Decisions 230
8.5.1 Bridging Game Mechanics & Causal Mechanisms with System-Based Design 230
8.5.2 Impact Frames: Defining Semantic Consistency in Dynamic Causal Model Traces via *Point of Maximum Action* Concept 232
8.5.3 Summary. 235
8.6 Proof-of-Concept: Learning a Mechanic with Diffusion Fine-Tuning 235
8.7 Scalability & Limitations 238
8.8 Summary 239

8.1 Introduction

Building on the causal generative learning principles developed in Chapter 7, this chapter introduces *Multiverse Mechanica* — a benchmark environment and methodological framework for studying **mechanic learning** and **causal consistency** in interactive deep generative AI world models. Whereas Chapter 7 focused on counterfactual contrastive training for static image diffusion, here we extend those ideas to the dynamic, agent-based setting of games, providing a structured domain to test whether models can learn the *rules* that govern a world rather than merely its visual appearance.

Motivation within the Thesis. The thesis argues that causal generative AI can endow robots with a capacity for **counterfactual cognition** aligned with Level 3 of Pearl's Ladder of Causation. In this framing, robots must not only recognise patterns (Level 1) and anticipate the effects of their actions (Level 2), but also imagine, evaluate, and learn from *alternative worlds* (Level 3). While previous chapters addressed decision-making and explanation through causal reasoning and counterfactual inference, this chapter focuses on the complementary challenge of *learning* these causal regularities in deep generative settings. Specifically, it investigates whether a deep generative model can learn the *structural causal model (SCM)* — both the graphical structure and the functional assignments — of a causally complex, uncertain system. This is essential for the thesis-wide aim of building generative models that support downstream reasoning, prediction, and explanation in robotics.

Gaps Addressed. Despite rapid advances in deep video and game world models [68–72], existing evaluations predominantly measure perceptual fidelity, sample quality, or short-horizon forecasting. Few benchmarks assess whether a model has learned the underlying *mechanics* — the causal rules that define how entities interact and change. Current claims of implicit 'mechanics learning' are largely post-hoc and observational, offering little evidence that models generalise under intervention or respect the causal dependencies that make a world coherent. In contrast, this chapter introduces a principled causal benchmark that exposes whether a generative model behaves consistently under controlled interventions. By generating parallel-world trajectories differing only by targeted mechanic-level changes, Multiverse Mechanica enables formal, quantitative assessment of **causal consistency** in learned world models. While

we defer full-scale benchmarking and baseline comparisons to future work, the present chapter establishes the causal formalism and the infrastructure required for such evaluations.

Link to Thesis Aims and Research Questions. The work directly advances several of the thesis research questions. It contributes to **Q1 - Modelling** by framing causal generative models as high-dimensional abstractions of world dynamics suited to the multi-modal sensory modalities and tasks of robotics. It contributes to **Q2 - Structure and Parametrisation** by exposing modular, human-interpretable causal mechanisms that can be explicitly specified or learned. And it addresses **Q5 - Counterfactual Reasoning** by operationalising Level 3 reasoning over aligned *parallel worlds*, providing a route to evaluate whether generative models understand and preserve causal relationships under in-context interventions. Collectively, these advances strengthen the thesis's overarching goal of developing causal generative AI methods that support trustworthy, reasoning-capable robot cognition.

Contributions of this chapter.

1. **Formal Causal Definition of Game Mechanics.** Mechanics are modelled as modular structural equations governing state transitions and their counterfactuals, enabling explicit in-context 'all-else-equal' reasoning about causal dependencies. *RQs:* **Q1 - Modelling**, **Q2 - Structure and Parametrisation**.

2. **Multiverse Mechanica Environment.** A controllable benchmark generator for Level 3 (counterfactual) datasets via aligned parallel-world simulations with shared exogenous noise. The environment formalises causal interventions at the level of mechanics, producing datasets and evaluation metrics for causal consistency. *RQs:* **Q1 - Modelling**, **Q5 - Counterfactual Reasoning**.

3. **Proof-of-Concept Mechanic Learning Study.** Leveraging these causal representations in its objective function, a deep generative model is fine-tuned to reproduce a specific mechanic while maintaining invariance in non-descendant features, demonstrating how causal objectives improve consistency behaviour and interpretability. *RQs:* **Q5 - Counterfactual Reasoning** (primary), **Q1 - Modelling** (supporting).

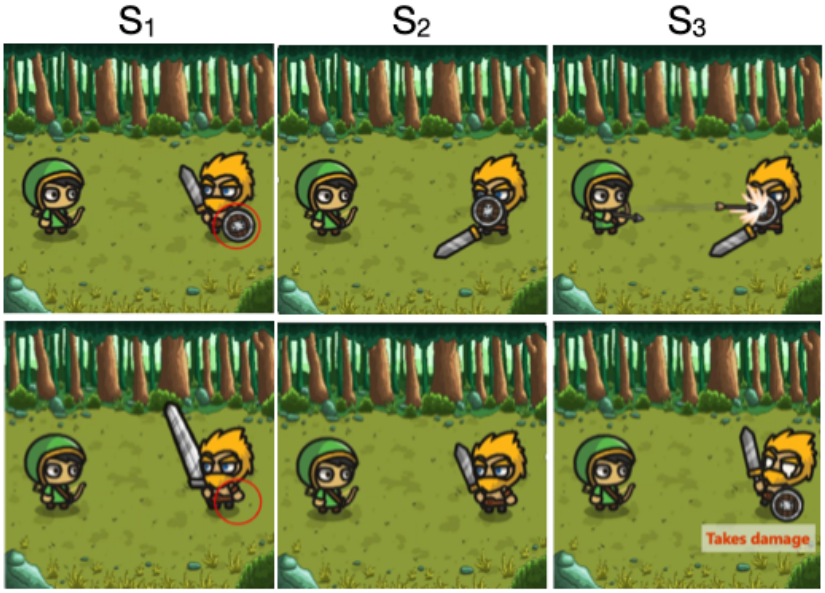


Figure 8.1: Clips from consistent contrasts sampled at their respective impact frames: each row contains two contrasting gameplay clips where differences are solely attributable to the shield mechanic. Left, middle, and right correspond to parallel world statements $\mathbf{S_1}$, $\mathbf{S_2}$, $\mathbf{S_3}$.

Significance for Robotics. Although presented in the language of game mechanics, the underlying contribution extends to robotics: learning the structure and assignment functions of an SCM for complex, uncertain, multi-modal systems. Robotic agents rely on precisely this form of world knowledge to reason about action consequences, physical interactions, and sensory feedback. By learning *structural* and *functional* components of world models rather than superficial correlations, models trained and evaluated within Multiverse Mechanica move closer to the causal world models required for robust, generalisable robot cognition and decision-making under uncertainty.

Chapter Structure. The remainder of this chapter is organised as follows. Sec. 8.2 provides preliminaries and contextualises our work in the context of the broader literature. Sec. 8.3 formalises mechanics as causal mechanisms and defines the minimal-change criterion. Sec. 8.4 describes the design of the *Multiverse Mechanica* testbed environment, implemented mechanics, data levels, and temporal alignment strategy. Sec. 8.6 reports the diffusion fine-tuning study and evaluation metrics. Sec. 8.7 discusses scalability, limitations, and future benchmarking directions. Finally, Sec. 8.8 summarises the contributions, connecting back to the thesis aims and forward to Chapter 9.

8.2 Background & Related Work

Defining Game Mechanics. Building on prior definitions [80], we define a *game mechanic* as a *modular subset* [76–79] of the *game rules* triggered by specific player/agent interactions [81, 82], producing changes in *game state* [83, 84] that shape gameplay visuals [85]. These subsets entail causal relations with preconditions and effects, representable as logic, finite-state machines, or transition functions [77–79, 86, 87]. We formalise this definition in Sec. 8.3.2.

Causal Framing. We adopt the causal hierarchy (level-1: observation, level-2: intervention, level-3: counterfactual) [33] and use causal DAGs and *structural causal models* (SCMs) [32] to model mechanics as mechanisms (rules) [160]. For focusing on a mechanic's variables, we reference *marginalised DAGs* (mDAGs) that preserve causal and interventional semantics while marginalising others [88] (see Sec. 3.1.9). The *causal consistency principle* states that variables not downstream of an intervention retain the same value across worlds [89, 90]; *counterfactual graphs* capture this cross-world consistency compactly (single nodes for shared variables; world-indexed nodes for affected ones) [89]. Construction details appear in Sec. 3.2.2; additional background is in Sec. 3.1.

For in-depth preliminaries on causal machine learning, readers are directed to Sec. 3.1.

Learning Game Mechanics. Empirical results highlight the challenges of learning game mechanics. A study performed by Gingerson, Amershi et al. highlights the continuing challenge of generating *consistent* gameplay with SOTA architectures — even when generations look plausible, they frequently break the rules of the mechanic. Chen, Xu, Zhang et al. report similar failures in spatial and numerical consistency, necessitating explicit corrective modules. More broadly, empirical studies of video prediction models show that while they excel in-distribution, they often rely on case-based mimicry and fail under distribution shift, violating simple physical principles [92, 93].

If we view a mechanic as a latent generative factor, then unsupervised learning of mechanics is provably impossible from video observations of gameplay alone [94] without strong inductive biases [161, 162]. Prior work in this area shows that causal representations are at best only partially identifiable from observational data without intervention data or strong causal inductive biases [95–97]. Our work builds on prior work that employs these causal approaches

to learning latent generative factors. But to our knowledge, our work is the first to apply this type of causal analysis to the problem of learning game mechanics during training, and generating consistent gameplay from a trained model.

Datasets, Testbeds & Environments. Existing testbeds, datasets, and environments for world models and video prediction largely emphasize intuitive reasoning about real-world Newtonian physics rather than explicitly defined game mechanics. For instance, IntPhys [98] probes intuitive physics by testing whether models respect basic object permanence and motion, while Physion [99] provides simulated videos of collisions and stability events to evaluate physical prediction. However, game mechanics can encompass non-realistic 'physics', such as spell casting and passing through portals. *Multiverse Mechanica* focuses on a broader set of game mechanics, and contributes a *playable generator* that emits data and artefacts that target learning of a formally defined ground-truth set of mechanics.

8.3 Formalising & Learning a Game Mechanic

In this section, we demonstrate the formalisation of a game mechanic as well as how we would learn that mechanic from data. Then, in Sec. 8.3.2, we provide a general mathematical description of this approach.

8.3.1 Illustrating Example

The Shield Mechanic Consider a scene from a stylized 90's fantasy turn-based combat game, where an archer battles a warrior. Like many games in this genre, there is a *shield mechanic*, as shown in the first row of Fig. 8.1, where the warrior may raise a shield to block incoming attacks.

8.3.1.1 Formalising the Shield Mechanic

How might we describe this shield mechanic in formal causal terms?

Step 1: Describe the Mechanic with Causal Logic. We start by completely describing the shield mechanic using a series of causal hypothetical statements of the form 'Given pre-conditions W, all else equal, if X, then Y'. Specifically, we focus on level-3 multiverse logic statements that employ conjunctions of conflicting conditions. Table 8.1 column 1 shows three statements, $\mathbf{S_1}$, $\mathbf{S_2}$, and $\mathbf{S_3}$, that fully describe the shield mechanic.

Descriptions in Causal Logic	Formal Counterfactual Notation	Consistent Contrast Sample Data
S_1 All else equal, if the opponent has a light weapon, they may equip a shield; if heavy, they cannot.	$\mathcal{S}_1 :\ P(S_{W=1}=1,\ S_{W=0}=0) \geq \epsilon_1$	$\mathcal{D}_1 = \{\omega_1,\ C,$ $(S_{W=1}, B_{W=1}, D_{W=1}, V_{W=1}),$ $(S_{W=0}, B_{W=0}, D_{W=0}, V_{W=0})\}$
S_2 All else equal, if the opponent has a shield, they may block; if no shield, they cannot.	$\mathcal{S}_2 :\ P(B_{S=1}=1,\ B_{S=0}=0) \geq \epsilon_2$	$\mathcal{D}_2 = \{\omega_2,\ C,\ W,$ $(B_{S=1}, D_{S=1}, V_{S=1}),$ $(B_{S=0}, D_{S=0}, V_{S=0})\}$
S_3 Given a shield, if a block succeeds then no damage; if it fails, damage occurs.	$\mathcal{S}_3 :$ $P(D_{B=1}=0,\ D_{B=0}=1 \mid S=1) \geq \epsilon_3$	$\mathcal{D}_3 = \{\omega_3,\ C,\ W,\ S,$ $(D_{B=1}, V_{B=1}),\ (D_{B=0}, V_{B=0})\}$

Table 8.1: Level-3 Shield Mechanic Multiverse Logic Statements: Conjunctions of conflicting conditions. The shield mechanic described as in natural language causal logic (column 1), which are then formalised with counterfactual notation (column 2), where strictly positive probabilities ($\epsilon_i > 0$) indicate bounded uncertainty due to other causal factors in the system. Column 3 shows *consistent-contrast* tuples (each row shares seed ω_i).

The columns of Fig. 8.1 correspond to **S_1**, **S_2**, and **S_3**. We could instead use *level-2* interventional statements, which are normally preferred because they are generally testable with experimental data. But the *level-3 parallel world* statements provide an additional constraint in the phrase 'all else equal'; that outcomes unaffected by the conditions must remain *consistent* across the clauses. As we will see below, we can use that constraint to operationalize consistency in game generation. Secondly, we can leverage the gaming setting's rare opportunity to generate level-3 data to validate level-3 statements.

Step 2: Rewrite as Counterfactual Expressions. We can rewrite **S_1**, **S_2**, and **S_3** as mathematical expressions using counterfactual notation, capturing the contrasts more compactly. For simplicity, let W denote the weapon type, S indicate whether a shield is equipped, B indicate whether the shield is used to block, and D indicate whether damage occurs. We treat these as binary variables for clarity, without loss of generality. Let $W = 1$ and $W = 0$ denote light and heavy weapon, respectively. S, B, and D, let 1 mean *True* and 0 mean *False*.

We use counterfactual notation to denote variables under the influence of intervention, such that Y under an intervention that sets X to x is written as $Y_{X=x}$. We can formalise the parallel world statements **S_1**, **S_2**, and **S_3** as shown in column 2 of Table 8.1, where $\epsilon_i; \forall i \in \{1, 2, 3\}$ denotes strictly positive probabilities ($\epsilon_i > 0$), indicating bounded uncertainty due to other causal factors in the system.

With this, our shield mechanic is described in formal mathematical terms.

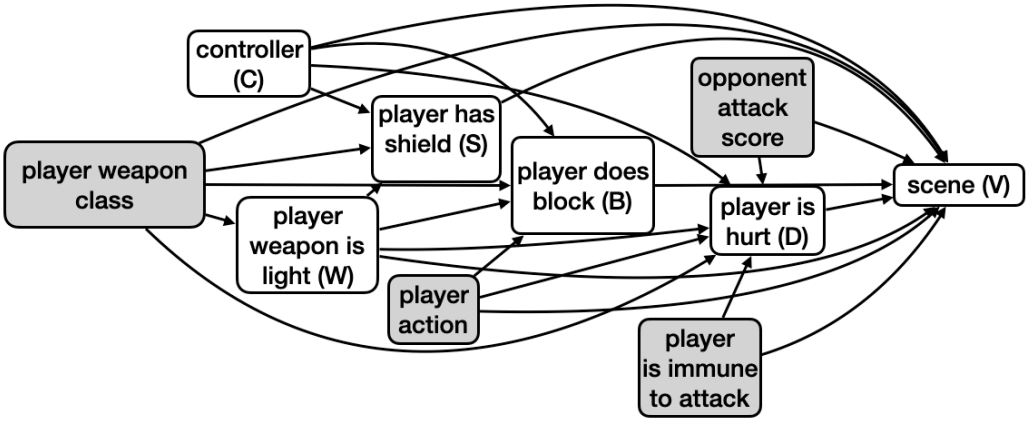


Figure 8.2: Shield Mechanic *Marginalised DAG (mDAG)*. A causal DAG marginalised to focus on the nodes specific to the shield mechanic and their latent confounders (gray nodes)

Step 3: Represent Mechanic with Causal Graphs Let V denote a full clip of gameplay. An outcome, denoted v, is a sequence of frames. Let C denote the controller input from a player at the start of the player's turn. Let G denote the full causal DAG for a single turn in the battle (see the full graph in Fig. C.2 in Appendix C.2). Let us assume we have access to this DAG, or that we could create it using knowledge of the game structure, analyzing causal dependence in the game's code [163, 164], or by applying causal discovery methods [165]. The variables implicated in our description of the shield mechanic are $Z = \{C, W, S, B, D, V\}$. The causal DAG G is quite large, so we derive the mDAG $G^{\mathcal{M}}$ that zooms in on Z (Fig. 8.2) by marginalising out the variables not in Z (see Sec. 3.1.9 for a description of the algorithm). Next, we can combine the mDAG with each counterfactual expression in the mechanic's description to construct the counterfactual graphs in Fig. 8.3.

The counterfactual graphs in Fig. 8.3 encode a representation of *causal consistency* — variables that are not downstream of interventions and thus are consistent across worlds are unique, while inconsistent variables have nodes indexed by each world. Thus, the counterfactual graphs are representations of the shield mechanic that explicitly describe what should remain consistent when generating gameplay depicting the shield mechanic.

8.3.1.2 Generating Shield Mechanic Data

We can generate level-3 parallel world data consistent with $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$ by creating parallel runs with identical initial conditions and random seeds. We can intervene separately in each run, producing clips of parallel *virtual* worlds that differ only in their respective interventions. We call the tuple of these clips, combined with the outcomes of other mechanic-related variables

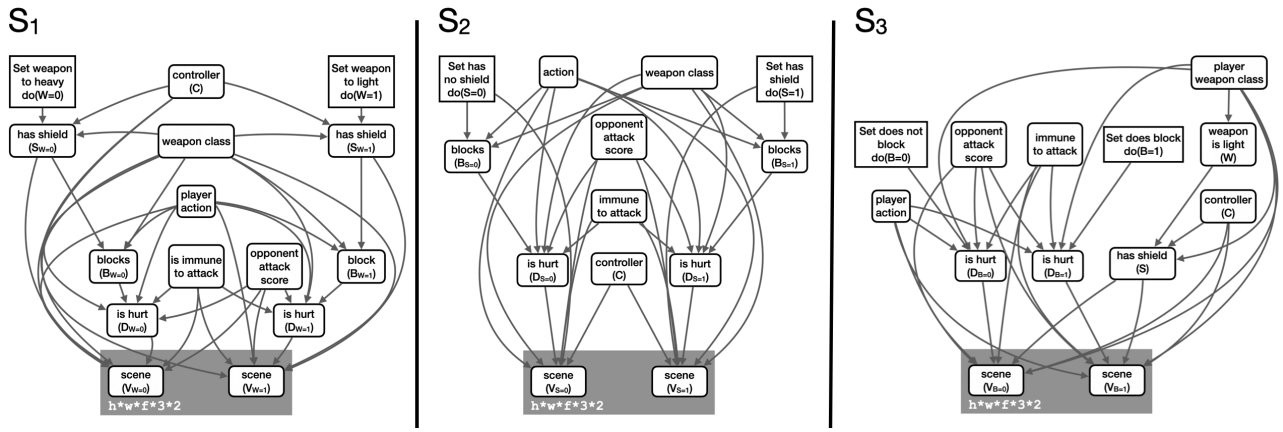


Figure 8.3: Counterfactual graphs for the shield mechanic. The video variables $V_{X=x}$ have shape height(h)*width(w)*frames(f)*channels(3)*worlds(2).

under these interventions and their shared initial condition/seed, *consistent contrasts*. Fig. 8.1 illustrates gameplay clips from the contrasts.

Let ω_1, ω_2, and ω_3 represent distinct sets of random seeds and initial conditions. Let C represent the controller input from the player. Let $V_{X=x}$ represent a video clip of gameplay under an intervention that sets X to x. Reading consistency from the graphs in Fig. 8.3, we see $W_{S=1} = W_{S=0} = W$, $W_{B=1} = W_{B=0} = W$, and $S_{B=1} = S_{B=0} = S$. Let $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$ represent samples of consistent contrasts for $\mathbf{S_1}$, $\mathbf{S_2}$, and $\mathbf{S_3}$ respectively, as shown in Table 8.1 column 3.

8.3.1.3 Learning the Shield Mechanic from Data

Let $P_1 = P(S_{W=1}, S_{W=0})$, $P_2 = P(B_{S=1}, B_{S=0})$, and $P_3 = P(D_{B=1}, D_{B=0}, S)$ denote the distributions constrained by $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$. We can estimate these distributions through repeated sampling of consistent contrasts $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$, then averaging over the sampling distributions to obtain the sampling distributions $\hat{P}_1$, $\hat{P}_1$, and $\hat{P}_3$ (see Sec. 3.2.4). In each case $\hat{P}_i$ converges almost surely to P_i. This provides a precise operationalisation of what it means for a generative model to learn the shield mechanic: learning constraints $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3\}$ on distributions $\{P_1, P_2, P_3\}$ or modelling $\{P_1, P_2, P_3\}$ directly.

However, in the canonical case of training game world models, we assume that only the controller inputs and the video outputs are observed during training. Here, the inference of $\hat{P}_i$ becomes a task of unsupervised learning of a latent vector $\mathcal{D} \setminus \{C, V_{X=x}, V_{X=x'}\}$ using $\{C, V_{X=x}, V_{X=x'}\}$ as features, where $V_{X=x}, V_{X=x'}$ is a vector of shape 2 * frame height * frame width * 3 RGB channels * number of frames. Without further assumptions, disentangling the components of $\mathcal{D} \setminus \{C, V_{X=x}, V_{X=x'}\}$ is generally infeasible. However, the counterfactual graphs

in Fig. 8.3 already disentangle these variables for us. Using these, the problem reduces to training a latent variable model.

8.3.2 Formal Framework for Game Mechanics

We assume a causal DAG $G = (\mathsf{V}, \mathsf{E})$ for a single step of gameplay with $\mathsf{V} = \{C_t, X_t, C_{t+1}, X_{t+1}, V_t, V_{t+1}\}$, and typical edges $X_t \to C_t$, $X_t \to X_{t+1}$, $C_t \to X_{t+1}$, $X_t \to V_t$, $X_{t+1} \to V_{t+1}$. For a given mechanic, we restrict to the variable subset $M = \{C_t, X_t^M, X_{t+1}^M, V_{t+1}\}$, $X_t^M \subseteq X_t$, $X_{t+1}^M \subseteq X_{t+1}$, and form the marginalised DAG G^M by marginalising variables outside M while preserving interventional semantics (Sec. 3.1.9).

