Doctoral Thesis

Object Empowerment: An Information-Theoretic Approach to Intrinsically Motivated Reinforcement Learning of Tool Use

by:

Faizan Rasheed

School of Physics, Engineering, and Computer Science

Submitted to the University of Hertfordshire in partial fulfilment of the requirement of the degree of Doctor of Philosophy

Supervisors:

Dr. Nicola Catenacci Volpi

Prof. Daniel Polani

April, 2026

Abstract

Artificial intelligence aspires to build agents that act autonomously, adapt to novel situations, and discover meaningful strategies without external instruction. A crucial challenge in this regard is to enable agents with intrinsic drives that promote purposeful behaviour even in the absence of explicit goals or dense rewards. Within cognitive science and reinforcement learning (RL), such *intrinsic motivations* have been linked to the agent's ability to influence its environment. Among these, *empowerment* stands out as a prominent form of intrinsic motivation. It is defined as the maximum mutual information between an agent's actions and resulting states. It formalises this notion by quantifying how much control an agent possesses over its future. It provides a principled, information-theoretic foundation for autonomous exploration and self-organised behaviour.

Building on this foundation, the present thesis advances empowerment from a general but unspecific model of control to a computational framework specifically designed for *tool use*. It argues that tool use, which is a hallmark of intelligent behaviour in both biological and artificial systems, can be understood as the process of maximising influence over task-relevant objects through intermediary entities (tools). To capture this formally, the thesis introduces *object empowerment*, a novel formulation that conditions the empowerment channel on manipulable objects, thereby isolating the agent's causal influence on specific environmental entities.

The framework is then extended to *learning tool–object interactions* by integrating object empowerment into RL as an intrinsic reward regulariser. This allows agents to autonomously discover functional dependencies between tools and objects, even under sparse-reward conditions. Subsequent chapters generalise the approach to environments with multiple tools and objects. This approach defines a *multi-object empowerment* model and a corresponding *tool–object empowerment matrix* that supports systematic tool comparison and selection. Finally, the thesis advances from tool selection to *tool characterisation* by introducing three empowerment-based measures: *persistence*, *latency*, and *reliability*. These measures quantify how long a tool remains effective, how quickly its effects manifest, and how robustly it performs under uncertainty, respectively.

Empirical validation across custom grid-world and MiniHack environments demonstrates that agents trained with object empowerment regularisation converge faster, explore more efficiently, and exhibit interpretable tool-use behaviours compared to standard RL baselines. These experiments reveal that empowerment not only facilitates exploration but also provides a transparent, quantitative account of causal structure in tool-mediated interaction.

Collectively, the thesis establishes object empowerment as a unifying principle for modelling and generating tool-use behaviour. By integrating information-theoretic control with RL, it bridges intrinsic motivation, causal reasoning, and autonomous skill acquisition. Thus, object empowerment offers more than an intrinsic drive: it constitutes a language for constructing agents that act not merely to explore, but to *understand and shape* their own possibilities for influence.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to Almighty ALLAH for His countless blessings and guidance throughout my academic journey. Without His grace and mercy, this achievement would not have been possible.

I extend my sincere thanks to my supervisor, Dr. Nicola Catenacci Volpi, for his continuous support, insightful feedback, and detail oriented supervision from the very first day of my PhD. I am also profoundly grateful to my co-supervisor, Prof. Daniel Polani, for his invaluable guidance, stimulating discussions, and generous support, particularly for facilitating my participation in international conferences to disseminate my research. It has been an honour to closely work with him, especially as he is the pioneer of Empowerment, the concept on which this thesis is based.

I am deeply thankful to the University of Hertfordshire for awarding me a fully funded scholarship, which made this research possible. My sincere appreciation goes to all my colleagues in the Adaptive Systems Research Group for their collaboration and encouragement, and to my friend Dr. Mubashir Ahmad from the Robotics Research Group, for his constant support, help, and guidance throughout this journey.

I owe my deepest gratitude to my mother Ismat Rasheed and my late father Rasheed Ahmed, whose dreams, love, and unwavering support have been the foundation of all my achievements. This milestone is truly dedicated to them.

My heartfelt thanks go to my beloved wife, Dr. Mahnoor Fatima, for her endless patience, encouragement, and support. She joined me in the UK after our marriage in the middle of my PhD and stood by me through every challenge with love and understanding.

I would also like to acknowledge my uncle, Barrister Mansoor ul Haq Ansari, whose support and inspiration during difficult times have meant more than words can express.

Finally, I extend my heartfelt gratitude to all my family members, including my brother, and my dear friends, for their constant support, companionship, and kindness. To everyone who has contributed, directly or indirectly, to my academic and personal growth during this journey, your encouragement and kindness have been invaluable, and I am truly grateful.

Author's Declaration

The thesis submitted for this research degree at the University of Hertfordshire contains no material as part of any other academic award at the same or another institute. Except where due reference is made in the text, the thesis represents the candidate's own research. All works presented in this dissertation are in accordance with the university's regulation and code of practice per the research program handbook.

This research was carried out in collaboration with the Adaptive Systems Research Group, School of Physics, Engineering, and Computer Science (SPECS), University of Hertfordshire, UK.

The research conducted during this PhD was actively disseminated through presentations at scientific conferences, seminars, and workshops, and the associated papers were published in their respective proceedings.

Peer-reviewed Work Contained in This Thesis

C1. **Faizan Rasheed**, Daniel Polani, and Nicola Catenacci Volpi, "Leveraging Empowerment to Model Tool Use in Reinforcement Learning," *IEEE 13th International Conference on Development and Learning (ICDL 2023)*, **Best Paper Finalist**.

C2. **Faizan Rasheed**, Daniel Polani, Kenzo Clauw, and Nicola Catenacci Volpi, "Object Empowerment-Driven Tool Selection in Reinforcement Learning," *IEEE 7th International Conference on Cognitive Machine Intelligence (CogMI 2025).*

Workshop Papers

W1. **Faizan Rasheed**, Daniel Polani, Kenzo Clauw, and Nicola Catenacci Volpi, "Object Empowerment-Driven Tool Selection for Exploration in Reinforcement Learning," *CoLLAs 2025 Workshop on Lifelong Learning in Cognitive Science.*

W2. **Faizan Rasheed**, Daniel Polani, Kenzo Clauw, and Nicola Catenacci Volpi, "Object Empowerment-Driven Tool Selection for Exploration in Reinforcement Learning," *18th European Workshop on Reinforcement Learning (EWRL 2025).*

W3. **Faizan Rasheed**, Daniel Polani, Kenzo Clauw, and Nicola Catenacci Volpi, "Object Empowerment as an Intrinsic Drive for Tool Selection in Reinforcement Learning," *7th International Workshop on Intrinsically Motivated Open-ended Learning (IMOL 2025).*

W4. **Faizan Rasheed**, Daniel Polani, and Nicola Catenacci Volpi, "Empowerment for Tool Comparison in Reinforcement Learning: A Preliminary Study," *Proceedings of Abstract, SPECS 2023, University of Hertfordshire, UK.*

Other Peer-reviewed Publications

O1. Abdul Ahad, Zheng Jiangbina, Mohammad Tahir, Ibraheem Shayea, Muhammad Aman Sheikh, and **Faizan Rasheed**, "6G and Intelligent Healthcare: Taxonomy, Technologies, Open Issues and Future Research Directions", *Internet of Things*, vol. 25, pp. 101068, 2024.

O2. **Faizan Rasheed**, Yasir Saleem, Kok-Lim Alvin Yau, Yung-Wey Chong, and Sye Loong Keoh, "The Role of Deep Learning in Parking Space Identification and Prediction Systems", *Computers, Materials & Continua*, vol. 75, pp. 761–784, 2023.

O3. **Faizan Rasheed**, Kok-Lim Alvin Yau, Rafidah Md Noor, and Yung-Wey Chong, "Deep Reinforcement Learning for Addressing Disruptions in Traffic Light Control", *Computers, Materials & Continua*, vol. 71, pp. 2225–2247, 2022.

Signed: **Date:**

Faizan Rasheed 1st November 2025

Note on Publications

The research presented in this thesis has been disseminated through peer-reviewed publications listed in the Author's Declaration. Table 1 summarises how these publications relate to specific chapters and contributions within the thesis.

While these publications report key components of the research, the thesis integrates them into a coherent framework and provides expanded experimental analysis and additional theoretical discussion that are not contained in the individual papers.

Table 1: Mapping between thesis chapters and related publications.

Chapter	**Publication(s)**	**Contribution(s)**
Chapter 4	C1	Introduces the formal definition of object empowerment to capture the agent's influence over manipulable objects.
Chapter 5	C1	Develops the tool-learning framework based on agent–tool–object decomposition and integrates object empowerment as an intrinsic regulariser to guide exploration in sparse-reward environments.
Chapter 6	C2, W1–W3	Introduces the multi-object empowerment formulation and a tool-selection mechanism that evaluates tools based on their empowerment over task-relevant objects.
Chapter 7	C2, W1–W3	Introduces the concept of tool persistence, quantifying how long a tool remains effective in influencing objects.

Contents

Abstract vi

Acknowledgements ix

Declaration x

Note on Publications xii

List of Figures xviii

List of Tables xxvii

1 Introduction 1

1.1 Research Hypothesis and Questions 4

1.2 Contributions of the Thesis 6

1.3 Structure of the Thesis 7

2 Literature Review 10

2.1 Defining Tools and Tool Use 11

2.2 Tool Use in Nature: Ethology and Anthropology 12

2.2.1 Tool Use in Non-Human Animals 12

2.2.2 Tool Use in Human Evolution 14

2.3 Affordance Theory, Tool Use, and Learning 16

2.4 Tool Characterisation and Classification 19

2.4.1 Anthropological and Historical Perspectives on Tool Classification . 19

2.4.2 Toward Computational Models of Tool Classification 20
2.5 Intrinsic Motivations in Reinforcement Learning 21
2.5.1 Curiosity-Driven Exploration 22
2.5.2 State Novelty and Count-Based Exploration 23
2.5.3 Competence-Based Intrinsic Motivation 24
2.5.4 Mutual Information-Based Intrinsic Motivation 26
2.5.5 Empowerment-Based Intrinsic Motivation 28
2.5.6 Intrinsic Motivation Applied to Tool Use 30
2.6 Hierarchical Reinforcement Learning and Tool Abstraction 32

3 Theoretical and Technical Background 33
3.1 Reinforcement Learning 34
3.1.1 Markov Decision Processes 34
3.1.2 Objective and Return 35
3.1.3 Main Families of RL Algorithms 35
3.1.4 Challenges in RL 36
3.1.5 RL Algorithms Used in This Thesis 36
3.1.6 Addressing Sparse Rewards through Intrinsic Motivation 40
3.2 Information-Theoretic Foundations 42
3.2.1 Entropy 42
3.2.2 Conditional Entropy 43
3.2.3 Mutual Information 43
3.2.4 Conditional Mutual Information 44
3.2.5 Channel Capacity and Communication Channels 46
3.3 Empowerment 47
3.3.1 The Action–Perception Loop 47
3.3.2 Computation of Empowerment in Non-Deterministic Environments . 49
3.3.3 Computation of Empowerment in Deterministic Environments . . . 52
3.3.4 Interpretation and Role in Intrinsic Motivation 53

4 Object Empowerment **55**

4.1 Formalism . . . 56

4.1.1 State space . . . 56

4.1.2 Action Space . . . 57

4.1.3 Transition Dynamics . . . 57

4.1.4 Agent's Object Empowerment . . . 58

4.1.5 Tool's Object Empowerment . . . 58

4.1.6 Object Empowerment in Deterministic Environments . . . 59

4.1.7 State-Average Object Empowerment . . . 60

4.2 Agent–Object Interaction Environment . . . 60

4.2.1 Environment Description . . . 61

4.2.2 Movable vs Non-Movable Objects . . . 62

4.3 Agent–Tool–Object Interaction Environment . . . 64

4.3.1 Environment Description . . . 64

4.3.2 Broom Tool: Proximal vs Distant Object Landscapes . . . 66

4.3.3 Picker Tool: Proximal vs Distant Object Landscapes . . . 68

4.4 Agent–Tool–Object Interaction in MiniHack . . . 75

4.4.1 Environment Description . . . 75

4.4.2 Destroyable vs. Indestructible Interaction Scenarios . . . 76

4.5 Summary . . . 81

5 Empowerment-Guided Learning of Tool–Object Interactions **82**

5.1 Tool-Learning Framework . . . 83

5.1.1 Reward Structure . . . 83

5.2 Experiments . . . 85

5.2.1 Experiment 1: Tools Comparison . . . 85

5.2.2 Experiment 2: Comparison of Intrinsic Motivation Mechanisms . . . 93

5.3 Influence of Horizon (h) and Weighting Factor (β) . . . 102

5.3.1 h-step Fully Observable Empowerment . . . 103

5.3.2 h-step Tool's Object Empowerment . . . 112
5.3.3 Behavioural Consequences of FOE and TOE . . . 119
5.4 Summary . . . 127

6 Learning Tool Selection 129
6.1 Tool-Learning Framework . . . 130
6.1.1 State Space . . . 130
6.1.2 Action Space . . . 131
6.1.3 Multi-Object Empowerment . . . 131
6.1.4 Tool Selection Mechanism . . . 131
6.2 Experiments . . . 133
6.2.1 Experiment 1: Tool Selection in a Single-Object Task . . . 133
6.2.2 Experiment 2: Tool Selection in a Multi-Object Task . . . 138
6.3 Summary . . . 144

7 Characterisation of Tools 146
7.1 Characterisation Measures . . . 147
7.1.1 Persistence of Tools . . . 147
7.1.2 Latency of Tools . . . 148
7.1.3 Reliability of Tools . . . 149
7.2 Experiments . . . 150
7.2.1 Experiment 1: Persistence of Tools . . . 151
7.2.2 Experiment 2: Latency of Tools . . . 156
7.2.3 Experiment 3: Reliability of Tools . . . 161
7.3 Summary . . . 169

8 Discussion and Conclusions 171
8.1 Overview . . . 171
8.2 Integration of Findings . . . 173
8.3 Limitations and Future Work . . . 174
8.3.1 Learning Empowerment from Interaction . . . 174

8.3.2 Generalisation Across Tools, Goals, and Environments 174
8.3.3 Comparison with Other Intrinsic Motivations 175
8.3.4 Computational Complexity 175
8.3.5 Extension to Continuous and Robotic Domains 176
8.3.6 Extensions of Tool Characterisation Framework 177
8.4 Conclusions 178

List of Figures

3.1 Schematic of the RL interaction loop between agent and environment. At each time step, the agent is in a state, selects an action, receives a reward, and transitions to a new state. 34

3.2 A2C architecture where the actor outputs the policy $\pi_\theta(a \mid s)$ and the critic outputs the state value $V^\pi(s)$. 37

3.3 Venn diagram representation of the relationship between entropies $H(X)$, $H(Y)$, conditional entropies $H(X|Y)$, $H(Y|X)$, and mutual information $I(X;Y)$. The overlapping region represents the mutual information—the reduction in uncertainty of one variable given the other. 44

3.4 Venn diagram representation of entropy, conditional mutual information, and multivariate mutual information among three random variables X, Y, and Z. Pairwise CMI terms such as $I(X;Y|Z)$ appear in the pairwise overlaps, while the shared region in the center corresponds to the multivariate mutual information $I(X;Y;Z)$. 45

3.5 Causal Bayesian network illustrating the action-perception loop over multiple time steps. Each action A_t influences the next environment state S_{t+1}, which generates observation O_{t+1} for the agent. The loop continues recursively, enabling the agent to iteratively interact with and influence its environment. 48

3.6 Empowerment landscapes in a 10×10 grid world for horizons $h = 1, 2, 5, 10$, based on the setup introduced by Klyubin et al. [1]. As the planning horizon increases, more states become reachable from each position, especially in open areas, resulting in higher empowerment values. Border regions and cells adjacent to walls exhibit lower values due to limited reachability. . . . 54

4.1 Agent–object grid world setup. The agent (red robot) can move freely within the grid boundaries, while the object (black box) can be pushed by the agent through its direct movement actions. 61

4.2 Agent–object empowerment landscapes for a *movable* object placed at the grid center (i.e., $(4, 4)$), computed for varying horizons. Longer horizons allow the agent to affect the object from a larger portion of the grid and in a greater number of possible ways, especially when the agent is closer to the object. 63

4.3 Agent–object empowerment landscape for a *non-movable* object fixed at $(4, 4)$, shown for $h = 1$. The same landscape occurs for any $h \geq 1$. 64

4.4 Two grid-world configurations, each with an agent (robot), an object (can), and a different type of tool (a broom or a picker). 65

4.5 Relative tool movements using $\mathcal{A}^{\mathfrak{T}}$ actions. 65

4.6 Illustrative tool–object interactions. Blue-bordered cells indicate the agent's starting position. In both subfigures, the lower part shows the outcome of a single action, while the upper part shows the outcome of a sequence of actions. With the picker, the agent can move the object without changing its own position, whereas with the broom, the agent must move its body to push the object. 70

4.7 Object empowerment landscapes for the broom tool at $h = 1$ in proximal (broom at $(5, 4)$) and distant (broom at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$. 71

4.8 Object empowerment landscapes for the broom tool at $h = 2$ in proximal (broom at $(5, 4)$) and distant (broom at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$. 71

4.9 Object empowerment landscapes for the broom tool at $h = 5$ in proximal (broom at $(5, 4)$) and distant (broom at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$. 72

4.10 Object empowerment landscapes for the broom tool at $h = 8$ in proximal (broom at $(5, 4)$) and distant (broom at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$. 72

4.11 Object empowerment landscapes for the picker tool at $h = 1$ in proximal (picker at $(5, 4)$) and distant (picker at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$. 73

4.12 Object empowerment landscapes for the picker tool at $h = 2$ in proximal (picker at $(5, 4)$) and distant (picker at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$. 73

4.13 Object empowerment landscapes for the picker tool at $h = 5$ in proximal (picker at $(5, 4)$) and distant (picker at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$. 74

4.14 Object empowerment landscapes for the picker tool at $h = 8$ in proximal (picker at $(5, 4)$) and distant (picker at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$. 74

4.15 MiniHack environment for object empowerment analysis. The agent (bottom-right) can pick up the axe and use it to destroy the tree. 76

4.16 Object empowerment landscapes for a destroyable object (tree) when the agent is equipped with the axe, for horizons $h = 3, 4, 5$. The tree is located at $(3, 3)$. 78

4.17 Object empowerment landscapes for a destroyable object (tree) when the agent is not equipped with the axe, which is initially placed at $(6, 6)$, for horizons $h = 8, 9, 10$. The tree is located at $(3, 3)$. 79

4.18 Empowerment landscape for an indestructible object (tree) across the grid. All values are zero, as no action sequence can affect the object. 80

5.1 Grid-world environment with the agent equipped with a broom. The can represents the movable object and the bin represents the goal position. . . . 86

5.2 Grid-world environment with the agent equipped with a picker. 86

5.3 Average number of episodes until convergence in Experiment 1. Error bars indicate standard deviation across 10 runs. Lower values denote faster convergence (episodes to reach the optimal return). 88

5.4 The average return received by agent in Experiment 1. 89

5.5 1-step OE landscapes with the can next to the picker and the broom in the experiment 1. 90

5.6 1-step FOE landscapes with the can next to the broom and the picker in the experiment 1. 91

5.7 1-step FOE landscapes with the can far away from the broom and the picker in the experiment 1. 92

5.8 Grid-world environment for experiment 2. The picker represents the tool, the can represents the movable object, and the bin represents the goal position. The goal of the task is to move the can onto the cell containing the bin. . . 95

5.9 Average number of episodes required for convergence in Experiment 2. Error bars indicate standard deviation across 10 runs. 96

5.10 The average return obtained by the five agents in Experiment 2. 97

5.11 1- and 5-step FOE in the third experiment. 99

5.12 1- and 6-step OE in the third experiment. 100

5.13 The average proportion of time steps during which the agent has the picker equipped (fuchsia, purple, and orange curves) and during which the can is attached to the picker (green, brown, and grey curves) in the third experiment. 101

5.14 1-step FOE landscape. 104

5.15 1-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 25,000 episodes. . . . 104

5.16 2-step FOE landscape. 105

5.17 2-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 30,000 episodes. . . . 105

5.18 3-step FOE landscape. 106

5.19 3-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 25,000 episodes. . . . 106

5.20 4-step FOE landscape. 107

5.21 4-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 25,000 episodes. . . . 107

5.22 5-step FOE landscape. 108

5.23 5-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 25,000 episodes. . . . 108

5.24 6-step FOE landscape. 109

5.25 6-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 20,000 episodes. . . . 110

5.26 7-step FOE landscape. 110

5.27 7-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 30,000 episodes. . . . 111

5.28 8-step FOE landscape. 111

5.29 8-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 35,000 episodes. . . . 112

5.30 1-step TOE landscape. 114

5.31 1-step TOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 25,000 episodes. 114

5.32 2-step TOE landscape. 115

5.33 2-step TOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 30,000 episodes. 115

5.34 6-step TOE landscape. 116

5.35 6-step TOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 20,000 episodes. 117

5.36 7-step TOE landscape. 117

5.37 7-step TOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 20,000 episodes. 118

5.38 8-step TOE landscape. 118

5.39 8-step TOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 30,000 episodes. 119

5.40 State visitation heatmap under high $\beta = 0.4$ with 1-step TOE. The agent remains concentrated around the object, ignoring the tool. 121

5.41 An example of trajectory under high $\beta = 0.4$ with 1-step TOE. The agent approaches the object and stays nearby, without engaging with the tool. . . 121

5.42 State visitation heatmap under high $\beta = 0.17$ with 6-step TOE. In contrast to low horizons, the agent's behaviour shifts toward the tool before interacting with the object. 122

5.43 An example of trajectory under high $\beta = 0.17$ with 6-step TOE. The agent first acquires the tool and then interacts with the object, remaining in its vicinity thereafter. 123

5.44 State visitation heatmap under high $\beta = 0.3$ with 1-step FOE. The agent remains near the tool and persistently engages in tool use. 123

5.45 An example of trajectory under high $\beta = 0.3$ with 1-step FOE. The agent predominantly stays around the tool location and repeatedly interacts with the tool. 124

5.46 Visitation heatmap under optimal parameters ($h = 6$, $\beta = 0.1$) with TOE. The agent balances exploration across task-relevant regions, focusing on both tool and object. 125

5.47 An example of trajectory under optimal parameters ($h = 6$, $\beta = 0.1$) with TOE. The agent effectively uses the tool to interact with the object, enabling task completion. 126

5.48 Visitation heatmap for a vanilla RL agent. The agent explores large parts of the environment, including many irrelevant states, reflecting inefficient exploration. 127

6.1 Initial state of the environment of Experiment 1. Black cells represent unobserved areas hidden from the current agent's field of view. 134

6.2 8-step axe-to-tree empowerment $\mathfrak{E}^{8}_{\mathfrak{T}_{\text{axe}^*}\mathfrak{O}_{\text{tree}^*}}$ landscape for all possible agent locations (in bits), when the axe is not equipped. 136

6.3 3-step axe-to-tree empowerment $\mathfrak{E}^{3}_{\mathfrak{T}_{\text{axe}^*}\mathfrak{O}_{\text{tree}^*}}$ landscape for all possible agent locations (in bits), when the axe is equipped. 136

6.4 8-step pickaxe-to-wall empowerment $\mathfrak{E}^{8}_{\mathfrak{T}_{\text{pickaxe}^*}\mathfrak{O}_{\text{wall}^*}}$ landscape for all possible agent locations (in bits), when the pickaxe is not equipped. 137

6.5 3-step pickaxe-to-wall empowerment $\mathfrak{E}^{3}_{\mathfrak{T}_{\text{pickaxe}^*}\mathfrak{O}_{\text{wall}^*}}$ landscape for all possible agent locations (in bits), when the pickaxe is equipped. 137

6.6 Learning performance in Experiment 1. The agent using axe-to-tree empowerment $\mathfrak{E}^{h}_{\mathfrak{T}_{\text{axe}^*}\mathfrak{O}_{\text{tree}^*}}$ as a regularizer ($\beta = 0.0009$, green) converges faster than standard PPO (blue). Shaded regions indicate standard deviation across 10 runs. 138

6.7 Initial state of the environment of Experiment 2. Black cells represent unobserved areas hidden from the current agent's field of view. 139

6.8 6-step axe-to-boulder-door empowerment landscape when the axe is not equipped. Empowerment peaks at the axe's location. 141

6.9 6-step axe-to-boulder–door empowerment landscape when the axe is equipped. Empowerment reflects the axe's ability to influence the door, with nonzero values marking states from which the agent can reach and destroy the door within six steps. These values appear both near the door and in other regions of the grid, indicating feasible access within the horizon but without highlighting a single optimal strike position. 141

6.10 5-step wand-to-boulder-door empowerment landscape when the wand is not equipped. 142

6.11 6-step wand-to-boulder–door empowerment landscape when the wand is equipped. Empowerment spans a wide area of the grid, with distinct peaks of 2.0 bits identifying optimal positions from which the agent can destroy both objects within six steps. Intermediate values indicate states where only one of the two objects can be influenced, reflecting the wand's long-range effect. 143

6.12 Learning performance in Experiment 2. The agent using wand-to-boulder–door empowerment as a regulariser ($\beta = 0.0009$, green) converges faster and more reliably than standard PPO (blue). Shaded regions represent standard deviation across 10 runs. 144

7.1 Initial state of the "persistence" experiment. The agent must either open the door with the key or destroy it with the axe to access and push the boulder onto the blue goal location. 151

7.2 7-step axe-to-boulder empowerment $\hat{\mathfrak{E}}^{7}_{\mathfrak{T}_{\text{axe}}\mathfrak{O}_{\text{bould}^*}}$ landscape when the axe is unequipped. Empowerment peaks at the axe's location, since seven steps suffice to equip it, destroy the door, and reach the boulder. 152

7.3 5-step axe-to-boulder empowerment $\hat{\mathfrak{E}}^{5}_{\mathfrak{T}_{\text{axe}}\mathfrak{O}_{\text{bould}^*}}$ landscape when the axe is equipped. Empowerment values appear in states from which the agent can clear the door and approach the boulder within the horizon. 152

7.4 5-step boulder empowerment landscape after the door has been cleared. The empowerment now acts as a gradient that guides the agent towards the boulder. 153

7.5 Temporal evolution of the door state and its associated object empowerment $\mathfrak{E}^{h\geq 3}_{\mathfrak{T}_j,\mathfrak{O}_{\text{door}}}$ for the key (red) and the axe (blue). The vertical axis enumerates the possible semantic states of the door (*open*, *closed*, *destroyed*), and $\mathcal{S}^{\text{door}}$ denotes the subset of the full state space in which the door still exists as a manipulable object. The horizontal axis shows discrete time steps, where t_1^{axe} marks the moment the agent uses the axe and irreversibly destroys the door, while $t_1^{\text{key}}, t_2^{\text{key}}, t_3^{\text{key}}$ correspond to repeated openings and closings of the door using the key. The colour bar on the right encodes the value of object empowerment in bits. For the key (red bars), empowerment remains at 1 bit for all $h \geq 3$, because the agent can always choose between two distinct future door states (open or closed) and can repeat this interaction indefinitely. For the axe (blue bar), empowerment is non-zero only once: after destruction there is no alternative future state of the door, so empowerment collapses to 0 bits and remains there permanently. Thus, the figure illustrates the difference between reversible (persistent) and irreversible (non-persistent) tool–object interactions. 154

7.6 Learning curve for the persistence experiment. The empowerment-regularised agent (green) learns faster and achieves higher performance than standard PPO (blue). Shaded regions represent standard deviation over 10 runs. . . 155

7.7 Initial state of the "latency" experiment. The agent (bottom right) must destroy the boulder (top left) using either the wand (bottom left) or the pickaxe (top right). Blue-grey bars represent static obstacles that restrict movement but cannot be destroyed. 157

7.8 Wand latency landscape for the boulder at $(0, 0)$. Latency grows with Manhattan distance, since the wand can only act when the agent is aligned in the same row or column as the boulder (diagonal positions do not allow activation). The maximum latency of 12 arises from the farthest starting state $(9, 9)$, which requires 9+3 steps: 9 movement steps to reach a valid orthogonal cell and 3 additional steps to apply the wand. Lower latency (blue) corresponds to states from which causal influence can be exerted more quickly. 158