The ground-truth SCM, consistent with G^M, induces a family of interventional and counterfactual distributions over the variables in M. We denote this family by

$$\mathcal{F}(M) \;=\; \Big\{\, P\Big(V_{t+1,X=x},\, Z_{X=x} \,\Big|\, E\Big) \;:\; X, Z \in M,\; x \in \mathcal{X}_X,\; E \text{ a conditioning event over } M \,\Big\}.$$

In practice, E corresponds to a predicate on the variables in M (e.g., $S{=}1$ for 'has shield').

We formalise a mechanic as the tuple $\langle G^M, \mathcal{M}\rangle$, where $\mathcal{M} = \{S_1, \ldots, S_k\}$ and each constraint $\mathcal{S}_i$: $P\Big(\bigwedge_{j=1}^{m_i} Y_{X=x_j} = y_j \,\Big|\, E\Big) \;\geq\; \epsilon_i$ binds counterfactuals of variables in M under interventions on $X \in M$, with $x_j \in \mathcal{X}_X$, E a conditioning event over variables in M, and $\epsilon_i \in (0, 1]$ (allowing non-deterministic relations due to factors outside M). For each $\mathcal{S}_i$ we denote the targeted distribution by P_i (e.g., $P_1 = P(S_{W=1}, S_{W=0})$ in the shield example).

Data & Estimation. A consistent-contrast dataset for $\mathcal{S}_i$ of size N is

$$\mathcal{D}_i^{(N)} = \Big\{ \Big(V_{X=x_1}(\omega_n), \ldots, V_{X=x_{m_i}}(\omega_n)\Big) \;:\; \omega_n \in \Omega,\; n = 1, \ldots, N \Big\},$$

optionally restricted to ω_n satisfying E. Let $\hat{P}_i^{(N)}$ be the empirical distribution induced by $\mathcal{D}_i^{(N)}$. Under i.i.d. sampling of seeds ω_n, $\hat{P}_i^{(N)} \xrightarrow{\text{a.s.}} P_i$. For each $\mathcal{S}_i$ (and associated P_i), we construct a *counterfactual graph* $G_i^{M,\text{cf}}$. In partially observed settings (video + controller only), G_i^{cf} specifies which latent variables are shared across worlds and which differ, reducing estimation to a well-posed latent variable problem aligned with the counterfactual graph's structure.

8.4 Multiverse Mechanica: A Playable Testbed for Learning Mechanics

We introduce *Multiverse Mechanica*, a fantasy-style battle game designed as a testbed for learning game mechanics. Unlike static datasets, Multiverse Mechanica is a playable game that emits the artefacts required to study and evaluate whether models capture the game's mechanics — not just gameplay visuals. Its design integrates three innovations: (i) native support for level-3 parallel-world contrasts with consistency under the same ω; (ii) per-mechanic mDAGs $G^{\mathcal{M}}$, parallel world and counterfactual graphs, and specifications of $\mathcal{M}$; and (iii) explicit visual grounding, where stance and scene variables are rendered into pixels.

8.4.1 Game Overview

Each episode consists of a pre-battle setup (character and equipment selection, random assignment of elemental buffs (e.g., fire, ice) followed by turn-based combat. The player occupies the left side of the screen, and the enemy occupies the right. On the player's turn to attack, a timing-based interaction yields an attack score; the enemy's turn samples an analogous attack score. Outcomes depend on weapons, defences, the attack score, and buffs. See Sec. 8.5 for additional details.

8.4.2 Implemented Mechanics (v1.0)

Version 1.0 of Multiverse Mechanica includes the following mechanics, each with associated $G^{\mathcal{M}}$ and parallel-world data sufficient to estimate $\mathcal{M}$. The **shield mechanic** focuses on equipping and blocking with a shield, as discussed in Sec. 8.3.1. In the **elemental immunity mechanic**, 'elemental' attributes (e.g., fire and ice) govern immunity and vulnerability to attacks. The **weapon range mechanic** governs melee vs. ranged combat. The **spell-casting mechanic** governs five sub-mechanics that allow players to give themselves an advantage in battle (e.g., gain increased attack power, dodge ability), or their opponent a disadvantage (e.g., disarm them or lower their defence) — projectiles, self-levitation, enemy-levitation, self-transformation, and enemy-transformation. See Appendix C.2 for detailed descriptions, including causal formalisations, DAGs, and illustrations.

8.4.3 Data Generation

Multiverse Mechanica is not a dataset but a generator. To generate data, an automated agent repeatedly plays the game to produce clips. Users can select a number N of generations. The generation process can randomly generate N clip examples, constituting level-1 data. The user can also specify interventions on specific game state variables and generate N clip examples where those interventions are applied, constituting level-2 data. Finally, the user can specify interventions and assign them to multiple game instances with a shared 'ω' (same random seed and initial conditions) and generate N consistent contrasts (tuples of clips), constituting level-3 data. Each mechanic has presets for level-2 and level-3 generation. Each clip is a 512x512 MP4 video averaging 4 seconds at 50 FPS. Each generated example is a tuple consisting of a clip, controller inputs, game-state variable outcomes, and a random seed for reproducibility. See Appendix C.3 for additional details related to data generation.

8.4.4 Visual Design Decisions

Impact Frames & Visual Conventions. We designed the game such that each turn contains an *impact frame* — the most visually and mechanically expressive phase of an interaction (e.g., the precise moment when an attack lands) Impact frames are not based on fixed time-points but are set according to specific conditions in the finite-state machines. See Sec. 8.5.2 for details.

Simple yet Information Dense Visuals. To facilitate rapid, inexpensive experimentation, we focus on the ability to run experiments with episodes that have a minimal number of frames. To this end, we use a simple art style that aligns with the representational biases of pre-trained vision models [166, 167] and animation conventions that emphasize dynamic information, such as speed lines ('zip ribbons') to depict fast motion, trajectory lines for projectiles, curved swipes for melee attacks, and burst lines and explosion visual effects for collisions or blocked strikes [168–170]. All characters and scenes are rendered in a stylized 'chibi' art style, sourced from *CraftPix.net* [171, 172], with simplified shapes and flat 2D cartoonish imagery.

In Sec. 8.6, we highlight this ability by limiting our analysis to single time-point snapshots at the impact frame, chosen as the impact frame of the clip. The images in Fig. 8.1 are all sampled at their respective impact frames.

8.4.5 Summary.

Multiverse Mechanica provides a compact yet expressive testbed for studying whether generative models can capture mechanics. Its design couples causal structure with visual grounding, leverages art and animation conventions for low-cost training, and enables reproducible creation of parallel-world contrasts.

8.5 Game Design Decisions

To ensure that the datasets produced by our simulation respect the causal assumptions of our model, we designed the game architecture with consistency guarantees as a primary objective. This section documents how these guarantees were realised in practice, focusing on two complementary aspects: (i) the modular, system-based design of game mechanics that mirrors the structure of the causal model, and (ii) the temporal alignment of captured frames to ensure semantic consistency in dynamic interactions.

8.5.1 Bridging Game Mechanics & Causal Mechanisms with System-Based Design

At the foundation of our data generation pipeline lies a design principle: game mechanics must be implemented in a way that respects and preserves the structure of the causal model they instantiate. To achieve this, we adopted a fully modular *Entity–Component–System* (ECS) architecture [173, 174], which enforces locality of causal mechanisms and supports reproducibility across runs.

8.5.1.1 Entities, Components, and Systems as Causal Units.

In our implementation, *entities* represent the units of analysis (e.g., characters, projectiles, platforms), *components* represent their attributes and state (e.g., position, animation phase, shield presence), and *systems* encapsulate the transformation rules that govern state evolution (e.g., combat resolution, kinematics, physics, or screenshot scheduling). This separation guarantees that each causal mechanism is expressed locally, without entanglement with unrelated processes.

Each system implements a distinct causal mechanism: for example, the `GameCombatSystem` maps action states of attacker and defender into outcomes such as *hit*, *block*, or *immune*, while the `GamePhysicsSystem` governs the motion of projectiles according to deterministic

physical rules. Systems are generally designed to operate frame-by-frame using component data as inputs, but they may also maintain local state when required (for example, the `GameAnimationSystem` manages a per-entity priority queue of animation requests). This design ensures that each causal transformation is encapsulated and modular, while still supporting the persistent state needed for realistic simulation.

8.5.1.2 Alignment with Causal Graph Structure

The ECS architecture was chosen deliberately to reflect the modular structure of our causal model. Nodes in the causal graph correspond to entity attributes (e.g., *weapon class*, *stance*, *elemental immunity*), while edges correspond to the update dependencies realised through system logic.

For example, the transition of a defender into a *blocking* or *hurt* state depends on multiple upstream components: the presence of a shield component, the character's action selection component (indicating whether the chosen action is to block), and downstream trigger flags such as `needs_to_block` or `is_hurt`. The shield and action selection components establish the potential for blocking, while the trigger flags are set based on situational context (e.g., proximity of an incoming attack). Only when both preconditions and triggers align does the `GameCombatSystem` update the defender's state to *blocking*; otherwise, the state transitions to *hurt*.

This design directly encodes the causal mechanism:

$$\text{shield component} + \text{action selection} \longrightarrow \{\texttt{needs\_to\_block}, \texttt{is\_hurt}\} \longrightarrow \text{outcome (block or hurt)}.$$

By structuring dependencies in this way, the system preserves the logic of the causal graph within the mechanics of the game engine. This mapping ensures that system update rules correspond closely to the assignment functions of the causal model.

8.5.1.3 Locality and Modularity for Consistency

By localizing mechanics in dedicated systems, the architecture prevents hidden confounding across game features. For instance, animation timing is managed exclusively by the `GameAnimationSystem`, while collision and trajectory updates are confined to the `GamePhysicsSystem`. This guarantees that modifying one mechanism (e.g., projectile gravity) does not inadvertently alter another (e.g., collision detection or blocking). Such modularity enforces a form of causal isolation, allowing dataset generation to reflect the true structure of the designed model.

8.5.1.4 Reproducibility and Connection to Interventions

The ECS structure also guarantees reproducibility: since each system applies deterministic update rules to the current component state, identical initial conditions yield identical traces. Importantly, the systems themselves do not support interventions in the sense of directly overriding assignment functions during simulation. Instead, interventions are handled at the model level: sampled values from the causal model are passed into the simulation as inputs that determine which branches of the `GameBehaviourTree` are executed (e.g., sampled values specifying whether a character *does block*). The behaviour tree then orchestrates the scene by triggering the appropriate system updates, while each system executes its assignment function deterministically given the requested state changes. In this way, the game engine acts as a faithful executor of causal mechanisms, while the intervention logic is confined to the sampling layer above.

Through this design, the simulation environment operates as a direct computational analogue of the causal model, where each mechanism is encapsulated in a corresponding system. This guarantees that generated training data inherits the same modularity and independence properties as the underlying causal graph, thereby supporting consistency-guaranteed counterfactual analysis.

8.5.2 Impact Frames: Defining Semantic Consistency in Dynamic Causal Model Traces via *Point of Maximum Action* Concept

While the system-based architecture guarantees local causal consistency at the level of game logic, temporal alignment must also be addressed to preserve counterfactual consistency in dynamic interactions. To this end, our system generates gameplay clips of contrasting player turns, each designed to include a canonical *impact frame*: the instant an attack connects, a shield block occurs, or a projectile visibly misses. Ensuring that these per-turn impact frames correspond to semantically equivalent points in the causal process is critical; otherwise, contrasts risk reflecting phase misalignment rather than true causal differences. We therefore formalise alignment using the *Point of Maximum Action* (PoMA) principle, which anchors impact frames to the most visually and mechanically expressive phase of the interaction.

8.5.2.1 Conceptual Framing: Temporal Alignment for Counterfactual Comparisons

In static structural causal models (SCMs), counterfactuals are evaluated at a single time index, and semantic alignment across factual and counterfactual worlds is immediate. In dynamic SCMs, the meaning of an event depends on *when* it occurs relative to the unfolding process. For example, a melee strike might connect later than a projectile impact due to differences in action duration. If frames are extracted at fixed indices, we risk capturing non-equivalent phases of these interactions (e.g., an attack wind-up in one turn versus a point of contact in another). If we extract training artefacts (e.g., impact frames) at fixed indices, we risk capturing non-equivalent phases of these interactions (such as a wind-up in one run versus a point of contact in another). This undermines the validity of counterfactual comparisons by introducing differences that are artefacts of temporal phasing rather than consequences of the intervention.

We therefore treat each player turn as a temporal *causal trace* and align counterfactual observations to the *most informative temporal locus* of the relevant event class. We formalise this with the principle of the *Point of Maximum Action* (PoMA).

8.5.2.2 Point of Maximum Action (PoMA).

Let $A(S_{t'}) \in \mathbb{R}_{\geq 0}$ score how *action-intense* the counterfactual state $S_{t'}$ is with respect to the target event class (e.g., impact, block). The PoMA alignment selects

$$t'_{\text{PoMA}} \;=\; \arg\max_{t' \in T'} A(S_{t'}).$$

PoMA frames are then extracted at t'_{PoMA}, aligning the impact frame to the point of maximum expressivity. This guarantees that contrasts correspond to the same semantic phase of the interaction, regardless of variation in action duration.

8.5.2.3 Four Alignment Methods: Brief Summary with Pros and Cons

We summarize four practical approaches for temporal alignment of dynamic counterfactuals. Each provides a distinct trade-off between simplicity, robustness, and semantic fidelity.

1) Constant Time Interval. *Rule.* Evaluate the counterfactual variable at the same nominal time as the factual: $t' \equiv t$. *Pros.* Conceptually simple; trivial to implement; deterministic. *Cons.* Fails when action durations differ; risks capturing non-equivalent phases (e.g., mid-swing vs. impact); unsuitable for dynamic interactions where event timing adapts to interventions.

2) Equilibrium-Based. *Rule.* Evaluate once the counterfactual dynamics have reached a steady state or absorbing condition (e.g., $\|S'_{t'+1} - S'_{t'}\| < \varepsilon$ or a domain predicate holds). *Pros.* Appropriate for tasks where long-run properties matter; robust to transient phasing differences; alignment invariant to small shifts in sequence length. *Cons.* Inapplicable to inherently transient events (e.g., impacts); some episodes may not converge; equilibrium may erase the very distinctions needed to analyze acute causal effects.

3) Point of Maximum Action (PoMA). *Rule.* Align to the counterfactual time of peak action intensity for the event class:

$$t'_{\text{PoMA}} \;=\; \arg\max_{t'} A(S'_{t'}).$$

Pros. Directly targets the most salient phase of the interaction; robust to differences in sequence length; naturally accommodates variable-duration actions by abstracting to their peak. *Cons.* Requires a well-defined intensity score A; can be non-trivial for abstract or multi-agent interactions; may require smoothing when peaks are brief or noisy.

4) Semantic Consistency. *Rule.* Align by maximizing semantic similarity between factual and counterfactual states. *Pros.* General and flexible in principle; useful in settings where semantic descriptors are available. *Cons.* Not used in our implementation. It requires an additional similarity metric and embedding design, which introduces complexity and potential bias.

In practice, we adopt the PoMA approach, extending it with event-specific scoring functions and event-defined windows to handle variable-duration interactions. Constant Time and Equilibrium serve primarily as conceptual baselines, while semantic similarity was considered but not implemented.

8.5.2.4 Implementation in Our Game: Event-Based Windows and Weighted Scoring

To instantiate these principles in a reproducible data pipeline, our engine implements an event-driven *GameScreenshotSystem* that schedules captures at semantically aligned moments:

1. **Event Detection.** The simulation raises structured events for interactions of interest (e.g., `melee_impact`, `projectile_hit`, `shield_block`). Each event is associated with the participating entities and their current states.

2. **Event-Based Scoring Windows.** For each interaction type, we define a start event and a stop event that bound a scoring window (e.g., `swing_start` → `impact`, or `projectile_cast` → `collision`). These windows are managed directly in the behaviour tree, ensuring that scoring only occurs during the semantically relevant phase of the interaction.

3. **Weighted Scoring.** Within each window, frames are scored according to event-specific weights. For example, projectile impact events may be given higher weight than projectile flight, and shield contact may be prioritized over shield raise. The capture frame is then chosen as

$$\widehat{t'} \in \arg\max_{t' \in \text{window}} A(S'_{t'}),$$

with deterministic tie-breaking for reproducibility.

By anchoring artefact capture to event-defined scoring windows and applying weighted intensity scoring, our pipeline produces semantically aligned visual data across simulations, despite natural variability in action durations. This guarantees that counterfactual comparisons reflect genuine causal differences, rather than artefacts of capturing frames at arbitrary, non-equivalent time indices.

8.5.3 Summary.

Taken together, these design choices preserve consistency at both structural and temporal levels. The ECS architecture ensures local mechanics map cleanly to modular causal mechanisms, while the screenshot system anchors each player turn contrast to a canonical *impact frame* selected via PoMA. By aligning contrasts at the most expressive phase of interaction, the system guarantees that observed differences reflect genuine causal effects rather than timing artefacts, producing impact frames that are both causally and semantically consistent.

8.6 Proof-of-Concept: Learning a Mechanic with Diffusion Fine-Tuning

One advantage of *Multiverse Mechanica*'s design (Sec. 8.4.4) is that mechanics are rendered into *impact frames* using simple yet information-dense visuals. This allows experiments with supervision from clips as short as a single frame. We illustrate this with a case study: fine-tuning a pre-trained image diffusion model to target mechanic learning.

We fine-tuned the latent diffusion model *OpenJourney-v4* [175] on $N = 1000$ impact frame *consistent contrasts* generated from the game. Each consistent contrast consists of paired images $(V_{X=x_0}, V_{X=x_1})$ from parallel worlds that share a random seed ω but differ by interventions on a mechanic variable X. This setup directly instantiates the causal consistency principle: non-descendant variables of X should remain invariant across the pair.

We introduce a **multiverse alignment** objective — a modification of the standard diffusion loss — to enforce the *causal consistency principle*. In a consistent contrast, all elements share a common sample $\omega \in \Omega$, mirroring the reverse process in deterministic diffusion variants that generate data from latent noise. We therefore initialise each contrast trajectory with the *same* noise: sample $\omega \sim \mathcal{N}(0, I)$ and set $Z_{T,X=x_0} = Z_{T,X=x_1} = \omega$.

Let $\{X = x_0, V_{X=x_0}, X = x_1, V_{X=x_1}\} \sim \hat{P}$, where $X \in \mathcal{X}_X$ is a game-state variable under intervention and $V_{X=x_j}$ is the corresponding impact frame snapshot under intervention $X = x_j$. With timesteps $t \in [0, T]$, let $Z_{t,X=x_0}, Z_{t,X=x_1}$ denote the noisy latent representations of $V_{X=x_0}$ and $V_{X=x_1}$, respectively. Conditioned on controller input c, the *denoiser* $\epsilon_\theta(\cdot)$ iteratively transforms the shared seed ω into a clean latent $Z_{0,X=x_j}$ that decodes into the impact frame $V_{X=x_j}$. Our task is to train the denoiser's weights θ with a loss that enforces causal consistency across contrasts.

The **multiverse alignment** loss has two components:

$$\mathcal{L} \;=\; \lambda_1 \mathcal{L}_1 \;+\; \lambda_2 \mathcal{L}_2, \quad \lambda_1, \lambda_2 \geq 0,\; \lambda_1 + \lambda_2 = 1.$$

$\mathcal{L}_1$: seed-consistency loss. Let Abduct_θ denote an inversion procedure that estimates the Gaussian seed from an observed impact-frame latent $Z_{0,X=x_j}$ and controller input c_j. Given $Z_{0,X=x_j}$ with controller input c_j, Abduct_θ uses the denoiser $\epsilon_\theta(\cdot)$ to trace $Z_{t,X=x_j}$ backward through the noise schedule, producing an estimate $\hat{\omega}_j$ of the exogenous seed ω. Under a deterministic sampler (e.g., DDIM), this corresponds to following the reverse trajectory from the observed latent to the initial noise (see Appendix C.5.10.5 for details). The seed-consistency loss then enforces agreement between abducted seeds across a consistent contrast:

$$\mathcal{L}_1(Z_{0,X=x_0}, c_0, Z_{0,X=x_1}, c_1) \;=\; \Big\| \mathsf{Abduct}_\theta(Z_{0,X=x_0}, c_0) - \mathsf{Abduct}_\theta(Z_{0,X=x_1}, c_1) \Big\|_2^2.$$

Remark. With deterministic sampling, shared ω implies shared non-descendant content. Minimizing $\mathcal{L}_1$ suppresses nuisance differences and attributes variation to mechanic-specific interventions.

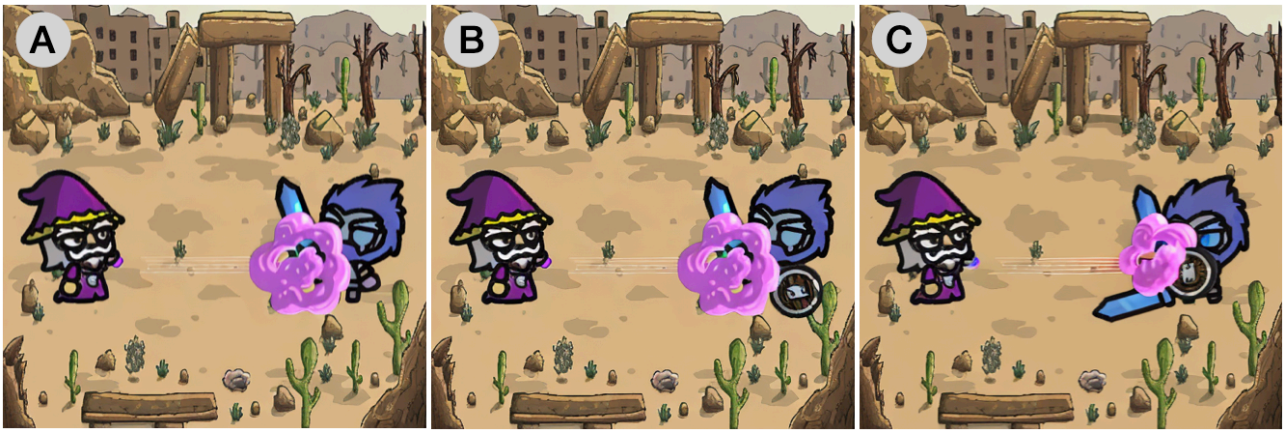


Figure 8.4: Proof-of-concept of shield mechanic learning with a diffusion model trained on game data. Mechanic learning is shown by sampling counterfactuals: unchanged image elements remain consistent with the original. (A) Held-out game image: wizard attacks warrior (blue due to an ice elemental buff). (B) Generated counterfactual with warrior holding a shield but not blocking ($V_{S=1,B=0}$). (C) Generated counterfactual with shield and block active ($V_{S=1,B=1}$) (aliased sword is a generation error).