7.9 Pickaxe latency landscape for the boulder at $(0, 0)$. Latency grows with spatial distance, reaching a maximum of 20 at the bottom-right corner. The pickaxe requires the agent to approach the boulder directly, resulting in high latency in distant regions. 158

7.10 3-step wand-to-boulder empowerment $\hat{\mathfrak{E}}^3_{\mathfrak{T}_{\text{wand}}\,\mathfrak{O}_{\text{bould}^*}}$ landscape (in bits) for all possible agent locations when the wand is not equipped. 159

7.11 3-step wand-to-boulder empowerment $\hat{\mathfrak{E}}^{3}_{\mathfrak{T}_{\text{wand}}\mathfrak{O}_{\text{bould}^*}}$ landscape (in bits) for all possible agent locations when the wand is equipped. 159

7.12 Learning curve for the latency experiment. The empowerment-regularised agent (green) learns faster and achieves higher performance than standard PPO (blue). Shaded regions represent standard deviation over 10 runs. . . 160

7.13 Initial state of the reliability experiment. The can must be brought into the bin. Three pickers are available and differ only by their noise level θ: green (0), blue (0.15), red (0.9). 162

7.14 6-step green_picker-to-can empowerment landscape $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}_{\text{green\_pick}}\mathfrak{O}_{\text{can}^*}}(\theta{=}0.0)$. Empowerment peaks (4.86 bits) near the fully reliable (green) picker and extends broadly toward the can, reflecting stable and deterministic tool dynamics. 164

7.15 6-step blue_picker-to-can empowerment landscape $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}_{\text{blue\_pick}}\mathfrak{O}_{\text{can}^*}}(\theta{=}0.15)$. Peak empowerment (3.81 bits) occurs both near the can and adjacent to the moderately reliable (blue) picker, reflecting partially degraded but functional causal control. 165

7.16 6-step red_picker-to-can empowerment landscape $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}_{\text{red\_pick}}\mathfrak{O}_{\text{can}^*}}(\theta{=}0.9)$. Maximum empowerment (3.81 bits) is limited to the can's vicinity, while values near the unreliable (red) picker drop to 2.40 bits, indicating unstable tool influence under high noise. 166

7.17 1-step classical empowerment landscape. The highest empowerment values (3.00 bits) occur near the most reliable (green) picker, followed by 2.70 bits near the moderately reliable (blue) picker, and 2.33 bits near the unreliable (red) picker. Notably, the empowerment around the unreliable picker is only marginally higher than the most of the environment (2.32 bits). This pattern mirrors the reliability hierarchy observed in object empowerment. 167

7.18 Learning performance comparison between PPO and PPO regularised with the 6-step green_picker-to-can empowerment $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}_{\text{green\_pick}}\mathfrak{O}_{\text{can}}}$ ($\beta = 0.004$). Curves show the mean extrinsic reward over training episodes. Shaded regions represent standard deviation over 10 runs. 168

List of Tables

1 Mapping between thesis chapters and related publications. xii

2.1 Comparative summary of tool-use behaviours documented across diverse non-human species. 15

5.1 β values that yielded the fastest convergence for each horizon h under FOE. 112

5.2 β values that yielded the fastest convergence for each horizon h under TOE. 119

6.1 Tool–object empowerment matrix $\mathbb{T}$ showing the state-averaged empowerment $\hat{\mathfrak{E}}^{h}_{\mathfrak{T}_j\mathfrak{O}_i}$ for each tool–object pair. Values indicate the degree of influence each tool has over each object, and i^* denotes the object of interest (i.e., task-relevant (target) object). 132

6.2 State-averaged tool-to-object empowerment $\hat{\mathfrak{E}}^{3}_{\mathfrak{T}_j\mathfrak{O}_i}$ in bits for each tool–object combination of Experiment 1. All object empowerment values are computed with the corresponding tool in the equipped state. 135

6.3 State-averaged tool-to-object empowerment $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}_j\mathfrak{O}_i}$ in bits for each tool–object combination of Experiment 2. The last column reports the multi-object empowerment $\hat{\mathfrak{E}}^{h}_{\mathfrak{T}_j\,\mathfrak{O}_{\text{bould}^*}\mathfrak{O}_{\text{door}^*}}$ values. All object empowerment values are computed with the corresponding tool in the equipped state. 139

7.1 Tool–object latency matrix $\mathbb{L}$ showing the state-averaged latency $\hat{\mathcal{L}}_{\mathfrak{T}_j,\mathfrak{O}_i}$ for each tool–object pair. Values indicate the state averaged minimum number of steps required for each tool to begin influencing each object, where i^* denotes the object of interest (i.e., task-relevant (target) object). 149

7.2 State-averaged latency $\hat{L}_{\mathfrak{T}_j\mathfrak{O}_i}$ for each tool–object pair in the latency experiment. Lower values indicate faster causal influence. 157

7.3 State-averaged empowerment $\hat{\mathfrak{E}}^{h}_{\mathfrak{T}_j\mathfrak{O}_i}(\theta)$ (in bits) for each tool–object pair under different noise levels θ. Lower values indicate reduced reliability due to stochastic tool dynamics. 162

Chapter 1

Introduction

The ability to use tools is often regarded as one of the defining hallmarks of intelligence [2]. From crows bending wires into hooks, to chimpanzees fashioning twigs for termite fishing, to humans constructing complex machines, tool use exemplifies purposeful, goal-directed interaction with the environment. Across species, it reflects not merely motor skill but a deeper capacity to understand causal structure, which recognises that an external object can extend one's own agency [3]. Such behaviour lies at the intersection of perception, action, and cognition, and has long inspired research in psychology, neuroscience, and artificial intelligence. It raises a fundamental question: how do intelligent systems, biological or artificial, come to understand what they can influence, and how that influence can be extended through tools?

In biological evolution, tool use did not emerge from external instruction but through intrinsic exploration. Animals and early humans were not told which objects were useful; they discovered affordances through curiosity, experimentation, and play. This capacity for open-ended exploration acted not just to gain an immediate reward, but to expand one's sphere of influence, which has been seen as a fundamental part of many kinds of adaptive behavior. In cognitive science, such self-driven activity is captured by the notion of *intrinsic motivation*, which is internal drives that promote learning and exploration even in the absence of extrinsic goals [4, 5]. Intrinsic motivation enables organisms to develop competencies, form causal models, and prepare for future challenges. It thus represents a cornerstone of autonomous intelligence [6].

Within computational modelling, intrinsic motivation has been formalised in various ways, such as through curiosity [7], novelty [8], or information gain [9]. Among these formulations, *empowerment* stands out for its direct grounding in information theory, control and its emphasis on the causal capabilities of the agent [1, 10]. Empowerment quantifies the amount of potential influence an agent has over future states of its environment; it

measures how much the agent *could* affect the world through its available actions, regardless of any specific goal. By formalising control as the channel capacity between actions and future states, empowerment bridges sensorimotor contingencies and decision-making under uncertainty, providing a mathematically elegant and intuitively meaningful notion of autonomy.

While alternative intrinsic motivation mechanisms such as curiosity or novelty focus on epistemic uncertainty or state visitation, they do not directly quantify an agent's causal capacity to influence specific entities in its environment. For modelling tool use, where the central question concerns how actions propagate through intermediate artefacts to affect manipulable objects, a measure grounded explicitly in causal influence is particularly appropriate. Empowerment provides such a measure by formalising control as the channel capacity between actions and future states, and by remaining independent of any externally specified task objective. This makes empowerment a particularly suitable conceptual starting point for the specific objective of modelling mediated tool–object interactions.

In this view, agents act to maximise their future influence; means to keep options open and maintain the ability to shape their environment. Such behaviour captures the essence of intrinsic motivation while remaining interpretable in terms of causal structure. However, while empowerment has been successfully applied to problems of control and exploration, it has not been studied in contexts that require the use of *tools*. Tool use involves a distinctive hierarchy of control: an agent acts upon a tool, which in turn acts upon an object. The causal pathway from action to outcome is therefore indirect and mediated. This layered structure poses new challenges for empowerment-based reasoning: how should an agent quantify its influence when it must first manipulate one entity to affect another?

Addressing this question forms the central motivation of the present thesis. If empowerment is to serve as a general principle of autonomous intelligence, it must be able to account for such mediated interactions, where agency is extended through instruments rather than exerted directly. Doing so requires new formulations that disentangle the causal contributions of agents, tools, and objects, while preserving the information-theoretic essence of empowerment. The research presented in this thesis develops precisely such formulations, providing a principled framework for modelling tool use as an emergent consequence of intrinsic motivation.

While empowerment provides a principled measure of control, it is often realised within a learning framework that allows agents to act, observe, and adapt through experience. *Reinforcement Learning* (RL) offers precisely a framework where agents learn through trial and error by interacting with an environment, receiving feedback in the form of rewards [11]. Formally, RL models the agent–environment interaction as a Markov Decision Process (MDP), where the objective is to learn a policy that maximises expected cumulative

reward. Through repeated experience, agents infer which actions yield beneficial outcomes, progressively shaping behaviour from feedback.

Within this thesis, RL is adopted as the learning framework within which empowerment is implemented and evaluated, serving as a flexible substrate for analysing how empowerment-based intrinsic regularisation shapes learning dynamics and tool-use behaviour. The choice of RL is further motivated by the sequential and interactive nature of tool use. Tool–object interactions require agents to make a sequence of decisions in which actions have delayed and mediated effects, often involving multiple stages such as locating, acquiring, and applying a tool. This naturally places the problem within the domain of sequential decision making under uncertainty [11]. Alternative frameworks offer only partial solutions in this context. Control-theoretic approaches [12] are well suited to continuous control in robotics but are less appropriate for the discrete environments considered in this thesis. Symbolic planning methods [13] can represent structured action sequences but typically assume known dynamics and do not address learning from interaction. Bayesian approaches provide principled models of uncertainty, but in practice overlap with RL formulations when applied to sequential decision problems [14]. In contrast, RL provides a unified framework that supports both learning and decision making under uncertainty, enabling agents to acquire behaviour through interaction while adapting to stochastic dynamics. Its recent success across a wide range of domains [15, 16] further supports its suitability as the computational framework within which empowerment-based intrinsic motivation can be studied. However, despite its success, conventional RL remains fundamentally limited by its reliance on *extrinsic rewards*. In many real-world and exploratory settings, rewards are sparse, delayed, or even absent, making learning unstable and inefficient. Biological organisms, in contrast, display remarkable learning even without explicit reinforcement, guided instead by internal drives such as curiosity and mastery. To enable artificial agents with similar autonomy, researchers have proposed *intrinsically motivated RL*, where internal signals supplement or replace external rewards [17, 18].

Empowerment naturally integrates within this paradigm. Rather than rewarding specific outcomes, empowerment rewards the *potential* to influence, acting as an intrinsic utility that favours states offering greater causal control over the future. When combined with RL, it provides a mathematically grounded intrinsic reward that aligns exploration with the discovery of controllable, meaningful interactions. In this thesis, empowerment serves not as an alternative to RL, but as its complement: an information-theoretic regulariser that guides policy learning toward states and behaviours rich in causal potential.

This synthesis of empowerment and RL provides a computational lens on tool use. By embedding empowerment within policy optimisation, agents can learn not merely to achieve given goals, but to understand how their actions propagate through tools to affect

the world. In this sense, empowerment bridges the gap between low-level control and high-level reasoning, offering a route toward intrinsically motivated, goal-independent skill acquisition.

1.1 Research Hypothesis and Questions

Before addressing specific empirical questions, this thesis begins with a more fundamental conceptual inquiry: *what constitutes a tool within the RL framework?* In natural and biological settings, a tool is an intermediary that extends an agent's capacity to act upon the world [19]. Translating this notion into computational terms requires a formalism that can express such mediated influence. While classical empowerment quantifies an agent's potential influence over future global states, it does not explicitly distinguish between incidental state changes and purposeful manipulation of specific objects. This thesis therefore extends empowerment to explicitly model agent–object interactions, enabling tools to be formalised as entities that transmit and amplify the causal link between an agent's actions and their effects on manipulable objects. This extension is reflected not only conceptually but also in the structure of the action space: classical empowerment considers all possible action sequences, whereas the proposed object-centred formulations restrict attention to action sequences that are relevant to object manipulation, such as agent-only actions for direct interaction and combined agent–tool actions for mediated effects. This interpretation forms the conceptual foundation upon which all subsequent questions and experiments are built.

The central hypothesis of this research is:

> **"Empowerment, when extended to explicitly model agent–object interactions, provides a sufficient and interpretable foundation for modelling tool use in artificial agents. This extension enables agents to discover, select, and characterise tools through their causal influence on manipulable objects, without relying on external supervision or task-specific rewards."**

To evaluate the central hypothesis, empowerment must be progressively extended, implemented within learning contexts, and examined for generality and interpretability. The following research questions structure this progression explicitly.

1. Since the central hypothesis requires empowerment to account for mediated object-level influence, it is first necessary to determine whether empowerment can be reformulated to explicitly capture object-level causal influence.

RQ1: How can empowerment be extended to explicitly capture the agent's influence over manipulable objects?
Classical empowerment quantifies control over global future states of the environment [1, 10]. However, this formulation does not distinguish between changes to task-relevant objects and incidental changes to the background state. In tool-use contexts, this distinction is critical, as meaningful control involves specific causal effects on manipulable entities. This question therefore seeks to formulate an *object-conditioned empowerment* measure that isolates the agent's influence on particular objects, establishing the foundation for studying empowerment as a model of purposeful interaction.

2. Having established a formal object-conditioned formulation, it is then necessary to examine whether this extension can function as an intrinsic mechanism for autonomous discovery of tool–object interactions.

   RQ2: Can object-conditioned empowerment serve as an intrinsic signal for discovering functional tool–object interactions?
   While intrinsic motivation has been used to guide exploration [20], its application to discovering functional dependencies between tools and objects remains underexplored. This question investigates whether object empowerment can act as an internal reward that encourages agents to interact with tools and objects in ways that increase their potential influence. By embedding object empowerment as a regulariser within RL, the goal is to determine whether agents can autonomously discover tool-use skills without explicit task rewards.

3. To understand the behavioural consequences of integrating object-conditioned empowerment within RL, its effects on learning dynamics must be empirically evaluated.

   RQ3: How do object-conditioned empowerment regularisers influence RL dynamics in sparse-reward environments?
   Empowerment has been successfully integrated into RL as an intrinsic drive for exploration and control [21–23]. However, its object-conditioned formulation (i.e., introduced in this thesis) extends this idea to capture causal influence over specific entities rather than global states. Sparse-reward domains such as MiniHack present ideal testbeds for examining how *object empowerment regularisation* affects convergence speed, policy stability, and behavioural interpretability when agents must discover and exploit tool–object relationships. This question therefore investigates whether object-empowerment-guided agents exhibit more efficient exploration and structured tool-use behaviours compared to baseline RL models.

4. If the proposed empowerment extension is to constitute a general foundation for modelling tool use, it must scale beyond single tool–object interactions to more complex

environments containing multiple tools and objects.

RQ4: How can empowerment be generalised to settings with multiple tools and objects to enable systematic selection?
Real-world environments rarely contain a single tool or object. Selecting the most effective tool for a given task requires comparing the agent's potential influence across multiple tool–object pairs. This question examines how to extend object empowerment into a *multi-object* formulation, yielding a structured tool–object empowerment matrix from which optimal tool choices can be derived. Such a matrix formalises tool selection as an empowerment maximisation problem, providing a principled alternative to heuristic or task-specific approaches.

5. Finally, if the proposed empowerment extension is to serve as an interpretable foundation, it should provide meaningful dimensions for characterising tools beyond mere selection.

   RQ5: Beyond selection, can empowerment also provide interpretable dimensions for characterising tools?
   Beyond knowing *which* tool to use, intelligent agents should also understand *why* certain tools are more effective, persistent, or reliable than others. This question explores whether empowerment can be decomposed into distinct dimensions, such as *persistence*, *latency*, and *reliability*, that collectively describe temporal and stochastic aspects of tool behaviour. By framing these as measurable empowerment-based quantities, this research aims to develop a principled language for comparing and characterising tools in terms of their causal and temporal properties.

Collectively, these research questions operationalise the central hypothesis by progressing from formal extension to learning integration, generalisation, and interpretability. Together, they articulate the overarching aim of this thesis: to develop a unified information-theoretic framework for modelling tool use grounded in empowerment. This framework spans progressively richer levels of abstraction, from object-specific influence to tool selection and characterisation.

1.2 Contributions of the Thesis

The thesis advances its central aim through a coherent set of contributions that extend empowerment from a general measure of controllability to a systematic framework for tool-mediated interaction and learning within RL. In what follows, the term *object empowerment* is used to refer to the object-conditioned extension of empowerment introduced in response

to Research Question RQ1. Each contribution directly addresses one or more of the research questions introduced in Section 1.1 and is developed in the corresponding chapters.

- **Object Empowerment (RQ1; Chapter 4):** Formulates empowerment conditioned on the state of manipulable objects, establishing a direct and interpretable link between intrinsic motivation and object-centred causal influence.
- **Multi-Object Empowerment (RQ4; Chapter 6):** Extends this formulation to environments with multiple tools and objects, yielding the *tool–object empowerment matrix*, from which a principled, intrinsic mechanism for tool selection is derived.
- **Tool Characterisation (RQ5; Chapter 7):** Introduces three empowerment-based dimensions, *persistence*, *latency*, and *reliability*, that formalise how long a tool remains effective, how quickly its influence propagates, and how robustly it performs under noise and uncertainty.
- **Tool-Learning Framework (RQ2–RQ3; Chapter 5):** Develops a unified computational framework that integrates empowerment with RL by decomposing the environment into agent, tool, and object subspaces.
- **Object Empowerment-Regularised Optimisation (RQ2–RQ3; Chapter 5):** Embeds object empowerment as an intrinsic regulariser in policy optimisation, integrating information-theoretic and reward-driven objectives to guide exploration toward causally informative states.

Together, these contributions establish a principled and extensible foundation for modelling tool use through empowerment. They provide both the conceptual formalism for understanding causal influence in structured environments and the methodological apparatus for implementing empowerment-driven behaviour in learning agents.

1.3 Structure of the Thesis

The remainder of this thesis is organised into seven chapters, each developing a successive layer of the empowerment-based framework for modelling tool use. In particular, Chapters 4–7 systematically address research questions RQ1–RQ5, progressively operationalising the central hypothesis.

Chapter 2 surveys the background and related work underpinning this research. It begins with the study of tool use in biological and artificial systems, drawing connections between ethological, psychological, and computational perspectives. It then reviews the literature on intrinsic motivation and information-theoretic approaches in RL, situating

empowerment within this broader context. The chapter concludes by identifying the conceptual and methodological gaps that motivate the development of empowerment-based formulations for tool-use modelling.

Chapter 3 provides the theoretical foundations necessary for the rest of the thesis. It introduces key concepts from RL, regularisation, and information theory, explaining how the notion of a communication channel can be repurposed to describe agent–environment interactions. This chapter establishes the formal background for empowerment as a measure of potential control and presents the mathematical tools required for its subsequent extensions.

Chapter 4 addresses RQ1 by introducing *object empowerment*, the first major contribution of the thesis. It extends classical empowerment to quantify the agent's influence over specific manipulable entities, thereby grounding the concept of causal control in object-directed interaction. The chapter also presents the first experimental results, visualising empowerment landscapes and demonstrating how they encode affordance structures in both simple grid worlds and MiniHack environments.

Chapter 5 addresses RQ2 and RQ3. It builds upon this foundation by introducing a framework for *learning tool–object interactions*. Here, object empowerment is integrated into RL as an intrinsic regulariser, allowing agents to autonomously discover meaningful dependencies between tools and objects without external supervision. The experiments highlight how empowerment-guided exploration facilitates tool acquisition and use in sparse-reward settings.

Chapter 6 addresses RQ4. It generalises the framework to environments containing multiple tools and objects. It formalises the *tool–object empowerment matrix*, a compact representation that quantifies each tool's potential causal influence over each object. From this matrix, a principled tool selection mechanism is derived, enabling agents to identify and prioritise the most effective tools intrinsically, based solely on empowerment structure. Empirical studies demonstrate how this mechanism leads to faster convergence and more interpretable behaviour in MiniHack-based tool-use tasks.

Chapter 7 addresses RQ5. It advances the framework from selection to *characterisation*. It introduces three empowerment-based dimensions, persistence, latency, and reliability, that capture complementary aspects of tool functionality: temporal continuity, temporal efficiency, and robustness under uncertainty. Through experiments in both MiniHack and custom grid worlds, the chapter shows how these measures jointly explain observed differences in tool utility and learning dynamics, thereby providing a more complete descriptive account of empowerment-driven tool use.

Chapter 8 concludes the thesis by integrating the theoretical, methodological, and

empirical findings. It discusses the broader implications of empowerment as a unifying principle for intrinsic motivation, causal reasoning, and autonomous behaviour. The chapter also outlines limitations of the current framework, such as computational cost, model assumptions, and challenges of generalisation, and highlights future directions for scaling empowerment-based models to more complex, continuous, and embodied domains.

In summary, the thesis proceeds from foundational theory to hierarchical application, gradually transforming empowerment from an abstract concept into a practical mechanism for discovery, selection, and characterisation of tools. Each chapter deepens the integration of empowerment with RL, culminating in a coherent and interpretable framework for intrinsically motivated, tool-using agents.

Chapter 2

Literature Review

This chapter reviews the multidisciplinary literature that provides the conceptual and empirical foundations for modelling tool use computationally. The discussion begins with reviewing biological and ethological studies, documenting tool-use behaviours across non-human species and tracing their cognitive and ecological underpinnings to establish the evolutionary and behavioural context within which computational models of tool use can be framed. Archaeological and anthropological perspectives on tool use in human evolution are then examined, highlighting how increasing technological and behavioural complexity in tool manufacture and use contributed to human cognitive and cultural development.

The chapter proceeds with reviewing affordance theory, which is a framework for modelling agent-object interactions. In general, affordance-based models enable agents to acquire functional representations of tools through interaction, learning how external objects extend their control over the environment.

While affordance theory offers a conceptual account of how agents perceive and act upon possibilities for interaction, computational frameworks such as reinforcement learning (RL) provide a formal means of realising these behaviours in artificial agents. In particular, intrinsic motivation theories translate the drive to explore and manipulate affordances into measurable learning signals. Thus, RL and intrinsic motivation can be viewed as algorithmic instantiations of the same exploratory and goal-directed processes that underpin tool use in natural systems.

Intrinsic motivation is introduced as a central mechanism for autonomous exploration and skill acquisition in RL. Multiple approaches are reviewed, including curiosity-driven exploration [7], novelty seeking [8], competence progress [24], and mutual information-based frameworks [25] that unify exploration, skill discovery, and information gain.

Special attention is given to empowerment, an information-theoretic intrinsic motivation that quantifies an agent's control over its environment and serves as a foundation for

tool-use modeling. The chapter also reviews studies relating intrinsic motivation directly to tool-use learning, showing that internal drives such as curiosity or empowerment can generate exploratory behaviour even in the absence of external rewards. Through this self-motivated interaction, agents gradually acquire competence in manipulating tools and develop simple forms of planning to achieve desired environmental outcomes.

The literature reviewed here provides context and motivation for the development of *object empowerment*, a novel intrinsic motivation framework that contributes to the modelling of tool use by quantifying an agent's capacity to influence task-relevant objects through their interactions with tools.

2.1 Defining Tools and Tool Use

Understanding what constitutes a *tool* is a fundamental step toward studying how agents, biological or artificial, can learn to use them. Tool use is commonly defined as *"The external employment of an unattached or manipulable attached environmental object to alter more efficiently the form, position, or condition of another object"* [26]. This definition highlights the causal chain between an agent, a tool, and a target object, establishing that tools function as intermediaries through which the agent exerts influence on its surroundings.

Refinements to this definition emphasize two additional aspects: intentionality and embodiment [27]. Intentionality distinguishes tool use from accidental interactions, implying that the agent purposefully manipulates the tool to achieve a desired outcome. Embodiment, on the other hand, refers to the functional role of tools as extensions of the organism or agent, temporarily incorporated into its action capabilities [27]. These refinements highlight that tool use is not merely mechanical manipulation but involves adaptive reasoning about cause, effect, and utility.

This conceptual framing directly motivated the development of the object empowerment framework, introduced in later chapters. In this formulation, the tool is explicitly modeled as a mediator of causal influence between the agent and a target object, capturing how the agent's control propagates through the environment via tool–object interactions. In this sense, object empowerment provides a computational instantiation of the very principle embedded in the classical ethological definition of tool use: an agent's capacity to act upon one object through another.

The notion of tool use has been studied across different disciplinary perspectives. In ethology, it is investigated as an evolved behavioural strategy that enhances adaptability and resource acquisition among non-human animal species. In anthropology, it represents a central feature of human technological and cultural development, reflecting cumulative transmission of knowledge and skill. Together, these perspectives reveal that tool use is

both a biological and cognitive phenomenon, one that bridges perception, motor control, and learning across species. The conceptual foundation outlined above provides the theoretical grounding for analysing tool use across biological and artificial systems. The next sections review its manifestations in nature, from non-human animals to human evolution.

2.2 Tool Use in Nature: Ethology and Anthropology

Tool use is a foundational aspect of intelligent behaviour observed across both human and non-human animal species. It reveals complex interactions between cognition, environment, and social–cultural transmission, whereby knowledge and skills related to tool use are passed across individuals and generations. This section introduces key biological foundations and evolutionary aspects of tool use, focusing on examples from non-human and human species that illustrate differences in complexity, intentionality, and the gradual emergence of cultural accumulation. Understanding these natural manifestations of tool use not only informs anthropology and ethology but also provides valuable inspiration for computational models of tool use in embodied cognition and intelligent behaviour.

2.2.1 Tool Use in Non-Human Animals

Although once considered a hallmark of uniquely human intelligence, tool use has been observed across a diverse range of animal taxa, including primates, birds, marine mammals, and even insects. This section surveys the breadth of non-human animal tool use, organised by major taxonomic groups. Each group reveals how ecological context, cognition, and evolution interact to shape tool-oriented behaviour.

Among primates, chimpanzees (*Pan troglodytes*) demonstrate the most diverse and sophisticated tool repertoire [28–30]. In the Tai Forest, Ivory Coast, West Africa, they use a combination of stones as hammer and anvil to crack nuts [28], and at Gombe, Tanzania, they use twigs for termite fishing [29]. In Bossou, Guinea, West Africa, they manufacture and reuse leaf sponges to drink water [30]. Capuchin monkeys (*Cebus apella*) use stones to break open palm nuts [31], and long-tailed macaques (*Macaca fascicularis*) have been observed using tools to access shellfish and floss their teeth [32], indicating a broad phylogenetic distribution of tool competence among primates. These behaviours show evidence of planning, tool transport (i.e., carrying tools in anticipation of future use), and manufacturing.

Tool use among birds is most famously represented by New Caledonian crows (*Corvus moneduloides*), which manufacture and use hook-shaped tools from twigs and pandan leaves to extract insects from crevices [33]. These crows not only shape tools with skill and

foresight, but have also demonstrated remarkable planning abilities, including *sequential tool use*, whereby one tool is used to obtain another [34]. They have also been shown to select tools of appropriate length and shape depending on the demands of the task, demonstrating a flexible understanding of tool properties [35]. Such behaviour reflects an ability to evaluate the affordances of available tools and to select the most effective one for achieving a specific goal (i.e., an ability that conceptually parallels the mechanism of tool selection modelled through object empowerment in this thesis). In addition to crows, captive Goffin's cockatoos (*Cacatua goffiniana*) have also been observed spontaneously manufacturing tools from sticks to retrieve rewards, demonstrating impressive flexibility and innovation in problem-solving [36]. These parrots, which do not use tools in the wild, still manage to display novel tool construction in captivity, indicating latent cognitive capacities. Green-backed herons (*Butorides striata*) are known to use bait, such as insects, bread, or feathers, to lure fish, a sophisticated form of tool-assisted hunting [37]. These examples illustrate that avian tool use is not only widespread but also marked by flexible strategies and ecological adaptation, consistent with the broader evolutionary functions of tool use discussed above.

Marine species have also evolved independent instances of tool use. Sea otters (*Enhydra lutris*) use rocks as anvils to break open shellfish on their chests [38]. Among bottlenose dolphins (*Tursiops Truncatus*), individuals in Shark Bay carry marine sponges to protect their snouts while foraging, a behaviour shown to be culturally transmitted, particularly through maternal lines [39, 40]. This culturally transmitted behaviour is one of the rare examples in non-human animals of social learning associated with tool use.