$\mathcal{L}_2$: structure-alignment loss. Let $S \subset \{1, \ldots, T\}$ be a subset of high-noise timesteps. For each $t \in S$, the denoiser predicts noise $\epsilon_\theta(Z_{t,X=x_j}, t, c_j)$. We align predictions across the contrast:

$$\mathcal{L}_2 \;=\; \sum_{t \in S} \Big\| \epsilon_\theta(Z_{t,X=x_0}, t, c_0) - \epsilon_\theta(Z_{t,X=x_1}, t, c_1) \Big\|_2^2.$$

Remark. Early reverse steps encode coarse layout. Aligning them enforces global semantic identity across contrasts, while leaving mechanic-specific differences to emerge in later, low-noise steps.

Choice of ℓ_2 Loss. We employ an ℓ_2 loss (i.e., $\| \cdot \|_2^2$) in both $\mathcal{L}_1$ and $\mathcal{L}_2$, consistent with standard diffusion model training under Gaussian noise assumptions, where the denoising objective corresponds to maximum likelihood estimation of a Gaussian conditional distribution. In the context of counterfactual contrastive learning, the ℓ_2 penalty enforces tight step-wise alignment of the denoising trajectories across parallel worlds, suppressing deviations that would otherwise accumulate over the diffusion process. This contrasts with an ℓ_1 formulation, which is more tolerant to local discrepancies and may permit misalignment across timesteps. A more detailed discussion of this design choice is provided in Sec. 7.4.

Evaluation Results. Fig. 8.4 shows qualitative counterfactual generations: the model preserves non-mechanic-related content while toggling the targeted shield mechanic. From a factual image v with $X = x$ we abduct the exogenous seed ω, then generate a counterfactual image v' with $X = x'$ while keeping ω fixed. This corresponds to sampling from $P(V_{X=x'} \mid X = x, V =$

v), i.e. counterfactual reconstruction with shared ω. This provides visual evidence that the fine-tuned model has learned aspects of the mechanic, not merely pixels.

To systematise this evaluation, we report quantitative results in Table 8.2. The causal consistency metrics provide a systematic approach to evaluating consistency [125, 176], and we include metrics for image quality and reconstruction quality [177, 178] to serve as table-stakes baselines. While the model demonstrates reasonable image-to-text alignment (CLIP score: 24.45) and strong image similarity (0.89) between factual and counterfactual pairs, the reconstruction metrics reveal challenges in perfectly inverting the diffusion process, with reconstruction PSNR at 10.73 dB. The high exogenous distance (0.67) suggests that our current implementation requires further optimization to achieve tighter alignment between parallel worlds. Despite these limitations, the model successfully generates semantically meaningful counterfactuals, as evidenced by the transfer CLIP score of 20.77.

Table 8.2: Evaluation metrics for diffusion fine-tuning on 1000 consistent-contrast pairs. Causal consistency is emphasized as the key criterion for mechanic learning; reconstruction and image quality provide baseline checks.

Metric Category	Metric	Value
Causal Consistency	CF Transfer CLIP Score	20.77
	Exogenous Distance (MSE)↓	0.671
Reconstruction	Reconstruction PSNR	10.73 dB
	Reconstruction SSIM	0.183
	Reconstruction CLIP Score	21.45
Image Quality	PSNR (Factual vs CF)	17.66 dB
	SSIM (Factual vs CF)	0.433
	CLIP Image-Text Score	24.45
	CLIP Image-Image Similarity	0.892

8.7 Scalability & Limitations

Our descriptions of mechanics assume parallel-world statements (Sec. 8.3.1) with discrete differences across worlds; the same logic extends to incremental changes, but we have not yet characterized which mechanics require that expressivity. In our proof-of-concept, we operate on impact frame snapshots (Sec. 8.6), which makes experiments tractable but sidesteps temporal dynamics; in principle, the approach applies to any generative video model capable

of conditioning across worlds. As future work, we plan to benchmark video-capable world models and longer horizons.

Contrast generation scales linearly with the number of worlds per mechanic and seeds, whereas composing multiple mechanics can grow contrasts combinatorially. Our simplified 2D domain reduces compute expense and aids clarity, but limits transfer to photorealistic 3D and other genres, where new rendering conditions and longer horizons demand more complex models. Evaluation of consistency currently relies on human or VLM-based checks; the robustness of automated evaluators in this setting remains an open challenge.

Benchmarking and baselines. This chapter deliberately focuses on establishing the causal formalism and proof-of-concept demonstration rather than full-scale benchmarking. We therefore omit baseline comparisons against existing world models or video generators, as direct evaluation would require carefully aligned parallel-world datasets and causal consistency metrics that are not yet standardised. Developing this benchmark — including standardised datasets, metrics, and baseline evaluations — is the next step in this research direction. The *Multiverse Mechanica* formulation presented here lays the conceptual and infrastructural foundation for those benchmarking studies, which will assess state-of-the-art generative models under controlled causal interventions and measure their ability to learn and reproduce mechanics faithfully across worlds.

8.8 Summary

This chapter advanced the thesis goal of developing causal generative AI models capable of reasoning over *parallel worlds* by introducing *Multiverse Mechanica* — a causal, playable testbed for studying whether generative world models learn **mechanics**, not merely visual appearance. Building on the counterfactual contrastive principles of Chapter 7, we formalised game mechanics as modular structural equations governing state transitions and counterfactual dependencies, and instantiated these as executable causal mechanisms within a controlled simulation environment.

Through this framework, we demonstrated how causal formalisation and consistent parallel-world generation can make *causal consistency* an evaluable property of deep generative models. A proof-of-concept diffusion fine-tuning study illustrated how causal objectives can improve consistency behaviour by enforcing shared exogenous noise and invariance of non-descendant

features across contrasts. Although full benchmarking and baseline comparisons are deferred to future work, the chapter established the conceptual and infrastructural foundations for such studies, setting the stage for standardised evaluation of world models under causal interventions. Together, these components provide a reproducible path to assessing whether deep generative world models **can learn mechanics — not just pixels**.

In the broader context of this thesis, the work contributes directly to **Q1 - Modelling**, **Q2 - Structure and Parametrisation**, and **Q5 - Counterfactual Reasoning**. It shows how the learning of causal mechanisms can be operationalised and tested within deep generative settings, bridging from the static diffusion models of the previous chapter to interactive, dynamic worlds where agents must internalise and reproduce causal rules. For robotics, this represents a step toward learning the structural and functional components of world models necessary for reasoning, prediction, and decision-making under uncertainty.

While demonstrated in a game-based environment, the framework directly maps to robotics scenarios in which actions induce structured changes in the physical world. For example, in a block-stacking task, a robot may consider a candidate action such as placing a block at a specific pose and must reason about the resulting scene — e.g., predicting the RGB observation of the resulting tower state under that intervention. This reasoning may also depend on the robot's viewpoint, as the perceived outcome can vary with camera position, requiring the agent to anticipate how the scene would appear from different angles when selecting actions or planning navigation. Such queries require predicting how specific elements of the scene change while non-descendant components (e.g., unrelated objects or background structure) remain invariant. The parallel-world formulation provides a mechanism for generating and comparing these counterfactual outcomes, enabling models to learn which aspects of a scene should change under intervention and which should remain stable across modalities and viewpoints. This positions *Multiverse Mechanica* as a controlled testbed for studying causal consistency in generative models with direct relevance to robot planning, perception, and simulation.

The research in this chapter has been accepted for publication at ICLR 2026 as *'Multiverse Mechanica: A Testbed for Learning Game Mechanics via Counterfactual Worlds'* [10].

9 Conclusion

Contents

9.1 Key Insights from the Thesis 242
9.1.1 Design Trade-Offs in Causal Model Construction 242
9.1.2 Causality as a Framework for Integrated Model, Data, and Inference Design in Robotics 243
9.2 Summary of Contributions 245
9.3 Future Work 247
9.4 Thesis Impact 249

In this thesis, we investigated how causal generative AI can enhance robot understanding and reasoning of system physicality under uncertainty and causal complexity. Through the unifying conceptual framework of *Robot Causal Reasoning*, we have shown how causal representations — embedded within probabilistic, generative, and decision-theoretic formulations — provide the computational foundation for robust robot cognition, spanning perception, inference, decision-making, and explanation.

By progressively ascending Pearl's Ladder of Causation, the thesis advanced from *interventional* to *counterfactual* reasoning, demonstrating that the ability to explicitly represent, infer, and simulate causal mechanisms enables robots to reason about uncertainty, mitigate confounding bias, and generate human-aligned explanations of their actions. Together, these advances contribute toward a causally grounded foundation for explanation, simulation, and imagination in embodied and generative AI systems.

Before summarising the contributions of the thesis, we first distil a set of key insights that emerge across the presented works, capturing the core design principles that unify the approaches developed throughout this thesis.

9.1 Key Insights from the Thesis

9.1.1 Design Trade-Offs in Causal Model Construction

A central design question emerging across the presented works is how much of a causal model should be specified *a priori* versus learned from data. Across the thesis, a consistent design pattern emerges: high-level causal structure is informed by domain knowledge, while functional relationships and stochastic effects are learned from data.

In robotics, variables, causal ordering, and key structural dependencies often reflect known properties of physical systems and task formulations. For example, in Chapters 4 and 5, the higher-level decision-making causal graph structure encodes known relationships between actions, states, and observations, enabling principled reasoning under intervention. In contrast, the conditional distributions governing state transitions, observations, and latent effects are learned from data, allowing the model to capture uncertainty, variability, and complex multi-modal interactions.

This reflects a fundamental trade-off between inductive bias and flexibility. Hand-specified structure constrains the hypothesis space, improving sample efficiency, interpretability, and robustness to confounding, but risks misspecification if the assumed structure is incomplete or incorrect (e.g., confounding variables are missing from the assumed causal graph). Conversely, fully learned models offer greater expressivity but require large amounts of diverse, interventional data to reliably recover causal relationships, particularly in partially observable settings.

The results of this thesis suggest that a hybrid approach is both practical and effective for robotics. By anchoring models in causal structure while learning parameterisations and high-dimensional representations, systems can achieve both causal interpretability and empirical performance. This balance becomes increasingly important as models scale to complex environments, where neither purely symbolic nor purely data-driven approaches are sufficient in isolation.

9.1.2 Causality as a Framework for Integrated Model, Data, and Inference Design in Robotics

A second key insight arising from this thesis is that causal modelling in robotics is inherently a **joint design problem** across multiple interdependent components. Across the presented works, four elements must be jointly specified in a coherent and mutually consistent manner: (1) the model formulation, (2) the causal treatment, (3) the data collection process, and (4) the causal estimand. Rather than relying on ad-hoc design choices or post-hoc empirical validation, this thesis demonstrates that causality provides a principled and formal framework for addressing this joint design problem, enabling systematic reasoning about identifiability, intervention, and counterfactual inference, and supporting guarantees such as causal consistency in generative models.

1. Model Formulation. The model formulation defines the variables, causal structure, and functional relationships that describe the system. This includes selecting which variables to include, specifying causal dependencies, and determining which mechanisms are assumed versus learned. Across the thesis, this ranges from CBNs and structured SCMs over decision-making processes in Chapters 4, 5, and 6, to physics-grounded simulation models in Chapters 5 and 6, and fully specified generative data models in Chapters 7 and 8.

2. Causal Treatment. The causal treatment determines how interventions are defined and applied, including which variables are manipulated and which variables are conditioned upon to block confounding paths. Crucially, causality provides a principled basis for identifying when such treatment is required and how it should be applied: by analysing the structure of the causal graph, one can detect confounding (e.g., unblocked back-door paths), determine appropriate adjustment strategies (e.g., conditioning, mediators, or instrumental variables), and formally derive valid estimators using do-calculus. This, in turn, informs the design trade-off between hand-specified structure and learned components in **robotics and machine learning applications**: the aspects of the system required for valid causal identification (e.g., variables, dependencies, and adjustment sets) must be explicitly modelled, while remaining functional relationships and stochastic effects can be learned from data.

In Chapter 4, this is realised through explicit $do(\cdot)$ interventions to remove confounding in transition estimation, and to represent deliberative action selection in Chapter 5. In later

chapters (Chapters 6, 7, and 8), this extends to counterfactual interventions over generative processes, enabling reasoning across parallel worlds.

3. Data Collection. The data collection process constrains what can be observed and learned in practice. This includes which variables are measurable, how data is generated (observational or interventional), and whether the available data supports identification of the desired causal quantities. Across the thesis, this spans learning transition probabilities from interaction data in Chapter 4, estimation of Gaussian sensor noise and block placement error models from simulated trials in Chapters 5 and 6, controlled simulation environments (e.g., PyBullet), structured synthetic datasets (e.g., dSprites in Chapter 7), and causally instrumented environments that emit parallel-world data (Chapter 8).

4. Causal Estimand. The causal estimand defines the target quantity of interest, such as associative, interventional, or counterfactual queries, and determines how it can be estimated from the available data. These correspond to Levels 1–3 of Pearl's Ladder of Causation and are instantiated throughout the thesis, from interventional effect estimation in Chapter 4 to interventional estimation of task success probabilities for candidate placement actions in Chapter 5, and to counterfactual reasoning and consistency objectives in Chapters 6, 7, and 8.

Across all chapters, these four components are tightly coupled. Choices in one dimension directly constrain the others: the assumed model structure determines which causal effects are identifiable; the available data limits which parameters can be learned; and the target estimand dictates the required form of intervention and conditioning. This interdependence highlights that causal modelling is not a sequence of independent design decisions, but a unified process that must be considered holistically.

Taken together, this perspective provides a **principled framework** for designing causal systems in robotics, ensuring that model structure, intervention design, data generation, and inference procedures are jointly aligned with the underlying causal assumptions and the intended reasoning tasks.

We now summarise our contributions to each of the six research questions posed in Chapter 1.

9.2 Summary of Contributions

Q1 - Modelling: *How can causal generative machine learning models be used as abstractions of uncertain and highly non-linear world dynamics and relationships?*

Chapters 4, 5, 6, 7, and 8 each contributed to this question by developing causal generative abstractions of robot–world interactions across multiple levels of representation. Chapter 4 introduced the CAR-DESPOT framework, which integrated structural causal models (SCMs) into online POMDP planning, enabling interventional reasoning over world dynamics with unobserved confounding. Chapter 5 formalised causal generative modelling for manipulation through the COBRA-PPM architecture, combining probabilistic programming, physics-based simulation, and Bayesian inference to represent uncertainty in physical interaction. Chapter 6 extended these ideas into the counterfactual domain, transforming the COBRA-PPM world model into an explanatory SCM capable of simulating alternative task outcomes. Chapter 7 explored causal generative modelling in the high-dimensional space of diffusion models, using contrastive learning to enforce causal consistency between factual and counterfactual samples. Finally, Chapter 8 generalised these principles through *Multiverse Mechanica*, a generative world model that learns, contrasts, and evaluates parallel causal worlds for simulation-based reasoning. Together, these contributions demonstrated that causal generative models provide powerful, interpretable abstractions that unify prediction, inference, and imagination in robot cognition.

Q2 - Structure and Parametrisation: *How can human-specified knowledge and data-driven learning be used to define the structure and learn the parameterisation of formal causal models?*

Chapters 4, 5, 7, and 8 addressed this question through complementary approaches to causal structure and parameter learning. In Chapter 4, CAR-DESPOT integrated expert-defined structural priors with data-driven parameter estimation, yielding a hybrid causal-probabilistic planner that maintained interpretability while adapting to real-world stochasticity. In Chapter 5, the COBRA-PPM architecture combined symbolic causal priors with empirical learning of stochastic effects in a probabilistic programming framework, balancing causal interpretability with expressivity. Chapters 7 and 8 extended these ideas to learned generative models, where structural and parametric causal knowledge emerged from consistency objectives and model-based contrastive reasoning. Across all contributions, the thesis established that

causal structure learning provides a tractable bridge between expert modelling and statistical estimation, enabling robots to build causal world models that evolve with experience.

Q3 - Confounder Bias: *How can causal inference and intervention-based treatment methods be used to address confounding bias in the robot decision-making process?*

This question was primarily addressed in Chapter 4, where the SCM-UCPOMDP formulation extended online POMDP planning to explicitly account for unobserved confounders. By applying causal intervention calculus ($do(\cdot)$) to evaluate action effects, CAR-DESPOT corrected confounded value estimates and produced more causally faithful action policies. This work provided the first causal extension of the AR-DESPOT framework, offering a theoretically grounded and empirically validated method for mitigating confounding bias in robot planning and decision-making.

Q4 - Decision Making: *How can causal inference be used to extract actionable insights for robot decision-making that improve task performance and assure robust autonomy?*

Chapters 5 and 6 contributed to this research question by showing how causal inference supports both predictive and explanatory decision-making. In Chapter 5, the COBRA-PPM framework used causal Bayesian inference for action selection under uncertainty, enabling the robot to anticipate and adapt to variable physical outcomes. Chapter 6 extended these ideas to the explanatory domain, applying counterfactual reasoning to evaluate decision quality post-hoc and to assign causal responsibility for observed task outcomes. Together, these contributions showed that causal reasoning enhances both prospective (planning) and retrospective (evaluation) aspects of decision-making, promoting autonomy that is robust, interpretable, and aligned with causal reality.

Q5 - Counterfactual Reasoning: *How can level-3 counterfactual knowledge and inference algorithms be used to build models that understand system physicality and generate alternative outcomes consistent with human expectations and causal intuitions?*

Chapters 6, 7, and 8 advanced this question through the design of algorithms and representations that operationalise counterfactual reasoning. Chapter 6 demonstrated how robot task execution can be represented in a twin-world causal model to simulate hypothetical outcomes and explain behaviour in human terms. Chapter 7 introduced *Counterfactual Contrastive*

Learning, enforcing consistency between factual and counterfactual trajectories in latent diffusion models. Chapter 8 generalised these concepts to generative world modelling, treating counterfactual reasoning as parallel world simulation and comparison. Together, these works provided a unified framework for integrating level-3 causal reasoning into both interpretable and generative AI systems.

Q6 - Counterfactual Explanations: *How can counterfactual reasoning be used to generate faithful, human-aligned explanations of robot actions and outcomes, and attribute causal responsibility in support of explainable and trustworthy autonomy?*

This question was directly addressed in Chapter 6, which presented causal explanation methods for robot task execution. Building upon the causal models introduced in COBRA-PPM, the system generated post-hoc counterfactual explanations through interventions that quantified variable responsibility and produced human-aligned narratives of robot decisions. Integrating this approach with the *Ethical Black Box* concept proposed under RoboTIPS, the work established a tangible mechanism for auditing robot behaviour and attributing causal responsibility across perception, planning, and actuation. This contribution represents a concrete step toward transparent and trustworthy autonomy grounded in causal reasoning.

9.3 Future Work

While the thesis establishes a coherent causal foundation for robot cognition, several open challenges remain that define promising directions for future research. These directions arise directly from the outcomes and limitations identified across Chapters 4–8, and reflect opportunities to extend causal reasoning in robotics along the dimensions of scale, adaptability, human alignment, and generative consistency.

Scalability and Generalisation. The integration of causal reasoning into large-scale, multi-agent, and continuous-control domains remains an open research frontier. Extending the SCM-UCPOMDP and COBRA-PPM formulations (Chapters 4 and 5) to hierarchical, distributed, or cooperative multi-robot settings would allow agents to reason about inter-agent dependencies, causal influence, and shared uncertainty. Such extensions would enable more robust and generalisable autonomy in dynamic environments, where causal effects propagate across multiple

agents and latent physical processes. A further challenge lies in scaling causal inference to high-dimensional sensorimotor spaces while retaining interpretability and tractable computation.

Continual and Interactive Causal Learning. Future work should also explore learning causal structure dynamically through interaction, allowing robots to refine their world models over time. This builds upon the hybrid causal–statistical formulations in Chapters 4 and 5, where causal graphs and parameters were defined *a priori* or learned offline. A key future goal is to enable continual causal discovery, active intervention selection, and online structure learning in embodied agents. Integrating these capabilities with the counterfactual generative frameworks introduced in Chapters 7 and 8 would allow robots to update their causal beliefs from experience and maintain causal coherence as their environments evolve.

Human Evaluation of Counterfactual Explanations. While Chapter 6 demonstrated that counterfactual reasoning provides a principled mechanism for robot self-explanation, further empirical evaluation is required to assess how such explanations align with human expectations and causal judgements. Human-participant studies could evaluate interpretability, perceived trustworthiness, and the extent to which causal attributions match human reasoning about responsibility and intent. This would support the development of explanation strategies that are not only technically valid but also socially and ethically grounded.

Human Consistency Judgement of Parallel-World Generations. Similarly, human studies are needed to evaluate how parallel-world simulations generated by counterfactual generative models (Chapters 7 and 8) align with human notions of causal and perceptual consistency. For example, future experiments using visual datasets such as *dSprites* or interactive environments like *Multiverse Mechanica* could measure how human observers judge the plausibility and coherence of generated factual–counterfactual pairs. Such studies would provide valuable grounding for assessing causal alignment in generative AI and for benchmarking human-level causal understanding in simulation-based reasoning systems.

Causally Consistent Video Generation and Temporal Reasoning. Finally, the causal consistency principles developed for diffusion models and generative world models (Chapters 7 and 8) can be extended to video-based simulation and training. Moving beyond static image generation, future work could investigate causal diffusion models that learn and maintain temporal

coherence across frames, enabling robots and generative agents to reason over dynamic scenes and evolving interactions. This would further support causal simulation as a foundation for policy learning, world prediction, and temporal counterfactual inference in embodied systems.

9.4 Thesis Impact

This thesis contributes to the emerging convergence of robotics, causal inference, and generative AI, advancing a unified view of *causal generative cognition* for autonomous systems. It demonstrates that causal generative models unify perception, reasoning, and explanation — providing the algorithmic and representational foundations for robust, interpretable, and human-aligned autonomy.

From a scientific perspective, the thesis advances the state of the art in robot reasoning and cognitive modelling by:

1. Introducing **causal extensions to probabilistic planning and decision-making under uncertainty**, through the CAR-DESPOT framework, enabling robots to reason about interventional structure and mitigate confounding bias;

2. Formalising **causal Bayesian reasoning architectures for manipulation**, via the COBRA-PPM model, which combines probabilistic programming and causal inference for robust task execution;

3. Extending **counterfactual inference to explanation and evaluation**, through the development of the first post-hoc causal explanation framework for physical robots, linking causal attribution to the Ethical Black Box paradigm; and

4. Developing **causal consistency methods for multi-modal generative AI**, including Counterfactual Contrastive Learning and Multiverse Mechanica, which operationalise causal reasoning in high-dimensional generative and simulation-based models.

Together, these advances establish a coherent causal hierarchy across all six research questions posed in Chapter 1 — spanning modelling, learning, inference, decision-making, and explanation. They contribute a scientifically grounded and computationally unified foundation for robot cognition under uncertainty, situating causal reasoning as the connective medium between symbolic and data-driven approaches to autonomy.

From a societal perspective, the ability for autonomous systems to reason causally, imagine alternatives, and explain their behaviour coherently represents a pivotal step toward **trustworthy cognitive robotics**. Such systems can not only act effectively in the physical world but also understand, communicate, and justify the causes and consequences of their actions. This capability underpins the development of safe, transparent, and ethically aligned autonomous systems in domains such as human-robot collaboration, healthcare, and responsible AI.