While tool use is often associated with vertebrates, a growing body of research highlights that various insect species also exhibit behaviours that meet established definition of tool use. These behaviours demonstrate that complex interactions with environmental objects are not exclusive to animals with large brains or advanced nervous systems [41]. Certain ant species, such as those in the genus *Aphaenogaster*, have been observed using debris like soil particles, leaves, or pine needles to absorb liquid food sources. The ants then transport these soaked materials back to their nests, effectively using the debris as tools to carry food they cannot transport internally due to anatomical constraints [42, 43]. Additionally, *Conomyrma bicolor* ants engage in a form of tool use for competitive interference. They collect small stones and drop them into the entrances of rival colonies, obstructing access and reducing competition for resources [44].

A comparative summary of documented tool-use behaviours across a range of non-human species, including primates, birds, marine mammals, and insects, is presented in Table 2.1.

Taken together, these findings demonstrate that tool use in non-human animals spans

a remarkable range of taxa and ecological contexts. From primates and birds to marine mammals and insects, animals have evolved creative strategies to exploit environmental resources through object manipulation. This variation in cognitive strategies and problem-solving mechanisms across species suggests that tool use has emerged convergently under distinct evolutionary pressures. Although such behaviours vary in complexity, many satisfy the established criteria for tool use (i.e., the deliberate manipulation of an external object to achieve a specific goal or alter another object's state [26, 27]). They also reveal important cognitive, ecological, and social underpinnings: for example, the cultural transmission of nut-cracking techniques among chimpanzees and sponge-carrying behaviour in dolphins illustrate how some tool-use traditions are maintained and passed across generations. These insights provide an essential comparative baseline for understanding how tool use evolved in humans, to which the discussion now turns.

2.2.2 Tool Use in Human Evolution

While many non-human animals exhibit tool-oriented behaviours, tool use in humans is distinguished by its complexity, intentional design, and cumulative cultural evolution. Archaeological and paleoanthropological evidence suggests that tool use was not merely an adaptation, but a driving force in the co-evolution of human cognition, social organization, and technological advancement [45].

Early Stone Tool Industries and Cognitive Implications

The earliest known stone tools belong to the Oldowan industry, dated to approximately 2.6 million years ago and associated with species such as *Homo habilis* and late *Australopithecus* [46]. These tools, simple flakes and cores, were used for cutting, scraping, and pounding, representing a major leap in ecological flexibility. Around 1.7 million years ago, the Acheulean industry emerged, characterized by the appearance of bifacial handaxes. These tools exhibit symmetry, standardization, and multi-stage production processes that suggest cognitive advancements in planning [47, 48]. Neuroarchaeological studies indicate that engaging in Acheulean tool manufacture activates regions of the brain associated with motor planning, spatial reasoning, and even language processing [49]. This supports the hypothesis that tool-making and language may have co-evolved, reinforcing one another through shared demands on sequencing, hierarchical processing, and social learning.

Cumulative Culture and Meta-Tool Use in Human Evolution

Human tool use is also distinguished by its transmission across generations, leading to cumulative improvements; a phenomenon often referred to as the "ratchet effect" [50].

Table 2.1: Comparative summary of tool-use behaviours documented across diverse non-human species.

Species	**behaviour**	**Tool Type**	**Function**	**Reference**
Pan troglodytes (Chimpanzee)	Nut cracking, termite fishing, leaf sponging	Stones, twigs, leaves	Foraging, water collection	[28–30]
Cebus apella (Capuchin monkey)	Nut cracking	Stones	Foraging	[31]
Macaca fascicularis (Long-tailed macaque)	Opening shellfish, dental flossing	Stones, hair	Foraging, grooming	[32]
Corvus moneduloides (New Caledonian crow)	Hook tool manufacturing	Sticks, leaves	Extracting prey	[33, 34]
Cacatua goffiniana (Goffin's cockatoo)	Tool manufacture for retrieval tasks	Wooden sticks	Problem-solving	[36]
Enhydra lutris (Sea otter)	Shell cracking	Rocks	Foraging	[38]
Tursiops truncatus (Bottlenose dolphin)	Snout protection while foraging	Marine sponges	Foraging tool for protection	[39, 40]
Aphaenogaster spp. (Ants)	Transporting liquid food	Leaves, pine needles	Foraging/transport	[42, 43]
Conomyrma bicolor (Ants)	Blocking rival colony entrances	Pebbles, stones	Competitive interference	[44]

In contrast to non-human primates, whose tool-use behaviours tend to remain relatively fixed over time, human societies build upon prior innovations. This accumulation is facilitated by teaching, imitation, and language, forming the foundation of human technological evolution.

An especially striking feature of human tool use is the ability to use tools to create other tools, so-called meta-tool use. For example, early humans developed bone or stone tools to shape wooden handles, leading to the emergence of composite tools such as spears and axes. This recursive use of tools indicates a significant cognitive shift, enabling abstraction, analogical reasoning, and goal-subgoal decomposition [51].

Tool use during human evolution offers a compelling model for the co-development of intelligence, culture, and embodiment. The emergence of recursive, planned tool construction, from shaping wooden handles to assembling composite tools, reflects a cognitive progression from reactive behaviour to deliberative reasoning. This trajectory highlights advanced capabilities such as abstraction, analogical reasoning, and hierarchical planning. These same principles provide critical inspiration for the development of artificial agents capable of flexible, open-ended tool use. As the discussion moves from naturalistic studies to computational models, the challenge becomes how to formalise affordances and tool representations in embodied systems; this is explored in the next section.

2.3 Affordance Theory, Tool Use, and Learning

Affordance theory, first introduced by J.J. Gibson, provides a foundational account of how agents perceive and interact with their environment [52]. Gibson coined the term *affordances* to describe the action possibilities that the environment offers to an agent, relative to the agent's capabilities. In the context of tool use, affordances refer to the potential actions that an agent can perform with a given object; effectively, what the tool affords the agent. In this view, tool affordances form a subset of general environmental affordances, distinguished by their potential to expand the agent's reach or effect space. While environmental affordances describe what is possible in the agent's current state, tool affordances describe what becomes possible when an external object is leveraged as an extension of the agent's body or intent. This distinction is crucial for artificial agents, as it allows the formalization and discovery of "tools" not just as manipulable objects, but as means to transform the agent's control over the environment.

These ideas have profoundly influenced robotics and AI, where affordances are increasingly used to enable agents to interpret and act on their environments. Building on these theoretical foundations, a number of studies in robotics and AI have operationalized affordances to enable intelligent tool-use behaviour. Stoytchev [53] applied the concept to

robotics, showing that tool affordances are not intrinsic properties but are learned through interaction. In his experiments, robots discovered tool affordances by engaging in exploratory behaviour, thereby learning that certain objects could extend their action possibilities. This aligns with the key idea of this thesis: an agent can discover the utility of objects as tools by interacting with its environment.

Sinapov et al. [54] introduced a behaviour-grounded approach in which a robot learns the consequences of tool-mediated actions, effectively acquiring a model of tool affordances. Their system allowed generalization to novel tools after a behavioural babbling phase, highlighting the importance of learning dynamics through experience. Similarly, Jain et al. [55] proposed a framework in which a robot learns to manipulate target objects using tools through autonomous exploration, collecting relational data between tools, actions, and their effects. This data was then modeled using a Bayesian Network to infer tool affordances.

Gonçalves et al. [56] developed a computational model of *multi-object affordances*, also based on Bayesian Networks, which captures interactions involving intermediate tools used on primary objects, closely reflecting the structure of tool-object relationships in the RL framework proposed in this thesis. Later, the same authors extended this work to model mutual affordances between objects, where one object plays the role of a tool and the other as a manipulable entity [57]. Saito et al. [58] proposed a deep learning-based approach to select appropriate tools for given tasks, offering another parallel to the setting considered in this thesis, where agents must compare tool utility under empowerment-driven exploration.

Recent advancements have explored integrating language into affordance learning. Ren et al. [59] introduced ATLA (Accelerated Learning of Tool Manipulation with Language), a meta-learning framework leveraging large language models to accelerate tool learning. By conditioning policies on language descriptions of tools, ATLA enables agents to adapt quickly to new tools across tasks like pushing, lifting, sweeping, and hammering. Furthermore, causal approaches have been applied to tool affordance learning. Brawer et al. [60] introduced a method where robots construct structural causal models through observation and self-supervised experimentation, enabling reasoning from causes to effects and vice versa. This approach allows robots to learn and utilize tool affordances effectively, even with novel tools. These newer directions, leveraging language and causal reasoning, highlight the growing interest in enabling agents to generalize affordance knowledge across modalities and contexts, a necessary step toward robust, open-ended tool use.

While these studies emphasize learning tool affordances through interaction and probabilistic reasoning, most focus on predicting object functionality or task success without quantifying the degree of control a tool enables. This thesis addresses this gap by proposing a novel information-theoretic framework grounded in empowerment. This formulation not

only facilitates the discovery of tools via interaction but also enables agents to compare them based on how much they expand the agent's potential to influence future states, offering a principled and scalable metric for tool affordance evaluation.

Beyond robotics, the concept of affordances has gained increasing relevance in RL. Since affordances characterize what actions are possible in a given state, they provide a natural framework for identifying and formalizing tools. In RL, a tool can be viewed as a "special" object that expands the agent's action repertoire, enabling outcomes that would otherwise be inaccessible. For example, if an agent must move a distant object and finds a stick nearby, using the stick to push the object reduces the number of steps needed; an instance of affordance enhancement.

Several RL studies have sought to formalize affordances to improve learning efficiency and generalization. Khetarpal et al. [61] proposed modeling affordances as mappings between environmental features and feasible actions. This enables agents to focus on task-relevant actions, thereby improving sample efficiency and transition model learning. Liao et al. [62] offered an integrative theory of affordance-formation based on RL principles, showing how agents can learn to associate perceptual cues with promising motor actions through interaction.

The BabyAI platform by Chevalier-Boisvert et al. [63], built on the MiniGrid environments [64], provides another compelling illustration of affordance learning. Agents are guided via natural language instructions to perform tasks such as retrieving a 'key' to open a 'door'. While the framework does not explicitly frame the key as a tool, the functional role it plays mirrors tool-like behaviour, where objects must be understood and used in sequence to accomplish a goal. The paper highlights how affordance learning supports structured, compositional behaviours in RL settings.

More recently, Liu et al. [65] introduced a RL framework that enables agents not only to learn how to use tools, but also to autonomously design them for task-specific manipulation. Their dual-policy architecture comprises a designer policy that generates tool shapes based on task context and a controller policy that learns to manipulate the designed tools. While this work represents a significant step toward general tool-use capabilities in RL, it does not explicitly address the problem of discovering tool affordances through intrinsic interaction; an area this thesis aims to investigate.

Formally defining affordances allows us to precisely characterize what constitutes a tool in RL environments. In the proposed framework, tools are not merely objects that can be picked up, they are entities that expand the agent's action possibilities, allowing it to bring about state transitions that were otherwise inaccessible. This expansion of control space directly relates to empowerment. By quantifying empowerment-based affordances, this thesis offers a principled mechanism for identifying and comparing tools based on how

significantly they enhance the agent's ability to influence its environment.

2.4 Tool Characterisation and Classification

While much of the literature on tool use focusses on behavioural observations and affordance learning, an important parallel thread concerns the systematic characterization and classification of tools. These frameworks, often developed in anthropology and archaeology, aim to capture the diversity, complexity, and cognitive demands of tool use across cultures and time periods. Understanding how tools have been categorized historically provides a valuable foundation for constructing formal models of tool use in artificial agents.

2.4.1 Anthropological and Historical Perspectives on Tool Classification

Tool classification has long been an essential concern in both anthropology and ethnography, serving as a basis for understanding technological evolution, cultural transmission, and environmental adaptation. These classifications not only document the structural diversity of tools but also encode information about the social and ecological contexts in which they are embedded.

Johnston's comprehensive survey [66] outlines a broad typology of tool forms, tracing their development from primitive implements to more elaborate constructions. He emphasizes functional adaptation as a primary force in tool evolution, with changes in shape and composition often corresponding to shifts in economic or subsistence patterns. The paper organizes tools by broad operational categories, such as pounding, cutting, grinding, or perforating, while also noting their increasing structural complexity (e.g., from hand-held to hafted tools). In contrast, Healey [67] provides an in-depth ethnographic case study of the Maring people of Papua New Guinea. Healey shows how indigenous tool taxonomies are shaped not only by function and form but also by cultural meanings and usage contexts. The Maring distinguish cutting tools based on factors such as material (stone vs. metal), usage domain (gardening, hunting, ritual), and degree of modification.

A seminal contribution to the structural analysis of tools comes from Oswalt [68], who introduced the concept of *technounits* to describe the internal structure of artifacts. A technounit is defined as a "structurally distinct and functionally integrated component" of a tool that contributes to its overall function or form. Oswalt proposed a comparative methodology for analyzing food-getting technologies in hunter-gatherer societies by quantifying the number and arrangement of these technounits.

Building on Oswalt's foundational framework, later researchers formalized two key metrics to compare toolkits across cultural and environmental contexts [69]:

- **Tool richness**: The total number of distinct tool types present within a given toolkit or cultural context.
- **Tool complexity**: The average number of technounits per tool, reflecting internal structural elaboration.

These measures have been applied in several archaeological and ethnographic studies to investigate how ecological risk, mobility, or population size influence technological organization. For example, Collard et al. [69] analyzed North American Paleoindian projectile points and found that higher tool complexity correlated with more challenging environmental conditions, suggesting a relationship between tool design and adaptive problem-solving.

This quantitative framework offers a principled way to assess the cognitive demands of tool manufacture and use, as well as the diversity and sophistication of tool systems. It also serves as a useful precursor to computational formalizations of tools in artificial agents, where internal structure and function can be analogized to the agent's capacity for representing and sequencing sub-actions.

2.4.2 Toward Computational Models of Tool Classification

While anthropological and ethnographic classifications offer rich descriptive accounts of tool diversity, their frameworks are often grounded in culturally specific categories or structural typologies that may not directly translate into computational models suitable for artificial agents. In robotics and RL, agents interact with tools not through cultural knowledge but through physical interaction, learning, and environment-driven feedback. This requires a shift toward formal, behaviour-grounded models of tool classification that can capture the functional utility of tools in dynamic and task-general contexts.

One key distinction between natural and artificial agents lies in how tool properties are discovered and represented. Rather than classifying tools based on morphology or human-defined typologies, artificial agents typically acquire knowledge through interaction, exploration, and task-based learning. This echoes the behaviour-grounded approaches proposed in robotics research, where affordances are learned through the observation of action-outcome relationships [53, 54]. In this view, a tool can be classified not solely by its form but by the range of state transitions it enables when manipulated by the agent.

Building on these ideas, Sinapov et al. [70] proposed a computational model in which a robot incrementally learns a hierarchical taxonomy of outcomes produced when interacting with different tools. Rather than relying on static features or predefined categories, their model clusters observed environmental outcomes into hierarchically organized classes, allowing the robot to represent functional tool properties at varying levels of abstraction.

Each tool is thus characterized by the diversity and structure of the environmental effects it affords during interaction. The key insight of their framework is that tools can be classified functionally by analyzing similarities between their learned outcome taxonomies, rather than relying on predefined human categorizations. Two tools that induce similar environmental transformations are considered functionally similar, regardless of differences in visual appearance or design. Using this method, the robot was able to successfully differentiate among six tools (e.g., sticks, hooks, arrows) and identify functional equivalence across some of them based purely on interaction data. Importantly, their distance metrics between outcome taxonomies provide a quantitative, task-grounded basis for tool comparison, which is directly relevant to formalizing tool affordances in RL agents.

Causal modeling frameworks also offer a powerful avenue for formalizing tool use. By learning causal relations between actions, tools, and environment dynamics, agents can reason about which tools are likely to produce desired effects [60]. Such models can capture higher-level relational knowledge that supports tool substitution, meta-tool use, and generalized problem-solving.

Altogether, these interaction-driven and causal frameworks mark an important shift from static taxonomies toward grounded models that capture the functional role of tools in embodied action. In the context of this thesis, they offer a useful starting point for developing computational models that unify tool classification, affordance learning, and information-theoretic formalizations based on empowerment.

2.5 Intrinsic Motivations in Reinforcement Learning

RL provides a computational framework for studying how agents can learn to act optimally through interaction with their environment. An agent observes a state, selects an action, and receives feedback in the form of a reward signal that guides its learning process [11]. While RL has achieved impressive results in domains with well-defined external rewards, many real-world and biological settings lack such explicit feedback, making it difficult for agents to discover useful behaviours purely through extrinsic reinforcement. In natural organisms, motivation plays a central role in guiding behaviour even in the absence of external rewards. The concept of *intrinsic motivation*, originating in psychology and cognitive science [4, 5], refers to activities that are performed for their own sake. For example, out of curiosity, the desire for mastery, or the drive to reduce uncertainty. Intrinsic motivation has been proposed as a key mechanism underlying spontaneous exploration, play, and lifelong learning in humans and animals, which reflects the biological tendency to seek novelty, challenge, and competence. This perspective has inspired computational researchers to model such internal drives within RL, enabling artificial agents

to self-organize and acquire skills without requiring dense or pre-specified rewards. In RL, intrinsic motivation provides internal reward mechanisms that encourage agents to explore novel situations, acquire diverse experiences, and build rich models of their environment that generalize across tasks [17, 18]. These mechanisms integrate psychological aspects such as curiosity, surprise, or control into quantifiable signals that shape behaviour. This section reviews several prominent families of intrinsic motivation frameworks, leading into empowerment-based approaches that provide the theoretical foundation for the contributions of this thesis.

2.5.1 Curiosity-Driven Exploration

Curiosity represents one of the most fundamental forms of intrinsic motivation, observed across both biological and artificial agents. In psychology, curiosity is broadly defined as the drive to seek novel, surprising, or informative experiences [71]. It promotes exploration and cognitive development by motivating individuals to engage with stimuli that reduce uncertainty or yield new knowledge. In RL, curiosity-based methods operationalize this principle by rewarding agents for encountering states that deviate from their current expectations, thus encouraging exploration even in the absence of extrinsic rewards. A seminal contribution by Schmidhuber [72] formalised curiosity as the maximization of prediction improvement over time, where agents are driven to explore situations that yield learning progress rather than those that are either fully predictable or completely random. Subsequent work has refined this idea into a family of curiosity-driven algorithms that use prediction error as a proxy for novelty or surprise. Pathak et al. [7], for instance, proposed the Intrinsic Curiosity Module (ICM), in which agents learn a forward dynamics model to predict the next state given the current state and action. Intrinsic reward is computed as the discrepancy between the predicted and actual next state, incentivizing the agent to visit states where its predictive model performs poorly. Unlike learning-progress approaches, which reward improvements in prediction accuracy over time, curiosity-based rewards depend solely on instantaneous prediction error, capturing a form of surprise rather than competence development. This class of methods has proven particularly effective in sparse-reward environments and visually rich domains, such as Super Mario Bros [73] and VizDoom [74], where curiosity enables sustained exploration without external supervision.

While curiosity-driven approaches effectively promote exploration by encouraging agents to encounter surprising or poorly predicted states, they primarily focus on improving predictive models of the environment rather than explicitly quantifying the agent's causal influence over specific entities within it. Consequently, although such methods support broad exploration, they do not directly capture the structured, object-centred interactions that are central to modelling tool use in this thesis. However, curiosity-driven signals

can be seen as complementary to empowerment-based approaches, as they promote the discovery of novel states that may later be exploited through structured, control-oriented intrinsic objectives.

2.5.2 State Novelty and Count-Based Exploration

Another major family of intrinsic motivation methods focuses on quantifying *state novelty*. Whereas curiosity is typically defined as the drive to reduce prediction error or surprise based on an internal dynamics model, novelty-based approaches directly reward agents for visiting previously unseen or infrequently encountered states, regardless of any predictive model. In essence, curiosity seeks informational gain through learning, while novelty maximization promotes state-space coverage through exploration.

Bellemare et al. [75] introduced pseudo-count-based exploration, wherein agents receive larger rewards for visiting states that are rarely encountered, encouraging systematic coverage of the state space. Since exact state counting is infeasible in high-dimensional or continuous observation spaces, Tang et al. [76] proposed a *hashing*-based approximation. In this framework, states are embedded into lower-dimensional hash codes, allowing approximate visitation counts to guide exploration bonuses.

Building on these ideas, Burda et al. [77] proposed *Random Network Distillation* (RND), where the agent predicts the output of a fixed random neural network given its current observation. Here, the prediction error does not reflect surprise or learning progress as in curiosity, but instead measures how familiar a state is with respect to the agent's accumulated experience. Highly familiar states yield low intrinsic rewards, while novel or rarely visited observations lead to higher errors and thus stronger exploratory drives.

More recently, non-parametric density estimation techniques such as APT [78], and RE3 [79], have used k-nearest neighbor (KNN) approaches to estimate state density and novelty. These methods reward the agent for visiting states far from previously visited observations in the feature space, promoting deep exploration and sample-efficient policy learning.

Further advancements have introduced representation learning-based novelty measures. For instance, Proto-RL [80] and [81] enable robust state representations that enhance novelty estimation. These representation-driven methods provide more stable and scalable exploration signals, particularly in high-dimensional visual RL domains.

Novelty-based exploration methods therefore prioritise broad coverage of the state space by encouraging agents to visit previously unseen or infrequently encountered states. While this strategy is highly effective for exploration, it does not explicitly distinguish whether state changes arise from meaningful interactions with task-relevant objects or from inciden-

tal environmental variation. In contrast, the empowerment-based approach adopted in this thesis focuses on measuring the agent's causal capacity to influence specific entities within the environment, providing a more structured basis for modelling tool–object interactions. A comparison with novelty-driven exploration, specifically count-based methods, is later investigated experimentally in Section 5.2.2. In this sense, novelty-based exploration can complement empowerment by facilitating broad state-space coverage, increasing the likelihood of discovering task-relevant interactions that can subsequently be exploited through object-centred intrinsic signals.

2.5.3 Competence-Based Intrinsic Motivation

In contrast to curiosity- or novelty-driven exploration, which encourage agents to seek out surprising or previously unseen states, another family of intrinsic motivation methods focuses on *competence progress*, inspired by developmental learning observed in humans and animals. Here, competence refers to the agent's ability to successfully perform a given task or achieve a goal, while learning progress quantifies how this competence improves over time. These approaches allow agents to autonomously structure and sequence their own learning, gradually shifting attention toward tasks where their performance is improving most. This process, often described as *self-organized curriculum learning*, enables the agent to decide which goals to pursue at each stage of development instead of following a pre-defined training schedule.

Competence-based intrinsic motivations are typically explored in open-ended and multi-goal RL settings, where agents face a variety of tasks or parameterized goals. In such contexts, intrinsic rewards are derived from observable improvements in task success or skill performance over time. When an agent performs better at a particular goal than it did previously, this improvement signals learning progress and produces an intrinsic reward. Conversely, if no improvement occurs, the intrinsic motivation fades, encouraging the agent to explore other goals.

Oudeyer et al. [82] proposed the *Intelligent Adaptive Curiosity (IAC)* framework, an influential developmental model in which agents actively select tasks based on their learning progress, prioritizing goals that currently yield the greatest improvement in performance. This formulation operates within an open-ended learning setting, where multiple potential goals or skills exist, and the agent must decide autonomously which ones to learn next. The resulting mechanism allows agents to self-direct exploration toward learnable regions of their environment, avoiding both tasks that are too easy (already mastered) and those that are too difficult (unlearnable). Through this process, agents construct internal learning sequences, or *curricula*, mirroring how human infants gradually progress from simple to complex motor and cognitive abilities. Curriculum learning, in this sense, refers to the

progressive acquisition of skills in a structured order that facilitates continual learning and retention.

Building on this foundation, Forestier et al. [83] extended intrinsically motivated goal exploration to tool-use scenarios. Their agents selected object-manipulation tasks according to competence progress, demonstrating that complex tool-use behaviours, such as using one object to control another, can emerge autonomously without any predefined reward or task specification. This work is particularly relevant to the present thesis, as it connects intrinsic motivation with the emergence of tool-mediated behaviour.

A key computational architecture in this area is the Self-Adaptive Goal Generation—Robust Intelligent Adaptive Curiosity (SAGG-RIAC) model proposed by Baranes and Oudeyer [84]. In SAGG-RIAC, agents generate parameterized goals in a continuous task space and estimate competence progress locally for each region. They then adapt their exploration focus to areas where competence is improving most rapidly, effectively balancing exploration and exploitation. Unlike earlier models, SAGG-RIAC introduced a hierarchical mechanism for autonomously partitioning the goal space and adaptively allocating exploration resources, enabling more scalable and robust learning in high-dimensional sensorimotor domains. This approach has since been widely adopted in developmental robotics as a general framework for self-directed skill acquisition in continuous and redundant action spaces.

More recent work has further extended these ideas toward open-ended goal discovery in intrinsically motivated systems. For example, the Goal-discovering Robotic Architecture for Intrinsically motivated Learning (GRAIL) [85] enables agents to autonomously discover and explore goals through intrinsically motivated interaction, while subsequent extensions such as C-GRAIL [86] introduce context-dependent goal learning within a RL framework. More recent developments, such as H-GRAIL [87], further address the challenge of open-ended and hierarchical goal learning in more complex environments. These approaches collectively highlight the importance of interdependent goals and structured learning progress in autonomous skill acquisition.

Competence-based intrinsic motivation shares important similarities with the approach developed in this thesis, particularly in its emphasis on structured, goal-directed behaviour and the autonomous discovery of meaningful interactions. However, while competence-progress methods organise learning around externally defined or internally generated goals, the empowerment-based framework focuses instead on quantifying the agent's causal influence over the environment itself. In this sense, competence-based approaches and empowerment can be viewed as complementary: the former structures learning over goal spaces, while the latter provides a principled measure of controllability that can guide interaction with objects and tools.

2.5.4 Mutual Information-Based Intrinsic Motivation

A distinct class of intrinsic motivation methods leverages *mutual information (MI)* (see Chapter 3.2.3 for a full formalisation) as a principled, information-theoretic signal to drive exploration and skill acquisition. Within this family, empowerment can be understood as a special case, corresponding to the maximum mutual information between an agent's actions and its future environmental states. While MI is widely employed in representation learning [88], its use as an intrinsic motivation differs in purpose. It measures how much an agent's internal variables, rather than merely encoding compact features. For instance, its actions, states, or latent policies, inform or influence aspects of the environment. This formulation encourages controllable, diverse, and predictable interactions between the agent and its surroundings.

Unsupervised Behaviour and Skill Discovery via Mutual Information

Unsupervised behaviour and skill discovery methods aim to enable agents to autonomously acquire reusable behavioural primitives, such as options, skills, or policies, without relying on external rewards. By maximising internal criteria like diversity or controllability, agents learn a repertoire of behaviours that can later be composed to solve downstream tasks more efficiently. These approaches are particularly valuable in sparse-reward or open-ended settings, where the goal is to develop general-purpose competencies rather than task-specific policies.

A foundational contribution in this direction is Variational Intrinsic Control (VIC) [89], which introduced one of the earliest variational formulations of empowerment. VIC maximises the MI between the agent's internally selected option variable (representing a temporally extended action sequence) and the resulting state at the option's termination. This encourages the agent to learn diverse and reliably distinguishable options, effectively expanding its behavioural repertoire. In this framework, options can be viewed as meta-actions that abstract over multiple primitive steps, allowing the agent to control its environment at a higher temporal scale. Although VIC is often discussed within the context of skill discovery, its objective directly corresponds to empowerment, measuring how much influence the agent can exert over future states through extended actions.

Building on this idea, DIAYN ("Diversity is All You Need") [90] also employs a MI objective to discover distinct and controllable behaviours. Rather than modelling temporally bounded options, DIAYN introduces skills, policies conditioned on latent identifiers, such that each skill induces a unique distribution of visited states. By maximising the mutual information between the skill variable and the agent's trajectories, DIAYN produces a set of behaviours that are distinguishable based on their environmental outcomes.

Conceptually, while VIC emphasises control through temporally extended options, DIAYN focuses on learning diverse behaviours at the policy level; both, however, can be interpreted as empowerment-related formulations that aim to maximise the agent's influence over the environment in different ways. These approaches are therefore closely aligned with empowerment, differing primarily in how the underlying mutual information objective is structured and optimised.

Beyond VIC and DIAYN, several unsupervised skill-discovery methods pursue related goals with different training signals and structures. DADS [91] leverages a learned dynamics model and maximises a predictability/diversity objective over state transitions, yielding skills that produce reliably distinguishable dynamics. VALOR [92] frames option discovery via variational inference, learning latent-conditioned policies together with an inference model that identifies the latent from trajectories, thereby producing identifiable, reusable options. HIDIO [93] introduces a hierarchical scheduler–worker architecture for skill discovery, optimising diversity/entropy objectives to learn temporally coordinated skills over full trajectories. These approaches illustrate complementary design choices, such as model-based prediction (DADS), variational identification of options (VALOR), and hierarchical coordination (HIDIO), within the broader landscape of unsupervised behaviour acquisition.

Information Gain and Model Uncertainty Approaches

Another prominent line of work applies mutual information to model-based exploration by rewarding the agent for information gain about its learned environment model. In Variational Information Maximizing Exploration (VIME) [9], agents receive intrinsic rewards proportional to the reduction in uncertainty about environment dynamics after each transition, quantified via mutual information between actions and model parameters. This encourages agents to explore transitions that provide the greatest learning signal for their predictive models.

Similarly, Exploration with Mutual Information (EMI) [94] estimates the MI between latent state representations and action sequences, guiding the agent toward behaviourally informative regions of the environment. These methods reward transitions that improve the agent's understanding of either the environment's structure or its own controllable behaviours.

Mutual Information State Intrinsic Control

A more recent direction has focused on maximizing MI between the agent's own state and the surrounding environment, leading to efficient intrinsic control behaviours. MUSIC (Mutual Information State Intrinsic Control) [25] formalises this idea by decomposing the

overall system state into two components: the agent's internal or controllable part (e.g., body configuration, effector states) and the external or environmental part (e.g., objects, surroundings). The agent is intrinsically rewarded for increasing the MI between these two components, thereby learning behaviours that maximise the mutual dependence between its own configuration and its environment. In practical terms, this means that the agent is encouraged to discover behaviours that reliably influence its surroundings, such as pushing, grasping, or manipulating objects, without any extrinsic rewards. MUSIC has demonstrated the emergence of complex skills such as pick-and-place behaviours in robotic environments without any extrinsic rewards.

Conceptually, MUSIC shares close ties with empowerment, as both aim to quantify an agent's potential influence over its environment through MI. While empowerment measures the MI between actions and future states, MUSIC focuses instead on MI between the agent's and environment's state components, offering a more tractable and scalable formulation for continuous-control tasks. This makes MUSIC particularly compatible with empowerment-based formulations, as both seek to maximise the agent's ability to influence its environment, albeit through different decompositions of the underlying system. This direct coupling between intrinsic motivation and environmental interaction makes MUSIC particularly suitable for domains where object manipulation and causal control are central.

These MI-based approaches highlight the importance of controllability and behavioural diversity as intrinsic drivers of learning. Many of these formulations are closely related to empowerment, differing primarily in how the mutual information objective is defined (e.g., actions vs. states vs. latent variables) and optimised in practice. While most methods focus on discovering reusable behaviours or improving predictive models of interaction, they do not explicitly structure the agent's influence over specific entities within the environment. In contrast, the approach developed in this thesis builds on the empowerment principle to quantify object-specific causal influence, enabling intrinsic evaluation of tool functionality through agent–tool–object interactions. Many of these MI-based frameworks share a close conceptual connection to empowerment, which itself is the maximum of the MI between the agent's actions and its future environmental states. In the next section, this thesis builds on this connection to review empowerment-based approaches in RL, which serve as the theoretical foundation for the contributions of this thesis.

2.5.5 Empowerment-Based Intrinsic Motivation

Among information-theoretic intrinsic motivation frameworks, *empowerment* has emerged as a particularly principled and general measure. Empowerment formalizes the agent's intrinsic drive to maximize its potential influence over the environment by quantifying the mutual information between its actions and the resulting future states (see Chapter 3.3 for

full formalisation). Rather than focusing on novelty, surprise, or external goal achievement, empowerment rewards agents for seeking states where they retain maximal optionality and control over (distinguishable) future outcomes.

The concept was originally introduced by Klyubin et al. [1, 10], who proposed that intelligent agents may act to place themselves in states that maximize the diversity of reachable and controllable future states. This perspective draws inspiration from biological systems, where organisms often seek to maintain high flexibility and adaptability in uncertain environments. Empowerment thereby serves as a task-independent, unsupervised measure of agency.

Later work extended the empowerment framework to continuous domains and high-dimensional problems [95, 96]. These papers introduced Monte Carlo sampling techniques to approximate the empowerment value in settings where exact computation of MI was infeasible. These studies marked an important step toward applying empowerment beyond small discrete environments, demonstrating how sampling-based estimation could preserve its information-theoretic interpretation in more realistic control problems. Later, Salge et al. [97] provided a comprehensive overview of these approaches, discussing analytical, sampling-based, and approximation techniques within a unified framework. Mohamed and Rezende [21] subsequently proposed a scalable variational information maximisation approach that enabled empowerment estimation in deep RL settings by representing both the policy and dynamics model with neural networks. This variational formulation made empowerment computationally tractable in complex, high-dimensional visual domains, effectively bridging classical information-theoretic formulations with modern deep RL. More recent work, such as Bharadhwaj et al. [22], has further explored empowerment in visual model-based RL, reinforcing its relevance for agents that must learn structured representations of controllability in complex environments.

Choi et al. [98] investigated empowerment for representation learning, proposing an empowerment-regularised objective that yields state abstractions aligned with controllability (i.e., preserving the agent's ability to influence future states). Zhao et al. [23] introduced an efficient online estimator for empowerment that scales to longer horizons via a variational formulation implemented with neural networks, making computation practical in complex domains. Related work has integrated empowerment with causal modelling to adapt intrinsic rewards toward interventions that produce predictable, high-impact changes in the environment [99, 100]. In human-centric settings, empowerment has been used to guide human-assistive agents without inferring explicit goals [101, 102]. These advances highlight the growing versatility of empowerment in tackling real-world learning challenges.

Beyond its theoretical appeal, empowerment has been empirically validated as an effective intrinsic drive in sparse-reward settings. Massari et al. [103] provided experimental

evidence that empowerment-based exploration can drive efficient learning even in the absence of external rewards. Recently, the integration of empowerment into hierarchical RL frameworks has further expanded its applicability by proposing *Hierarchical Empowerment* [104].

Empowerment has also been successful for robotics applications. For instance, Dai et al. [105] proposed an empowerment-based solution for manipulation tasks under sparse reward settings. Cao et al. [106] recently extended empowerment into a causality-aware framework for embodied agents. Their *Causal Action Empowerment (CAE)* method integrates causal structure learning with empowerment to identify controllable state variables, prioritize high-impact actions, and improve sample efficiency. Empowerment has also been explored in multi-agent coordination and communication scenarios [107], as well as social navigation [108].

In the context of skill learning, [109] demonstrates how empowerment can be leveraged to acquire skills in RL tasks. Recent work by Lidayan et al. [110] demonstrates how empowerment-driven exploration enables agents to autonomously discover diverse tool-like interaction strategies in open-ended simulated environments, further underscoring the suitability of empowerment for structured, multi-object interaction tasks.

Empowerment differs from many other intrinsic motivation frameworks in that it explicitly models controllability and influence rather than novelty or surprise. This makes it particularly suitable for domains where agents interact with structured environments containing tools and manipulable objects, and where the objective is not merely exploration but the maintenance and expansion of the agent's capacity to act. The work presented in this thesis builds directly on empowerment-based intrinsic motivation by extending it to explicitly model agent–object interactions through *object empowerment*. In this formulation, empowerment is conditioned on the state of specific manipulable entities within the environment. This allows artificial agents to evaluate the functional utility of tools by quantifying their potential influence over task-relevant objects. This object-centred perspective provides a principled information-theoretic basis for modelling tool use, tool affordances, and flexible problem-solving in reinforcement learning systems.

2.5.6 Intrinsic Motivation Applied to Tool Use

While most intrinsic motivation frameworks are formulated in general-purpose RL settings, few studies have specifically explored their application to tool-use learning. These works demonstrate how intrinsic motivation can drive exploration and facilitate the acquisition of tool-related skills in the absence of dense external supervision.

Seepanomwan et al. [111] proposed an intrinsic motivation architecture for a humanoid

robot tasked with learning new motor skills that enable tool use. In their setup, the robot first autonomously explores motor behaviours using intrinsic rewards based on novelty and competence progress, gradually acquiring control over a ball-on-table manipulation task that required tool use. Once a sufficient skill repertoire was acquired, a goal-directed phase allowed the robot to exploit these learned behaviours for task completion.

Building upon this, Seepanomwan et al. [112] studied tool-use development using a simulated iCub humanoid robot. They compared two hypotheses: one focusing on the gradual development of goal-directed planning abilities, and another emphasizing intrinsic motivation as the driver for self-organized acquisition of affordances, skills, and forward models. Their results showed that intrinsic motivation could account for the rapid emergence of tool-use abilities, supporting the critical role of intrinsic motivation in tool-use development.

Forestier et al. [113] further extended these ideas by introducing an intrinsically motivated goal exploration architecture where agents select multiple learning goals autonomously. Crucially, knowledge acquired while exploring one goal (e.g., learning to use a tool) can be transferred to facilitate learning of related goals (e.g., solving tasks that require tool manipulation). This transfer of experience across multiple interconnected goals represents a critical prerequisite for developing more general tool-use competencies.

Taken together, these studies highlight how intrinsic motivation, through mechanisms such as competence progress and novelty-driven exploration, can serve as an effective driver for acquiring tool-use behaviours. These studies suggest that combining intrinsic motivation with goal exploration, competence-based learning, and planning mechanisms can support the emergence of increasingly flexible and adaptive tool-use capabilities. Importantly, many of these approaches operate in multi-goal settings where learning one objective can influence or facilitate progress on others, reflecting forms of interdependent goal learning [82, 84–87, 113]. In such settings, tool use can be understood as a structured form of skill acquisition involving interdependent sub-goals, for example, acquiring a tool before using it to manipulate an object. The approach developed in this thesis aligns with this perspective, but provides a distinct formulation in which such interdependencies emerge through object-centred empowerment. Rather than relying on predefined or externally parameterised goals, object empowerment intrinsically guides the agent toward sequences of actions that increase its causal influence over task-relevant objects, thereby enabling the autonomous resolution of interdependent goals in tool-use scenarios. This thesis builds on these foundations by introducing an information-theoretic intrinsic motivation framework based on empowerment, namely *object empowerment*, allowing artificial agents to evaluate the functional utility of tools via their capacity to influence task-relevant objects during interaction.

2.6 Hierarchical Reinforcement Learning and Tool Abstraction

Beyond intrinsic motivation frameworks, another influential line of research for structuring complex behaviour in reinforcement learning is hierarchical reinforcement learning (HRL). HRL addresses the challenge of long-horizon decision-making by introducing temporal abstraction and multi-level policy structures. Rather than selecting only primitive actions, HRL methods define higher-level entities such as options, subpolicies, or managers that operate over extended time scales [114, 115]. These abstractions allow agents to decompose complex tasks into structured components and to reuse learned behaviours across contexts.

The options framework [114] formalises temporally extended actions as policies with initiation conditions and termination rules. More recent approaches, such as the Option-Critic architecture [116], learn such abstractions end-to-end, while architectures like FeUdal Networks [117] and HIRO [118] employ manager–worker decompositions to learn hierarchical control policies. These methods aim to improve exploration efficiency and credit assignment in environments with sparse rewards.

From the perspective of tool use, HRL offers a natural interpretation in which tools may correspond to temporally extended skills or subpolicies that achieve specific environmental transformations. In this sense, HRL provides a mechanism for structuring behaviour into reusable components that can capture multi-step interactions, including those involving tools.

The approach developed in this thesis focuses on a complementary aspect, namely the intrinsic evaluation of agent–environment interactions in terms of object-level causal influence. By extending empowerment to quantify such influence, the framework provides a measure of how actions affect specific objects within the environment. While HRL structures behaviour through hierarchical policies, object empowerment provides an intrinsic signal that can guide which interactions are meaningful or effective. These perspectives can be viewed as complementary, with HRL addressing how behaviours are organised and executed, and empowerment addressing how their effects are intrinsically evaluated.

Chapter 3

Theoretical and Technical Background

This chapter presents the foundational concepts that underpin the development of the empowerment-based RL approaches explored in subsequent chapters. It begins with a formal introduction to the RL paradigm, which frames learning as a sequential decision-making process driven by trial-and-error interactions with an environment. Core elements such as Markov Decision Processes (MDPs), value functions, and the exploration–exploitation trade-off are discussed to establish a principled understanding of the RL framework. The chapter also introduces the main families of RL algorithms, including value-based methods (e.g., Q-learning), policy-based methods (e.g., REINFORCE [119]), and actor–critic approaches, which combine both paradigms. Special emphasis is given to widely used actor–critic variants such as Advantage Actor-Critic (A2C) [120] and Proximal Policy Optimization (PPO) [121], which are employed in the experiments as baselines in sparse-reward environments.

The chapter then shifts focus to the role of *intrinsic motivation* in tackling sparse or delayed reward scenarios. In this context, reward regularisation techniques are introduced as mechanisms to integrate task-agnostic objectives, such as novelty, curiosity, and empowerment, into the learning process. These methods serve to encourage exploration and facilitate the discovery of meaningful behavior in the absence of dense external feedback.

To support a formal treatment of the intrinsic motivation empowerment, the chapter introduces key information-theoretic concepts, including entropy, mutual information, and channel capacity. These concepts form the basis for defining empowerment as a measure of an agent's potential influence over its future observations. The action–perception loop is formalised from an information-theoretic standpoint, leading to the definition of empowerment and its variants for both stochastic and deterministic environments.

3.1 Reinforcement Learning

Reinforcement learning (RL) provides a mathematical framework for modeling sequential decision-making problems where an agent interacts with an environment to achieve a goal [11]. At each time step, the agent observes the current state of the environment, selects an action, receives feedback in the form of rewards, and transitions to a new state as shown in Figure 3.1 Over time, the agent aims to learn a policy that maximizes its cumulative reward.

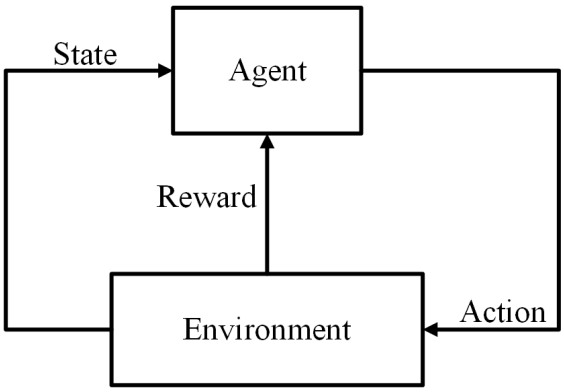


Figure 3.1: Schematic of the RL interaction loop between agent and environment. At each time step, the agent is in a state, selects an action, receives a reward, and transitions to a new state.

3.1.1 Markov Decision Processes

Formally, RL problems are modeled as Markov Decision Processes (MDPs) [122], defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where:

- $\mathcal{S}$ is the set of possible states that describe the environment.
- $\mathcal{A}$ is the set of possible actions available to the agent.
- $P(s'|s, a)$ is the transition probability distribution, specifying the probability of transitioning to state $s' \in \mathcal{S}$ when taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$.
- $R(s, a)$ is the reward function, providing a real-valued scalar feedback (which can be continuous and negative) for performing action a in state s.
- $\gamma \in (0, 1]$ is the discount factor, determining how future rewards are weighted relative to immediate rewards.

At each discrete time step t, the agent is in the current state s_t, selects an action a_t, receives a reward $r_t = R(s_t, a_t, s_{t+1})$, and transitions to a new state $s_{t+1} \sim P(\cdot|s_t, a_t)$. The agent's behavior is governed by a policy $\pi(a|s)$, which defines the probability distribution over actions conditioned on the current state.

3.1.2 Objective and Return

The goal of the agent is to learn a policy that maximizes the expected cumulative discounted reward [11]. The cumulative discounted reward G_t, or return, at time t is defined as follows:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k} \tag{3.1.1}$$

The expected return under a policy π is called value function [11]:

$$V^\pi(s) = \mathbb{E}_\pi \left[G_t \,|\, s_t = s \right] \tag{3.1.2}$$

which measures the expected cumulative reward when starting from state s and following policy π thereafter.

Similarly, the state-action value function, or Q-function, is defined as [11]:

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[G_t \,|\, s_t = s, a_t = a \right] \tag{3.1.3}$$

The objective of RL is to find an optimal policy π^* that maximizes the expected return from any initial state.

3.1.3 Main Families of RL Algorithms

RL algorithms can broadly be categorized into three main families: value-based methods, policy-based methods, and actor-critic methods. Each family offers unique strategies for learning optimal behaviors in sequential decision-making problems.

- **Value-based Methods:** These approaches focus on estimating value functions, such as the state-value function $V^\pi(s)$ or the action-value function $Q^\pi(s, a)$, and derive policies by acting greedily with respect to these estimates. Classical algorithms include Q-learning and Deep Q-Networks (DQN) [15], which approximate Q-values using deep neural networks.
- **Policy-based Methods:** These methods directly learn a parameterized policy $\pi_\theta(a \mid s)$ without explicitly estimating value functions. Policy gradients are computed using

algorithms like REINFORCE [119], which optimize the expected return by adjusting the parameters θ via gradient ascent.

- **Actor-Critic Methods:** These combine the strengths of both value-based and policy-based methods. The actor updates the policy based on feedback from the critic, which evaluates actions by estimating value functions. This family includes algorithms such as Advantage Actor-Critic (A2C) [120, 123] and Proximal Policy Optimization (PPO) [121], which improve learning stability and efficiency.

3.1.4 Challenges in RL

Despite the remarkable progress achieved in recent years, RL continues to face several enduring challenges that limit its scalability, reliability, and applicability to complex real-world domains. One of the most prominent difficulties is *sample inefficiency*: many RL algorithms require a large number of interactions with the environment to learn effective policies [15]. This issue becomes particularly acute in high-dimensional continuous control or sparse-reward settings, where informative feedback is rare. A second core challenge concerns the *exploration–exploitation trade-off* [11]. Striking a balance between exploring new strategies and exploiting known rewarding behaviors is a persistent challenge. Ineffective exploration can lead to suboptimal policies. The *credit assignment problem* further complicates learning in long-horizon tasks, where delayed rewards make it difficult to infer which past actions were responsible for success or failure [124]. Generalization is also a persistent challenge: agents trained in one environment often fail to adapt to even slightly perturbed conditions or unseen configurations [125]. This limits transfer and reusability of learned behaviors. Finally, *sparse and delayed rewards* represent one of the hardest obstacles in RL [17, 18]. When external feedback is rare, agents struggle to discover useful behaviors without additional guidance. To address this, auxiliary objectives, reward shaping, or intrinsic motivation mechanisms (e.g., empowerment) are often introduced to encourage exploration and provide denser learning signals.

3.1.5 RL Algorithms Used in This Thesis

The experimental studies in this thesis rely on policy-gradient methods, particularly the A2C and PPO. These approaches were selected for their conceptual clarity, ease of integration with intrinsic-motivation signals such as empowerment, and their strong empirical performance in both discrete and continuous control domains. This section briefly outlines their theoretical formulation and training dynamics, before they are applied as baseline learning mechanisms in the experiments.

Advantage Actor-Critic (A2C)

The *Advantage Actor-Critic (A2C)* algorithm [120, 123] is a synchronous variant of the actor-critic family of RL algorithms, and serves as one of the baseline methods used in this thesis. It combines the strengths of both value-based and policy-based RL. It uses two neural networks: an actor, which learns the policy $\pi_\theta(a \mid s)$, and a critic, which estimates the value function $V^\pi(s)$. The overall structure is illustrated in Fig. 3.2.

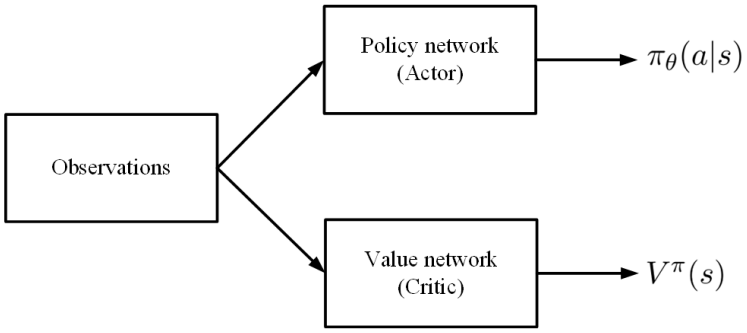


Figure 3.2: A2C architecture where the actor outputs the policy $\pi_\theta(a \mid s)$ and the critic outputs the state value $V^\pi(s)$.

Unlike REINFORCE [119], a pure policy-gradient method that suffers from high variance due to its reliance on Monte Carlo estimates, A2C incorporates a value baseline to stabilize learning. Specifically, it estimates the *advantage function* $A^\pi(s_t, a_t)$, which measures how much better or worse an action is compared to the average action at state s_t:

$$A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t) \tag{3.1.4}$$

Here, the action-value function $Q^\pi(s_t, a_t)$ is approximated using n-step returns, denoted by $\tilde{Q}^\pi(s_t, a_t)$ and defined as:

$$\tilde{Q}^\pi(s_t, a_t) = \sum_{k=0}^{n-1} \gamma^k r_{t+k+1} + \gamma^n V^\pi(s_{t+n}) \tag{3.1.5}$$

This estimate aggregates the immediate rewards $r_{t+1}, r_{t+2}, \ldots$ obtained after taking action a_t in state s_t, followed by following policy π. It serves as a finite-horizon surrogate for the true return G_t, without implying equality.

The loss for the critic is computed using the mean squared error between the predicted state value $V^\pi(s_t)$ (from the critic network) and the bootstrapped target $\tilde{G}_t$ (computed from sampled trajectories):

$$L_{\text{val}}(\theta_C) = \frac{1}{T}\sum_{t=0}^{T}\left(V^{\pi}(s_t) - \tilde{G}_t\right)^2 \tag{3.1.6}$$

The policy loss for the actor follows directly from the policy gradient theorem, where $\log \pi_{\theta_A}(a_t \mid s_t)$ encourages the policy to increase the probability of actions with positive advantage and decrease it otherwise:

$$L_{\text{pol}}(\theta_A) = -\frac{1}{T}\sum_{t=0}^{T} A^{\pi}(s_t, a_t) \log \pi_{\theta_A}(a_t \mid s_t) \tag{3.1.7}$$

At each time step t, the agent observes state s_t and selects an action a_t from its policy network. After executing the action, it receives a reward R_t and the next state s_{t+1}. These interactions form trajectories (s_t, a_t, R_t, s_{t+1}) used to update both actor and critic networks. The following algorithm summarizes the A2C update process:

Algorithm 1 Advantage Actor-Critic (A2C)

Initialize actor learning rate α_A, critic learning rate α_C
Initialize actor parameters θ_A, critic parameters θ_C
for each episode **do**
    Collect trajectory $\{(s_t, a_t, R_t, s_{t+1})\}_{t=0}^{T}$ using current policy π_{θ_A}
    for each step $t = 0$ to T **do**
        Estimate value $V^{\pi}(s_t)$ using critic network θ_C
        Estimate bootstrapped return $\tilde{G}_t = \sum_{k=0}^{n-1} \gamma^k r_{t+k+1} + \gamma^n V^{\pi}(s_{t+n})$
        Compute advantage: $A^{\pi}(s_t, a_t) = \tilde{G}_t - V^{\pi}(s_t)$
    end for
    Compute value loss: $L_{\text{val}}(\theta_C) = \frac{1}{T}\sum_{t=0}^{T}(V^{\pi}(s_t) - \tilde{G}_t)^2$
    Compute policy loss: $L_{\text{pol}}(\theta_A) = -\frac{1}{T}\sum_{t=0}^{T} A^{\pi}(s_t, a_t) \log \pi_{\theta_A}(a_t \mid s_t)$
    Update critic: $\theta_C \leftarrow \theta_C - \alpha_C \nabla_{\theta_C} L_{\text{val}}$
    Update actor: $\theta_A \leftarrow \theta_A - \alpha_A \nabla_{\theta_A} L_{\text{pol}}$
end for

Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) [121] is a popular on-policy reinforcement learning algorithm that builds upon the actor-critic framework, introducing stability and robustness in policy updates through a surrogate objective function with a clipping mechanism. PPO is widely used in deep RL due to its balance of implementation simplicity, sample efficiency, and reliable performance across diverse environments.

Similar to A2C, PPO maintains two networks: an actor that represents the policy $\pi_\theta(a \mid s)$ and a critic that estimates the value function $V^{\pi}(s)$. However, unlike traditional

policy gradient methods, PPO avoids large and potentially destructive policy updates by penalizing deviations from the old policy during optimization. This is achieved using a clipped objective function that restricts the policy update within a small trust region.

Let the probability ratio between the new policy π_θ and the old policy $\pi_{\theta_{\text{old}}}$ be defined as:

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \tag{3.1.8}$$

The surrogate objective function used in PPO is given by:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta) A_t,\, \text{clip}\big(r_t(\theta), 1-\epsilon, 1+\epsilon\big) \cdot A_t \right) \right] \tag{3.1.9}$$

Here, the expectation $\mathbb{E}_t$ denotes an average over time steps in a batch of trajectories. The term A_t represents the advantage estimate at time step t, and ϵ is a hyperparameter (typically between 0.1 and 0.3) that determines the clipping range. The function $\text{clip}(r, 1-\epsilon, 1+\epsilon)$ truncates the value of r so that it remains within the interval $[1-\epsilon, 1+\epsilon]$. The min operator in the objective ensures that the update remains conservative: if the policy update would lead to an overly large policy change (reflected by a large value of $r_t(\theta)$), the clipped version is used instead. This mechanism limits the extent to which the new policy can deviate from the previous one, thereby promoting training stability.

The overall PPO objective combines the policy loss L^{CLIP}, value function loss, and an entropy bonus to encourage exploration:

$$L^{\text{PPO}}(\theta) = L^{\text{CLIP}}(\theta) - c_1 L^{\text{val}}(\phi) + c_2 \mathcal{H}[\pi_\theta] \tag{3.1.10}$$

Here, $L^{\text{val}}(\phi)$ is the mean squared error loss for the critic, and $\mathcal{H}[\pi_\theta]$ is the entropy of the policy used to promote exploration. Coefficients c_1 and c_2 balance the relative importance of the different loss components.

The entropy bonus encourages the policy to remain stochastic and is computed as:

$$\mathcal{H}[\pi_\theta] = \mathbb{E}_{a \sim \pi_\theta} \left[-\log \pi_\theta(a \mid s) \right] \tag{3.1.11}$$

where higher entropy corresponds to more exploratory behavior, preventing premature convergence to suboptimal deterministic policies.

At each iteration, PPO collects trajectories using the current policy, computes advantage estimates (often using Generalized Advantage Estimation (GAE) [126]), and performs

multiple epochs of stochastic gradient updates using mini-batches. Although the surrogate objective function in Equation 3.1.9 is maximized in principle (i.e., gradient ascent), in practice it is often implemented as a minimization problem by negating the objective and applying gradient descent. The key difference from A2C lies in the clipped surrogate objective and repeated updates using the same batch of data.

In this thesis, PPO is used as the main learning algorithm for training agents across several experiments. Although the grid-world environments are relatively small, neural network function approximators are retained as part of the standard implementations in both RLlib [127] and Stable-Baselines3 [128], which offer efficient parallelism and reproducibility across custom environments.