In this way, the thesis contributes both conceptual and technical progress toward *assured autonomy*: intelligent systems whose actions are not only effective but also causally intelligible, accountable, and aligned with human reasoning about the world. By uniting causal inference with generative world modelling, the work lays the foundation for the next generation of robots and AI agents capable of reasoning, learning, and explaining through cause and effect.

References

[1] Yuchen Xiao et al. 'Online Planning for Target Object Search in Clutter under Partial Observability'. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2019, pp. 8241–8247.

[2] Shutterstock. *Businessman hand take one block from tower of wooden blocks and tower collapses.* 2018. URL: `https://www.shutterstock.com/image-photo/businessman-hand-take-one-block-tower-1027021183` (visited on 23/03/2026).

[3] Kevin Smith. *Tutorial: Probabilistic Programming.* MIT CBMM Summer Course 2018. 2018. URL: `https://www.youtube.com/watch?v=9SEIYh5BCjc` (visited on 23/03/2026).

[4] Lars Kunze and Michael Beetz. 'Envisioning the qualitative effects of robot manipulation actions using simulation-based projections'. In: *Artificial Intelligence* 221 (2015), pp. 1–23.

[5] Javier Alonso-Mora et al. 'Reactive mission and motion planning with deadlock resolution avoiding dynamic obstacles'. In: *Autonomous Robots* 42.4 (2018), pp. 801–824.

[6] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect.* Basic books, 2018.

[7] Ricardo Cannizzaro and Lars Kunze. 'CAR-DESPOT: Causally-Informed Online POMDP Planning for Robots in Confounded Environments'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Apr. 2023.

[8] Ricardo Cannizzaro, Jonathan Routley and Lars Kunze. 'Towards a Causal Probabilistic Framework for Prediction, Action-Selection & Explanations for Robot Block-Stacking Tasks'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2023 Workshop on Causality for Robotics*. 2023.

[9] Ricardo Cannizzaro et al. 'COBRA-PPM: A Causal Bayesian Reasoning Architecture Using Probabilistic Programming for Robot Manipulation Under Uncertainty'. In: *Proceedings of the 12th European Conference on Mobile Robots (ECMR)*. Padua, Italy, Sept. 2025.

[10] Ricardo Cannizzaro et al. 'Multiverse Mechanica: A Testbed for Learning Game Mechanics via Counterfactual Worlds'. In: *The Fourteenth International Conference on Learning Representations*. 2026.

[11] Ricardo Cannizzaro et al. 'Towards Probabilistic Causal Discovery, Inference & Explanations for Autonomous Drones in Mine Surveying Tasks'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2023 Workshop on Causality for Robotics*. 2023.

[12] Calum Imrie et al. 'Aloft: Self-Adaptive Drone Controller Testbed'. In: *Proceedings - 2024 IEEE/ACM 19th Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2024* (Apr. 2024), pp. 70–76.

[13] Ricardo Cannizzaro et al. *Team ORIon - 2022 Team Description Paper*. 2022.

[14] Samuel Sze et al. *ORIon-UTBMan 2023 Team Description Paper*. HAL-04500704. RoboCup, 2023. URL: `https://hal.science/hal-04500704`.

[15] Engineering and Physical Sciences Research Council (EPSRC). *RAILS: Responsible AI for Long-term Trustworthy Autonomous Systems*. 2022. URL: https://gtr.ukri.org/projects?ref=EP%2fW011344%2f1 (visited on 02/10/2025).

[16] Engineering and Physical Sciences Research Council (EPSRC). *RoboTIPS: Developing Responsible Robots for the Digital Economy*. 2019. URL: https://gtr.ukri.org/projects?ref=EP%2fS005099%2f1 (visited on 02/10/2025).

[17] Microsoft Research. *Societal Resilience: Building a More Resilient Society Through Mission-Driven Research and Applied Technology*. 2025. URL: https://www.microsoft.com/en-us/research/group/societal-resilience/ (visited on 02/10/2025).

[18] Microsoft Research. *AI Interaction and Learning (AIIL) Group: Focusing on Symbiotic AI Systems that Combine Human and AI Collaboration for Greater Value*. 2025. URL: https://www.microsoft.com/en-us/research/group/ai-interaction-and-learning/ (visited on 02/10/2025).

[19] Assuring Autonomy International Programme (AAIP). *Assuring Autonomy International Programme (AAIP). A GBP 12 million initiative funded by Lloyd's Register Foundation and the University of York to address the global challenge of safety assurance of Robotics and Autonomous Systems*. Programme ran 2018–2023. 2018. URL: https://www.york.ac.uk/assuring-autonomy/about/aaip/ (visited on 02/10/2025).

[20] Leslie Pack Kaelbling, Michael L. Littman and Anthony R. Cassandra. 'Planning and acting in partially observable stochastic domains'. In: *Artificial Intelligence* 101 (1-2 1998), pp. 99–134.

[21] Joelle Pineau, Geoffrey Gordon and Sebastian Thrun. 'Anytime point-based approximations for large POMDPs'. In: *Journal of Artificial Intelligence Research* 27 (2006), pp. 335–380.

[22] Oliver Brock, Jeff Trinkle and Fabio Ramos. 'SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces'. In: *Robotics: Science and Systems IV*. MIT Press, 2009, pp. 65–72.

[23] David Silver and Joel Veness. 'Monte-Carlo planning in large POMDPs'. In: *Advances in neural information processing systems* 23 (2010).

[24] Adhiraj Somani et al. 'DESPOT: Online POMDP planning with regularization'. In: *Advances in Neural Information Processing Systems*. Vol. 26. 2013.

[25] Hanna Kurniawati and Vinay Yadav. 'An Online POMDP Solver for Uncertainty Planning in Dynamic Environment'. In: *Robotics Research: The 16th International Symposium ISRR*. Cham: Springer International Publishing, 2016, pp. 611–629.

[26] Junzhe Zhang and Elias Bareinboim. *Markov Decision Processes with Unobserved Confounders: A Causal Approach*. Tech. rep. Technical Report R-23, Purdue AI Lab, 2016.

[27] Elias Bareinboim, Andrew Forney and Judea Pearl. 'Bandits with Unobserved Confounders: A Causal Approach'. In: *Advances in Neural Information Processing Systems*. 2015, pp. 1342–1350.

[28] Pedro A Ortega et al. 'Shaking the Foundations: Delusions in Sequence Models for Interaction and Control'. In: *arXiv preprint* (2021). arXiv: 2110.10819.

[29] Judea Pearl. 'The Seven Tools of Causal Inference, with Reflections on Machine Learning'. In: *Commun. ACM* 62 (3 Feb. 2019), pp. 54–60.

[30] Brenden M. Lake et al. 'Building machines that learn and think like people'. In: *Behavioral and Brain Sciences* 40 (Nov. 2017), e253.

[31] Thomas Hellström. 'The relevance of causation in robotics: A review, categorization, and analysis'. In: *Paladyn, Journal of Behavioral Robotics* 12 (1 2021), pp. 238–255.

[32] Judea Pearl. *Causality: Models, reasoning, and inference, second edition*. Cambridge university press, 2009.

[33] Elias Bareinboim et al. 'On Pearl's Hierarchy and the Foundations of Causal Inference'. In: *Probabilistic and Causal Inference: The Works of Judea Pearl*. Ed. by Hector Geffner, Rina Dechter and Joseph Y. Halpern. New York, NY, USA: Association for Computing Machinery, 2022, pp. 507–556.

[34] M Beetz and H Grosskreutz. 'Causal Models of Mobile Service Robot Behavior'. In: *Proceedings of AIPS* (1998), pp. 163–170.

[35] Erdi Aker et al. 'Causal reasoning for planning and coordination of multiple housekeeping robots'. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 6645 LNAI. Springer. 2011, pp. 311–316.

[36] Constantin Uhde et al. 'The robot as scientist: Using mental simulation to test causal hypotheses extracted from human activities in virtual reality'. In: *IEEE International Conference on Intelligent Robots and Systems*. 2020, pp. 8081–8086.

[37] Maximilian Diehl and Karinne Ramirez-Amaro. 'Why Did I Fail? A Causal-Based Method to Find Explanations for Robot Failures'. In: *IEEE Robotics and Automation Letters* 7 (4 2022), pp. 8925–8932.

[38] Maximilian Diehl and Karinne Ramirez-Amaro. 'A causal-based approach to explain, predict and prevent failures in robotic tasks'. In: *Robotics and Autonomous Systems* 162 (2023), p. 104376.

[39] Rhys Peter Matthew Howard and Lars Kunze. 'Simulation-Based Counterfactual Causal Discovery on Real World Driver Behaviour'. In: *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*. 2023.

[40] Ossama Ahmed et al. 'CausalWorld: A Robotic Manipulation Benchmark for Causal Structure and Transfer Learning'. In: *arXiv preprint* (2020). arXiv: 2010.04296.

[41] Michael Beetz et al. 'Cognition-Enabled Autonomous Robot Control for the Realization of Home Chore Task Intelligence'. In: *Proceedings of the IEEE* 100.8 (2012), pp. 2454–2471.

[42] Niloy Ganguly et al. 'A Review of the Role of Causality in Developing Trustworthy AI Systems'. In: *arXiv preprint* (2023). arXiv: 2302.06975.

[43] Pericle Salvini Alan F.T. Winfield Anouk van Maris and Marina Jirotka2. 'An Ethical Black Box for Social Robots: A Draft Open Standard'. In: *7th International Conference on Robot Ethics and Standards (ICRES)*. 2022.

[44] Rhys Peter Matthew Howard, Nick Hawes and Lars Kunze. 'Generating Causal Explanations of Vehicular Agent Behavioural Interactions with Learnt Reward Profiles'. In: *2025 IEEE International Conference on Robotics and Automation (ICRA)*. 2025, pp. 10416–10423.

[45] Rhys Peter Matthew Howard and Lars Kunze. 'Extending Structural Causal Models for Autonomous Vehicles to Simplify Temporal System Construction & Enable Dynamic Interactions Between Agents'. In: *Proceedings of the Fourth Conference on Causal Learning and Reasoning*. Ed. by Biwei Huang and Mathias Drton. Vol. 275. Proceedings of Machine Learning Research. PMLR, May 2025, pp. 1477–1505.

[46] Eli Bingham et al. 'Pyro: Deep Universal Probabilistic Programming'. In: *Journal of Machine Learning Research* 20 (2019), pp. 1–6.

[47] Loic Matthey et al. *dSprites: Disentanglement Testing Sprites Dataset*. 2017. URL: https://github.com/deepmind/dsprites-dataset/ (visited on 02/10/2025).

[48] T. Kloek and H. K. van Dijk. 'Bayesian Estimates of Equation System Parameters: An Application of Integration by Monte Carlo'. In: *Econometrica* 46.1 (1978), pp. 1–19.

[49] David Wingate and Theophane Weber. 'Automated Variational Inference in Probabilistic Programming'. In: *arXiv preprint* (2013). arXiv: 1301.1299.

[50] Toyota Motor Corporation. *Human Support Robot (HSR)*. 2024. URL: https://mag.toyota.co.uk/toyota-human-support-robot (visited on 15/03/2024).

[51] Nate Koenig and Andrew Howard. 'Design and use paradigms for Gazebo, an open-source multi-robot simulator'. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. IEEE. 2004, 2149–2154 vol.3.

[52] David Coleman et al. 'Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study'. In: *Journal of Software Engineering for Robotics* 5.1 (May 2014), pp. 3–16.

[53] Paul Voigt and Axel Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer International Publishing, Jan. 2017.

[54] Daniel Omeiza et al. 'Explanations in Autonomous Driving: A Survey'. In: *IEEE Transactions on Intelligent Transportation Systems* 23 (2021), pp. 10142–10162.

[55] Matthew Gadd et al. 'Sense–Assess–eXplain (SAX): Building Trust in Autonomous Vehicles in Challenging Real-World Driving Scenarios'. In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. 2020, pp. 150–155.

[56] L Kunze et al. 'Towards explainable and trustworthy collaborative robots through embodied question answering'. In: *IEEE International Conference on Robotics & Automation (ICRA) 2022 Workshop on collaborative robots and the work of the future*. 2022.

[57] Sarthak Jain and Byron C. Wallace. 'Attention is not Explanation'. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*. Association for Computational Linguistics, 2019, pp. 3543–3556.

[58] Sarah Wiegreffe and Yuval Pinter. 'Attention is not not Explanation'. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Jan. 2019, pp. 11–20.

[59] Richa Nahata et al. 'Assessing and Explaining Collision Risk in Dynamic Environments for Autonomous Driving Safety'. In: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*. Vol. 2021-Septe. IEEE. 2021, pp. 223–230.

[60] Daniel Omeiza et al. 'Why Not Explain? Effects of Explanations on Human Perceptions of Autonomous Driving'. In: *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts, ARSO*. Vol. 2021-July. 2021, pp. 194–199.

[61] David Lewis. *Counterfactuals*. Cambridge, MA: Harvard University Press, 1973.

[62] Tobias Gerstenberg. 'What would have happened? Counterfactuals, hypotheticals and causal judgements'. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 377 (1866 Dec. 2022).

[63] Joseph Y. Halpern. *Actual Causality*. MIT Press, 2016.

[64] Chad R. Samuelson and Joshua G. Mangelson. 'Embedding Scientific Knowledge via Visual Dirichlet Forests for Inference in Underwater Robotics'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2023 Workshop on Causality for Robotics*. 2023.

[65] Luca Castri et al. 'ROS-Causal: A ROS-based Causal Analysis Framework for Human-Robot Interaction Applications'. In: *Causal-HRI: Causal Learning for Human-Robot Interaction" workshop at the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Mar. 2024. eprint: 2402.16068.

[66] Aditya Ramesh et al. 'Zero-Shot Text-to-Image Generation'. In: *Proceedings of Machine Learning Research* 139 (Feb. 2021), pp. 8821–8831.

[67] Judea Pearl. 'On the Consistency Rule in Causal Inference: Axiom, Definition, Assumption, or Theorem?' In: *Epidemiology* 21 (6 Aug. 2010), pp. 872–875.

[68] Jake Bruce et al. 'Genie: Generative interactive environments'. In: *Forty-first International Conference on Machine Learning*. 2024.

[69] Jack Parker-Holder and Shlomi Fruchter. *Genie 3: A New Frontier for World Models*. 2025. URL: `https://deepmind.google/discover/blog/genie-3-a-new-frontier-for-world-models/` (visited on 24/09/2025).

[70] Decart et al. *Oasis: A Universe in a Transformer*. 2024. URL: `https://oasis-model.github.io/` (visited on 24/09/2025).

[71] Xianglong He et al. 'Matrix-Game 2.0: An Open-Source, Real-Time, and Streaming Interactive World Model'. In: *arXiv preprint* (2025). arXiv: `2508.13009`.

[72] Haoxuan Che et al. 'GameGen-X: Interactive Open-world Game Video Generation'. In: *International Conference on Learning Representations*. 2025.

[73] J. Gingerson, S. Amershi et al. 'World and Human Action Models Towards Gameplay Ideation'. In: *Nature* (2024). Microsoft Research technical report also available.

[74] David Ha and Jürgen Schmidhuber. 'World Models'. In: *arXiv preprint* (2018). arXiv: `1803.10122`.

[75] Seung Wook Kim et al. 'Learning to Simulate Dynamic Environments with GameGAN'. In: *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[76] Staffan Björk and Jussi Holopainen. *Patterns in Game Design*. Hingham, MA: Charles River Media, 2004.

[77] Tom Schaul. 'A Video Game Description Language for Model-based or Interactive Learning'. In: *Proceedings of the IEEE Conference on Computational Intelligence in Games*. 2013.

[78] Michael Thielscher. 'The General Game Playing Description Language is Universal'. In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*. 2011.

[79] Alexander Zook and Mark O. Riedl. 'Automatic Game Design via Mechanic Generation'. In: *arXiv preprint* (2019). arXiv: `1908.01420`.

[80] Priscilla Lo, David Thue and Elin Carstensdottir. 'What is a game mechanic?' In: *International Conference on Entertainment Computing*. Springer. 2021, pp. 336–347.

[81] Sus Lundgren and Staffan Bjork. 'Game mechanics: Describing computer-augmented games in terms of interaction'. In: *Proceedings of TIDSE*. Vol. 3. 2003.

[82] Tracy Fullerton, Chris Swain and Steven Hoffman. *Game design workshop: Designing, prototyping, & playtesting games*. CRC Press, 2004.

[83] Aki Järvinen. 'Games without Frontiers: Theories and Methods for Game Studies and Design'. PhD thesis. University of Tampere, 2008.

[84] Carlo Fabricatore. 'Gameplay and Game Mechanics: A Key to Quality in Videogames'. In: *Proc. OECD Expert Meeting on Videogames and Education*. 2007.

[85] Robin Hunicke, Marc LeBlanc and Robert Zubek. 'MDA: A Formal Approach to Game Design and Game Research'. In: *Proc. AAAI Workshop on Challenges in Game AI*. 2004.

[86] Alexander Zook and Mark Riedl. 'Automatic game design via mechanic generation'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 28. 2014.

[87] Joris Dormans. 'Engineering Emergence: Applied Theory for Game Design'. PhD thesis. University of Amsterdam, 2012.

[88] Robin J Evans. 'Graphs for margins of Bayesian networks'. In: *Scandinavian Journal of Statistics* 43.3 (2016), pp. 625–648.

[89] Ilya Shpitser and Judea Pearl. 'What Counterfactuals Can Be Tested'. In: *arXiv preprint* (2012). arXiv: 1206.5294.

[90] Judea Pearl. 'On the consistency rule in causal inference: axiom, definition, assumption, or theorem?' In: *Epidemiology* 21.6 (2010), pp. 872–875.

[91] Z. Chen, T. Xu, Y. Zhang et al. 'Model as a Game: On Numerical and Spatial Consistency for Generative Games'. In: *arXiv preprint* (2025). arXiv: 2503.21172.

[92] D. Kang, Y. Wu et al. 'How Far Is Video Generation from World Models?' In: *arXiv preprint* (2024). arXiv: 2402.19014.

[93] J. Riochet, H. Le Borgne, E. Ricci et al. 'IntPhys 2: A Benchmark for Physical Consistency in Video Prediction'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[94] F. Locatello et al. 'Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations'. In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. 2019.

[95] P. Spirtes, C. Glymour and R. Scheines. *Causation, Prediction, and Search*. 2nd. MIT Press, 2000.

[96] Elias Bareinboim and Judea Pearl. 'Causal Inference and the Data-Fusion Problem'. In: *Proceedings of the National Academy of Sciences* 113.27 (2016), pp. 7345–7352.

[97] B. Schölkopf et al. 'Towards Causal Representation Learning'. In: *Proceedings of the IEEE* 109.5 (2021), pp. 612–634.

[98] Ronan Riochet et al. 'IntPhys: A Framework and Benchmark for Visual Intuitive Physics Reasoning'. In: *arXiv preprint* (2018). arXiv: 1803.07616.

[99] Daniel M. Bear et al. 'Physion: Evaluating Physical Prediction from Vision in Humans and Machines'. In: *arXiv preprint* (2021). arXiv: 2106.08261.

[100] Mariya Hendriksen et al. 'Adapting Vision-Language Models for Evaluating World Models'. In: *arXiv preprint* (2025). arXiv: 2506.17967.

[101] Miguel Monteiro et al. 'Measuring Axiomatic Soundness of Counterfactual Image Models'. In: *arXiv preprint* (2023). arXiv: 2303.01274.

[102] Judea Pearl et al. *Probabilistic reasoning in intelligent systems : networks of plausible inference*. eng. Revised second printing. Morgan Kaufmann Series in Representation and Reasoning. San Francisco, California: Morgan Kaufmann Publishers, Inc., 1988.

[103] Thomas Bayes. 'An Essay Towards Solving a Problem in the Doctrine of Chances'. In: *Philosophical Transactions of the Royal Society of London* 53 (1763). Communicated by Richard Price, pp. 370–418.

[104] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[105] Nicholas Metropolis et al. 'Equation of State Calculations by Fast Computing Machines'. In: *Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092.

[106] W. K. Hastings. 'Monte Carlo Sampling Methods Using Markov Chains and Their Applications'. In: *Biometrika* 57.1 (1970), pp. 97–109.

[107] Michael I. Jordan et al. 'An Introduction to Variational Methods for Graphical Models'. In: *Machine Learning* 37.2 (1999), pp. 183–233.

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

[109] Paul W. Holland. 'Statistics and Causal Inference'. In: *Journal of the American Statistical Association* 81.396 (1986), pp. 945–960.

[110] James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. 2nd ed. New York: Springer, 1985.

[111] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 4th. Upper Saddle River, NJ: Pearson, 2021.

[112] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge, MA: MIT Press, 2018.

[113] Hanna Kurniawati. 'Partially Observable Markov Decision Processes and Robotics'. In: *Annual Review of Control, Robotics, and Autonomous Systems* 5.1 (2022), pp. 253–277.

[114] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York: John Wiley & Sons, 1994.

[115] Richard E. Bellman. *Dynamic Programming*. Princeton, NJ: Princeton University Press, 1957.

[116] S. Ross et al. 'Online Planning Algorithms for POMDPs'. In: *Journal of Artificial Intelligence Research* 32 (2008), pp. 663–704.

[117] Nan Ye et al. 'DESPOT: Online POMDP planning with regularization'. In: *Journal of Artificial Intelligence Research* 58 (2017), pp. 231–266.

[118] Cameron B. Browne et al. 'A Survey of Monte Carlo Tree Search Methods'. In: *IEEE Transactions on Computational Intelligence and AI in Games* 4.1 (2012), pp. 1–43.

[119] Robert Moss. *The Four Stages of the Monte Carlo Tree Search Algorithm*. Licensed under CC BY-SA 4.0. 2021. URL: `https://commons.wikimedia.org/wiki/File:MCTS_Algorithm.png` (visited on 20/03/2026).

[120] Herbert Robbins. 'Some Aspects of the Sequential Design of Experiments'. In: *Bulletin of the American Mathematical Society* 58.5 (1952), pp. 527–535.

[121] Tor Lattimore, Torsten Schäfer and Csaba Szepesvári. 'Causal Bandits: Learning Good Interventions via Causal Inference'. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2016, pp. 1181–1189.

[122] Diederik P. Kingma and Max Welling. 'Auto-Encoding Variational Bayes'. In: *arXiv preprint* (2014). arXiv: `1312.6114`.

[123] Danilo Jimenez Rezende, Shakir Mohamed and Daan Wierstra. 'Stochastic Backpropagation and Approximate Inference in Deep Generative Models'. In: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32. Proceedings of Machine Learning Research 2. 2014, pp. 1278–1286.

[124] Ashish Vaswani et al. 'Attention Is All You Need'. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2017, pp. 5998–6008.

[125] Alec Radford et al. 'Learning Transferable Visual Models From Natural Language Supervision'. In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*. Vol. 139. PMLR, 2021, pp. 8748–8763.

[126] Jonathan Ho, Ajay Jain and Pieter Abbeel. 'Denoising Diffusion Probabilistic Models'. In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020.

[127] Yang Song et al. 'Score-Based Generative Modeling through Stochastic Differential Equations'. In: *arXiv preprint* (2021). arXiv: 2011.13456.

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

[129] Jiaming Song, Chenlin Meng and Stefano Ermon. 'Denoising Diffusion Implicit Models'. In: *arXiv preprint* (2021). arXiv: 2010.02502.

[130] Robin Rombach et al. 'High-Resolution Image Synthesis with Latent Diffusion Models'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 10684–10695.

[131] Jonathan Ho and Tim Salimans. 'Classifier-Free Diffusion Guidance'. In: *arXiv preprint* (2022). arXiv: 2207.12598.

[132] Min Chen et al. 'POMDP-lite for Robust Robot Planning under Uncertainty'. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. May 2016, pp. 5427–5433.

[133] Trey Smith and Reid Simmons. 'Heuristic Search Value Iteration for POMDPs'. In: *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI)*. Banff, Canada: AUAI Press, 2004, pp. 520–527.

[134] Richard S Sutton. 'Integrated architectures for learning, planning, and reacting based on approximating dynamic programming'. In: *Proceedings of the 7th International Conference on Machine Learning (1990)*. Elsevier, 1990, pp. 216–224.

[135] Greg Brockman et al. 'OpenAI Gym'. In: *arXiv preprint* (2016). arXiv: 1606.01540.

[136] Mark Towers et al. 'Gymnasium: A Standard Interface for Reinforcement Learning Environments'. In: *arXiv preprint* (2024). arXiv: 2407.17032.

[137] Rajesh Ranganath, Sean Gerrish and David M. Blei. 'Black box variational inference'. In: *Journal of machine learning research* 33 (2014), pp. 814–822.

[138] Diederik P. Kingma and Jimmy Ba. 'Adam: A Method for Stochastic Optimization'. In: *arXiv preprint* (2015). arXiv: 1412.6980.

[139] Kurtland Chua et al. 'Deep reinforcement learning in a handful of trials using probabilistic dynamics models'. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 4759–4770.

[140] Danijar Hafner et al. 'Mastering Diverse Domains through World Models'. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Jan. 2023.

[141] Danijar Hafner et al. 'Learning Latent Dynamics for Planning from Pixels'. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 2555–2565.

[142] Julian Schrittwieser et al. 'Mastering Atari, Go, chess and shogi by planning with a learned model'. In: *Nature 2020 588:7839* 588 (7839 Dec. 2020), pp. 604–609.

[143] Markku Suomalainen, Yiannis Karayiannidis and Ville Kyrki. 'A survey of robot manipulation in contact'. In: *Robotics and Autonomous Systems* 156 (2022), p. 104224.

[144] Jack Collins et al. 'RAMP: A Benchmark for Evaluating Robotic Assembly Manipulation and Planning'. In: *IEEE Robotics and Automation Letters* 9.1 (2024), pp. 9–16.

[145] Jianlan Luo et al. 'Robust Multi-Modal Policies for Industrial Assembly via Reinforcement Learning and Demonstrations: A Large-Scale Study'. In: *Proceedings of Robotics: Science and Systems (RSS)*. 2021.

[146] Richard D Smallwood and Edward J Sondik. 'The Optimal Control of Partially Observable Markov Processes over a Finite Horizon'. In: *Operations research* 21 (5 1973), pp. 1071–1088.

[147] Haoyu Bai et al. 'Intention-aware online POMDP planning for autonomous driving in a crowd'. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 454–460.

[148] Matthew Budd et al. 'Bayesian reinforcement learning for single-episode missions in partially unknown environments'. In: *6th Conference on Robot Learning (CoRL 2022)*. OpenReview, 2022.

[149] Erwin Coumans and Yunfei Bai. *PyBullet, a Python module for physics simulation for games, robotics and machine learning*. http://pybullet.org, accessed 2024-03-15. 2016.

[150] R. Tyrrell Rockafellar and Stanislav Uryasev. 'Optimization of Conditional Value-at-Risk'. In: *Journal of Risk* 3 (2000), pp. 21–41.

[151] Santiago Garrido-Jurado et al. 'Automatic generation and detection of highly reliable fiducial markers under occlusion'. In: *Pattern Recognition* 47.6 (2014), pp. 2280–2292.

[152] William J Youden. 'Index for rating diagnostic tests'. In: *Cancer* 3.1 (1950), pp. 32–35.

[153] P Salvini et al. 'Human involvement in autonomous decision-making systems. Lessons learned from three case studies in aviation, social care and road vehicles'. In: *Frontiers in Political Science* 5 (2023).

[154] Lara Radojcic. *Causal Reasoning for Action Selection and Explanations in a Collaborative Robot Block Stacking Task*. Tech. rep. Engineering Science Fourth Year Project Final Report, University of Oxford, 2024.

[155] Pericle Salvini. *RoboTIPS: Developing Responsible Robotics for the Digital Economy*. 2024. URL: `https://www.robotips.co.uk/` (visited on 30/09/2024).

[156] S.J. Tobin. 'Attribution'. In: *Encyclopedia of Human Behavior*. Ed. by V.S. Ramachandran. 2nd ed. San Diego: Academic Press, 2012, pp. 236–242.

[157] Emre Kcman et al. 'Causal Reasoning and Large Language Models: Opening a New Frontier for Causality'. In: *arXiv preprint* (2023). arXiv: `2305.00050`.

[158] C. W. J. Granger. 'Investigating Causal Relations by Econometric Models and Cross-spectral Methods'. In: *Econometrica* 37.3 (1969), pp. 424–438.

[159] Diederik P. Kingma and Max Welling. 'Auto-Encoding Variational Bayes'. In: *CoRR* abs/1312.6114 (2013).

[160] Stephan Bongers, Tineke Blom and Joris M. Mooij. 'Causal Modeling of Dynamical Systems'. In: *arXiv preprint* (2018). arXiv: `1803.08784`.

[161] T. M. Mitchell. *The Need for Biases in Learning Generalizations*. Technical Report CBM-TR-117, Rutgers University, 1980.

[162] D. H. Wolpert. 'The Lack of A Priori Distinctions Between Learning Algorithms'. In: *Neural Computation* 8.7 (1996), pp. 1341–1390.

[163] Glynn Winskel. 'Event structures'. In: *Advanced course on Petri nets*. Springer. 1986, pp. 325–392.

[164] Alfred V. Aho et al. *Compilers: Principles, Techniques, and Tools*. 2nd. Boston, MA: Addison-Wesley, 2006.

[165] Clark Glymour, Kun Zhang and Peter Spirtes. 'Review of causal discovery methods based on graphical models'. In: *Frontiers in genetics* 10 (2019), p. 524.

[166] luciI. *Chibi Artstyle Model*. Community submission on Civitai. 2024. URL: https://civitai.com/models/22820/chibi-artstyle (visited on 24/09/2025).

[167] Lvmin Zhang, Anyi Rao and Maneesh Agrawala. 'Adding Conditional Control to Text-to-Image Diffusion Models'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2023, pp. 3813–3824.

[168] Scott McCloud. *Understanding Comics: The Invisible Art*. Reprint edition. Harper Perennial, 2020.

[169] Will Eisner. *Comics & Sequential Art*. Revised edition. First published 1985; expanded edition 1990; revised edition 2008. New York: W. W. Norton & Company, 2008, p. 192.

[170] Neil Cohn. *The Visual Language of Comics: Introduction to the Structure and Cognition of Sequential Images*. Bloomsbury Advances in Semiotics. London: Bloomsbury Academic, 2013.

[171] CraftPix.net. *Chibi Game Character Sprites Collection*. 2025. URL: https://craftpix.net/sets/chibi-game-character-sprites-collection/ (visited on 11/10/2025).

[172] CraftPix.net. *Tiny Fantasy Characters Sprite Collection*. 2025. URL: https://craftpix.net/sets/tiny-fantasy-characters-sprite-collection/ (visited on 11/10/2025).

[173] Robert Nystrom. *Game Programming Patterns*. Genever Benning, 2014. Chap. 14, Component.

[174] Jason Gregory. *Game Engine Architecture*. 3rd ed. Boca Raton, FL: A K Peters/CRC Press, 2018, pp. 1043–1062.

[175] PromptHero. *OpenJourney v4*. 2022. URL: https://huggingface.co/prompthero/openjourney-v4 (visited on 20/03/2026).

[176] Jack Hessel et al. 'CLIPScore: A Reference-free Evaluation Metric for Image Captioning'. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2021, pp. 7514–7528.

[177] Zhou Wang et al. 'Image quality assessment: From error visibility to structural similarity'. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612.

[178] Stephen Wolf, Margaret Pinson et al. *Reference Algorithm for Computing Peak Signal to Noise Ratio (PSNR) of a Video Sequence with a Constant Delay*. Tech. rep. NTIA/ITS / COM-9 C.6. Proposed reference implementation and calibration procedures for PSNR. National Telecommunications and Information Administration (NTIA) / ITS, 2009.

[179] Farama Foundation. *Frozen Lake Environment Gymnasium*. 2024. URL: https://gymnasium.farama.org/environments/toy_text/frozen_lake/ (visited on 23/10/2024).

[180] Gorilla. *Gorilla Experiment Builder*. 2024. URL: https://gorilla.sc/experiment-builder (visited on 23/10/2024).

[181] Rensis Likert. *A technique for the measurement of attitudes. Arch*. 1st ed. Vol. 140. Albany : State University of New York Press, 1932.

[182] Tero Karras et al. 'Elucidating the design space of diffusion-based generative models'. In: *Advances in neural information processing systems* 35 (2022), pp. 26565–26577.

[183] Eli Bingham et al. 'Pyro: Deep universal probabilistic programming'. In: *Journal of machine learning research* 20.28 (2019), pp. 1–6.

[184] Charles Tapley Hoyt et al. 'Causal Identification with Y_0'. In: *arXiv preprint* (2025). arXiv: 2508.03167.

Appendices

A

Human-Participant Study Design

This appendix presents the design of a proposed human-participant study, referenced in Sec. 6.7.2, intended to evaluate how counterfactual explanations influence human understanding and trust in robot autonomy. Although the study was not conducted as part of this thesis, its design is included here for completeness and to support future empirical evaluation.

The purpose of this section is to outline the experimental structure, hypotheses, and evaluation domains required to empirically assess the interpretability and usefulness of responsibility-based counterfactual explanations. In particular, the study is designed to test whether counterfactual explanations align more closely with human causal attributions than interventional or descriptive explanations, and whether they support improved trust calibration.

Hypotheses. Three hypotheses are proposed for demonstrating that counterfactual explanations are better aligned with human causal reasoning and can increase trust in autonomous systems:

- **H1:** Counterfactual-based explanations for robot task execution have better alignment with human causal attributions than hypothetical or interventional explanations.
- **H2:** In settings involving artificial agents, human causal attribution is biased toward prioritising agent decision-making errors (intent) before perception and action errors (non-agent randomness), even when the observed outcome is the same.
- **H3:** Counterfactual-based explanations can help increase human confidence, understanding, and trust in robot autonomy.

Study Objectives. To test these hypotheses, the study will measure:

1. How accurately participants can predict robot behaviour after receiving different explanation types (interpretability);
2. How confident participants are in the robot's reliability (trust calibration); and
3. How well each explanation format supports causal reasoning about success and failure (human causal alignment).

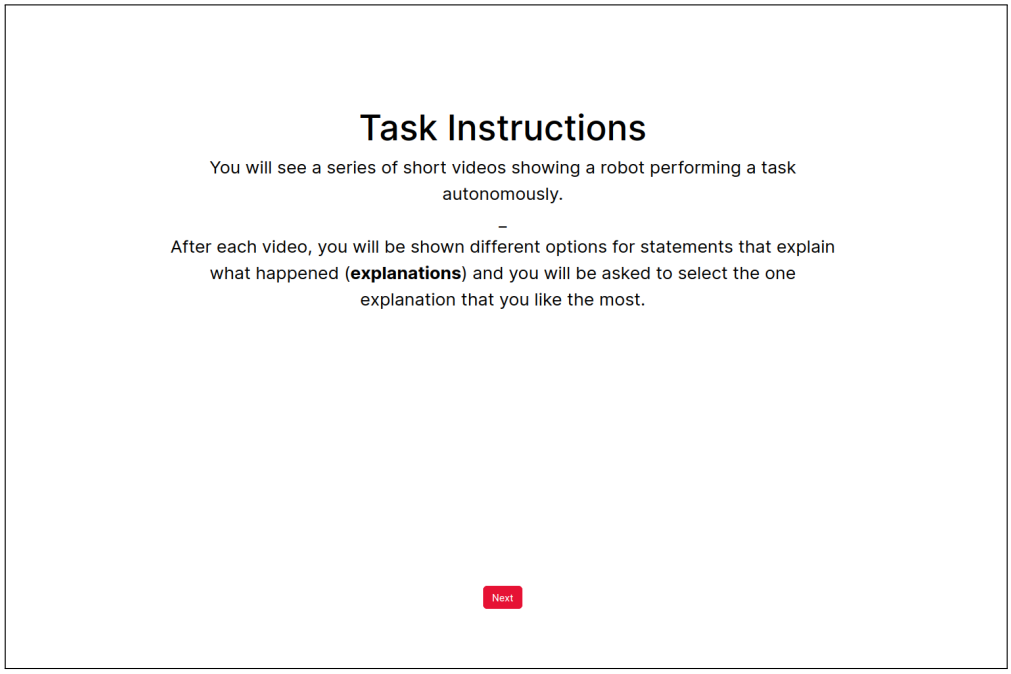
Experimental Domains. Two task domains are proposed for evaluation:

D1: Frozen Lake. The first domain is the *Frozen Lake* problem, widely used in reinforcement learning and planning research [179]. An agent must traverse a 2D grid world to reach a goal while avoiding holes, under deterministic or stochastic (slippery) surface conditions. This environment enables rapid generation of many episodes, allowing large-scale quantitative testing. We will generate explanations of success and failure that consider: (i) the agent's choice of action or learned policy; (ii) whether a slip event occurred; and (iii) whether the episode ended due to timeout. This domain is well suited for large-scale online human-subject evaluation.

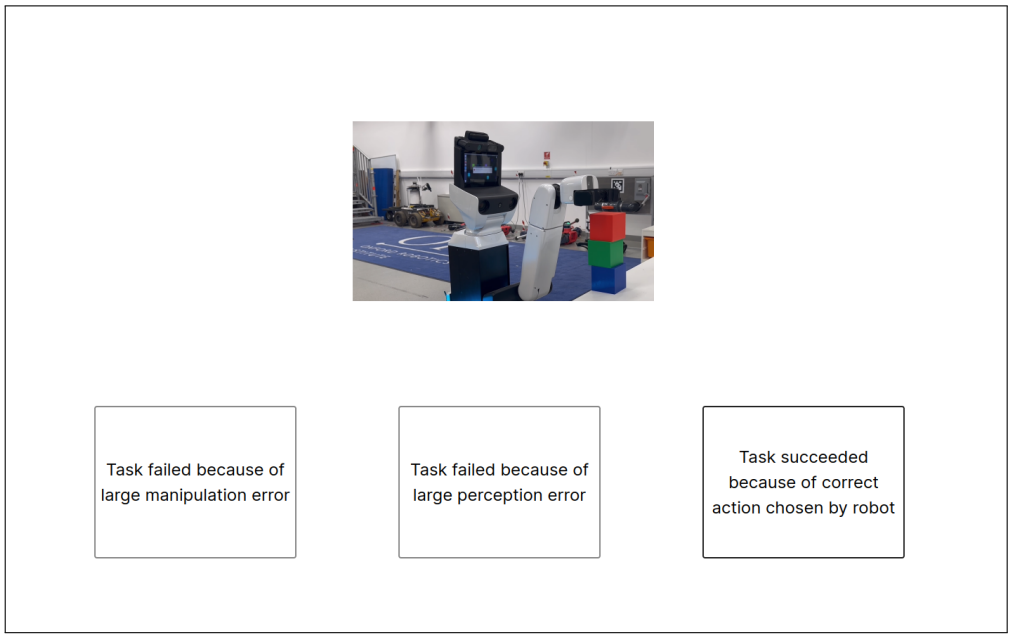
D2: Robot Block Stacking Task. The second domain is the robot block stacking task presented earlier in this chapter. Due to its higher complexity and slower simulation cycle in Gazebo, this domain will be used for smaller-scale qualitative evaluation on both simulated and real Toyota HSR robots. We will generate explanations of success and failure outcomes that consider: (i) the agent's choice of action parameterisation variables; (ii) the magnitude of perception error; and (iii) the magnitude of manipulation error. This domain provides a more realistic robotic setting for testing counterfactual explanation generation and user interpretation.

Proposed Experiments. Two complementary human-participant experiments are planned:

E1: Multiple cause variables of different types. Participants will be shown short videos of robot task episodes, each accompanied by several natural-language text explanations generated by different methods (e.g., interventional, counterfactual). They will be asked to select which explanation they find most convincing or intuitive. This experiment will measure preferences and perceived causal alignment, addressing hypotheses H1 and H2. An illustration of the planned online experiment interface, implemented in the Gorilla platform [180], is shown in Fig. A.1.



(a) Draft instructions, shown to participants at the start of the experiment.



(b) Example trial showing a robot task video and multiple candidate explanations from different methods.

Figure A.1: Initial design of the explanation preference task for the human-participant study, implemented using the Gorilla online experiment builder [180]. Participants view robot task episodes and select or rate explanations generated by different methods.

E2: Human confidence, understanding, and trust in robot autonomy. In this experiment, participants will be divided into groups that each receive explanations of one type: none (control), hypothetical, hypothetical agent-action prioritised, counterfactual, or counterfactual agent-action prioritised. Participants will first review a short description of the robot task and its expected performance (e.g., typical success rate). Before viewing any task episodes, participants will rate their initial confidence, understanding, and trust in the robot (e.g., using a 5-point Likert scale [181]). They will then observe a sequence of robot task episodes paired with explanations generated by their assigned method. Afterward, participants will again rate their confidence, understanding, and trust, and may also provide qualitative feedback. The change in these ratings across groups will reveal which explanation method produces the greatest increase in understanding and trust, addressing hypothesis H3.

Expected outcomes. We hypothesise that participants exposed to counterfactual explanations will demonstrate higher comprehension accuracy and better trust calibration than those exposed to interventional or baseline conditions. In particular, participants receiving responsibility-based explanations should show improved reasoning about the causal dependencies between actions and outcomes, and a clearer understanding of why tasks succeed or fail. We expect that results of this future study will provide empirical evidence on how counterfactual explanations influence human causal reasoning and trust in robot autonomy, informing future development of adaptive, context-aware, and ethically transparent robot explainer systems.

B

Prompt Specification for Caption Generation

This appendix provides the exact prompting procedure used to generate natural-language captions for constructing the *Counterfactual dSprites* dataset in Ch. 7. Captions are generated using GPT-4 via a structured, multi-stage prompting scheme conditioned on symbolic latent variables from the dSprites dataset.

B.1 Overview

The prompting procedure consists of two sequential stages:

1. Generation of a caption from latent variables
2. Generation of a minimally edited caption under modified latent variables

Each stage is conditioned on structured inputs derived directly from the underlying latent variables. For clarity and consistency, all continuous latent variables (scale, orientation, and position) are represented using values rounded to two decimal places in the prompt.

B.2 Stage 1: Caption Generation

The model is prompted to generate a single-sentence description of an image from its latent variables:

> You are an AI assistant that creates captions for dSprites images.
> You will be presented with a Python list of tuples describing an image.

Possible values are:
Shape: square (1.0), ellipse (2.0), heart (3.0)
Scale: 6 values linearly spaced in [0.5, 1]
Orientation: 40 values in [0, 2π]
Position X: 32 values in [0, 1]
Position Y: 32 values in [0, 1]

Based on these values, create a one sentence description of the image. Use simple language and do not refer explicitly to numeric values.

Examples:

Values: [('shape': 3.0), ('scale': 1.00), ('orientation': 6.12), ('position_x': 0.03), ('position_y': 0.52)]
Caption: A large heart shape, oriented slightly to the right, positioned near the left edge and centrally along the vertical axis.

Values: [('shape': 2.0), ('scale': 0.90), ('orientation': 4.35), ('position_x': 0.03), ('position_y': 0.00)]
Caption: A moderately large ellipse, tilted to the left, located very close to the left edge and at the very top of the image.

Now here are the values: {values}
Provide the caption.

B.3 Stage 2: Minimally Edited Caption Generation

Given a modified set of latent variables, the model is prompted to generate a new caption that preserves the original phrasing except for the attributes that have changed:

Now here are the values for another image. Some of the values have changed.

Create a caption that uses the exact same wording as the original caption except for the elements that have changed.

Examples:

Original Values: [('shape': 3.0), ('scale': 1.00), ('orientation': 6.12), ('position_x': 0.03), ('position_y': 0.52)]
Original Caption: A large heart shape, oriented slightly to the right, positioned near the left edge and centrally along the vertical axis.
New Values: [('shape': 1.0), ('scale': 1.00), ('orientation': 6.12), ('position_x': 0.03), ('position_y': 0.52)]
New Caption: A large square shape, oriented slightly to the right, positioned near the left edge and centrally along the vertical axis.

Original Values: [('shape': 2.0), ('scale': 0.90), ('orientation': 4.35), ('position_x': 0.03), ('position_y': 0.00)]
Original Caption: A moderately large ellipse, tilted to the left, located very close to the left edge and at the very top of the image.
New Values: [('shape': 2.0), ('scale': 0.20), ('orientation': 4.35), ('position_x': 0.03), ('position_y': 0.00)]
New Caption: A very small ellipse, tilted to the left, located very close to the left edge and at the very top of the image.

Now here were the original values: {values}
Here was the original caption: {caption}
Now here are the new values: {new_values}
Provide the new caption.

B.4 Discussion

This structured prompting procedure enforces a tight correspondence between symbolic latent variables and natural-language descriptions. In particular, Stage 2 explicitly constrains the generated caption to preserve lexical structure, ensuring that only the modified attributes are updated while all other elements remain unchanged.

This design reduces linguistic variability and supports controlled evaluation of whether generated images reflect the intended edits while preserving invariant factors, corresponding to non-descendant variables as defined in Sec. 7.2.

C
Multiverse Mechanica

C.1 Proof-of-Concept Dataset

Here, we describe the dataset used for the proof-of-concept training presented in Sec. 8.6, used to learn the **shield gameplay mechanic**.

C.1.1 Generation

We generate a level-3 dataset, as described in Sec. 8.4.3. We automatically derive parallel world interventions based on the full game DAG, targeting the variables relevant to the parallel world contrast statements for the shield mechanic. The set of derived interventions define multiple parallel world tuples, enumerated such that we sufficiently cover the support of joint distribution of the mDAG induced by the query variables associated with the level-3 parallel world statements for the mechanic.