The following algorithm outlines the core PPO update procedure:

Algorithm 2 Proximal Policy Optimization (PPO)

Initialize actor learning rate α_θ, critic learning rate α_ϕ
Initialize actor parameters θ, critic parameters ϕ
for each iteration **do**
    Collect trajectories $\{(s_t, a_t, R_t, s_{t+1})\}$ using current policy π_θ
    Estimate bootstrapped returns $\tilde{G}_t$ and advantages A_t using GAE [126]
    Compute old log probabilities: $\log \pi_{\theta_{\text{old}}}(a_t \mid s_t)$
    for each epoch **do**
        for each minibatch **do**
            Compute ratio: $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$
            Compute clipped surrogate loss $L^{\text{CLIP}}(\theta)$ using Equation 3.1.9
            Compute value loss: $L^{\text{val}}(\phi) = \left(V^\pi(s_t) - \tilde{G}_t\right)^2$
            Compute entropy bonus: $\mathcal{H}[\pi_\theta] = -\sum_a \pi_\theta(a \mid s_t) \log \pi_\theta(a \mid s_t)$
            Update critic: $\phi \leftarrow \phi - \alpha_\phi \nabla_\phi L^{\text{val}}(\phi)$
            Update actor: $\theta \leftarrow \theta + \alpha_\theta \nabla_\theta \left(L^{\text{CLIP}}(\theta) + c_2 \mathcal{H}[\pi_\theta]\right)$
        end for
    end for
end for

3.1.6 Addressing Sparse Rewards through Intrinsic Motivation

Sparse rewards are a major obstacle in RL, especially in complex environments where agents receive little or no feedback for long sequences of actions. In such cases, conventional reward-driven exploration often fails, as agents struggle to discover strategies that yield meaningful extrinsic rewards. To address this, many works introduce intrinsic motivation mechanisms that augment extrinsic rewards with exploration signals that are largely

independent of the external task. These mechanisms are designed to encourage the agent to seek novel or unpredictable experiences, often quantified through measures of novelty [8], curiosity [7], information gain [9], or empowerment [10]. Such quantities can serve as *reward regularisers* that bias the agent towards behaviors that improve long-term exploration or competence.

A common method to incorporate intrinsic motivation is through *reward augmentation*, where the extrinsic reward R from the environment is combined with an intrinsic signal M, yielding a regularised reward $\hat{R}$:

$$\hat{R} \coloneqq R + \beta M, \tag{3.1.12}$$

where β is a weighting coefficient that determines the influence of the intrinsic reward on learning. This formulation has been widely adopted across several streams of exploration research as follows. The regulariser M can take various forms:

- **Intrinsic Curiosity Module (ICM)** rewards the agent for encountering outcomes that its internal predictive model fails to anticipate, thereby encouraging exploration of unfamiliar dynamics. The forward model, trained jointly with the policy, predicts the next state given the current state and action, and the prediction error serves as the intrinsic reward [7].
- **Count-Based Exploration** encourages the agent to visit novel regions of the state space by assigning higher intrinsic rewards to states that have been encountered less frequently. In high-dimensional environments, pseudo-count estimators are used to approximate the notion of novelty [75].
- **Variational Information Maximizing Exploration (VIME)** promotes curiosity-driven behavior by rewarding transitions that reduce the agent's uncertainty about the environment dynamics. In practice, this is implemented within a model-based RL framework, where the intrinsic reward is proportional to the information gain about the model's parameters [9].
- **Empowerment** encourages exploration toward states where the agent has maximal potential influence over its future. It measures the channel capacity, i.e., the maximum possible mutual information between actions and successor states, thus driving the agent to seek controllable and information-rich regions of the environment [1].

These intrinsic signals are often added at the reward level (as in Equation 3.1.12), but can also appear as auxiliary losses during training, or influence policy regularisation

directly (e.g., entropy regularization [129] or Kullback–Leibler (KL)-divergence penalties in PPO [121]).

This thesis adopts an information-theoretic intrinsic reward based on empowerment, which measures the agent's potential to influence future states through its actions. This intrinsic signal serves as a dense and task-independent guide in environments where extrinsic rewards are sparse. This thesis integrates empowerment into the reward structure using Equation 3.1.12, encouraging the agent to navigate toward states with high controllability and utility. Object empowerment is introduced and utilised as a novel extension of empowerment that quantifies the agent's potential to influence specific target objects in the environment. This object-centric variant proves particularly effective in tool-based scenarios, where interactions are mediated through tools and directed at manipulable objects. The regularisation strength of both empowerment forms is controlled by the coefficient β, which is tuned depending on the environment and task. By integrating traditional empowerment and object empowerment into the learning process, this work aims to promote structured exploration and accelerate the acquisition of tool-use behaviours, even in the absence of immediate external feedback.

3.2 Information-Theoretic Foundations

Information theory [130], pioneered by Claude Shannon in 1948 [131], provides a mathematical framework for quantifying uncertainty, measuring information, and analyzing the capacity of communication channels. Originally developed to study efficient signal transmission, its concepts have since found widespread application in statistics, machine learning, artificial intelligence, and many other fields. In the context of RL, information theory forms the basis for several intrinsic motivation frameworks [9, 25, 90], including empowerment [21]. This section introduces the core information-theoretic notions that underpin the development of the empowerment-based approaches presented in the subsequent chapters.

3.2.1 Entropy

Entropy is a measure of the uncertainty associated with a random variable. Let X be a discrete random variable with probability mass function $p(x)$. The entropy $H(X)$ is defined as:

$$H(X) = -\sum_{x} p(x) \log p(x) \tag{3.2.1}$$

Entropy measures the expected number of bits required to optimally encode the out-

come of X under its probability distribution. It represents the average level of *unpredictability* of the variable's outcomes. That means higher entropy implies that the outcome of X is more uncertain, while lower entropy indicates that it is more predictable. Entropy is always non-negative ($H(X) \geq 0$) and reaches its minimum value of 0 when the outcome of X is certain (i.e., $p(x) = 1$ for one value of x). It attains its maximum value when all outcomes are equally likely, that is, for a uniform distribution over n possible outcomes, $H(X) = \log n$ bits.

3.2.2 Conditional Entropy

The conditional entropy $H(Y|X)$ quantifies the remaining uncertainty about a discrete random variable Y given that the value of another discrete random variable X is known. Let $p(x, y)$ denote their joint probability distribution and $p(y|x)$ the corresponding conditional probability. The conditional entropy is defined as:

$$H(Y|X) = -\sum_{x} p(x) \sum_{y} p(y|x) \log p(y|x). \tag{3.2.2}$$

Intuitively, $H(Y|X)$ measures the expected uncertainty of Y averaged over all possible values of X. If X completely determines Y, then $H(Y|X) = 0$; if X provides no information about Y, then $H(Y|X) = H(Y)$. Thus, conditional entropy expresses how much uncertainty about Y remains after X is known.

3.2.3 Mutual Information

Mutual information captures how much knowing one random variable reduces uncertainty about another. Formally, for discrete random variables X and Y with joint probability distribution $p(x, y)$ and marginals $p(x)$ and $p(y)$, mutual information is defined as:

$$I(X;Y) = \sum_{x} \sum_{y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{3.2.3}$$

The mutual information $I(X;Y)$ between two variables X and Y is also expressed as:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X). \tag{3.2.4}$$

It quantifies the amount of information shared between X and Y. Mutual information is always non-negative and equals zero if and only if X and Y are statistically independent.

Figure 3.3 provides an intuitive Venn diagram-based representation of the relationship between entropy, conditional entropy, and mutual information. This visualisation helps convey how information is shared between two random variables and how uncertainty is

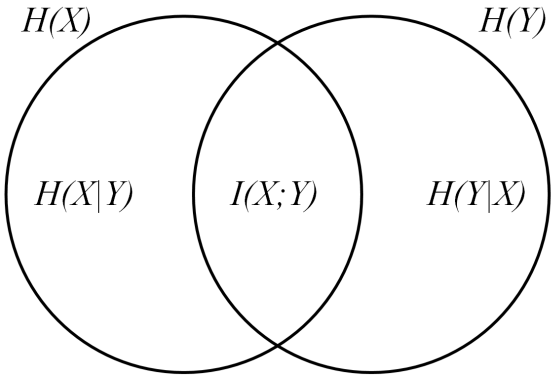


Figure 3.3: Venn diagram representation of the relationship between entropies $H(X)$, $H(Y)$, conditional entropies $H(X|Y)$, $H(Y|X)$, and mutual information $I(X;Y)$. The overlapping region represents the mutual information—the reduction in uncertainty of one variable given the other.

partitioned among their joint and individual components. The total uncertainty of random variables X and Y are represented by the circles $H(X)$ and $H(Y)$, respectively. The intersection between the two circles corresponds to the mutual information $I(X;Y)$, which represents the portion of uncertainty that is common to both variables. The non-intersecting regions correspond to the conditional entropies $H(X|Y)$ and $H(Y|X)$, indicating the residual uncertainty in each variable once the other is known.

3.2.4 Conditional Mutual Information

While mutual information measures the overall dependency between two variables, conditional mutual information (CMI) quantifies the mutual dependence between two variables given knowledge of a third. Formally, the conditional mutual information between X and Y given Z is defined as:

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z) \tag{3.2.5}$$

This measures how much knowing Y helps reduce uncertainty about X when Z is already known. It reflects the additional information that Y provides about X, beyond what is already contained in Z.

Figure 3.4 visualizes the relationships among entropies, conditional entropies, and CMI in a three-variable setting. For example, the fact that $I(X;Y|Z)$ represents the amount of information shared between X and Y once Z is known is reflected in the intersection between X and Y excluding any shared information with Z. The central intersection, labeled $I(X;Y;Z)$, denotes the "multivariate mutual information" or "interaction information",

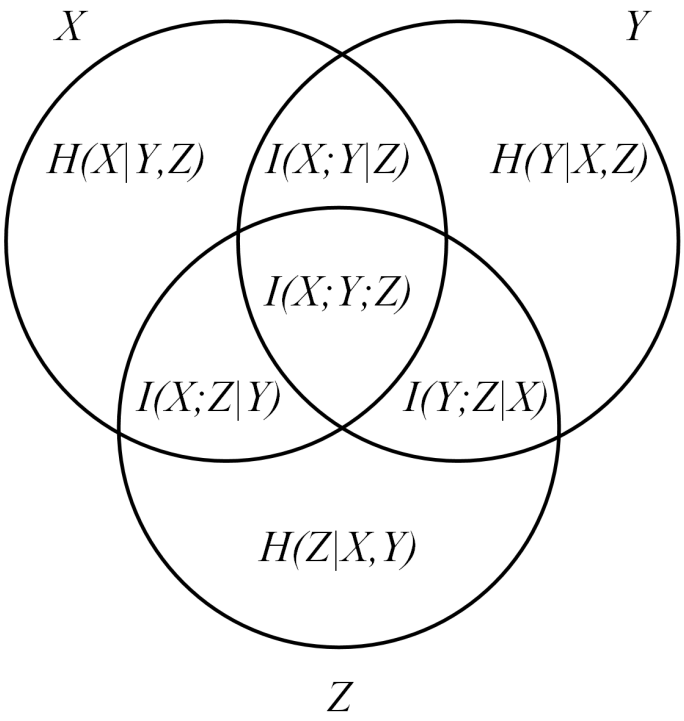


Figure 3.4: Venn diagram representation of entropy, conditional mutual information, and multivariate mutual information among three random variables X, Y, and Z. Pairwise CMI terms such as $I(X;Y|Z)$ appear in the pairwise overlaps, while the shared region in the center corresponds to the multivariate mutual information $I(X;Y;Z)$.

which can be either positive or negative depending on how the dependencies among variables combine. A positive value indicates redundancy, meaning that the three variables share overlapping information (e.g., knowing Z explains part of what X and Y share). A negative value indicates synergy, where the joint knowledge of X and Y provides more information about Z than either variable alone. This property makes interaction information a non-intuitive but powerful extension of pairwise mutual information.

CMI could be useful in sequential decision-making contexts, where one often seeks to measure dependencies between variables while accounting for prior knowledge—such as past actions or observations. In the context of empowerment, CMI can be used to evaluate the influence of actions on future states while conditioning on the current state—thus enabling temporally extended notions of controllability and agency. CMI will also play a role in the later formalisation of object empowerment and its decomposition across environmental components.

3.2.5 Channel Capacity and Communication Channels

In information theory, a *communication channel* refers to a probabilistic mapping from an input variable X to an output variable Y [130]. Formally, a channel is defined by a conditional probability distribution $p(y \mid x)$, which specifies the probability that an input symbol $x \in \mathcal{X}$ produces an output symbol $y \in \mathcal{Y}$. This formulation encompasses both deterministic channels (where $p(y \mid x)$ is a delta function) and stochastic channels with noise.

The *channel capacity* is the maximum information rate, which is the maximum amount of information (in bits) that can be transmitted per channel use, at which messages can be communicated reliably over such a channel. Mathematically, it is defined as the maximum mutual information between the input and output over all possible input distributions $p(x)$:

$$C = \max_{p(x)} I(X;Y) \tag{3.2.6}$$

In this classical framework, a sender encodes a message into input symbols $x \in \mathcal{X}$, which are transmitted through a channel that introduces probabilistic noise according to $p(y|x)$. The receiver observes the corresponding outputs $y \in \mathcal{Y}$. The term "reliably" indicates that information can be transmitted with an arbitrarily low probability of error, provided that the transmission rate does not exceed the channel capacity. This value, measured in bits per transmission, represents the theoretical upper bound on how much information can be communicated without error, assuming optimal encoding and decoding strategies (see [130] for more details).

Types of Channels: Communication theory categorizes channels into several types, based on their structural and noise properties:

- **Deterministic Channels:** Each input symbol maps to a unique output symbol; there is no randomness in the transmission. These channels have capacity equal to the entropy of the output.
- **Noisy Channels:** The output is a probabilistic function of the input (e.g., Binary Symmetric Channel). These reflect real-world communication systems with transmission errors.
- **Memoryless Channels:** The channel transformation $p(y \mid x)$ is independent across time steps; the output depends only on the current input and not on past inputs or outputs. The actuation channel used in empowerment is of this type, since the environment dynamics $P(s' \mid s, a)$ in an MDP depend only on the current state–action pair and remain stationary over time.

- **Channels with Memory:** The output depends not only on the current input but also on previous inputs/outputs.

These distinctions are particularly relevant when information-theoretic concepts in RL are expressed in terms of communication channels. For example, the interaction between an agent and its environment can be interpreted as a noisy, memoryless channel, where the agent's actions correspond to channel inputs and the resulting states to channel outputs.

The information-theoretic quantities introduced in this section (i.e., entropy, conditional entropy, mutual information, conditional mutual information, and channel capacity) constitute the mathematical backbone of the empowerment-based frameworks. These measures not only allow one to quantify uncertainty and dependency but also provide a rigorous means of formalising agency and control in autonomous agents. The next section develops this idea in detail and lays the foundation for the definition of object empowerment (see Chapter 4), which models how agents can evaluate and manipulate tools and objects within their environment.

3.3 Empowerment

Empowerment is an information-theoretic measure that quantifies the degree of influence an agent can exert on its environment through its actions [1, 10, 97]. It is formalised as the Shannon channel capacity of the agent's *action–perception loop* [132], capturing the maximal information that can flow from the agent's actions to its future percepts. This corresponds to the original formulation of empowerment, where the output of the actuation channel represents the agent's sensor variables (i.e., future perceptions) rather than the full environmental state.

3.3.1 The Action–Perception Loop

At the heart of empowerment lies the concept of the action–perception loop, which models the closed interaction cycle between an agent and its environment. At each time step t, the agent is in state $S_t \in \mathcal{S}$, selects an action $A_t \in \mathcal{A}$, the environment transitions to a new state S_{t+1}, and the agent receives an observation $O_{t+1} \in \mathcal{O}$ that informs its next action.

From an information-theoretic viewpoint, this loop can be interpreted as a *communication channel*, where the agent "sends" information into the environment via its actions, and "receives" information through its observations. Empowerment then quantifies the *channel capacity* of this loop, i.e., the maximal amount of information the agent can inject into its future percepts via its choice of actions. This formalisation highlights empowerment as

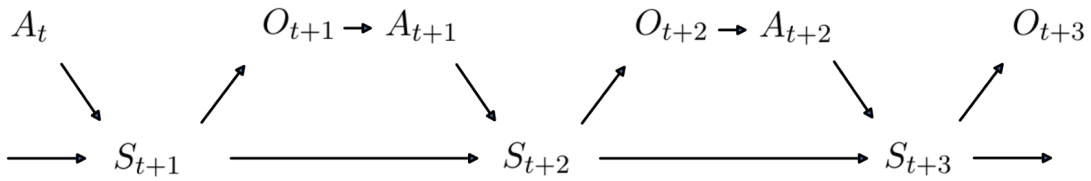


Figure 3.5: Causal Bayesian network illustrating the action-perception loop over multiple time steps. Each action A_t influences the next environment state S_{t+1}, which generates observation O_{t+1} for the agent. The loop continues recursively, enabling the agent to iteratively interact with and influence its environment.

a measure of the agent's potential control over its future sensorimotor experience across multiple time steps [10, 97].

To illustrate this loop visually, Figure 3.5 depicts the agent–environment interaction as a *Causal Bayesian network* [133]. Each action A_t influences the next state S_{t+1}, which determines the observation O_{t+1}, closing the loop as the agent selects A_{t+1} based on current percepts. This causal framing will later facilitate the definition of the h-step actuation channel used in empowerment.

Let $a_t \in \mathcal{A}$ denote a single action at time t, and let $a_{t:t+h-1} = (a_t, a_{t+1}, \ldots, a_{t+h-1}) \in \mathcal{A}^h$ denote an action sequence of length h. In empowerment, such sequences are regarded as random variables A_t^h, representing all possible futures the agent could generate from a given starting state. Importantly, empowerment considers what the agent *could* do, not what it *will* do.

To formalise the h-step action–perception channel, the probability distribution over observations induced by the execution of action sequences from an initial state is considered. Let $P(O \mid S)$ be the observation model, describing the probability of perceiving o given that the true state is s. The induced h-step channel is then given by the conditional distribution:

$$P(O_{t+h} \mid A_t^h = a_{t:t+h-1}, S_t = s).$$

This triple $(A_t^h, P(O_{t+h} \mid A_t^h, S_t = s), O_{t+h})$ defines the h-step actuation channel, with input A_t^h, output O_{t+h}, and channel kernel $P(O_{t+h} \mid A_t^h, S_t = s)$.

Empowerment at state s with horizon h is the Shannon capacity of this channel:

$$\mathfrak{E}^h(s) = \max_{P(a_{t:t+h-1}|s)} I(O_{t+h}; A_t^h \mid S_t = s) \tag{3.3.1}$$

It quantifies, in bits, the maximum amount of information an agent can inject into its future observations via the selection of action sequences of length h. Intuitively, empowerment measures how much "freedom of choice" the agent has in shaping its perceptual future. High empowerment reflects greater control over what the agent will eventually observe,

regardless of any specific task.

Empowerment is defined under an *open-loop* assumption: the agent selects an entire action sequence in advance, without adjusting based on intermediate feedback.

3.3.2 Computation of Empowerment in Non-Deterministic Environments

To compute empowerment, the mutual information maximisation problem in Equation 3.3.1 must be solved. A widely used method for this purpose is the *Blahut–Arimoto (BA)* algorithm [134, 135], an iterative algorithm originally developed to compute channel capacity in information theory. The formulation and equations presented in this section closely follow the approach of Jung et al. [95], where the BA algorithm has been adapted for empowerment computation in stochastic control systems.

The BA algorithm iteratively estimates the optimal input distribution over action sequences, denoted $P(A_t^h \mid s)$, which maximizes the mutual information between these sequences and the resulting future observations, conditioned on the current state s. This formulation computes the empowerment value by treating the action-perception loop as a communication channel and optimizing its channel capacity.

Suppose the set of action sequences of length h is finite and denoted $\mathcal{A}^h = \{a_{t:t+h-1}^{(1)}, \ldots, a_{t:t+h-1}^{(n)}\}$, and let $s \in \mathcal{S}$ be the current state of the environment. The goal is to compute the mutual information between A_t^h and O_{t+h}, given $S_t = s$.

Let $p_k(A_t^h)$ denote the input distribution over action sequences at iteration k. Strictly speaking, this distribution depends on the current state s (i.e., $p_k(A_t^h \mid s)$), but we omit the explicit conditioning on s for notational clarity since s is fixed throughout the computation. It is given as:

$$p_k(A_t^h) \equiv \left(p_k^{(1)}, \ldots, p_k^{(n)}\right) \quad \text{with} \quad p_k^{(v)} = \Pr(A_t^h = a_{t:t+h-1}^{(v)})$$

Define the conditional distribution over future observations as $p(o \mid s, a_{t:t+h-1}^{(v)})$ for each action sequence $a_{t:t+h-1}^{(v)}$. Then, for each $a_{t:t+h-1}^{(v)} \in \mathcal{A}^h$, the algorithm computes:

$$d_{v,k} := \int_{\mathcal{O}} p(o \mid s, a_{t:t+h-1}^{(v)}) \log \left[\frac{p(o \mid s, a_{t:t+h-1}^{(v)})}{\sum_{i=1}^{n} p(o \mid s, a_{t:t+h-1}^{(i)}) \cdot p_k^{(i)}} \right] \mathrm{d}o \tag{3.3.2}$$

For discrete observation spaces, this simplifies to:

$$d_{v,k} := \sum_{o \in \mathcal{O}} p(o \mid s, a_{t:t+h-1}^{(v)}) \log \left[\frac{p(o \mid s, a_{t:t+h-1}^{(v)})}{\sum_{i=1}^{n} p(o \mid s, a_{t:t+h-1}^{(i)}) \cdot p_k^{(i)}} \right] \tag{3.3.3}$$

These divergence terms $d_{v,k}$ are associated with each action sequence $a^{(v)}_{t:t+h-1}$. Specifically, each $d_{v,k}$ is a KL divergence between the outcome distribution conditioned on choosing $a^{(v)}_{t:t+h-1}$ and the expected outcome distribution under the current input distribution p_k. Intuitively, a larger $d_{v,k}$ indicates that the action sequence leads to more distinguishable or informative future observations, given the current estimate of how likely each action sequence is to be executed. They are then used to update the input distribution:

$$p^{(v)}_k := \frac{1}{z_k} p^{(v)}_{k-1} \exp(d_{v,k}) \tag{3.3.4}$$

where the normalization constant is:

$$z_k := \sum_{v=1}^{n} p^{(v)}_{k-1} \exp(d_{v,k}) \tag{3.3.5}$$

After a fixed number of iterations leading to convergence, the empowerment estimate is:

$$\mathfrak{E}_k(s) = \sum_{v=1}^{n} p^{(v)}_k \cdot d_{v,k} \tag{3.3.6}$$

Algorithm 3 BA Algorithm for Empowerment [95]

Require: Transition model $p(o \mid s, a^{(v)}_{t:t+h-1})$, set of action sequences of length h: $\{a^{(1)}_{t:t+h-1}, \ldots, a^{(n)}_{t:t+h-1}\}$, initial state s, threshold ϵ, maximum iterations $K_{\max}$

Ensure: Estimated empowerment $\mathfrak{E}(s)$

1: Initialize input distribution: $p_0(a^{(v)}_{t:t+h-1}) \leftarrow \frac{1}{n}$ for all $v = 1, \ldots, n$

2: $k \leftarrow 0$

3: **repeat**

4: **for** each $v = 1$ to n **do**

5: Compute:

$$d_{v,k} \leftarrow \sum_{o \in \mathcal{O}} p(o \mid s, a^{(v)}_{t:t+h-1}) \log \frac{p(o \mid s, a^{(v)}_{t:t+h-1})}{\sum_{i=1}^{n} p(o \mid s, a^{(i)}_{t:t+h-1}) \cdot p_k(a^{(i)}_{t:t+h-1})}$$

6: **end for**

7: Compute normalization constant:

$$z_k \leftarrow \sum_{v=1}^{n} p_k(a^{(v)}_{t:t+h-1}) \cdot \exp(d_{v,k})$$

8: **for** each $v = 1$ to n **do**

9: Update input distribution:

$$p_{k+1}(a^{(v)}_{t:t+h-1}) \leftarrow \frac{1}{z_k} \cdot p_k(a^{(v)}_{t:t+h-1}) \cdot \exp(d_{v,k})$$

10: **end for**

11: Estimate empowerment:

$$\mathfrak{E}_{k+1}(s) \leftarrow \sum_{v=1}^{n} p_{k+1}(a^{(v)}_{t:t+h-1}) \cdot d_{v,k}$$

12: $k \leftarrow k + 1$

13: **until** $|\mathfrak{E}_k(s) - \mathfrak{E}_{k-1}(s)| < \epsilon$ **or** $k \geq K_{\max}$

14: **return** $\mathfrak{E}_k(s)$

Algorithm 3 outlines the full BA procedure for empowerment computation in environments with discrete and finite action and observation spaces. It operates iteratively, alternating between computing the per-action information contributions $d_{v,k}$ and updating the input distribution $p_k(a^{(v)}_{t:t+h-1})$. At each step, the algorithm refines the estimate of the optimal action distribution that maximizes mutual information between actions and future percepts.

The core iterative procedure of the BA algorithm can be summarised as follows:

- Compute the divergence term $d_{v,k}$ for each action sequence $a^{(v)}_{t:t+h-1}$, representing its contribution to mutual information under the current input distribution.
- Update the input distribution $p_k(a^{(v)}_{t:t+h-1})$ using these divergences and a normalising constant z_k.
- Estimate the empowerment at iteration k as the expected divergence under the updated distribution.

The process repeats until convergence, i.e., when the change in empowerment estimate is smaller than a pre-defined threshold ϵ, or a maximum number of iterations $K_{\max}$ is reached.

Practical Considerations: The BA algorithm provides a principled framework to estimate empowerment in stochastic and partially observable settings. However, its computational demands grow rapidly with the size of the action space (which scales exponentially with the horizon h) and the observation space. This is because, at each iteration, the agent must evaluate the full conditional probability distribution $p(o \mid s, a^{(v)}_{t:t+h-1})$ for every action sequence and recompute the input distribution via repeated updates over all sequences (as seen in Equations 3.3.3 and 3.3.4).[1]

In practice, some approximation strategies are used to alleviate this computational burden:

- *Monte Carlo Estimation:* When the observation or action spaces are continuous or extremely large, empirical sampling can be used to approximate the conditional distributions and mutual information terms [96].
- *Variational Approaches:* Recent work has introduced neural variational bounds to approximate empowerment, bypassing the need to enumerate all action sequences explicitly [89, 136].

These approximations trade off accuracy for tractability and can introduce estimation bias, particularly in environments with high-dimensional state or observation spaces or in the presence of significant stochasticity. The extent of this bias depends on the choice of approximation method and the structure of the underlying dynamics.

3.3.3 Computation of Empowerment in Deterministic Environments

In deterministic, fully observable environments, where transitions and observations are not stochastic, the empowerment formulation simplifies significantly. Each action sequence

[1] See [95] for a detailed complexity discussion.

maps to a unique outcome. Let $\mathcal{O}^h(s)$ be the set of unique observations reachable from state s by executing all h-step action sequences. Then empowerment reduces to:

$$\mathfrak{E}^h(s) = \log_2 |\mathcal{O}^h(s)| \tag{3.3.7}$$

Here, $|\cdot|$ denotes cardinality. This expression counts the number of distinct outcomes the agent can reach within h steps, providing a direct measure of future diversity.