We assign these interventions to multiple game instances with a shared 'ω' (same random seed and initial conditions) and generate $N = 1000$ consistent contrasts, constituting level-3 data. Since we are training on text and images (i.e., not gameplay clips), for each we generate only a subset of training artefacts consisting of these two modalities: 1) the impact frame as a 512x512 PNG image, and 2) the game-state variable outcomes (converted into a caption in the pre-processing method, outlined below).

Figure C.1: Impact frame generated by Multiverse Mechanica for a world in a parallel-world tuple.

C.1.2 Pre-Processing

To align our artefact modalities with the text-to-image latent diffusion architecture used in our proof-of-concept, we must perform a pre-processing step to convert from the game-state variable outcomes — a dictionary of variable-name/value pairs — to a text caption. We implement a templated captioning process for each parallel world tuple, by which plain-text captions are deterministically constructed based on the values of the game-state variables.

Consider, for example, the following impact frame image and game-state variable artefacts generated by Multiverse Mechanica for one of the worlds in a parallel world tuple.

For this world, Multiverse Mechanica has generated the impact frame image shown in Fig. C.1 and the following game-play state:

```
gameplay_state = {
    'background': 'forest',
    'player action': 'melee attack',
    'player class': 'warrior',
    'player does block': False,
    'player element': 'ice',
    'player has shield': False,
    'player is hurt': False,
    'player is immune to attack': False,
```

```
    'player␣weapon': 'short␣sword',
    'player␣weapon␣class': 'light',
    'player␣weapon␣element': 'ice',
    'player␣weapon␣is␣light': True,
    'player␣weapon␣range': 'melee',
    'opponent␣action': 'defend',
    'opponent␣class': 'warrior',
    'opponent␣does␣block': True,
    'opponent␣element': 'none',
    'opponent␣has␣shield': True,
    'opponent␣is␣hurt': False,
    'opponent␣is␣immune␣to␣attack': False,
    'opponent␣weapon': 'short␣sword',
    'opponent␣weapon␣class': 'light',
    'opponent␣weapon␣element': 'fire',
    'opponent␣weapon␣is␣light': True,
    'opponent␣weapon␣range': 'melee',
}
```

Given this game-play state, our pre-processing step constructs the following caption:

> 'A 1-on-1 battle between two warriors. The warrior on the left has a ice element buff. The warrior on the left's weapon is a short sword with a ice buff. The warrior on the left is launching a melee attack. The warrior on the right's weapon is a short sword with a fire buff and they have a shield. The warrior on the right is blocking with their shield.'

The captions generated by the pre-processing step is combined with the generated image to thus create training data suitable for text-to-image latent diffusion architecture used in our proof-of-concept.

C.2 Mechanics Implemented in Multiverse Mechanica v1.0

In this section, we describe the ground truth causal structure and mechanics Multiverse Mechanica. We describe each game mechanic in terms of:

- *level-2* interventional statements: A set of causal hypothetical statements of the form 'Given preconditions W, all else equal, if X, then Y.'
- A set of causal Markov kernels encompassing the relevant conditional probability distributions in the causal DAG.

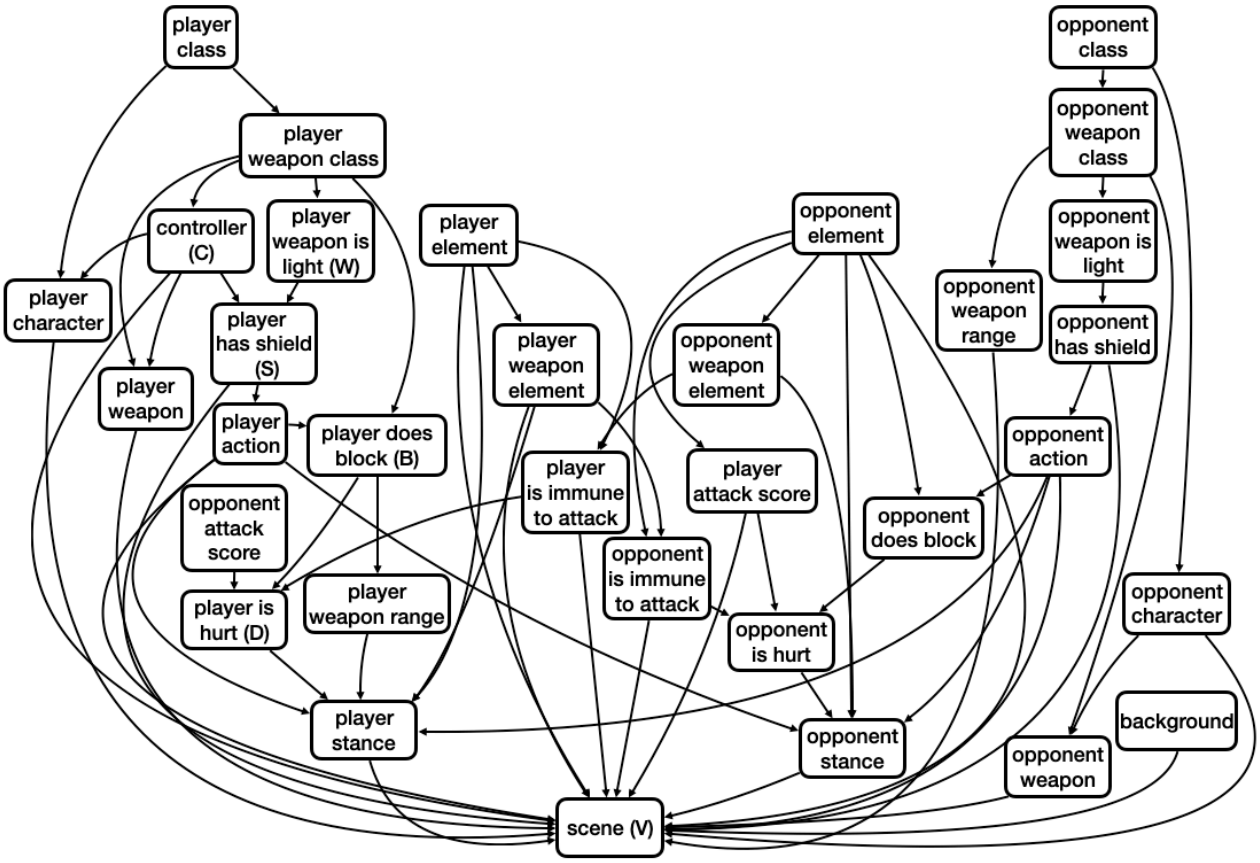


Figure C.2: Full causal DAG of a turn in Multiverse Mechanica

- *level-3 parallel world* statements: A set of counterfactual statements of the form 'Given preconditions W, all else equal, if X, then Y AND if X' then Y'.'

- A set of counterfactual probability expressions defining the induced counterfactual outcome probabilities.

Particularly, in the case of the level-3 parallel world statements, we can enumerate counterfactual worlds in which we make a change and the target outcome variable(s) change, with guarantees on consistency, with respect to the game mechanic.

We formulate the counterfactual cases by taking interventions to change variables (**bold**), from the factual case (Case 0). Downstream changes are shown in *italics*.

Full Causal DAG Fig. C.2 shows the full causal DAG of a turn in Multiverse Mechanica v1.0.

C.2.1 Shield Mechanic

The warrior character can equip a shield they can use to block incoming attacks, if they have a free hand and perform the 'defend' action.

(a) Factual: The warrior is equipped with a one-handed short-sword and a shield, and blocks the arrow with a shield.

(b) Counterfactual: The warrior is equipped with a two-handed long-sword, and no shield, and is unable to block the attack.

Figure C.3: Example counterfactual contrast statement for the shield mechanic.

C.2.1.1 level-2 Interventional Definition

We can completely define this mechanic by a set of contrasting statements:

- If a player has a small weapon, they may hold a shield, otherwise they cannot.
- If a player has a shield, they may block incoming attacks, otherwise they cannot.
- If a player blocks an incoming attack, they avoid taking damage, otherwise they may take damage.

Involved variables:

- `player action` : [idle, melee attack, defend]
- `opponent weapon class` : [small, large]
- `opponent has shield` : [True, False]

- `opponent action` : [idle, melee attack, defend]
- `opponent is hurt` : [True, False]

Latent variables:

- `opponent weapon is heavy` : [one-handed, two-handed]

C.2.1.2 Causal Markov Kernels

In our model, *block* is one of the actions. Thus, we can reason about the shield game mechanic in terms of the player/opponent action being sampled as *block*.

The causal Markov kernel for the opponent action variable encompasses the shield game mechanic:

$$P\Big(\text{opponent action} \;\Big|\; \text{opponent has shield, opponent weapon}\Big).$$

To reason about the opponent blocking, we consider the probability that the action variable is *block*:

$$P\Big(\text{opponent action} = \text{block} \;\Big|\; \text{opponent has shield, opponent weapon}\Big).$$

To reason about the opponent getting hurt, we consider the probability that the opponent gets hurt, given the opponent action and player action:

$$P\Big(\text{opponent is hurt} \;\Big|\; \text{opponent action, opponent has shield, opponent weapon}\Big).$$

An important detail to take note of is that the causal DAG does not have a variable to represent the character's outcome (e.g. getting hit, blocking, dodging) — it only has a notion of final stance. However, the stance variable is currently unused after sampling; the battle video game handles the underlying outcome logic and renders the scene.

C.2.1.3 level-3 Counterfactual Parallel World Statements

We can rewrite the level-2 interventional statements as the following level-3 parallel world statements:

1. Given a character had a heavy weapon and they did not have a shield; if the character would have had a light weapon instead, then they could have equipped a shield.

2. Given a character had not had a shield and did not block; if the character had a shield instead, then they could have blocked.

3. Given a character had a shield but did not block and took damage; if the character had blocked instead, then they would have avoided taking damage from the attack.

C.2.1.4 Counterfactual Probability Expressions

We can formulate these natural-language statements in mathematical notation.

Notation. Let:

- WC = weapon class,
- HS = has shield,
- B = does block,
- H = is hurt.

Then we have:

$$P\Big(HS_{WC=\text{Light}} = \text{True} \,\Big|\, WC = \text{Heavy},\ HS = \text{False}\Big) > 0,$$

$$P\Big(B_{HS=\text{True}} = \text{True} \,\Big|\, HS = \text{False},\ B = \text{False}\Big) > 0,$$

$$P\Big(H_{B=\text{True}} = \text{False} \,\Big|\, HS = \text{True},\ B = \text{False},\ H = \text{True}\Big) = 1.$$

C.2.2 Elemental Immunity Mechanic

Under the elemental immunity game mechanic, players may have elemental attributes; either fire or ice. Weapons may also have elemental attributes. If the attacking player weapon element and opponent element are the same (and non-none), the opponent is granted elemental immunity, and thus avoids taking damage from the attack, since they are already imbued with the element the attacker is using to attack them.

The elemental attributes of characters are indicated by variations from their base appearance:

- *none*: Characters appear in their standard appearances
- *fire*: Characters appear with orange shading and/or red outline

- *ice*: Characters appear with light blue shading and/or blue outline

The elemental attributes of weapons are similarly indicated by variations from their base appearance:

- *none*: Weapons appear in their standard appearances
- *fire*: Weapons appear with orange shading and/or red outline. Melee weapon slash visual effects appear with red shading. Ranged weapon projectiles (e.g., arrows, spell attacks) and explosions appear with red shading.
- *ice*: Weapons appear with light blue and/or blue outline. Melee weapon slash visual effects appear with blue shading. Ranged weapon projectiles and explosions appear with blue shading.

(a) Factual: The fire wizard is immune to the warrior's melee attack with a fire-type short sword.

(b) Counterfactual: The fire wizard takes damage from the warrior's melee attack with an ice-type short sword.

Figure C.4: Example counterfactual contrast statement for the elemental immunity mechanic.

C.2.2.1 level-2 Interventional Definition

We can completely define this mechanic by a set of contrasting statements:

- If a player has an elemental attribute, they may wield a weapon with the same elemental attribute but they cannot wield one of an opposing element in the fire-ice dichotomy. For

example, a player with a fire-elemental attribute may wield a fire-elemental weapon (or non-elemental weapon); they cannot wield an ice-elemental weapon.

- If a player does not have an elemental attribute, they may wield any elemental-type weapon; otherwise the rule above applies.
- If a player is hit by an incoming attack from a weapon with the same elemental attribute, they are immune and thus avoid taking damage; otherwise they may take damage.

Involved variables:

- `player element` : [none, fire, ice]
- `player weapon element` : [none, fire, ice]
- `opponent element` : [none, fire, ice]
- `opponent is immune to attack` : [True, False]
- `opponent is hurt` : [True, False]

Latent variables: None.

C.2.2.2 Causal Markov Kernels

The element of the player and the opponent are both independent variables:

$$P(\text{player element}),$$

$$P(\text{opponent element}).$$

The causal Markov kernel for the player weapon elemental is as follows:

$$P\Big(\text{player weapon element} \;\Big|\; \text{player element}\Big).$$

To reason about the opponent being immune to the attack, we consider the conditional probability given the opponent's element and player's weapon element:

$$P\Big(\text{opponent is immune to attack} \;\Big|\; \text{player weapon element, opponent element}\Big).$$

To reason about the opponent getting hurt, we consider the probability that the opponent gets hurt, given the opponent being immune:

$$P\Big(\text{opponent is hurt} \;\Big|\; \text{opponent is immune to attack}\Big).$$

C.2.2.3 level-3 Counterfactual Parallel World Statements

We can rewrite the level-2 interventional statements as the following level-3 parallel world statements:

1. Given a character had a fire elemental attribute and they had a fire elemental weapon; if the character would have had an ice elemental attribute instead, then they could have had an ice- or none-elemental weapon but not a fire elemental.

2. Given a character had a fire elemental attribute and they had a fire elemental weapon; if the character would have had no elemental attribute instead, then they could have had any elemental or non-elemental type weapon.

3. Given a character had an ice elemental attribute and was hit with an attack from an ice elemental weapon, and thus was immune and did not take damage; if the character had a fire elemental attribute instead, then they would have taken damage from the attack.

C.2.2.4 Counterfactual Probability Expressions

We can formulate these natural language statements in mathematical notation.

Notation. Let:

- $PE =$ player element,
- $PWE =$ player weapon element,
- $OE =$ opponent element,
- $I =$ is immune to attack,
- $H =$ is hurt.

Then we have:

Statement 1.

$$P\Big(PWE_{PE=\text{Ice}} = \text{Fire} \,\Big|\, PE = \text{Fire},\ PWE = \text{Fire}\Big) = 0,$$

$$P\Big(PWE_{PE=\text{Ice}} = \text{Ice} \,\Big|\, PE = \text{Fire},\ PWE = \text{Fire}\Big) > 0,$$

$$P\Big(PWE_{PE=\text{Ice}} = \text{None} \,\Big|\, PE = \text{Fire},\ PWE = \text{Fire}\Big) > 0.$$

Statement 2.

$$P\Big(PWE_{PE=\text{None}} = \text{Fire} \,\Big|\, PE = \text{Fire},\ PWE = \text{Fire}\Big) > 0,$$

$$P\Big(PWE_{PE=\text{None}} = \text{Ice} \,\Big|\, PE = \text{Fire},\ PWE = \text{Fire}\Big) > 0,$$

$$P\Big(PWE_{PE=\text{None}} = \text{None} \,\Big|\, PE = \text{Fire},\ PWE = \text{Fire}\Big) > 0.$$

Statement 3.

$$P\Big(H_{OE=\text{Fire}} = \text{True} \,\Big|\, PWE = \text{Ice},\ OE = \text{Ice}\Big) = 1.$$

C.2.3 Weapon Range Mechanic

Under the weapon range game mechanic, weapon classes have range attributes; either melee or ranged. Each weapon class is either a melee or ranged type. For example, the light sword, the heavy sword, and the dagger are melee weapons; while the bow, the staff, and the throwing knife are ranged weapons. The weapon range attribute affects primarily visual elements of the battle scene:

1. **Stance:** a character attacking with a melee weapon will be depicted in the snapshot physically swinging the melee weapon at the opponent's location, while a character attacking with a ranged weapon will be depicted shooting/throwing/casting from their position.

2. **Scene:** a character attacking with a melee weapon will be shown in the game scene (video) first approaching the opponent's position before physically swinging the melee weapon, while a character attacking with a ranged weapon will be depicted shooting/-throwing/casting from their position and the emitted projectile will be shown travelling towards the opponent from left to right (possibly on a parabolic trajectory, if affected by gravity, e.g. the arrow shot from the bow).

(a) Factual: The assassin performs a melee attack on a wizard with a sword.

(b) Counterfactual: The assassin performs a ranged attack on a wizard with a throwing knife.

Figure C.5: Example counterfactual contrast statement for the weapon range mechanic.

C.2.3.1 level-2 Interventional Definition

We can completely define this mechanic by a set of contrasting statements:

- If a weapon is one of the weapon classes *light sword*, *heavy sword*, or *dagger*, it is a melee weapon; otherwise if it is *bow*, *staff*, or *throwing knife* it is a ranged weapon.
- If a player wields a melee weapon, they will be depicted in the snapshot as performing a melee attack at the opponent's location; otherwise they will be depicted as performing a ranged attack from their own location.
- If a player wields a melee weapon, they will be shown in the game scene (video) first approaching the opponent's position before physically swinging the melee weapon; otherwise they will be depicted shooting/throwing/casting from their position and the emitted projectile will be shown travelling towards the opponent from left to right.

Involved variables:

- `player weapon class` : [light sword, heavy sword, dagger, bow, staff, throwing knife]
- `player weapon range` : [melee, ranged]
- `player stance` : pixels in the rendered image snapshot

- `scene` : pixels in the rendered gameplay video

Latent variables: None.

C.2.3.2 Causal Markov Kernels

The causal Markov kernel for the player weapon range is as follows:

$$P\Big(\text{player weapon range} \;\Big|\; \text{player weapon class}\Big).$$

To reason about the player stance rendered in the image snapshot, we consider the conditional probability given the player weapon and player weapon range:

$$P\Big(\text{stance} \;\Big|\; \text{player weapon, player weapon range}\Big).$$

Similarly, to reason about the scene rendered in the gameplay video, we consider the conditional probability given the player weapon, player weapon range, and player stance:

$$P\Big(\text{scene} \;\Big|\; \text{player weapon, player weapon range, stance}\Big).$$

C.2.3.3 level-3 Counterfactual Parallel World Statements

We can rewrite the level-2 interventional statements as the following level-3 parallel world statements:

1. Given a character had a light sword and they had a melee weapon; if the character would have had a bow instead, then they would have had a ranged weapon.

2. Given a character had a melee weapon and they were depicted in the snapshot as performing a melee attack at the opponent's location; if the character would have had a ranged weapon instead, then they would have been depicted as performing a ranged attack from their own location.

3. Given a character had a melee weapon and they were shown in the game scene video first approaching the opponent's position before physically swinging the melee weapon; if the character would have had a ranged weapon instead, then they would have been depicted shooting/throwing/casting from their position and the emitted projectile would have been shown travelling towards the opponent from left to right.

C.2.3.4 Counterfactual Probability Expressions

We can formulate these natural language statements in mathematical notation.

Notation. Let:

- PWC = player weapon class,
- PWR = player weapon range,
- PS = player stance,
- S = scene.

Then we have:

Statement 1.

$$P\Big(PWR_{PWC=\text{bow}} = \text{ranged} \,\Big|\, PWC = \text{light sword},\ PWR = \text{melee}\Big) = 1.$$

Statement 2.

$$P\Big(PS_{PWR=\text{ranged}} = \text{depicted performing ranged attack} \,\Big|\, PWR = \text{melee},\ PS = \text{depicted performing melee attack}\Big) = 1, \tag{C.1}$$

$$P\Big(PS_{PWR=\text{ranged}} = \text{depicted performing melee attack} \,\Big|\, PWR = \text{melee},\ PS = \text{depicted performing melee attack}\Big) = 0. \tag{C.2}$$

Statement 3.

$$P\Big(S_{PWR=\text{ranged}} = \text{shown performing ranged attack} \,\Big|\, PWR = \text{melee},\ PS = \text{shown approaching the opponent and performing melee attack}\Big) = 1, \tag{C.3}$$

$$P\Big(S_{PWR=\text{ranged}} = \text{shown performing melee attack} \,\Big|\, PWR = \text{melee},\ PS = \text{shown approaching the opponent and performing melee attack}\Big) = 0. \tag{C.4}$$

C.2.4 Spell-Casting Mechanic

Under the spell-casting mechanic, the wizard character may perform one of five spells:

1. **Spawn Magic Projectile Spell to Perform Ranged Attack**
2. **Summon Cloud Platform Spell to Dodge Attack**
3. **Self-Transform Spell to Increase Melee Strength**
4. **Opponent Transform Spell to Lower Enemy Defence**
5. **Levitation Spell to Disarm Opponent**

None of these spell actions can be mitigated by the enemy. Even in the case of the self-transform spell, the sheer size of the transformed wizard (into a golem) renders any block action by the opponent useless. Each spell action choice affects the *stance* and *scene* variables in the rendered game.

C.2.4.1 level-2 Interventional Definition.

All spells share the same base level-2 contrasting statements:

- If the player character is a wizard, they may wield a magical staff, otherwise they cannot.
- If the character is equipped with a magical staff, they may cast spells, otherwise they cannot.
- If the character casts a spell, they gain a particular offensive or defencive benefit to help them in the battle, otherwise they do not.

C.2.4.2 Causal Markov Kernels

Notation. Let:

- PC = player character
- PW = player weapon
- PA = player action
- PF = player form

- PS = player stance
- OC = opponent character
- OW = opponent weapon
- OA = opponent action (e.g., defend)
- OF = opponent form
- OS = opponent stance
- B = opponent successfully blocks (semantic variable, only true if the defend action actually succeeds)
- D = opponent dodges
- H = opponent is hurt
- Stance and Scene are the rendered variables

Latent variables:

- OIA = opponent incoming attack indicator

Shared enabling kernels.

$$P(PW = \text{staff} \mid PC = \text{wizard}) > 0$$

$$P(PA \in \{\text{spell actions}\} \mid PW \neq \text{staff}) = 0$$

$$P(PA \mid PW = \text{staff}, \text{state}) \text{ is exogenous (agent policy)}$$

Shared state & rendering kernels.

$$P(PF, PS, OF, OS \mid PA, OA, PC, PW, OC, OW)$$

$$P(\text{Stance} \mid PF, PS, OF, OS, PA, OA)$$

$$P(\text{Scene} \mid \text{Stance}, PF, PS, OF, OS, PA, OA)$$

Shared interaction/outcome kernels.

$$P(B \mid PF, OA)$$

$$P(D \mid PS, OIA)$$

$$P(H \mid B, D, PA)$$

C.2.4.3 level-3 Counterfactual Parallel World Statements.

All spells also share a common set of counterfactual statements:

1. Given a character was not a wizard and therefore could not wield a magical staff; if the character had been a wizard instead, then they could have wielded a magical staff.
2. Given a character was not equipped with a magical staff and therefore could not cast spells; if the character had been equipped with a staff instead, then they could have cast spells.
3. Given a character did not cast a spell and therefore received no benefit in the battle; if the character had cast a spell instead, then they would have gained the corresponding offensive or defencive benefit.

C.2.4.4 Counterfactual Probability Expressions (Shared).

Not wizard $\rightarrow$ *Wizard* $\Rightarrow$ *Staff access*

$$P\Big(PW_{PC=\text{wizard}} = \text{staff} \,\Big|\, PC \neq \text{wizard},\ PW \neq \text{staff}\Big) > 0.$$

No staff $\rightarrow$ *Staff* $\Rightarrow$ *Spell casting possible*

$$P\Big(PA_{PW=\text{staff}} \in \{\text{spell actions}\} \,\Big|\, PW \neq \text{staff},\ PA \notin \{\text{spell actions}\}\Big) > 0.$$

C.2.5 Spawn Magic Projectile Spell to Perform Ranged Attack

The wizard can cast a spell to conjure and launch a magical projectile directly at the opponent. This action is a ranged attack, and its visual and mechanical dynamics follow the same principles as other ranged weapons described in Sec. C.2.3.

C.2.5.1 level-2 Interventional Definition.

- If the wizard casts the spawn projectile spell, a magical projectile is launched towards the opponent (i.e., $OIA = \text{True}$), otherwise no projectile is spawned ($OIA = \text{False}$).
- If a magical projectile strikes the opponent and is not successfully blocked or dodged, the opponent is hurt, otherwise they are not.

Involved variables:

- PA = player action
- OA = opponent action (e.g., defend, dodge)
- PS = player stance
- B = opponent successfully blocks
- D = opponent dodges
- H = opponent is hurt

Latent variables:

- OIA = opponent incoming attack indicator

C.2.5.2 Causal Markov Kernels.

$$P(OIA \mid PA)$$

$$P(B \mid PF, OA)$$

$$P(D \mid PS, OIA)$$

$$P(H \mid B, D, PA)$$

C.2.5.3 level-3 Counterfactual Parallel World Statements.

1. Given the wizard idled and there was no incoming attack; if the wizard had cast the spawn projectile spell instead, then the opponent would have faced an incoming attack ($OIA = \text{True}$).
2. Given the wizard cast the spawn projectile spell and the projectile hit (i.e., no successful block or dodge) and the opponent was hurt; if the opponent had successfully blocked or dodged instead, then they would not have been hurt.

C.2.5.4 Counterfactual Probability Expressions.

Idle → Projectile spell ⇒ Incoming attack present

$$P\Big(OIA_{PA=\text{projectile spell}} = \text{True} \,\Big|\, PA = \text{idle}, OIA = \text{False}\Big) = 1.$$

Projectile hits → Successful block ⇒ Not hurt

$$P\Big(H_{B=\text{True}} = \text{False} \,\Big|\, PA = \text{projectile spell}, OIA = \text{True}, B = \text{False}, H = \text{True}\Big) = 1.$$

Projectile hits → Successful dodge ⇒ Not hurt

$$P\Big(H_{D=\text{True}} = \text{False} \,\Big|\, PA = \text{projectile spell}, OIA = \text{True}, D = \text{False}, H = \text{True}\Big) = 1.$$

C.2.5.5 Summon Cloud Platform Spell to Dodge Attack

The wizard has a spell they can use to summon a levitating cloud platform and use it to raise themselves up above the battlefield to dodge incoming enemy attacks.

level-2 Interventional Definition.

- If the wizard casts the cloud platform spell, a platform is summoned; the wizard moves onto it and is levitated upwards above the battlefield, otherwise they remain on the ground.
- If the wizard is in an elevated position on a platform during an incoming attack, they successfully dodge and avoid damage, otherwise if they are on the ground they remain vulnerable and may take damage.

Involved variables:

- PA = player action
- OA = opponent action
- OS = opponent stance (grounded, elevated)
- H = opponent is hurt

Latent variables: None.

Causal Markov Kernels.

$$P(OS \mid PA)$$

$$P(D \mid OS, OIA),$$

$$P(H \mid D).$$

level-3 Counterfactual Parallel World Statements.

1. Given the wizard performed the idle action and remained on the ground; if the wizard had cast the cloud platform spell instead, then they would have been elevated.
2. Given the wizard remained on the ground during an incoming attack and was hurt; if the wizard had cast the cloud platform spell instead, then they would have been elevated and dodged the attack, and thus not been hurt.

Counterfactual Probability Expressions.

Idle → Cloud Platform ⇒ Elevated stance

$$P\Big(OS_{PA=\text{cloud platform}} = \text{elevated} \,\Big|\, PA = \text{idle},\, OS = \text{grounded}\Big) = 1.$$

Grounded, hurt during incoming attack → Cloud Platform ⇒ Dodge, not hurt

$$P\Big(H_{OS=\text{elevated}} = \text{False} \,\Big|\, PA = \text{idle},\, OS = \text{grounded},\, OIA = \text{True},\, H = \text{True}\Big) = 1.$$

C.2.5.6 Self-Transform Spell to Increase Melee Strength

The wizard can cast a spell to transform themselves into a large golem. In this form, their melee attacks are enormously strengthened and cannot be blocked by opponents.

level-2 Interventional Definition.

- If the wizard casts the self-transform spell, they transform into a golem, otherwise they remain in their normal form.
- If the wizard is transformed into a golem, their melee attack cannot be blocked due to sheer size and strength, otherwise it can.

(a) Factual: The wizard idles and is struck by the archer's arrow.

(b) Counterfactual: The wizard summons a magical cloud platform to gain an elevated position and dodge the archer's arrow.

Figure C.6: Example counterfactual contrast statement for the summon cloud platform spell mechanic.

- If the wizard's golem-form melee attack is not successfully blocked, the opponent will be hurt, otherwise they will not be.

Involved variables:

- PA = player action
- OA = opponent action (e.g., defend)
- B = opponent successfully blocks
- H = opponent is hurt

Latent variables:

- PF = player form (normal, golem)

Causal Markov Kernels.

$$P(PF \mid PA)$$

$$P(B \mid PF, OA)$$

$$P(H \mid B, PA)$$

level-3 Counterfactual Parallel World Statements.

1. Given the wizard performed the idle action and their form remained unchanged; if the wizard had cast the self-transform spell instead, then they would have transformed into a mighty golem.

2. Given the wizard cast a projectile spell and it was blocked by the opponent; if the wizard had cast the self-transform spell instead, then they would have transformed into a golem and the melee attack would not have been blocked by the opponent.

3. Given the wizard cast a projectile spell and it was blocked by the opponent and thus they were not hurt; if the wizard had cast the self-transform spell instead, then they would have transformed into a golem and the melee attack would have bypassed the block and hurt the opponent.

Counterfactual Probability Expressions.

Idle $\rightarrow$ *Self-Transform* $\Rightarrow$ *Golem form*

$$P\Big(PF_{PA=\text{self-transform}} = \text{golem} \,\Big|\, PA = \text{idle},\ PF = \text{normal}\Big) = 1.$$

Magic projectile attack blocked $\rightarrow$ *Self-Transform* $\Rightarrow$ *Block attempt failed*

$$P\Big(B_{PF=\text{golem}} = \text{True} \,\Big|\, PA = \text{magic projectile attack},\ PF = \text{normal},\ OA = \text{defend}, B = \text{True}\Big) = 0.$$

Magic projectile attack blocked, not hurt $\rightarrow$ *Self-Transform* $\Rightarrow$ *Hurt despite block action*

$$P\Big(H_{PF=\text{golem}} = \text{True} \,\Big|\, PA = \text{magic projectile attack},\ PF = \text{normal},\ OA = \text{defend},\ H = \text{False}\Big) = 1.$$

C.2.5.7 Opponent Transform Spell to Lower Enemy Defence

The wizard can cast a spell to transform an opponent into a weak or harmless creature (i.e., a snail), disarming them and preventing them from defending against incoming attacks.

(a) Factual: The wizard casts a magic projectile spell but the warrior blocks it.

(b) Counterfactual: The wizard casts a spell to transform itself into a large golem to perform a powerful melee attack.

Figure C.7: Example counterfactual contrast statement for the self-transform spell mechanic.

level-2 Interventional Definition.

- If the wizard casts no spell, the opponent retains their normal form and can block or defend as usual.
- If the wizard casts the opponent transform spell, the opponent is transformed (e.g. into a snail), otherwise they remain in their normal form.
- If the opponent is transformed, they are disarmed and cannot block; otherwise they may block.

Involved variables:

- PA = player action
- OA = opponent action
- B = opponent successfully blocks
- H = opponent is hurt

Latent variables:

- OF = opponent form (normal, transformed)

Causal Markov Kernels.

$$P(OF \mid PA),$$

$$P(H \mid OF, B).$$

level-3 Counterfactual Parallel World Statements.

1. Given the wizard cast a projectile spell and the opponent remained in their normal form; if the wizard had cast the opponent transform spell instead, then the opponent would have been transformed into a snail.

2. Given the wizard cast a projectile spell and it was blocked by the opponent, and thus the opponent was not hurt; if the wizard had cast the opponent transform spell instead, then the opponent would have been unable to block and would have been hurt by a follow-up attack.

Counterfactual Probability Expressions.

Idle or projectile spell → Opponent Transform ⇒ Opponent form changes

$$P\Big(OF_{PA=\text{opponent transform}} = \text{transformed} \,\Big|\, PA \neq \text{opponent transform},\ OF = \text{normal}\Big) = 1.$$

Projectile spell blocked, not hurt → Opponent Transform ⇒ Block disabled (opponent disarmed)

$$P\Big(B_{OF=\text{transformed}} = \text{True} \,\Big|\, PA = \text{magic projectile attack},\ OF = \text{normal},\ B = \text{True}\Big) = 0.$$

C.2.5.8 Levitation Spell to Disarm Opponent

The wizard can cast a spell that lifts the opponent into the air, leaving them unable to defend or block effectively, and rendering them disarmed.

(a) Factual: The wizard casts a magic projectile spell but the warrior blocks it.

(b) Counterfactual: The wizard casts a spell to transform the warrior into a snail to disarm them.

Figure C.8: Example counterfactual contrast statement for the opponent transform spell mechanic.

level-2 Interventional Definition.

- If the wizard casts no spell, the opponent remains grounded and may block normally.
- If the wizard casts the levitation spell, the opponent is lifted into the air, otherwise they remain grounded.
- If the opponent is levitated, they cannot block and are effectively disarmed; otherwise they may block.

Involved variables:

- PA = player action
- OA = opponent action
- OS = opponent stance (grounded, levitating)
- B = opponent successfully blocks
- H = opponent is hurt

Latent variables: None.

Causal Markov Kernels.

$$P(OS \mid PA),$$

$$P(H \mid OS, B)$$

level-3 Counterfactual Parallel World Statements.

1. Given the wizard cast a projectile spell and the opponent remained grounded; if the wizard had cast the levitation spell instead, then the opponent would have been levitated.

2. Given the wizard cast a projectile spell and it was blocked by the opponent, and thus the opponent was not hurt; if the wizard had cast the levitation spell instead, then the opponent would have been unable to block and left vulnerable.

Counterfactual Probability Expressions.

Idle or projectile spell → Levitation ⇒ Opponent stance changes

$$P\Big(OS_{PA=\text{levitation}} = \text{levitating} \,\Big|\, PA \neq \text{levitation},\, OS = \text{grounded}\Big) = 1.$$

Projectile spell blocked, not hurt → Levitation ⇒ Block disabled (opponent disarmed)

$$P\Big(B_{OS=\text{levitating}} = \text{True} \,\Big|\, PA = \text{magic projectile attack},\, OS = \text{grounded},\, B = \text{True}\Big) = 0.$$

C.3 Specifications of Generated Video Clips

Generated video clips have following technical specifications:

C.3.1 Resolution & Format

All video clips are rendered at a resolution of 512×512 pixels in a square aspect ratio. Videos are encoded in MP4 format using the H.264 codec with yuv420p pixel format to ensure broad compatibility across video players and analysis frameworks.

(a) Factual: The wizard casts a magic projectile spell but the warrior blocks it.

(b) Counterfactual: The wizard casts a levitation spell to lift the warrior off the ground, disarming them.

Figure C.9: Example counterfactual contrast statement for the opponent levitation spell mechanic.

C.3.2 Frame Rate, Duration & Timing

Videos are captured at 50 frames per second (FPS) using a fixed-interval delta time method to ensure consistent temporal sampling across all generated clips. This approach decouples game simulation time from wall-clock time, enabling reproducible frame timing essential for dataset generation and comparative analysis.

Video clip duration is variable and event-driven, spanning the complete battle sequence from initialisation to completion. Clip length is determined by the termination of both player behaviour trees, typically ranging from several seconds to longer sequences depending on the complexity of actions performed (e.g., melee attacks, spell casting, projectile interactions, defensive maneuvers).

C.3.3 Dataset Organisation

The system generates consistent contrasts, enabling direct comparison between observed battle outcomes and alternative scenarios under modified conditions. Each video file follows the naming convention `battle_XXXXXX-TIMESTAMP.mp4`, where `XXXXXX` represents a zero-padded 6-digit battle identifier. Accompanying JSON metadata files provide technical details including encoding parameters, frame timing information, and battle configuration data.

C.3.4 Implementation

Video generation utilizes the imageio library with PyAV backend for efficient encoding. The rendering pipeline captures frame sequences from the game's screenshot buffer system, which also maintains temporal consistency in impact frames through the behaviour tree execution framework.

C.3.5 Variables generated

In addition to clips, each generated example includes:

- Controller inputs C_t
- Full or partial state X_t
- Mechanic-specific CL-DAG $G^{\mathcal{M}}$
- Set of mechanic-specific parallel world DAGs
- Set of mechanic-specific counterfactual DAGs
- Seed / ω identifiers

C.4 Suggested Metrics for Evaluating Performance

For future work on benchmarking SOTA models on our dataset, we propose tasks that can be used to evaluate mechanic learning with Multiverse Mechanica.

Table C.1 summarizes the four tasks for evaluating mechanic-learning with *Multiverse Mechanica*. We expand here on how each metric is operationalized and its relation to prior work.

C.4.1 Mechanic Inference.

The fully- and partially-observed inference tasks require estimating the distributions $\hat{P}_i$ implied by mechanic-specific constraints $\mathcal{M}$ and comparing them against the ground truth distributions P_i. We adopt KL-divergence as the primary quantitative measure, explicitly reporting $D_{\mathrm{KL}}(P_i \,\|\, \hat{P}_i)$ for each mechanic. This aligns with classical practice in distributional evaluation, where divergence to the ground truth provides a direct measure of whether the model has captured the stochastic relations induced by the mechanic.

Task	Description	Metric
Fully observed mechanic inference	Infer $\mathbf{P_i}$ using clips, game state variables, and controller inputs	$D_{\mathrm{KL}}\big(P_i \,\|\, \hat{P}_i\big)$
Partial (canonical) mechanic inference	Infer $\mathbf{P_i}$ using clips and controller inputs, assuming game state is unobserved	$D_{\mathrm{KL}}\big(P_i \,\|\, \hat{P}_i\big)$
Generate consistent contrasts	Generate multiple consistent contrast examples for each d_i associated with the mechanic, and validate that they are visually accurate and consistent	Human or VLM validation
Counterfactual abduction	Generate $\{v_{X=x}, v_{X=x'}\}$ from game, obtain $\hat{v}_{X=x'} = \mathbb{E}[V_{X=x'} \mid V_{X=x} = v_{X=x}]$ using the model, then compare $\hat{v}_{X=x'}$ to $v_{X=x'}$	Human or VLM validation

Table C.1: Tasks that can be used to evaluate mechanic learning with Multiverse Mechanica.

C.4.2 Consistency in Contrast Generation.

The contrast-generation and counterfactual-abduction tasks require evaluating whether pairs of clips $(v_{X=x}, v_{X=x'})$ are *consistent*, i.e. differing only in outcomes attributable to the intervention variable X while holding non-descendant factors fixed. Prior work has typically relied on human evaluation: for example, Gingerson, Amershi et al. collected large-scale human judgments of whether generated gameplay videos adhered to intended mechanics. Recent work has investigated whether VLMs can serve as automated evaluators of model generations. Hendriksen et al. show that pre-trained VLMs can be adapted for evaluating world model rollouts, e.g. scoring whether predicted videos match textual descriptions of target outcomes. However, to our knowledge, no prior work has specifically used VLMs to evaluate *consistency* across parallel-world contrasts, as defined by causal counterfactual principles. This remains an open direction, and *Multiverse Mechanica* provides level-3 parallel-world data where such evaluations can be systematically explored.

Work by Monteiro et al. introduced *composition* metrics that effectively evaluate consistency in image-based counterfactuals by checking whether attributes remain unchanged under controlled edits. These metrics capture aspects of counterfactual faithfulness that parallel our notion of consistency. However, they have not yet been extended to video data, nor applied to generative modelling of game mechanics. We view *Multiverse Mechanica* as a platform to develop such extensions, allowing both human and VLM-based evaluation methods to be compared side-by-side.

C.4.3 Summary.

In short, our metrics combine classical distributional divergences for inference tasks with human- or VLM-based consistency checks for contrastive generation tasks. This dual approach ensures that models are evaluated not only on reproducing distributions of state variables but also on capturing the causal consistency of gameplay dynamics under controlled interventions.

C.5 Theoretical & Implementation Details for Proof-of-Concept

C.5.1 Background on Diffusion Models and Reverse-Sampling

Denoising diffusion probabilistic models (DDPMs) define a forward Markov chain that gradually adds Gaussian noise to an image and a learned reverse process that denoises it step by step. Given a data sample $x_0 \sim q(x_0)$, the forward process produces x_t by directly adding noise to the data: $q(x_t|x_{t-1}) = \mathcal{N}\Big(x_t; \sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I\Big)$ for $t = 1, \ldots, T$. Here $0 < \alpha_t < 1$ are predefined variances. In the reverse generation, one starts from pure noise $x_T \sim \mathcal{N}(0, I)$ and iteratively denoises to x_0 using a model ϵ_θ trained to predict the injected noise. At each step, the model predicts $\hat{\epsilon} \approx \epsilon$ such that x_{t-1} can be estimated by removing noise: e.g., $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\Big(x_t - (1-\alpha_t)\hat{\epsilon}\Big)$ (with additional variance for stochastic sampling). In practice, one can also use the continuous-time formulation and solve a reverse stochastic differential equation or its deterministic counterpart (as in DDIM), yielding a mapping from an initial noise u directly to an image $x = G(u, c)$. Importantly, this exogenous noise u acts as the stochastic latent that accounts for random variation in generated images. Using a deterministic sampler (e.g.,setting $\eta = 0$ in DDIM), one obtains a one-to-one mapping between u and the output x, and can invert a given image to its corresponding u for a particular conditioning c.

We will leverage this invertibility to extract the latent noise u_a from the original image x and the latent noise u_b from the counterfactual image x_{cf}, where x denotes the factual image generated under the original conditioning, x_{cf} represents the counterfactual image generated under modified conditioning, u_a is the inverted latent noise corresponding to the factual image x, and u_b is the inverted latent noise corresponding to the counterfactual image x_{cf}.

C.5.2 Background on Causal Counterfactuals in Image Generation

In causal terms, we can view the generative model as a structural causal model $x := f_\theta(u, c)$, where c is a cause (e.g.,textual description or a set of discrete random variables) and u is an unobserved exogenous variable accounting for randomness. A counterfactual image aims to answer: 'What would the image look like if we change c from c_A to c_B, while keeping all other latent factors the same?' The classical procedure for generating such counterfactuals is the three-step abduction-action-prediction: (1) Abduction: infer the exogenous noise u_a that produced the factual image x under c_A (this u_a captures the instance-specific variations of x); (2) Action: intervene by setting the prompt to c_B (while keeping u_a fixed); (3) Prediction: generate the new image as $x_{\text{cf}} = f_\theta(u_a, c_B)$. This procedure, if the model perfectly captures the true causal mechanism, would change only the aspects of the image directly affected by c and leave other details intact (satisfying causal consistency that no non-descendant features of c change). In practice, directly using u_a with a new prompt c_B can produce a reasonable edited image, but it may fail or produce artefacts if c_B demands alterations that conflict with the original latent factors.

By contrast, many non-SCM image editing approaches do not explicitly enforce the same latent noise. For example, one might simply prompt the model with c_B and generate a new sample (different u), or apply heuristics like latent interpolation, attention refocusing, or mask-based noising of only certain regions. These approaches can produce plausible results, but often lack guarantees that only the intended changes occur — the model might inadvertently change unrelated details because the random draw u or other generation conditions differ. Our goal is to incorporate causal principles into the diffusion editing process to maximize counterfactual faithfulness (only c-dependent changes) while still allowing the model flexibility to implement the edit realistically.

C.5.3 Contrastive Training via Alignment Losses

In this section, we provide the necessary background to help understand our method.

C.5.4 Notation

We first define the notation used in our counterfactual editing framework. Let x denote the original (factual) image, and let x_{cf} denote the counterfactual image we aim to generate (the

edited image after an intervention). We model image generation via a diffusion model as $x = G(u, c)$, where u is an initial exogenous noise (drawn from a Gaussian prior, typically $u \sim \mathcal{N}(0, I)$) and c is the conditioning (in our case, a text prompt). We use c_A for the original prompt and c_B for the counterfactual prompt. Using an inversion technique (e.g., reverse ODE or deterministic DDIM inversion), we can obtain u_a as the noise that generates x under c_A, and similarly u_b as the noise corresponding to x_{cf} under c_B. We denote by x_t the (noisy) latent image at diffusion timestep t when evolving toward x (with $x_0 = x$ and $x_T = u_a$ for the forward noising process), and likewise $x_{\text{cf},t}$ for the counterfactual trajectory. The diffusion model's denoiser is denoted $\epsilon_\theta(x_t, c, t)$, which predicts the added noise at step t for latent x_t and conditioning c. For brevity, we write $\epsilon(x_t, c, t)$ when θ is clear from context. Finally, $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_{\text{text}}$, and $\mathcal{L}_{\text{sub}}$ will denote different loss terms introduced below.

C.5.5 Method 1: L_1 — Consistency Alignment

Our first new loss function enforces consistency alignment between the factual and counterfactual generations. We obtain the noise u_a and u_b corresponding to x and x_{cf} respectively (via the inversion process described above). The consistency alignment loss is then defined as,

$$\mathcal{L}_1 = \|u_a - u_b\|_2^2, \quad u_* = H_\theta^{T \leftarrow 0}(x_*, c_*) \tag{C.5}$$

where $H_\theta^{T \leftarrow 0}$ represents the inversion function that maps from image space back to noise space at timestep T.