In fully observable domains where $\mathcal{O} = \mathcal{S}$, the observation model is the identity, and empowerment becomes:

$$\mathfrak{E}^h_S(s) = \log_2 |\mathcal{S}^h(s)| \tag{3.3.8}$$

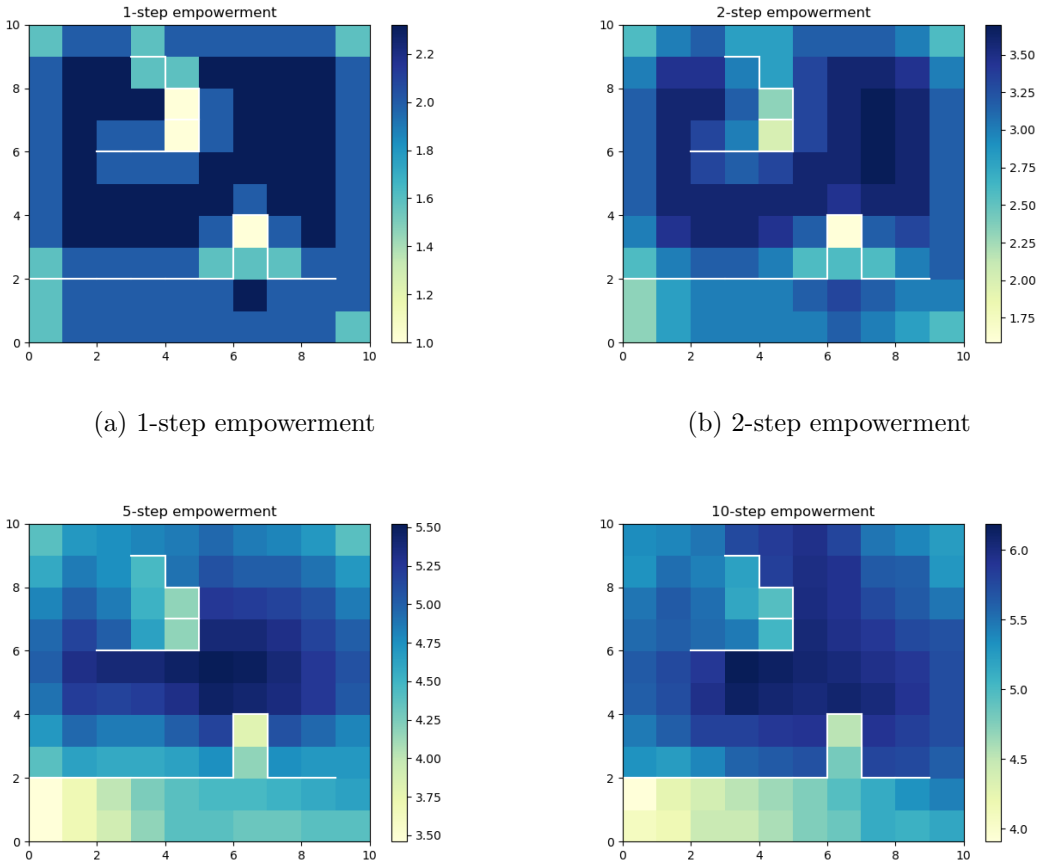
where $\mathcal{S}^h(s)$ is the set of distinct states reachable from s via h-step action sequences.

To visualize how empowerment varies spatially and across horizons, the canonical grid-world setup by Klyubin et al. [1] is reproduced (see Figure 3.6). This 10×10 grid allows the agent to move in four directions or remain stationary. Some adjacent cells are separated by walls (white colour lines), restricting movement. Empowerment is computed for each grid cell based on the number of reachable final states encountered after all possible action sequences of length h have been executed. Figure 3.6 shows the resulting landscapes for $h \in \{1, 2, 5, 10\}$. Even for relatively short horizons (e.g., $h = 2$), the essential structure of the empowerment landscape is apparent: central regions exhibit higher empowerment due to greater maneuverability, while walls and corners constrain reachable future trajectories and reduce empowerment. As the horizon increases, the landscape becomes smoother and more uniform. In the case of infinite horizon, every state becomes reachable from every other, resulting in high but uninformative empowerment values that are identical across the grid. These visualizations exemplify the core intuition behind empowerment: the agent's potential to influence its own future.

3.3.4 Interpretation and Role in Intrinsic Motivation

Empowerment provides a general-purpose, task-independent signal of how much control or influence an agent has in a given state. High empowerment states afford the agent a rich set of distinguishable future observations; low empowerment states offer limited options or are dominated by environmental noise. Unlike goal-directed rewards, empowerment requires no explicit objective. Its maximisation encourages agents to stay in situations where their actions make a difference.

This information-theoretic notion of agency has been used in various intrinsic motivation frameworks to support exploration, behavioral diversity, and skill acquisition



(a) 1-step empowerment

(b) 2-step empowerment

(c) 5-step empowerment

(d) 10-step empowerment

Figure 3.6: Empowerment landscapes in a 10×10 grid world for horizons $h = 1, 2, 5, 10$, based on the setup introduced by Klyubin et al. [1]. As the planning horizon increases, more states become reachable from each position, especially in open areas, resulting in higher empowerment values. Border regions and cells adjacent to walls exhibit lower values due to limited reachability.

[21, 105, 106, 110]. In this thesis, empowerment serves as the foundation for a novel extension: *object empowerment*, which will model how tools alter an agent's control over specific objects in its environment. The next chapter introduces this extension and its relevance to tool use.

Chapter 4

Object Empowerment

This chapter contributes to the overall aim of the thesis by introducing *object empowerment* as an object-conditioned extension of classical empowerment. In doing so, it directly addresses research question RQ1, which asks how empowerment can be reformulated to explicitly capture the agent's influence over manipulable objects. Parts of the formulation and illustrative results presented in this chapter are based on publication C1. The thesis extends this work by presenting the concept in a broader theoretical context and by providing additional analysis of object-specific controllability.

Traditional formulations of empowerment measure the degree of influence an agent has over its entire state space, as perceived through its sensors. While this perspective captures a general sense of agency, it often fails to account for the structured and compositional nature of real-world environments. In many practical scenarios, agents operate in environments composed of multiple discrete objects or entities, and the extent of their influence depends on what their sensory channels can observe or affect. For example, an agent may be equipped with a tool that allows it to affect only a particular subset of the environment, while the rest remains unchanged or out of reach.

This observation motivates the need for a more focused measure of agency, one that evaluates an agent's potential to influence specific elements within its environment. To address this, the concept of *Object Empowerment* is introduced. Object empowerment quantifies the agent's capacity to control or affect a particular object (or class of objects) over a given planning horizon, independent of the rest of the state space. Rather than measuring how many distinct future states are reachable, it instead measures how many distinct future states of a target object the agent can induce reliably through its actions. This object-centric view of empowerment enables a more nuanced understanding of the agent's interactions with objects in structured environments. It is particularly relevant for a formalisation of the use of tools, which selectively influence only certain parts of the

world. By isolating the controllability of individual objects, object empowerment provides an interpretable, intrinsic signal that can guide exploration and decision-making in tasks where an agent needs to interact with them.

The remainder of this chapter formalises the concept of object empowerment, and presents illustrative landscapes that reveal how the agent's object-specific influence varies spatially and temporally within an environment.

4.1 Formalism

4.1.1 State space

In all environments considered here there is an agent with state $\mathcal{S}^{\mathfrak{A}}$ (e.g., position or pose), one or more *objects* with states $\mathcal{S}^{\mathfrak{O}}$ (e.g., location, color, intact/destroyed), and *tools* with states $\mathcal{S}^{\mathfrak{T}}$ (e.g., equipped/unequipped, orientation, charge). Tools represent a special class of objects the agent can equip and use to change the state of other objects in ways that may be impossible otherwise (e.g., a key opens a door; a hammer drives a nail; a pickaxe breaks a boulder). Not all tools can affect all objects, and two tools that affect the same object may do so via different dynamics. For simplicity, a setting with one agent, one tool, and one target object is considered here. The extension to multiple tools and objects is discussed later in Chapter 6.

Following the formulation introduced above, a specific type of action–perception loop $(\mathcal{S}, \mathcal{O}, \mathcal{A}, P)$ is defined for object empowerment. Conceptually, this loop is similar to a Partially Observable Markov Decision Process (POMDP). However, there is an important difference: although a sensor (observation) is introduced to define empowerment, the underlying task itself remains fully observable to the agent. In other words, the agent always knows its complete state for control and planning. Partial observability is used here only to define the *empowerment channel* through an object-focused sensor, not to limit the agent's knowledge of the environment.

Object empowerment is now formalised as a variant of empowerment that quantifies an agent's potential to influence a target object's state. Let the full environment state space $\mathcal{S}$ be decomposed as:

$$\mathcal{S} = \mathcal{S}^{\mathfrak{A}} \times \mathcal{S}^{\mathfrak{T}} \times \mathcal{S}^{\mathfrak{O}} \times \mathcal{S}^{\mathfrak{W}} \tag{4.1.1}$$

- $\mathcal{S}^{\mathfrak{A}}$: state space of the agent (e.g., its location in the environment).
- $\mathcal{S}^{\mathfrak{T}}$: state space of tools, if present (e.g., its position or whether it is equipped by the agent).
- $\mathcal{S}^{\mathfrak{O}}$: state space of the object of interest (e.g., its location or condition).

- $\mathcal{S}^{\mathfrak{W}}$: other static components of the environment, such as walls or goal positions.

An *object sensor* is defined that observes only the object-related component of object $\mathfrak{O}$ of the global state. The observation space of the object sensor for object $\mathfrak{O}$ is defined as $\mathcal{O}^{\mathfrak{O}} = \mathcal{S}^{\mathfrak{O}}$. The observation model is then given by:

$$P(\hat{o} \mid s) \coloneqq \delta_{\hat{o},s^{\mathfrak{O}}} \quad , \tag{4.1.2}$$

where δ denotes the Kronecker delta function, which equals 1 when the observation $\hat{o}$ matches the object state $s^{\mathfrak{O}}$, and 0 otherwise. The term "object empowerment" is used to denote the empowerment of the actuation channel that employs the object sensor.

4.1.2 Action Space

Among all the actions in $\mathcal{A}$ that an agent can perform, the subsets of actions executed while using a tool $\mathfrak{T}$ are defined. The following subsets of $\mathcal{A}$ are distinguished:

- $\mathcal{A}^{\mathfrak{A}} \subseteq \mathcal{A}$: actions executed directly by the agent, independently of tools.
- $\mathcal{A}^{\mathfrak{T}} \subseteq \mathcal{A}$: actions corresponding to the use of a tool $\mathfrak{T}$.
- $\mathcal{A}^{\mathfrak{A}\mathfrak{T}} \coloneqq \mathcal{A}^{\mathfrak{A}} \cup \mathcal{A}^{\mathfrak{T}}$: the combined set of agent actions and those specific to tool $\mathfrak{T}$.

For instance, in a navigation task, $\mathcal{A}^{\mathfrak{A}}$ may include the action "north", while $\mathcal{A}^{\mathfrak{T}}$ could contain "chop" once the agent equips a tool (e.g., an axe). Otherwise, when the tool is not equipped, executing actions from $\mathcal{A}^{\mathfrak{T}}$ produces no effect on the environment. For simplicity of notation, $\mathcal{A}^{\mathfrak{A}\mathfrak{T}}$ is denoted as $\mathcal{A}^{\mathfrak{T}}$ in the following.

4.1.3 Transition Dynamics

The environment evolves according to the transition model $P(s' \mid s, a)$, which defines the probability of reaching a future state s' after taking an action a in the current state s. These transitions depend on the agent's motion, the tool's state, and their interactions with the object.

In general, three types of transitions can be distinguished:

- **Agent transitions:** those that change the agent's own state $\mathcal{S}^{\mathfrak{A}}$, such as movement or navigation actions (e.g., moving north or south).
- **Tool transitions:** those that modify the tool's state $\mathcal{S}^{\mathfrak{T}}$, such as picking up, unequipping , or rotating a tool.

- **Tool–object interactions:** those that result from using an equipped tool on an object, changing the object's state $\mathcal{S}^{\mathfrak{O}}$ (e.g., chopping a tree or breaking a boulder).

For the purpose of defining empowerment, the transition probability distributions associated with the two actuation channels introduced in Sections 4.1.4 and 4.1.5 are considered. The first describes transitions produced by the agent's own actions, while the second represents transitions that occur when the agent acts through an equipped tool. Other transitions, such as those that change the tool's equipped status or reposition it in the environment, are part of the general transition model $P(s' \mid s, a)$ but are not explicitly formalised here.

4.1.4 Agent's Object Empowerment

Using the state, action, observation, and transition components defined above, the *agent's object empowerment* is defined as the channel capacity between the agent's action sequences and the resulting object state observed after h steps. Formally, the actuation channel consists of the triple $(\mathcal{A}_t^{\mathfrak{A}^h}, P(O_{t+h}^{\mathfrak{O}} | \mathcal{A}_t^{\mathfrak{A}^h}, S_t = s), O_{t+h}^{\mathfrak{O}})$, with the input being the h-step agent action sequence and the output being the object state perceived after h steps. Its empowerment is defined as:

$$\mathfrak{E}_{\mathfrak{A}\mathfrak{O}}^h(s) \coloneqq \max_{P(a^{\mathfrak{A}^h})} I\left(\mathcal{A}_t^{\mathfrak{A}^h}; O_{t+h}^{\mathfrak{O}} \mid S_t = s\right) \tag{4.1.3}$$

This empowerment value, measured in bits, represents how much information the agent's actions can transmit to the object's future state. In simple terms, it indicates how many distinct object outcomes the agent can reliably produce through its actions over the h-step horizon.

4.1.5 Tool's Object Empowerment

Similarly, *tool-object empowerment* measures the potential influence of the tool's actuation on the object. In this case, the actuation channel is described by the triple $(\mathcal{A}_t^{\mathfrak{T}^h}, P(O_{t+h}^{\mathfrak{O}} | \mathcal{A}_t^{\mathfrak{T}^h}, S_t = s), O_{t+h}^{\mathfrak{O}})$, where the input consists of h-step sequences of tool actions. Its empowerment is formally defined as:

$$\mathfrak{E}_{\mathfrak{T}\mathfrak{O}}^h(s) \coloneqq \max_{P(a^{\mathfrak{T}^h})} I\left(\mathcal{A}_t^{\mathfrak{T}^h}; O_{t+h}^{\mathfrak{O}} \mid S_t = s\right) \tag{4.1.4}$$

This empowerment value, measured in bits, quantifies how much information the tool's actions can transmit to the object's future state. In simple terms, it indicates how many distinct object outcomes the tool can reliably produce when used by the agent over the h-step horizon.

The prior assumption, which posits that the agent must be capable of interacting with the object when using the tool, can be formalized by the condition $\mathfrak{E}^h_{\mathfrak{T}\mathfrak{O}} > 0$. Furthermore, $\mathfrak{E}^h_{\mathfrak{T}\mathfrak{O}}$ can be used to quantify how much the tool is effective in influencing the state of the object. In principle, different pairs of tools and objects could be considered and compared with respect to the magnitude of this quantity. A tool is considered useful when $\mathfrak{E}^h_{\mathfrak{A}\mathfrak{O}} < \mathfrak{E}^h_{\mathfrak{T}\mathfrak{O}}$; if this condition does not hold, the tool encumbers the agent with respect to object manipulation. In other words, when this condition is satisfied, by using the tool the agent has more impact on the object than by not using it. Note how these statements can be expressed both with respect to a single state $s \in \mathcal{S}$, using the per-state empowerment values $\mathfrak{E}^h_{\mathfrak{A}\mathfrak{O}}(s)$ and $\mathfrak{E}^h_{\mathfrak{T}\mathfrak{O}}(s)$, or in terms of the whole MDP, averaging empowerment over an uniform distribution of states, yielding average empowerment values $\hat{\mathfrak{E}}^h_{\mathfrak{A}\mathfrak{O}}$ and $\hat{\mathfrak{E}}^h_{\mathfrak{T}\mathfrak{O}}$.

4.1.6 Object Empowerment in Deterministic Environments

In deterministic environments, where a given action sequence from a specific state always leads to a single outcome, object empowerment simplifies to counting the number of distinct object states that can be reached after h steps. This makes empowerment a purely combinatorial quantity, avoiding the need for probabilistic modeling.

Given an initial state $s \in \mathcal{S}$, the deterministic *agent-object empowerment* is defined as:

$$\mathfrak{E}^h_{\mathfrak{A}\mathfrak{O}}(s) \coloneqq \log_2 \left| \left\{ o^{\mathfrak{O}} \;\middle|\; o^{\mathfrak{O}} = T^h_{\mathfrak{O}}(s, a^{\mathfrak{A}^h}),\ a^{\mathfrak{A}^h} \in \mathcal{A}^{\mathfrak{A}^h} \right\} \right| \tag{4.1.5}$$

where $T^h_{\mathfrak{O}}(s, a^{\mathfrak{A}^h})$ is the deterministic transition function that returns the observed state $o^{\mathfrak{O}}$ of the object $\mathfrak{O}$ given that the agent executes the action sequence $\mathcal{A}^{\mathfrak{A}^h}$ in state s, and $\mathcal{A}^{\mathfrak{A}^h}$ denotes all possible h-step agent action sequences.

Similarly, the deterministic *tool-object empowerment* is given by:

$$\mathfrak{E}^h_{\mathfrak{T}\mathfrak{O}}(s) \coloneqq \log_2 \left| \left\{ o^{\mathfrak{O}} \;\middle|\; o^{\mathfrak{O}} = T^{\mathfrak{O}^h}(s, a^{\mathfrak{T}^h}),\ a^{\mathfrak{T}^h} \in \mathcal{A}^{\mathfrak{T}^h} \right\} \right| \tag{4.1.6}$$

where $\mathcal{A}^{\mathfrak{T}^h}$ represents all h-step tool-related action sequences.

These deterministic formulations provide an intuitive interpretation of object empowerment: they quantify how many distinct ways the agent, either directly or through a tool, can alter the object's state over a planning horizon of h steps.

The empowerment values introduced so far describe an agent's, either directly or through a tool, potential influence from a specific state. However, for comparing entities across an entire environment, it is useful to consider their average influence over all states, as described next.

4.1.7 State-Average Object Empowerment

While object empowerment is defined for each state individually, it is often useful to compute its average across all valid states of the environment. This *state-average object empowerment* provides a scalar summary of the overall influence that an agent or a tool can exert on the object:

$$\hat{\mathfrak{E}}^{h}_{X\mathfrak{O}} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \mathfrak{E}^{h}_{X\mathfrak{O}}(s), \tag{4.1.7}$$

where $X \in \{\mathfrak{A}, \mathfrak{T}\}$ denotes either the agent or a specific tool, and $|\mathcal{S}|$ is the total number of reachable states. This measure captures the average controllability of an object over the entire state space, enabling a compact comparison between entities, such as the agent and multiple tools, without focusing on local spatial differences.

4.2 Agent–Object Interaction Environment

The analysis of object empowerment begins with the most minimal and controlled setting: a two-dimensional grid world containing only an agent and an object. This simplified environment removes any auxiliary elements such as tools, or stochastic dynamics, allowing us to isolate the direct influence the agent can exert on the object through its own actions. By studying this case first, a baseline is established for interpreting empowerment values in more complex scenarios introduced later. The deterministic nature of the environment ensures that changes in entity states arise solely from the agent's deliberate actions, allowing empowerment values to be interpreted without the confounding effects of stochastic transitions. Immediately following the environment description, the corresponding empowerment landscapes are presented to illustrate how the agent's capacity to affect the object varies across the grid.

The grid size used in this chapter is 10×10, which provides a balance between analytical clarity and spatial expressiveness. A smaller grid would restrict the range of possible spatial relationships between entities in the environment, limiting the variability of empowerment values and making it difficult to observe meaningful gradients in the resulting landscapes. Conversely, substantially larger grids would increase the state space without adding conceptual insight while significantly increasing the computational cost of multi-step empowerment calculations. The 10×10 configuration therefore offers a sufficiently rich spatial structure for analysing empowerment while remaining computationally tractable for systematic evaluation. This choice is also consistent with early empowerment studies, which used a 10×10 grid world to analyse agent influence and controllability in discrete environments [1].

4.2.1 Environment Description

The environment is a 10×10 grid world composed of two entities: an agent whose states are denoted by $s^{\mathfrak{A}} \in \mathcal{W}$ and represented as a red robot in Figure 4.1, and an object whose states are denoted by $s^{\mathfrak{O}} \in \mathcal{W}$ and depicted as a black box. Here, $\mathcal{W}$ denotes the set of all valid grid cells in the world. Each entity occupies a unique cell, and no two entities can occupy the same position simultaneously. The boundaries of the grid world are impassable for both the agent and the object.

The state space of the environment is defined as

$$\mathcal{S} = \mathcal{S}^{\mathfrak{A}} \times \mathcal{S}^{\mathfrak{O}},$$

where $\mathcal{S}^{\mathfrak{A}}$ and $\mathcal{S}^{\mathfrak{O}}$ correspond to all possible positions of the agent and the object, respectively. The available action set for the agent is

$$\mathcal{A}^{\mathfrak{A}} = \{\uparrow_A, \rightarrow_A, \downarrow_A, \leftarrow_A\},$$

representing one-cell movements in the cardinal directions of north, east, south, and west, respectively. If the target cell in the chosen direction is within bounds and unoccupied, the agent moves into that cell; otherwise, the state remains unchanged.

The object is modeled as a movable entity that responds to pushes by the agent. If the agent is adjacent to the object (i.e., $d_M(s^{\mathfrak{A}}, s^{\mathfrak{O}}) = 1$, where d_M denotes the Manhattan distance) and attempts to move towards it, the object is pushed one cell in the same direction—provided that its destination cell is within bounds and unoccupied. If the push is obstructed by the grid boundary, the action has no effect, and no transition occurs.

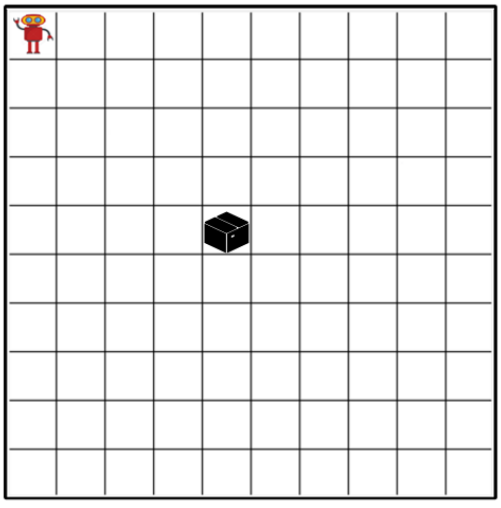


Figure 4.1: Agent–object grid world setup. The agent (red robot) can move freely within the grid boundaries, while the object (black box) can be pushed by the agent through its direct movement actions.

All transitions in this grid world are deterministic; given a state and an action, the next state is uniquely defined.

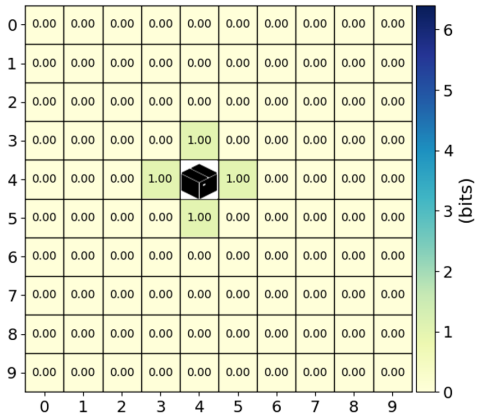
4.2.2 Movable vs Non-Movable Objects

Object empowerment is first compared in two variants of the agent–object grid world: one where the object is movable and one where it is fixed in place. This contrast highlights how an object's inherent mobility directly determines the agent's potential influence over it.

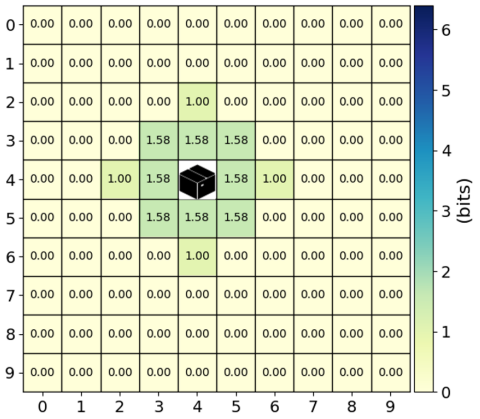
In the *movable* case, the object behaves as described in Section 4.2.1: it can be pushed by the agent into adjacent free cells within the grid boundaries. In the *non-movable* case, the object's position is fixed for the entire episode, and no agent action can alter its location.

For the movable object, Figure 4.2 shows the agent–object empowerment $\mathfrak{E}^h_{\mathfrak{AO}}$ landscapes for different horizons $h \in \{1, 2, 5, 10\}$. Each cell in the landscapes represents the value of object empowerment corresponding to the agent being in that particular state. At $h = 1$, non-zero empowerment is confined to positions immediately adjacent to the object, where a single push is possible. In these cells, the maximum empowerment value is 1 bit, as indicated by the color bar in the figure. This corresponds to two possible object locations resulting from the agent's action: the object remaining in its original position if the agent moves without interaction, or being displaced by one cell when pushed. Empowerment is zero bits in all other cells, where the agent cannot interact with the object and thus cannot change its state. As h increases, regions of high empowerment expand outward from the location of the box, reflecting the agent's ability to reach and manipulate the object from greater distances within the allowed number of steps. Moreover, when the agent is close to the object, a longer planning horizon enables it to move the box to a greater number of possible future positions, thereby increasing the empowerment values in the surrounding interaction region.

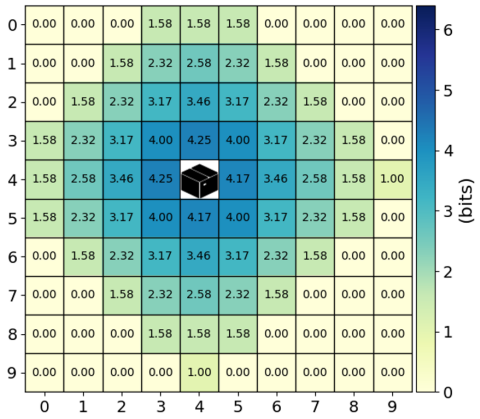
In contrast, the empowerment landscape for a non-movable object is identically zero across all states and horizons, as no sequence of actions can alter the object's position. Since this behaviour is invariant to the horizon, only the case for $h = 1$ is shown in Figure 4.3; the same landscape would be obtained for any $h \geq 1$. This serves as a baseline, illustrating that when an object is not physically manipulable, its empowerment is trivially null regardless of the agent's capabilities.



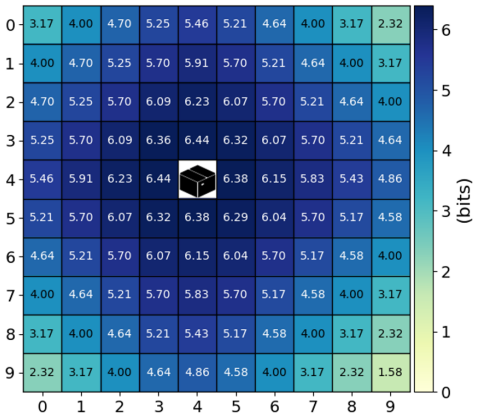
(a) $h = 1$



(b) $h = 2$



(c) $h = 5$



(d) $h = 10$

Figure 4.2: Agent–object empowerment landscapes for a *movable* object placed at the grid center (i.e., $(4, 4)$), computed for varying horizons. Longer horizons allow the agent to affect the object from a larger portion of the grid and in a greater number of possible ways, especially when the agent is closer to the object.

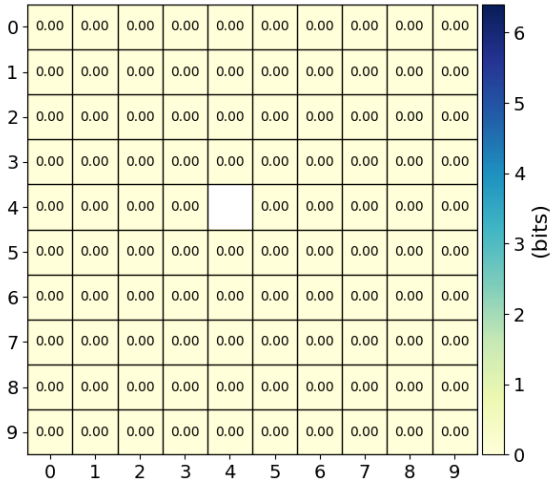


Figure 4.3: Agent–object empowerment landscape for a *non-movable* object fixed at $(4, 4)$, shown for $h = 1$. The same landscape occurs for any $h \geq 1$.