Given the diffusion model $x = f_\theta(u, c)$, we have:

$$x = f_\theta(u_a, c_A) \tag{C.6}$$

$$x_{\text{cf}} = f_\theta(u_b, c_B) \tag{C.7}$$

The inversion takes the image back to the noise, which yields:

$$u_a = H_\theta^{T \leftarrow 0}(x, c_A) \tag{C.8}$$

$$u_b = H_\theta^{T \leftarrow 0}(x_{\text{cf}}, c_B) \tag{C.9}$$

The $\mathcal{L}_1$ is added to the training loss as a regularization term to enforce ***exogenous invariance***. In the ideal case where $u_b = u_a$, the counterfactual generation becomes:

$$x_{\text{cf}} = f_\theta(u_a, c_B) \tag{C.10}$$

which ensures that all variations between x and x_{cf} are attributed solely to the conditioning change $c_A \to c_B$, while preserving the exogenous factors encoded in u_a.

This loss directly penalizes any differences between the underlying noise vectors of the original and edited image. The motivation is to ensure that x and x_{cf} share the same source of variation, so that as much of the scene's random details as possible remain unchanged. This approach extracts editing-related information from the seed, enabling differences to be more expressed by the conditioning c rather than random variations. The primary change affects the reverse mapping/generator's decomposition of conditions, belonging to 'seed-level' invariance.

Intuitively, if u_a and u_b are identical, the only differences between x_{cf} and x will come from the changed conditioning c_B vs c_A. In the ideal case, $\mathcal{L}_1 = 0$ means we are generating the counterfactual with the exact same 'random seed' as the factual image (the pure SCM counterfactual). This encourages maximal consistency: the background, lighting, style, and other incidental attributes should stay the same unless the new prompt explicitly demands their change. Enforcing $\mathcal{L}_1$ provides strong alignment that leads to stable edits. It prevents the edited image from drifting in appearance or composition: the counterfactual will tend to have the same objects and layout as the original, only differing in the aspects dictated by the prompt change. This is beneficial for preserving identity (e.g.,the same person's face before and after an edit) and ensuring only the intended attributes change.

However, this strict constraint can also have a few limitations. If the counterfactual prompt c_B is significantly different from c_A, using exactly the same noise u_a might overly constrain the generation, resulting in artefacts or an incomplete edit. The model might struggle to reconcile u_a (which was optimal for the original content) with the new prompt, leading to implausible images or failure to fully achieve the desired change.

C.5.6 Method 2: L_2 — Structure Preservation at High-Noise

As a more relaxed alternative, we propose to align the diffusion model's behaviour for the two images at the high-noise stages of generation, rather than forcing the initial noises to be identical. Concretely, let S be a set of diffusion time steps focusing on the high-noise region (e.g.,the latter half or last third of the diffusion schedule, when x_t is still highly noisy). We define the structure preservation loss $\mathcal{L}_2$ as:

$$\mathcal{L}_2 = \sum_{t \in S} \|\epsilon_\theta(x_t, c_A, t) - \epsilon_\theta(x_{\text{cf},t}, c_B, t)\|_2. \tag{C.11}$$

$\mathcal{L}_2$ measures the disparity between the model's denoising predictions for the factual versus counterfactual image trajectories, but only at very noisy states (where x_t is mostly noise). By penalizing this difference, we encourage the denoiser's reaction to the two inputs to be the same in the early stages of generation. This effectively steers $x_{\text{cf},t}$ to evolve in a similar direction as x_t while the image is still coarse and noisy, ensuring the two generation processes start out aligned in terms of global structure. Importantly, L_2 does not enforce that the latent noises x_t themselves are exactly equal, only that the predicted noise residuals (or equivalently, the score vectors) are similar. This distinction makes L_2 a partial relaxation of the L_1 constraint. It nudges the counterfactual to have a similar high-level appearance without locking in all the exact stochastic details.

When using L_2, the model is free to adjust u_b as needed, but it will still preserve large-scale aspects of u_a. For example, if x depicts a particular scene layout, L_2 will bias x_{cf} to keep that layout, since early denoising steps (which shape the overall composition) will be similar for both. As t gets smaller (less noise), x_{cf} can gradually diverge more to realise the new content c_B specifies. This approach maintains structure and identity better than an unconstrained edit, while granting more flexibility than L_1 for the model to incorporate the new prompt. In essence, L_2 focuses on aligning the coarse-grained features (which are determined in high-noise stages) and lets the fine details emerge freely.

One might consider applying a spatial mask so that structure preservation is enforced only on certain regions (for instance, only aligning the background areas that should remain unchanged). In our approach, we generally did not require an explicit mask for L_2. Since L_2 operates on high-noise (low-detail) states, it inherently affects global structure more than specific fine features. We found that a well-balanced L_2 encourages overall consistency without needing per-pixel restrictions — the model naturally preserves unedited parts of the image. However, if a particular application demands strict locality (e.g.,editing only a small region while leaving everything else exactly as is), a mask could be introduced to further ensure no influence of L_2 on the region to be changed (or conversely, to focus L_2 only on the region to preserve). In summary, L_2 already provides a soft, global consistency constraint, and masking is an optional refinement rather than a necessity in most cases.

C.5.7 Abduction-Action-Prediction and Its Diffusion Emulation

C.5.7.1 SCM Setup.

Let $\mathcal{M} = (\mathbf{U}, \mathbf{V}, \mathbf{F}, P(\mathbf{U}))$ be a structural causal model with exogenous variables $\mathbf{U}$, endogenous variables $\mathbf{V}$, structural assignments $\mathbf{F}$, and exogenous distribution $P(\mathbf{U})$ [32]. We single out: (i) the *mechanic variables* $X \subseteq \mathbf{V}$ that we will intervene on; (ii) the *controller input* C; and (iii) the visual variable V whose realisation is a impact frame snapshot v. Throughout, we explicitly treat our generation seed ω as a realisation of the SCM exogenous variables, i.e., $\omega \sim P(\mathbf{U})$, and we treat all other shared rendering conditions as part of ω (while C is held fixed explicitly).

C.5.7.2 Abduction-Action-Prediction (AAP).

Given a factual observation v_0 obtained under $(X{=}x_0, C{=}c)$, the AAP recipe for the two-world case proceeds as:

(Abduction) Infer $P(\mathbf{U} \mid V{=}v_0, X{=}x_0, C{=}c)$, choose a representative $\hat{\omega}$.

(Action) Form the intervened model $\mathcal{M}_{X=x_1}$ while keeping $C{=}c$ and $\mathbf{U}{=}\hat{\omega}$ fixed.

(Prediction) Evaluate the counterfactual $V_{X=x_1}(\hat{\omega})$.

Equivalently, in distributional form,

$$\hat{v}_1 \;\sim\; P\Big(V_1 \,\Big|\, V_0{=}v_0,\, X_0{=}x_0,\, C{=}c\Big),$$

which is shorthand for propagating $\hat{\omega} \sim P(\mathbf{U} \mid V_0{=}v_0, X_0{=}x_0, C{=}c)$ through $\mathcal{M}_{X=x_1}$.

C.5.8 Diffusion-Based Emulation of AAP

In our latent diffusion setting, we emulate abduction-action-prediction (AAP) by identifying the SCM exogenous variables with the model's initial latent noise:

$$\mathbf{U} \;\longleftrightarrow\; \omega \sim \mathcal{N}(0, I).$$

Abduction (DDIM inversion). We estimate $\hat{\omega}$ from a factual impact frame $V_{X=x_0}$ under (x_0, c) via deterministic sampler inversion (DDIM, $\eta{=}0$). Let $Z_{t,X=x_0}$ denote its noisy latents across $t \in [0, T]$. At each step, the denoiser $\epsilon_\theta(Z_{t,X=x_0}, t, c_0)$ predicts the noise, and inversion equations (Appendix C.5.10.5) are used to propagate forward in the schedule, yielding an estimate $\hat{\omega} \;=\; \mathsf{Abduct}_\theta(Z_{0,X=x_0}, c_0)$.

Action. Hold c fixed and change the mechanic state from x_0 to x_1.

Prediction (deterministic reverse). Initialise the trajectory with $Z_{T,X=x_1} := \hat{\omega}$ and run the deterministic reverse process conditioned on (x_1, c) to obtain a counterfactual latent $Z_{0,X=x_1}$, then decode it to the counterfactual impact frame $\hat{V}_{X=x_1}$.

Algorithm C.1 Diffusion-based AAP via DDIM (η=0)

Require: Diffusion model (ϵ_θ, scheduler, decoder), factual $(V_{X=x_0}, x_0, c)$, target x_1

1: **Abduction:** $\hat{\omega} \leftarrow \mathsf{Abduct}_\theta(Z_{0,X=x_0}, c_0)$ // DDIM inversion in latent space

2: **Action:** Keep c fixed, set $X := x_1$

3: **Prediction:** $Z_{T,X=x_1} := \hat{\omega}$; for $t = T, \ldots, 1$: $Z_{t-1,X=x_1} \leftarrow \mathrm{DDIMStep}\big(Z_{t,X=x_1}, t, x_1, c; \epsilon_\theta\big)$; $\hat{V}_{X=x_1} \leftarrow \mathrm{decoder}(Z_{0,X=x_1})$

4: **return** $\hat{V}_{X=x_1}$

C.5.8.1 Connection to Our Training Losses.

The AAP framing clarifies the roles of our loss components. (i) *Exogenous alignment* $\mathcal{L}_1$ encourages shared-seed invariance by driving

$$\mathsf{Abduct}_\theta(Z_{0,X=x_0}, c_0) \;\approx\; \mathsf{Abduct}_\theta(Z_{0,X=x_1}, c_1),$$

for elements of the same contrast that share the true ω. This ensures the abduction step produces consistent seeds across worlds. (ii) *Structure preservation* $\mathcal{L}_2$ aligns denoiser outputs $\epsilon_\theta(Z_{t,X=x_j}, t, c_j)$ at high-noise steps $t \in S$, enforcing agreement on coarse global structure during the early reverse process. This mirrors AAP's assumption that exogenous factors (ω) are held fixed while only X changes.

C.5.8.2 SCM $\leftrightarrow$ Diffusion mapping (two-world case).

SCM concept		Diffusion instantiation
Exogenous $\mathbf{U}$	$\leftrightarrow$	Initial latent noise seed $\omega \sim \mathcal{N}(0, I)$
Abduction $P(\mathbf{U} \mid V_{X=x_0}, x_0, c)$	$\leftrightarrow$	DDIM (η=0) latent inversion $\hat{\omega} = \mathsf{Abduct}_\theta(Z_{0,X=x_0}, c_0)$
Action $X = x_1$	$\leftrightarrow$	Condition reverse process on (x_1, c)
Prediction $V_{X=x_1}(\hat{\omega})$	$\leftrightarrow$	Deterministic reverse to $Z_{0,X=x_1}$ then decode $\hat{V}_{X=x_1}$
Causal consistency	$\leftrightarrow$	Shared seed $\hat{\omega}$; early-step structure preservation ($\mathcal{L}_2$)

Abduction via DDIM inversion yields an *estimate* $\hat{\omega}$ whose fidelity depends on the schedule and conditioning; see Appendix C.5.10.6 for caveats and tuning guidance.

C.5.9 Additional Regularizers

Beyond the core losses L_1 and L_2, our framework can incorporate additional terms to improve consistency and fidelity:

C.5.9.1 Subspace Consistency Loss.

We can encourage the factual and counterfactual images to remain close in certain intermediate representations of the diffusion model. For example, one may align hidden latents or cross-attention maps at corresponding diffusion steps. By penalizing differences in these subspaces (e.g.,the model's multi-head attention maps for background tokens, or feature maps in a particular UNet layer), we enforce that the internal generation pathways for x and x_{cf} stay similar. This helps preserve layout and identity at a semantic level, complementing the pixel-space alignment enforced by L_1/L_2. Formally, if $F_t(x)$ denotes some feature (such as a latent embedding or attention tensor) computed during the denoising of x at step t, we can define a loss $L_{\text{sub}} = \sum_{t\in\mathcal{T}} \|F_t(x) - F_t(x_{\text{cf}})\|_2$ for some chosen set of layers or timesteps $\mathcal{T}$. This subspace consistency loss encourages the edited image to differ only minimally in features unrelated to the intervention.

Text Consistency Loss. To ensure the edited image indeed reflects the counterfactual prompt c_B, we include a text-image consistency term. We rely on a pre-trained image-text similarity model (such as CLIP) to measure alignment between x_{cf} and the description c_B. Let $\text{sim}(x_{\text{cf}}, c_B)$ be a similarity score (higher means the image matches the prompt better). We define a loss $L_{\text{text}} = -\text{sim}(x_{\text{cf}}, c_B)$ (or equivalently $1 - \text{sim}$, depending on the normalization) so that minimizing L_{text} maximizes the agreement between the generated image and the desired attributes. This ensures that while preserving content, we do not under-shoot the edit: the new image should clearly exhibit the prompted change. The text consistency loss guides the generation to remain faithful to the user's request, especially when L_1 or L_2 are pulling towards the original image. It helps avoid the outcome where the edit is so conservative that the difference between x_{cf} and x is hard to discern.

C.5.10 Loss Combination

Our full counterfactual editing objective combines these components in a weighted sum:

$$L_{\text{total}} = \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_{\text{text}} + \lambda_4 L_{\text{sub}}, \tag{C.12}$$

where λ_i are tunable weights that control the influence of each loss term. In practice, we choose these weights to balance identity preservation against effective editing. Typical values and trade-offs are as follows:

λ_1 **(Consistency Alignment):** This is often kept relatively small (e.g., λ_1 in the range 0 to 0.5) unless the edit is very minor. A small λ_1 nudges the initial noise vectors closer without forcing identity completely. Increasing λ_1 leads to more literal counterfactuals (very high consistency with the original image's details), but if set too high it may prevent the new attributes from appearing strongly. There is a trade-off between maintaining background/identity (higher λ_1 favors this) and allowing change (lower λ_1 gives more freedom).

λ_2 **(Structure Preservation):** We usually give L_2 a moderate to high weight (on the order of 1.0) as it is the principal mechanism to preserve structure. λ_2 in a range roughly 0.5 to 2.0 works well. A larger λ_2 tightly constrains the high-level layout and style to match the original, which is good for identity preservation; however, if λ_2 is excessively large, it can act almost as strictly as L_1, potentially impeding necessary changes. Reducing λ_2 allows the counterfactual generation to deviate more in composition if needed, but too low λ_2 might result in unwanted alterations in background or other objects.

λ_3 **(Text Consistency):** This weight should be high enough to ensure the edit actually happens (especially for subtle changes), but not so high that it overrides the preservation losses. In practice λ_3 is often set around 0.5 to 1.0 (assuming similarity is scaled to a comparable range) so that the image aligns with the prompt without artefacts. If λ_3 is set too low, the edit might be too conservative (the model might simply regenerate the original image to satisfy L_1/L_2). If λ_3 is too high, the model may introduce exaggerated or incorrect features to satisfy the prompt, possibly compromising identity or visual quality.

λ_4 **(Subspace Consistency):** If used, this is typically a small auxiliary weight (e.g.,0.1). Since L_{sub} operates on internal features, it can strongly bind the generation if overweighted. A modest λ_4 helps reinforce structural consistency without conflicting with the primary losses. Tuning λ_4 involves checking that it indeed improves preservation of details like face identity or

scene layout, without, for example, freezing the image in an early-state that ignores the new prompt. In some cases, we might set $\lambda_4 = 0$ (i.e., not use this term) if we find L_1 and L_2 are sufficient; when used, it serves as an extra regulariser.

In summary, L_1 and L_2 provide a spectrum between strict and loose alignment, λ_3 drives the fidelity to the requested counterfactual change, and λ_4 can bolster consistency on a feature level. We recommend starting with a balanced combination (for instance, $\lambda_1 = 0.2, \lambda_2 = 1.0, \lambda_3 = 0.5, \lambda_4 = 0.1$).

(a) Factual: A 1-on-1 battle. Foe 1 type is archer, element is fire, weapon is bow, weapon element is none, has shield is false, action is shoot. Foe 2 type is warrior, element is fire, weapon is long sword, weapon element is none, has shield is false, action is idle. Foe turn is foe 1.

(b) Counterfactual: A 1-on-1 battle. Foe 1 type is warrior, element is fire, weapon is long sword, weapon element is none, has shield is false, action is idle. Foe 2 type is warrior, element is fire, weapon is long sword, weapon element is none, has shield is false, action is idle. Foe turn is foe 1.

Figure C.10: Example counterfactual image tuple generated from fine-tuned text-to-image diffusion model.

C.5.10.1 Diffusion Backbone.

We use a pre-trained Stable Diffusion variant with a deterministic sampler [1], which makes the mapping between exogenous noise u and image v approximately invertible. This enables abduction of u from an image and consistent reuse across parallel worlds. Classifier-free guidance is applied with scale 7.5, and the scheduler uses $T = 50$ steps.

[1]Specifically we use DDIM $\eta = 0$, also the deterministic samplers in [182]

C.5.10.2 Conditioning.

In the main text we describe conditioning directly on game state variables (e.g., shield, weapon type, block outcome). For implementation, these variables and outcomes are converted into natural-language captions for compatibility with CLIP text encoders. This is a nuisance parameterisation: the underlying conditioning remains the game state variables.

C.5.10.3 Training Setup.

We fine-tune the network backbone (UNet) only, keeping the VAE and text encoder frozen. Batch size is 4, training runs for 50 epochs, with a cosine learning-rate schedule, weight decay 0.01, and gradient clipping. Images are peak-action snapshots extracted at canonical times in each episode to minimize temporal ambiguity.

C.5.10.4 Alignment Losses.

We mainly use two loss functions for fine-tuning:

- $\mathcal{L}_1$: consistency alignment. After inverting both factual and counterfactual images to latent noise (u_a, u_b), we penalize $\|u_a - u_b\|^2$, enforcing invariance of exogenous factors. Note that this loss function is very expensive due to it needs to inverse two sampling passes for each counterfactual data pair. In practice, we only apply $\mathcal{L}_1$ to one data point per batch of training data.

- $\mathcal{L}_2$: structure preservation at high noisy region (large SNRs). At early diffusion timesteps, we penalize discrepancies between denoiser predictions for factual vs. counterfactual trajectories, encouraging global structural consistency.

Both terms can be weighted with coefficients λ_1, λ_2.

C.5.10.5 Seed Abduction with Deterministic Diffusion.

Our multiverse alignment objective requires that consistent contrasts share the same exogenous noise ω. In deterministic samplers (e.g., DDIM), this means that two parallel reverse processes $(Z_{t,X=x_0})_{t=0}^T$ and $(Z_{t,X=x_1})_{t=0}^T$ can be initialised with the same ω, ensuring non-descendant content is consistent.

Given an observed impact frame $V_{X=x}$ with controller input c, let $Z_{0,X=x}$ denote its clean latent before decoding and $Z_{t,X=x}$ the noisy latent at step t. At each step, the denoiser $\epsilon_\theta(\cdot)$

predicts noise $\epsilon_\theta(Z_{t,X=x}, t, c)$, which is then used to trace the trajectory backward through the noise schedule. Iterating these updates recovers an estimate $\hat{\omega} = \mathsf{Abduct}_\theta(V_{X=x}, c)$.

Applying this inversion procedure to both members of a contrast yields $\hat{\omega}_{x_0}$ and $\hat{\omega}_{x_1}$, which ideally coincide. The seed-consistency loss $\mathcal{L}_1$ penalizes their distance, providing a concrete operationalisation of the causal consistency principle within deterministic diffusion.

C.5.10.6 Caveats.

Deterministic inversion. We use DDIM with $\eta = 0$ for Abduct_θ, yielding an approximately bijective mapping between seed and latent trajectory under fixed conditioning and schedule. In practice, invertibility is approximate and sensitive to: (i) the precise noise schedule; (ii) classifier-free guidance settings; and (iii) conditioning (x, c). Hence $\hat{\omega}$ should be treated as a consistent *estimate* rather than a ground-truth latent.

C.5.10.7 Choosing the High-Noise Set S for $\mathcal{L}_2$.

We select S as either (i) the last k steps of the schedule (empirically $k \in [\frac{T}{3}, \frac{T}{2}]$), or (ii) all t with $\mathrm{SNR}(t) \leq \tau$ for a threshold τ. A weighted variant uses $w_t \propto \mathrm{SNR}(t)^{-\gamma}$ and

$$\mathcal{L}_2^{\mathrm{w}} = \sum_t w_t \left\| \epsilon_\theta(z_{0,t}, t, x_0, c) - \epsilon_\theta(z_{1,t}, t, x_1, c) \right\|_2^2.$$

Under mild conditions, aligning score predictions at high-noise is connected to alignment in data space via Stein's identity.

C.5.10.8 Scope.

This is a feasibility study. We claim no pixel-level counterfactual identification and provide only qualitative illustrations. Future work may extend this approach to video sequences and more complex mechanics.

C.6 Software Dependencies

C.6.1 Game Engine.

The game itself is implemented in `Pygame`, a lightweight Python library for 2D graphics and interaction. We chose Pygame because it enables rapid prototyping of turn-based combat mechanics, frame-accurate rendering of impact frames, and reproducible control of random seeds, all within a Python environment that integrates smoothly with machine learning workflows.

C.6.2 Causal Modelling.

To formalize and simulate the causal generative process underlying gameplay, we use the `Pyro` probabilistic programming library [183]. Pyro provides the primitives required to implement SCMs consistent with the game's causal DAG, including stochastic functions for exogenous variables, deterministic assignments for endogenous variables, and intervention operators. This allows us to align the game engine's execution trace with an explicit causal model, and to sample parallel-world contrasts in a principled manner.

C.6.3 Reproducibility.

Both components are integrated in a unified Python codebase, ensuring that gameplay, causal modelling, and data generation can be run deterministically from a single seed.

C.6.4 Graph Libraries.

We generate mDAGs, parallel-world graphs and counterfactual graphs using the `YO` library [184].

C.6.5 Graph Serialisation.

All graphs are serialised as directed graphs in JSON using a node-link format. Nodes are represented as JSON objects with keys for the node identifier and attributes, while edges are represented as objects with source and target identifiers and an edge type field. In the case of mDAGs and parallel world graphs, exogenous nodes are marked with the attribute `"exogenous": true`. For example, the above example of an mDAG with observed nodes $\{A, B, C\}$, one directed edge $A \to B$, and one hyper-edge $\{B, C\}$, is converted to a CL-DAG and serialised as follows:

```
{
  "nodes": [
    {"id": "A"},
    {"id": "B"},
    {"id": "C"},
    {"id": "N_{B,C}", "exogenous": true}
  ],
  "links": [
```

```
    {"source": "A", "target": "B", "type": "directed"},
    {"source": "N_{B,C}", "target": "B", "type": "directed"},
    {"source": "N_{B,C}", "target": "C", "type": "directed"}
  ]
}
```