4.3 Agent–Tool–Object Interaction Environment

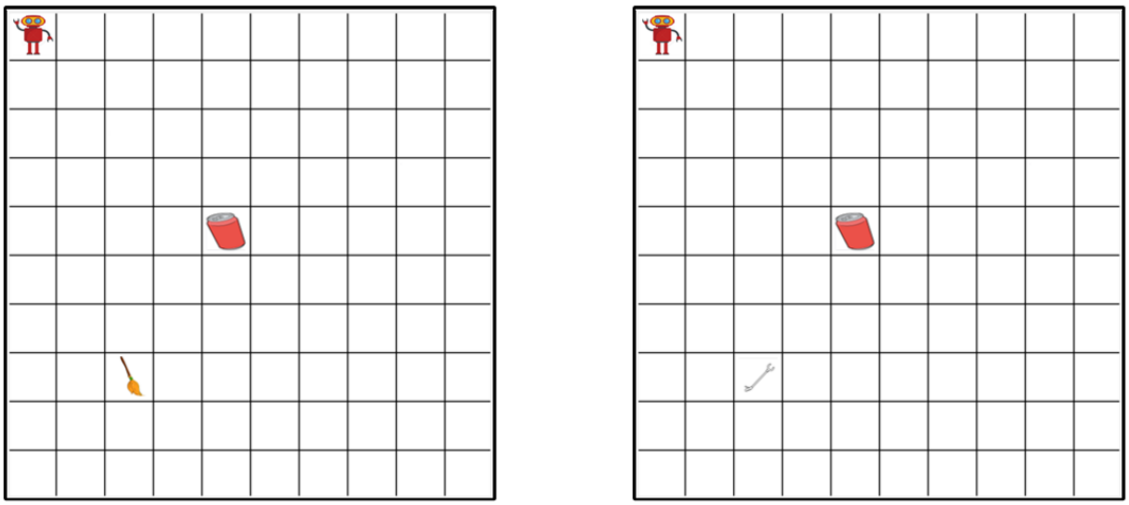
The basic agent–object scenario is now extended by introducing a tool into the grid world. This addition enables investigation of how the presence of manipulable intermediary entities affects object empowerment, both when the agent interacts directly with the object and when it does so through tool use. By considering different types of tools, each with distinct interaction dynamics, a broader range of tool–object relationships can be captured and their impact on the agent's ability to influence the object can be quantified.

4.3.1 Environment Description

Two variants of a 10×10 deterministic grid world (Figures 4.4a and 4.4b) are considered, containing three entities: an agent (depicted as a red robot), a tool (shown as either a broom or a picker), and an object (a can). Also in this case, each entity occupies a unique grid cell at all times, and no two entities may share the same cell. The grid is bounded, and none of the entities can move outside its limits. All transitions are deterministic: given a state and an action, the next state is uniquely defined.

The environment state $s \in \mathcal{S}$ is represented as a tuple $(s^{\mathfrak{A}}, s^{\mathfrak{T}}, s^{\mathfrak{O}})$, where $s^{\mathfrak{A}} \in \mathcal{W}$, $s^{\mathfrak{T}} \in \mathcal{W}$, and $s^{\mathfrak{O}} \in \mathcal{W}$ denote the positions of the agent, the tool, and the object, respectively. Here $\mathcal{W}$ is the set of all valid grid cells.

The action set is $\mathcal{A} = \mathcal{A}^{\mathfrak{A}} \cup \mathcal{A}^{\mathfrak{T}}$, where:



(a) Grid world with broom tool.

(b) Grid world with picker tool.

Figure 4.4: Two grid-world configurations, each with an agent (robot), an object (can), and a different type of tool (a broom or a picker).

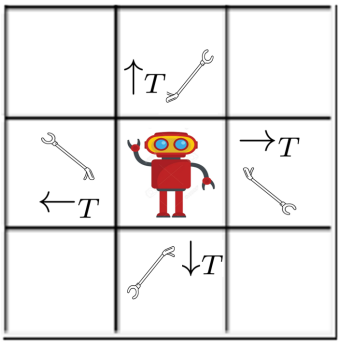


Figure 4.5: Relative tool movements using $\mathcal{A}^{\mathfrak{T}}$ actions.

- $\mathcal{A}^{\mathfrak{A}} = \{\uparrow_A, \rightarrow_A, \downarrow_A, \leftarrow_A\}$ moves the agent's body by one cell in a cardinal direction.
- $\mathcal{A}^{\mathfrak{T}} = \{\uparrow_T, \rightarrow_T, \downarrow_T, \leftarrow_T\}$ repositions the equipped tool relative to the agent (Figure 4.5).

A tool becomes *equipped* when it is in a cell adjacent to the agent ($d_M(s^{\mathfrak{T}}, s^{\mathfrak{A}}) = 1$). Pickup occurs automatically and persists thereafter; once equipped, the tool stays with the agent, moving along with it during navigation actions. Tool-specific actions from $\mathcal{A}^{\mathfrak{T}}$ can then be used to change the tool's state or interact with objects. When the tool is not equipped, these tool actions have no effect. The agent may influence the object either directly (by pushing it with its movement actions) or indirectly through an equipped tool.

The example transitions of both tools are given in Figure 4.6. This configuration supports the computation of both *agent–object empowerment* (direct manipulation) and

tool–object empowerment (manipulation through an equipped tool). Comparing the resulting landscapes reveals how different tool dynamics shape the agent's capacity to influence the object.

Two distinct tool–object interaction models are considered:

- **Broom:** This tool allows the agent to influence the object at short range when the tool's tip is *aligned* and adjacent to the object (i.e., when $d_M(s^{\mathfrak{T}}, s^{\mathfrak{O}}) = 1$ along the same row or column). When the agent pushes in the direction of the object, the broom extends the agent's effective body and causes the object to slide by one cell in that direction. However, the broom's effect is not persistent — if the agent moves away, the object remains in its current cell. The broom can be reoriented using $\mathcal{A}^{\mathfrak{T}}$ actions, allowing the agent to interact with the object from different directions. This tool provides the agent with a limited extension of reach and enables side pushing that would otherwise be impossible using only body movements.
- **Picker:** The picker enables *persistent coupling* with the object. When the equipped tool is aligned and adjacent to the object ($d_M(s^{\mathfrak{T}}, s^{\mathfrak{O}}) = 1$), the object becomes "attached" to the picker. Once attached, if the agent moves using $\mathcal{A}^{\mathfrak{A}}$, it carries both the tool and the object with it. Similarly, when the agent executes a rotation action from $\mathcal{A}^{\mathfrak{T}}$, both the tool and the attached object rotate together around the agent's position. In this way, the picker allows the agent to transport the object quickly and directly across the grid, requiring fewer steps than pushing it with its body or with the broom.

These two interaction models differ significantly in their dynamics and, consequently, in the empowerment values they produce. The broom extends the agent's range of influence but requires constant re-alignment for each interaction, leading to relatively low and localised empowerment values. In contrast, the picker's "sticky" dynamics allow the agent to move the object more directly and efficiently once attached, resulting in higher empowerment values and a broader spatial distribution of controllability. Both tools, however, increase the agent's influence over the object compared to using body (movements) actions alone.

4.3.2 Broom Tool: Proximal vs Distant Object Landscapes

This subsection investigates the effect of tool–object proximity on object empowerment using the broom tool. Two configurations are considered: (*i*) the broom placed directly adjacent to the object (proximal case), and (*ii*) the broom placed several grid cells away from the object (distant case), as shown in Figure 4.4a. For each configuration, the object

empowerment landscapes are computed for different horizons h, allowing us to examine how empowerment values propagate through the grid world. Each cell in the landscape represents the value of object empowerment corresponding to the agent being in that particular state. Cells adjacent to the tool indicate positions from which the agent can equip and use the broom to influence the object.

When the broom is positioned adjacent to the object, the 1-step object empowerment landscape (Figure 4.7a) shows that empowerment values are uniform and maximal (1.0 bits) in all cells adjacent to the object and broom positions. This indicates that the agent can now also use the broom to move the object to an additional possible location beyond its original one when the tool is equipped. In other words, the broom extends the agent's ability to influence the object, allowing one extra reachable outcome compared to using body movements alone. This also indicates that the agent has an equal capacity to influence the object regardless of whether it is adjacent to the broom (i.e., has equipped the broom) or to the object itself (i.e., has not equipped the broom).

As the horizon increases to $h = 2$ (Figure 4.8a), empowerment values begin to spread outward while remaining highest near the object (i.e., 2.0 bits). At this horizon, it is observed that the agent can influence the object more effectively through its direct movement actions $\mathcal{A}^{\mathfrak{A}}$ compared to positions where it must rely on both movement and tool actions ($\mathcal{A}^{\mathfrak{A}}$ and $\mathcal{A}^{\mathfrak{T}}$). In other words, empowerment values are higher in cells where the agent can directly interact with the object than in those where it must first use the broom to do so. A similar pattern persists as the planning horizon increases, although the region of non-zero empowerment gradually expands. By $h = 5$ (Figure 4.9a), empowerment exhibits a broader spatial distribution, yet the peak values remain concentrated around the object's location, reaching more than 4.0 bits. The same overall trend is observed at $h = 8$ (Figure 4.10a), where the maximal values reach 6.0 bits next to the object.

When the broom is placed far from the object, the 1-step landscape (Figure 4.7b) shows high empowerment (i.e., 1.0 bits) only in cells adjacent to the object due only to the agent interaction with the can. The broom's location in this configuration has no immediate impact on the empowerment values because the agent cannot affect the object in a single step by using the broom because this is too far to allow an interaction with the can in one step. At $h = 2$ (Figure 4.8b), the object can still only be influenced through the agent's movement actions $\mathcal{A}^{\mathfrak{A}}$, since the broom remains too far to contribute. The resulting empowerment values are symmetric around the object, with a maximum of 2.0 bits. By $h = 5$ (Figure 4.9b), empowerment spreads more broadly across the grid; however, the peak values (4.0 bits) continue to cluster near the object rather than around the broom's position. At this horizon, the agent begins to partially affect the object through limited use of the broom, with local maxima near the broom reaching about 2.6 bits. This

trend continues for $h = 8$ (Figure 4.10b), where the overall influence region expands, but the highest empowerment values (6.0 bits) remain concentrated near the object, and the maximal values close to the broom rise to approximately 5.0 bits.

For the broom tool, proximity to the object affects both the *magnitude* and the *location* of empowerment peaks, particularly at short horizons. At $h = 1$, the maximal empowerment values (1.0 bit) are identical in the proximal and distant cases, but their spatial locations differ—being centred near the broom in the proximal setup and around the object in the distant one. At higher horizons ($h = 5$ and $h = 8$), empowerment extends over a larger spatial region, and the two sources of influence, the agent's direct interaction with the object and the potential interaction through the broom, begin to overlap. However, when the broom is far from the object, all observed empowerment values arise solely from the agent's direct interaction with the object, since the broom cannot yet affect it. As the horizon increases, this distinction becomes blurred because the agent may eventually reach the broom, equip it, and use it within the same planning window. Overall, these results confirm that the broom's influence is *local*, it only increases empowerment when the object lies within its immediate range. Such locality is consistent with its physical nature and contrasts with tools that exert influence from a distance, which are further discussed in Chapter 7.1.2.

4.3.3 Picker Tool: Proximal vs Distant Object Landscapes

This subsection investigates the effect of tool-object proximity on object empowerment using the picker tool. The two configurations are the same as before: (*i*) the picker placed directly adjacent to the object (proximal case), and (*ii*) the picker placed several grid cells away from the object (distant case), like in Figure 4.4b. For each configuration, the object empowerment landscapes are computed for different horizons h, revealing how the influence of the picker differs from that of the broom tool.

When the picker is positioned adjacent to the object, the 1-step object empowerment landscape (Figure 4.11a) reveals a distinct asymmetry compared to the broom. Cells next to the picker exhibit a higher empowerment value (3.0 bits), while cells on the opposite side of the object have lower values (1.0 bits). This higher value arises from the picker's persistent coupling dynamics: once the tool is equipped and aligned with the object, the object becomes attached to it. From this configuration, the agent can move the object in four different ways through its movement actions $\mathcal{A}^{\mathfrak{A}}$ and in four additional ways through its rotation actions $\mathcal{A}^{\mathfrak{T}}$, resulting in eight possible next object states. The corresponding information content, $\log_2(8) = 3.0$ bits, quantifies the agent's expanded ability to affect the object when using the picker. This indicates that equipping the picker offers a greater immediate capacity to affect the object than interacting with it through the agent's own

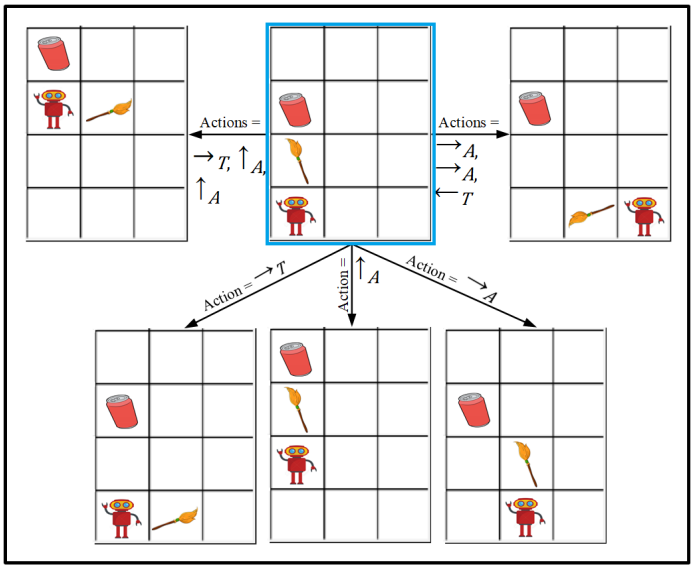
actions without using the tool.

At $h = 2$ (Figure 4.12a), high empowerment remains concentrated near the picker, with peak values around 4.5 bits, while the opposite side of the object continues to show lower values. By $h = 5$ (Figure 4.13a), the high-value region around the picker expands further, with peak values reaching 6.0 bits. At $h = 8$ (Figure 4.14a), the empowerment distribution becomes even broader, but the highest values (about 6.5 bits) still occur in proximity to the picker rather than the object. This is because, in these regions, the agent can combine both its movement actions $\mathcal{A}^{\mathfrak{A}}$ and tool actions $\mathcal{A}^{\mathfrak{T}}$ to influence the object, whereas positions closer to the object but farther from the picker rely mainly on the agent's actions $\mathcal{A}^{\mathfrak{A}}$ alone.

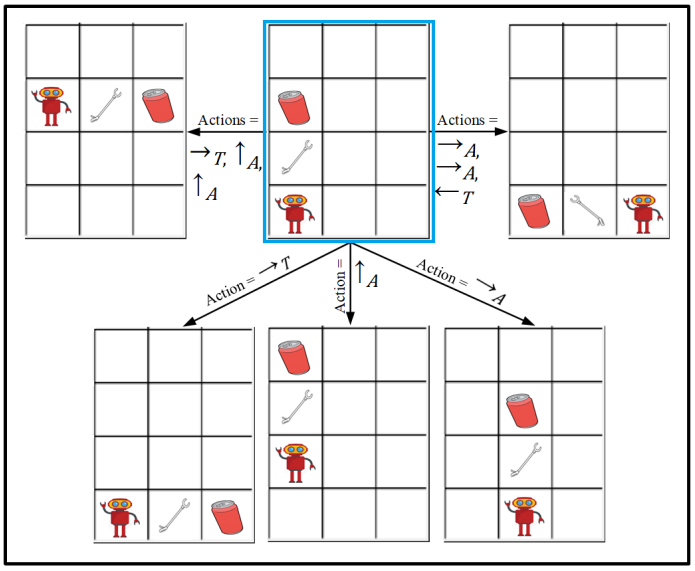
When the picker is placed far from the object, the 1-step landscape (Figure 4.11b) resembles that of the broom: high empowerment (1.0 bit) is found only in the cells adjacent to the object because the tools are too far to interact with it. In this configuration, the observed empowerment values result solely from the agent's actions, as neither tool can influence the can at such distance. At $h = 2$ (Figure 4.12b), the object can still be influenced only through the agent's movement actions $\mathcal{A}^{\mathfrak{A}}$, since the picker remains too far to contribute. The resulting empowerment values are symmetric around the object, with a maximum of 2.0 bits, similar to the broom case.

However, unlike the broom, at $h = 5$ (Figure 4.13b) a notable shift is observed: the region around the picker now reaches values of about 5.0 bits, comparable to the immediate surroundings of the object (4.0 bits). This shift occurs because, with a longer horizon, the agent has enough steps to move while carrying the picker, reach the object, and interact with it. The increase is notable since the picker enables more direct and efficient manipulation of the object, allowing it to be moved in more ways and in fewer steps than with the broom. This trend becomes even more evident at $h = 8$ (Figure 4.14b), where the area near the picker holds the highest empowerment values (6.3 bits), surpassing those around the object.

For the picker tool, proximity affects both the *extent* and the *location of peak empowerment values* as the horizon increases. When the picker is close to the object, empowerment values are high and concentrated around their interaction area. As the distance increases, these values initially drop because the agent requires more steps to reach and use the tool effectively. However, for longer horizons, the agent has enough steps to move with the picker, reach the object, and manipulate it. Unlike the broom, which only exerts a brief, local influence when directly in contact with the object, the picker's stronger and more persistent dynamics allow its impact to extend further within the same number of steps. As a result, for sufficiently large horizons, empowerment near the picker's location becomes more dominant than around the object's location, even when the tool starts farther away.

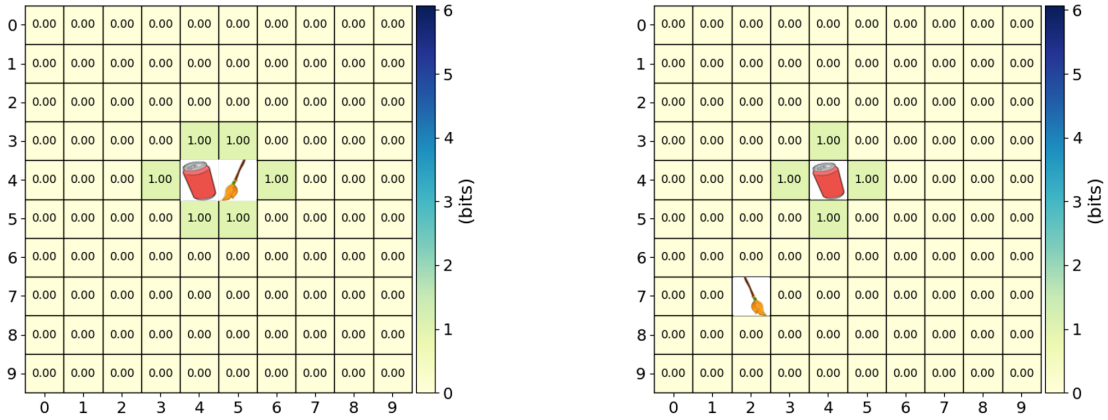


(a) Example transitions with the broom tool.



(b) Example transitions with the picker tool.

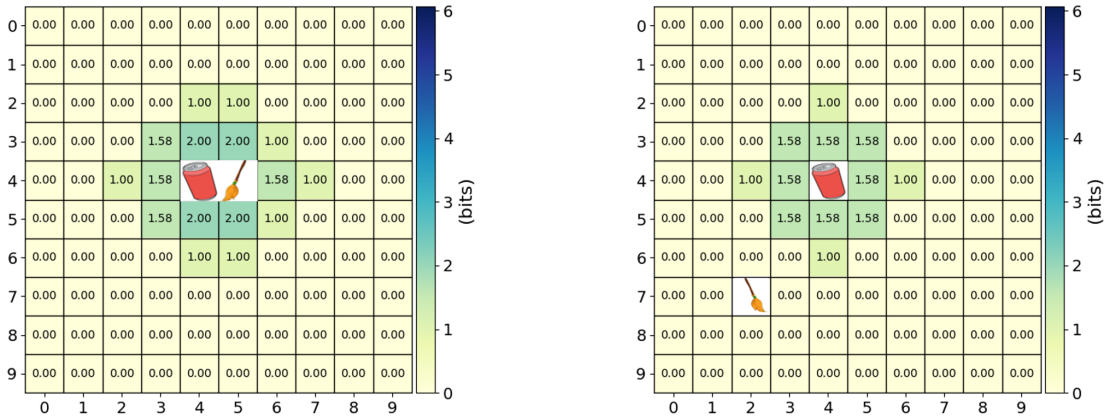
Figure 4.6: Illustrative tool–object interactions. Blue-bordered cells indicate the agent's starting position. In both subfigures, the lower part shows the outcome of a single action, while the upper part shows the outcome of a sequence of actions. With the picker, the agent can move the object without changing its own position, whereas with the broom, the agent must move its body to push the object.



(a) Proximal

(b) Distant

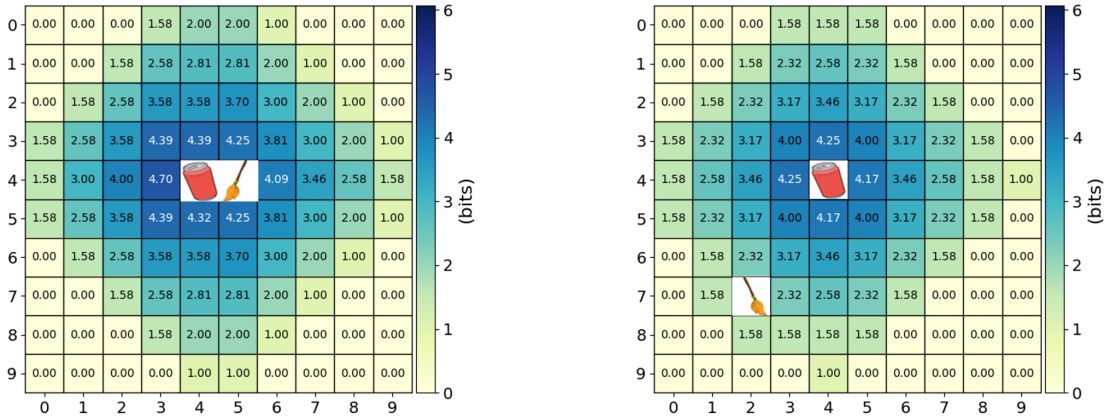
Figure 4.7: Object empowerment landscapes for the broom tool at $h = 1$ in proximal (broom at $(5, 4)$) and distant (broom at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$.



(a) Proximal

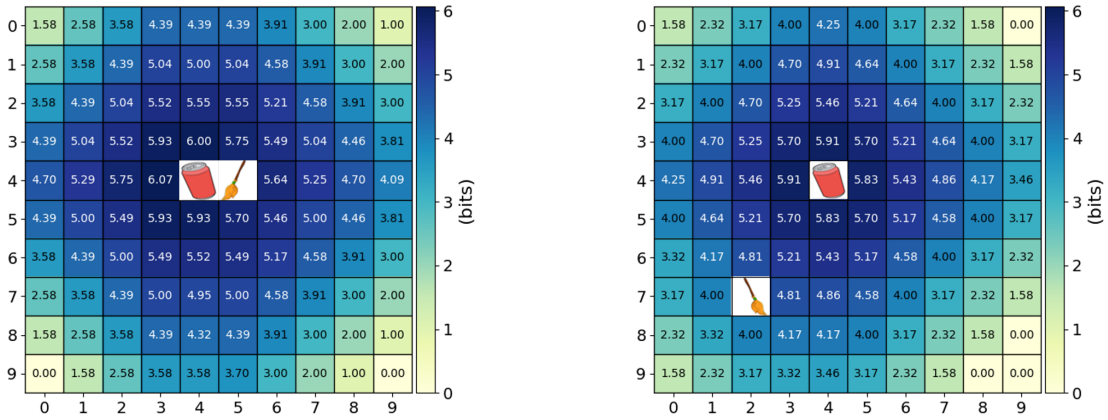
(b) Distant

Figure 4.8: Object empowerment landscapes for the broom tool at $h = 2$ in proximal (broom at $(5, 4)$) and distant (broom at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$.



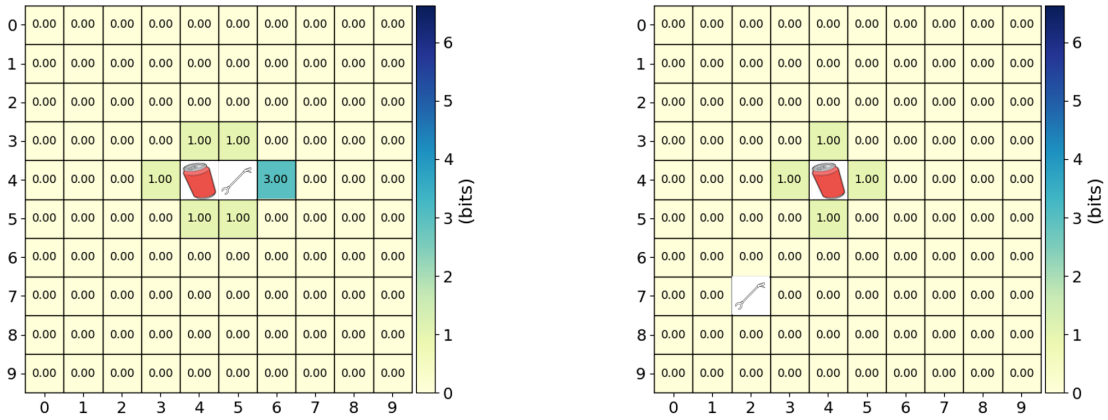
(a) Proximal (b) Distant

Figure 4.9: Object empowerment landscapes for the broom tool at $h = 5$ in proximal (broom at $(5, 4)$) and distant (broom at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$.



(a) Proximal (b) Distant

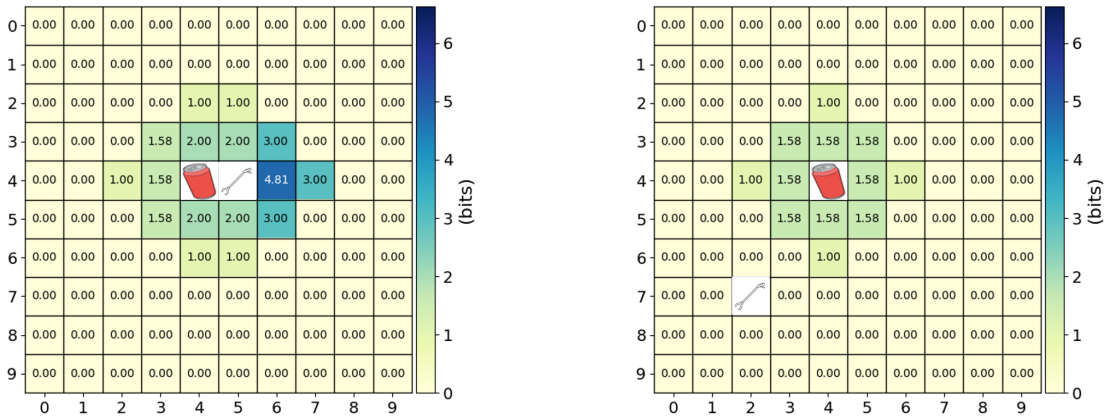
Figure 4.10: Object empowerment landscapes for the broom tool at $h = 8$ in proximal (broom at $(5, 4)$) and distant (broom at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$.



(a) Proximal

(b) Distant

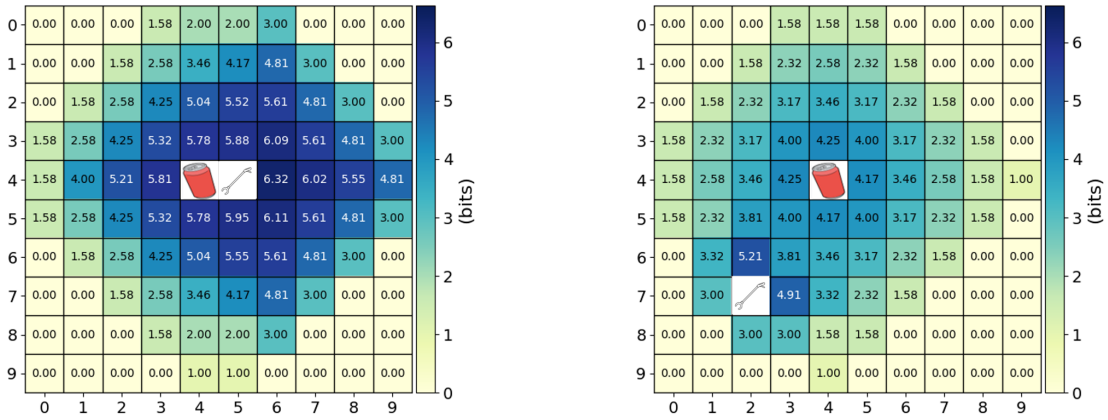
Figure 4.11: Object empowerment landscapes for the picker tool at $h = 1$ in proximal (picker at $(5, 4)$) and distant (picker at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$.



(a) Proximal

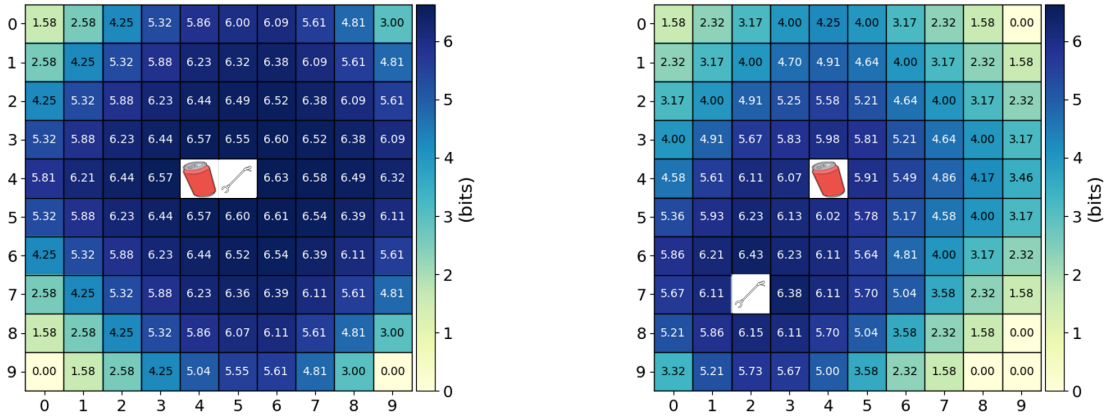
(b) Distant

Figure 4.12: Object empowerment landscapes for the picker tool at $h = 2$ in proximal (picker at $(5, 4)$) and distant (picker at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$.



(a) Proximal (b) Distant

Figure 4.13: Object empowerment landscapes for the picker tool at $h = 5$ in proximal (picker at $(5, 4)$) and distant (picker at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$.



(a) Proximal (b) Distant

Figure 4.14: Object empowerment landscapes for the picker tool at $h = 8$ in proximal (picker at $(5, 4)$) and distant (picker at $(2, 7)$) configurations. In both cases, the object is located at $(4, 4)$.

4.4 Agent–Tool–Object Interaction in MiniHack

MiniHack [137] is a RL environment built on top of the classic NetHack game [138]. It provides a rich, procedurally generated world in which an agent can move, explore, and interact with various entities such as walls, doors, tools, and objects. Actions include navigation, tool use, and object manipulation, enabling the study of complex sequential behaviours such as tool use and goal-directed interaction.

MiniHack offers multi-modal observations at each time step: (i) a matrix of glyphs representing the visual layout of the environment, (ii) an encoded textual message describing the most recent in-game event (e.g., *"You cut down the tree."*), and (iii) a list of inventory contents representing the tools currently held by the agent. These observations collectively form the input to the agent's policy and value networks, which will be leveraged later in the thesis. This section focuses on the computation and analysis of object empowerment in MiniHack.

4.4.1 Environment Description

The MiniHack environment employed here consists of an agent, an object $\mathfrak{O}$ (a tree), and a tool $\mathfrak{T}$ (an axe), as shown in Figure 4.15. In this setup, the tree is non-movable object and can be chopped by the agent equipped with the axe.

The environment is a 10×10 grid, with each tile corresponding to a glyph in the observation matrix. The state space $\mathcal{S}$ is:

$$\mathcal{S} = \left(s^{\mathfrak{A}},\, s^{\mathfrak{T}},\, s^{\mathfrak{O}}\right)$$

where $s^{\mathfrak{A}} \in \mathcal{W}$ is the agent's grid location, $s^{\mathfrak{T}} \in \mathcal{W}$ is the tool's position, $s^{\mathfrak{O}} \in \mathcal{W}$ is the object's position. In addition, each tool state includes two binary flags: an *equipped* flag indicating whether the tool is currently held by the agent (i.e., stored in the inventory), and a *hidden* flag denoting whether the tool is visible from the agent's point of view. Similarly, each object state includes a *hidden* flag, which specifies whether the object is visible from the agent's point of view, and a *destroyed* flag, which indicates whether the object has been destroyed by the agent (e.g., chopped).

The action space $\mathcal{A}$ includes:

- Agent movement actions $\mathcal{A}^{\mathfrak{A}}$ (north, south, east, west).
- Tool actions $\mathcal{A}^{\mathfrak{T}}$, which are only available when the tool is equipped.

Tool actions are based on the MiniHack game mechanics, where the use of a tool involves three sequential transitions: first, the agent decides to use a some tool by executing the

Figure 4.15: MiniHack environment for object empowerment analysis. The agent (bottom-right) can pick up the axe and use it to destroy the tree.

"apply" action; then, it selects which tool from its inventory to use via tool identifier actions; finally, it specifies one of the four cardinal directions in which to apply the tool ("apply" $\rightarrow$ "choose" $\rightarrow$ "direction"). For instance, applying an axe to the north may destroy a tree located in that direction, producing the in-game message *"You cut down the tree."*. A tool is equipped automatically when the agent enters the tool's cell (i.e., $s^{\mathfrak{A}} = s^{\mathfrak{T}}$), after which it is transferred to the agent's inventory.[1] The grid-world dynamics T is deterministic, so object empowerment is computed using Equation 4.1.6.

4.4.2 Destroyable vs. Indestructible Interaction Scenarios

In MiniHack, the same object can exhibit different interaction outcomes depending on whether the agent possesses the appropriate tool. To illustrate this, a single object type is considered: the *tree*, which can be either *destroyable* or effectively *indestructible* depending on the agent's current capabilities. When the agent equips the correct tool (an axe), the tree becomes destroyable and can be chopped down through the corresponding tool actions. In contrast, when the axe is not equipped or absent from the environment, the same tree behaves as an indestructible object, remaining unaffected by the agent's actions.

These two conditions, *with* and *without* the appropriate tool, allow us to analyse how tool possession changes the agent's potential to influence the object. Specifically, the resulting object empowerment landscapes are compared to examine how the tree's destroyability, mediated by tool availability, shapes both the spatial distribution and magnitude of object

[1] Note that equipping in MiniHack environments differs from the earlier grid-world setups, where a tool became equipped when it was in a cell adjacent to the agent (i.e., $d_M(s^{\mathfrak{T}}, s^{\mathfrak{A}}) = 1$).

empowerment across the environment.

Destroyable Object: Tool Equipped

When the agent is equipped with the appropriate tool (in this case, the axe), it becomes capable of destroying the tree. As described earlier, the MiniHack tool-use mechanic requires a minimum of three actions to interact with the object: (1) applying the tool, (2) selecting the tool from the inventory, and (3) specifying the direction of use. This means that even if the agent is already adjacent to the tree, at least $h = 3$ is required for object empowerment to be non-zero.

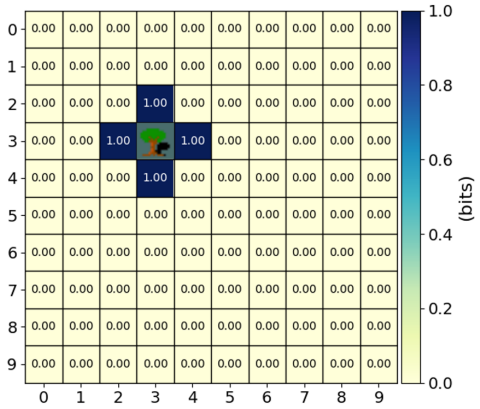
Figure 4.16 shows the resulting object empowerment landscapes for horizons $h = 3, 4, 5$. In all cases, the maximum empowerment value reaches 1.0 bit at positions from which the tree can be destroyed within the horizon. This 1.0 bit value corresponds to two distinguishable outcome states: *destroyed* or *not destroyed*. At $h = 3$, these positions correspond exactly to the four cardinal cells adjacent to the tree. As h increases, the set of positions with non-zero empowerment expands outward from the tree location, reflecting the agent's ability to reach one of the adjacent positions before executing the three-step interaction. It shows that only the spatial extent of empowered states grows with larger horizons.

Destroyable Object: Tool Not Equipped

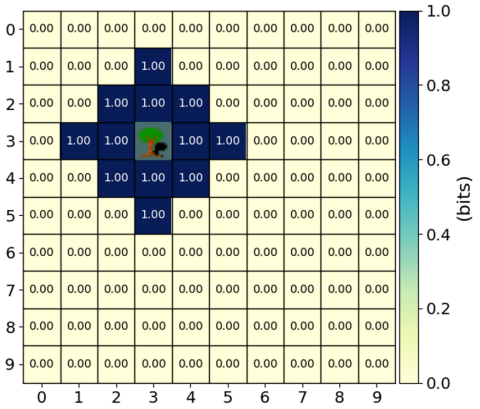
When the tool is not initially equipped, the agent must first navigate to the tool's location before it can interact with the object. In the current setup, the axe is placed at coordinates $(6, 6)$, requiring at least five Manhattan movement steps to reach a cell adjacent to the tree's position. Once the axe is equipped, the agent still needs to perform the three-step interaction sequence described earlier (apply, select tool, select direction) to destroy the tree. In total, the agent therefore requires a minimum of eight steps before the first possible interaction with the tree.

Figure 4.17 shows the resulting object empowerment landscapes for horizons $h = 8, 9, 10$. For $h < 8$, the empowerment landscape remains entirely zero because the agent cannot both equip the axe and reach the tree within the given horizon. At $h = 8$, the only empowered state is the position containing the axe itself, since this is the point where the agent first gains the ability to affect the object in subsequent steps. For $h = 9$ and $h = 10$, the set of empowered states begins to expand outward from the axe's location. This progression reflects the increasing number of positions from which the agent can eventually destroy the tree as h grows.

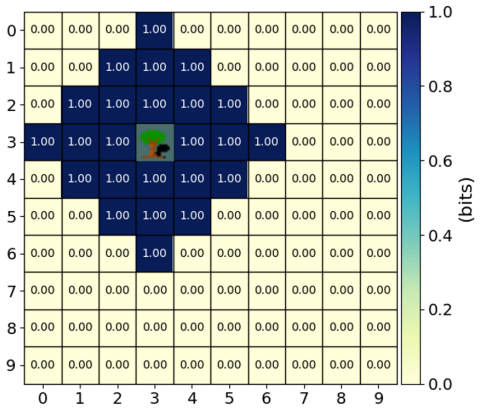
Also, in this case, while the spatial extent of empowered states increases with the horizon, the magnitude of the maximum value does not change.



(a) $h = 3$

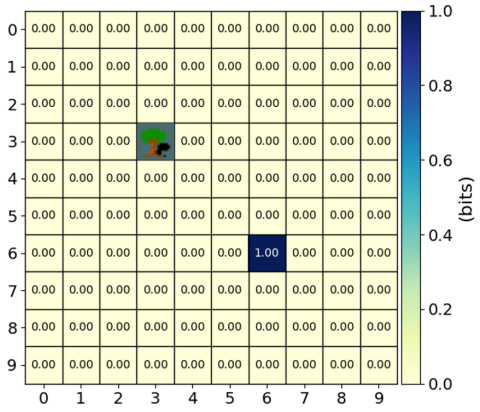


(b) $h = 4$

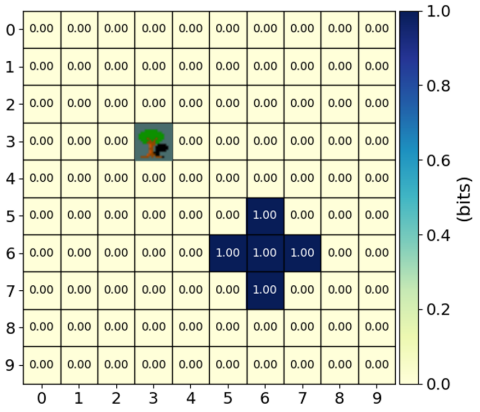


(c) $h = 5$

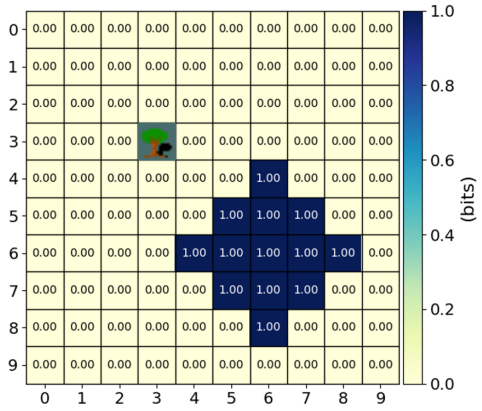
Figure 4.16: Object empowerment landscapes for a destroyable object (tree) when the agent is equipped with the axe, for horizons $h = 3, 4, 5$. The tree is located at $(3, 3)$.



(a) $h = 8$



(b) $h = 9$



(c) $h = 10$

Figure 4.17: Object empowerment landscapes for a destroyable object (tree) when the agent is not equipped with the axe, which is initially placed at $(6, 6)$, for horizons $h = 8, 9, 10$. The tree is located at $(3, 3)$.

Indestructible Object

An object is indestructible when it cannot be modified or removed under any circumstances within the environment. For example, a tree becomes indestructible if the agent attempts to interact with it using any tool other than an appropriate tool (like an axe), or if the appropriate tool is not present in the environment at all. In such cases, no sequence of actions, regardless of the horizon, can change the object state.

Consequently, the empowerment values remain zero across the entire state space for all horizons, as there are no states from which the agent can influence the object. This is illustrated in Figure 4.18, where the empowerment landscape is uniformly zero.

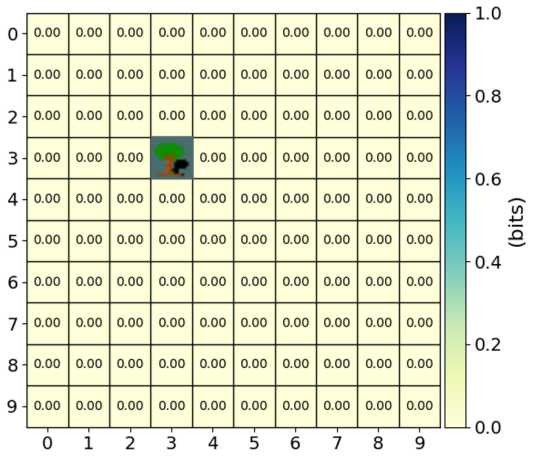


Figure 4.18: Empowerment landscape for an indestructible object (tree) across the grid. All values are zero, as no action sequence can affect the object.

Across the three scenarios: tool equipped, tool not equipped, and indestructible object, the object empowerment landscapes reveal how the agent's ability to influence the object depends critically on both tool accessibility and object properties. When a tool, which can modify the state of the object, is equipped, empowered states appear immediately around the object and expand outward from the tool location with increasing horizons. When the tool is not equipped, object empowerment emerges only after the agent can reach and equip the tool, delaying the onset of non-zero values. In contrast, when objects are indestructible, object empowerment remains zero everywhere, reflecting the absence of any action sequence capable of altering the object's state. Together, these cases highlight how object empowerment encodes diverse affordances arising from different agent–tool–object relationships.

4.5 Summary

This chapter developed and analysed the concept of *object empowerment* as an object-centric extension of the empowerment framework, capturing an agent's potential influence over specific objects in its environment. To the best of my knowledge, no prior work has formalised or computed object empowerment in the manner presented here.

The chapter began by introducing the formalism of object empowerment, defining it mathematically in terms of the agent–object relationship, and showing how it can be computed in deterministic environments. The concept was first applied to a simple grid-world setting without tools, comparing *movable* and *non-movable* objects. This comparison revealed how object properties fundamentally shape empowerment landscapes, with movable objects producing broader regions of non-zero empowerment, whereas objects that cannot be modified (e.g., moved) yield empowerment values of zero across all states.

The framework was then extended to scenarios involving tools, using two representative examples: the *broom* and the *picker*. For each, both *proximal* and *distant* configurations were examined, highlighting how tool capabilities, placement, and accessibility alter the spatial structure of empowered states.

The analysis was subsequently applied to the MiniHack environment, a more complex setting. Here, *destroyable* and *indestructible* objects were considered, with additional distinction between cases where the relevant tool was already equipped and where it had to be acquired first. These scenarios illustrated how object empowerment encodes affordances arising from both agent–tool–object relationships.

Overall, the presented results demonstrate that object empowerment provides a principled, environment-agnostic means of quantifying an agent's potential to influence objects. While the analysis presented in this chapter demonstrates how object empowerment captures an agent's potential influence over individual objects, the results have primarily focused on analysing empowerment landscapes rather than using them directly to guide behaviour. An important next question is therefore how this object-centric notion of empowerment can be incorporated into learning agents so that it actively shapes exploration and decision-making during interaction with the environment. The following chapter builds on these insights to investigate how object empowerment can be integrated into RL agents to guide exploration and tool use.

Chapter 5

Empowerment-Guided Learning of Tool–Object Interactions

This chapter builds on the object empowerment formulation introduced in Chapter 4 by investigating how it can be incorporated into RL to guide behaviour during tool–object interaction. In particular, it addresses research questions RQ2 and RQ3. The first examines whether object-conditioned empowerment can function as an intrinsic signal that enables agents to autonomously discover functional tool–object interactions, while the second investigates how such empowerment regularisation influences learning dynamics in sparse-reward environments. Parts of the framework and experiments presented in this chapter are based on publication C1; the thesis extends this work by providing a more detailed methodological description and a broader analysis of the resulting learning behaviour.

This chapter applies the previously introduced formalism of object empowerment within RL tasks involving tools and objects. The goal is to use object empowerment as an intrinsic motivation that guides policy optimisation toward states where the agent's actions have a greater and more reliable influence on objects. In the tool-use learning setting, the critical addition lies in how these components are interpreted for decision-making. A central challenge in such settings is *tool discovery*. This is the process by which an agent identifies and learns that a particular tool can be used to affect an object. Building upon the definitions established in earlier chapters, the chapter evaluates the framework in a grid-world scenarios. Subsequently, the results are analysed in terms of learning performance and the emergence of the behaviors that led to such good performance. The subsequent experiments demonstrate that empowerment-regularised agents achieve faster convergence and higher performance compared to baseline RL without intrinsic motivation. Finally, the role of the horizon parameter (h) and the weighting factor ((β) see Equation 3.1.12) is examined, with a comparative analysis of empowerment values obtained under different

configurations.

5.1 Tool-Learning Framework

Having established the formalism of object empowerment in the previous chapter, this section focuses on its practical use within RL tasks involving tools. Here, object empowerment is employed as an intrinsic motivational signal that biases policy optimisation toward states where the agent's actions exert stronger and more reliable effects on objects. The state and action spaces remain as defined earlier (see Section 4.4).

5.1.1 Reward Structure

The objective in this section is to examine whether intrinsic motivational signals can guide the agent toward meaningful tool–object interactions and improve learning performance in sparse-reward environments. To this end, the extrinsic reward R is regularised by adding an intrinsic term, following the general formulation introduced in Equation (3.1.12). Four specific instantiations of the intrinsic term M are considered:

- **Object Empowerment (OE):** Intrinsic motivation is derived from the agent's potential influence over the object's state $\mathcal{S}^{\mathfrak{O}}$, measured through object sensors:

$$\hat{R} \coloneqq R + \beta_{\text{OE}}\, \mathfrak{E}^{h}_{\mathfrak{IO}}(s), \tag{5.1.1}$$

  where $\mathfrak{E}^{h}_{\mathfrak{IO}}(s)$ is the h-step object empowerment defined in Section 4.1.6. This formulation encourages the agent to seek states where its actions have a direct and predictable effect on the object, thereby promoting purposeful object-centred exploration.

- **Fully Observable Empowerment (FOE):** Intrinsic motivation is instead based on the agent's overall influence on the complete environment state $\mathcal{S}$:

$$\hat{R} \coloneqq R + \beta_{\text{FOE}}\, \mathfrak{E}^{h}_{S}(s), \tag{5.1.2}$$

  where $\mathfrak{E}^{h}_{S}(s)$ denotes empowerment computed under full observability of the environment. This formulation encourages exploration of regions where the agent's general controllability is high, rather than specifically object-focused interactions.

- **Count-Based Exploration (CBE):** Intrinsic motivation is derived from state novelty through a visitation-count bonus:

$$\hat{R} \coloneqq R + \beta_{\text{CBE}}\, \frac{1}{\sqrt{N(s)}}, \tag{5.1.3}$$

where $N(s)$ denotes the number of times state s has been visited during training. In the discrete grid-world environments considered here, the state space is sufficiently small to maintain exact visitation counts for each state. This formulation follows the principle of count-based exploration widely used in RL to encourage exploration of rarely visited states [75], independently of whether those states correspond to meaningful tool–object interactions.

- **Object Empowerment + Count-Based Exploration (OE+CBE):** Intrinsic motivation is derived from a combination of object-centred causal influence and state novelty:

$$\hat{R} \coloneqq R + \beta_{\text{OE}}\, \mathfrak{E}^{h}_{\mathfrak{T}\mathfrak{O}}(s) + \beta_{\text{CBE}}\, \frac{1}{\sqrt{N(s)}}. \tag{5.1.4}$$

  This formulation combines novelty-driven exploration with object-specific causal guidance, allowing the agent to both explore unfamiliar states and prioritise those that enable meaningful tool–object interactions.

The comparison between OE, FOE, CBE, and OE+CBE enables an evaluation of how different intrinsic motivations, and their combination, shape tool-use behaviour. FOE and CBE both encourage broad exploratory behaviour: FOE by favouring states with high overall controllability, and CBE by rewarding rarely visited states. In contrast, OE provides object-centred guidance by favouring states from which the agent can exert greater causal influence over the object. The combined OE+CBE formulation tests whether novelty-driven exploration and object-specific causal guidance can complement each other during learning. These hypotheses are examined empirically in the experiments that follow.[1]

To evaluate these hypotheses, the learning process is implemented using the A2C algorithm (see Section 3.1.5). The regularised reward $\hat{R}$, incorporating one of the above intrinsic formulations, drives exploration toward states that are either novel, highly controllable, or object-relevant depending on the choice of intrinsic signal. This makes the learning process more effective in sparse-reward environments, where useful tool-use behaviours would otherwise emerge only after prolonged exploration.

Taken together with the structured state decomposition (Equation 4.1.1) and the distinction between agent-only and tool-mediated actions (see Section 4.1.2), these reward formulations establish the basis for analysing how different intrinsic motivations influence tool-use learning in the experiments that follow.

[1] For brevity, later sections use the symbol β generically when referring to intrinsic-reward weighting parameters, although the specific coefficients differ across reward formulations.

5.2 Experiments

To evaluate the proposed tool-learning framework, two distinct grid-world environments are considered, both using the transition dynamics of tools and objects as defined earlier in Section 4.3. The first experiment, termed Tools Comparison, investigates two different tools, the picker and the broom, under conditions where the agent begins already equipped with them. In this RL setting, the agent must solve tasks that involve manipulating the state of an object, where successful interaction depends on the specific tool dynamics. The picker and broom differ in how they allow the agent to affect the object, providing two distinct forms of tool-mediated interaction. This setup enables a direct comparison of how FOE and OE guide behaviour with different tool dynamics, without requiring tool discovery. The second experiment, termed Fully Observable vs. Object Empowerment, focuses on a single tool (the picker) and examines how learning differs when the intrinsic reward is based on either FOE or OE. In this setup, the tool is not initially equipped by the agent, requiring it to first acquire the tool before using it to interact with the object. However, the agent can still interact with the object directly, even without the tool, though such interactions are generally less effective. Together, these experiments highlight both the tool-specific behavioural patterns that emerge from FOE- and OE-driven learning.

5.2.1 Experiment 1: Tools Comparison

The first experiment investigates how different tools influence the agent's ability to interact with an object. To not include the effect of tool discovery in the analysis, the agent begins each episode already equipped with a tool, positioned adjacently at its start state. The environment, illustrated in Figure 5.1, consists of a 10×10 grid with five main components:

- The agent (a robot), initialized near the tool, ensuring that tool use is possible from the beginning of the episode.
- The tool (either a broom or a picker), which mediates interaction between the agent and the object.
- A movable object (a can), which can be acted upon directly or through the tool.
- A goal location (the bin), representing the target destination for the object. The goal of the task is to move the can onto the cell containing the bin.
- A barrier of walls, which constrains direct movement and requires specific tool-mediated interactions to reposition the object effectively.

Two tool variants are considered: a broom (Figure 5.1) and a picker (Figure 5.2). Both tools follow the same transition dynamics as defined previously in Section 4.3 under Chap-

ter 4. By comparing these conditions, the experiment highlights how the different affordances provided by the tools alter empowerment landscapes and, consequently, the learned behaviors in this environment.

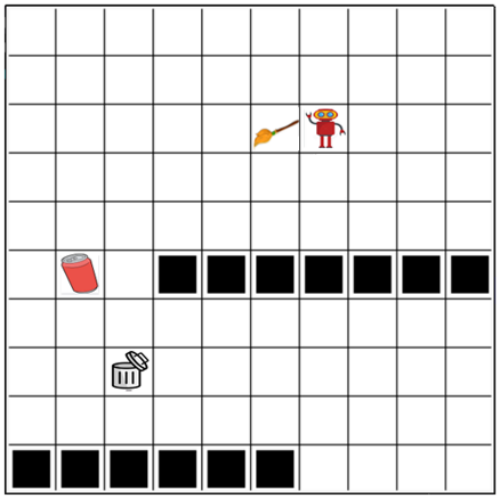


Figure 5.1: Grid-world environment with the agent equipped with a broom. The can represents the movable object and the bin represents the goal position.

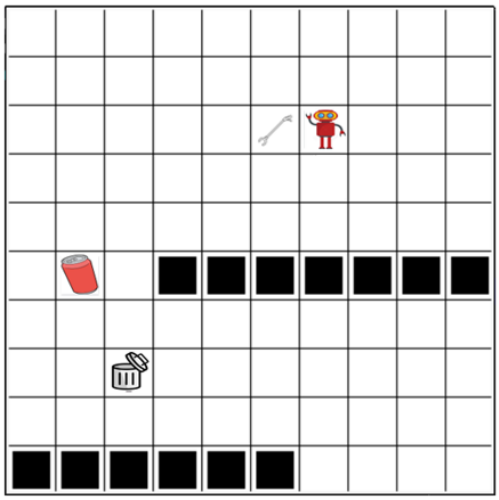


Figure 5.2: Grid-world environment with the agent equipped with a picker.

Each episode begins from the initial configuration shown in Figures 5.1 and 5.2. The objective is to move the can onto the cell containing the bin, which terminates the episode immediately upon success (the goal state is absorbing). A small step penalty of -1 is applied at each timestep to encourage efficient solutions, and a terminal reward of 0 is given when the can reaches the bin. Consequently, the optimal policy corresponds to reaching the goal in the minimum number of steps, resulting in the returns reported in this section (e.g., -10 with the picker and -12 with the broom). Episodes are capped at

a maximum of 200 steps to prevent indefinite exploration when the agent fails to solve the task.

In this experiment, empowerment is used to compare the impact that two different tools have on the state of the object. Only one of the two tools is used within each simulation. The optimal return obtained by moving the can into the waste bin with the picker is -10, while with the broom it is -12. The lower performance with the broom arises from its more restrictive interaction dynamics: it can only push the object from adjacent cells, without the ability to attach or carry it as the picker does. As a result, the agent equipped with the broom must rely on precise positional alignment to influence the can, making it less effective than the picker for completing the task. This difference can be formalised using the notion of *state-average object empowerment* defined in Equation 4.1.7. In this experiment, the average empowerment associated with the broom ($\hat{\mathfrak{E}}^3_{\mathfrak{T}_{\text{broom}}\mathfrak{O}} = 0.28\,\text{bits}$)[2] is only slightly higher than the average empowerment that would be obtained if no tool were available in the environment. Means, if the agent relied solely on its own movement actions ($\hat{\mathfrak{E}}^3_{\mathfrak{A}\mathfrak{O}} = 0.23\,\text{bits}$). This similarity reflects their limited and comparable capacity to move the can, since the broom, although equipped, only enables pushing from adjacent cells and does not introduce new, more efficient manipulation possibilities. In contrast, the picker exhibits a substantially higher state-average object empowerment ($\hat{\mathfrak{E}}^3_{\mathfrak{T}_{\text{picker}}\mathfrak{O}} = 0.52\,\text{bits}$), as its coupling mechanism enables carrying and rotating the object, resulting in a broader range of reachable object states. These empowerment values align with the learning results that follow, where the picker leads to faster convergence, consistent with its greater potential to influence the object.

Results

A comparative analysis was conducted with four RL agents: (i) a baseline *vanilla agent* (trained without intrinsic regularisation) and (ii) an OE-regularised agent, both equipped with the picker; and similarly, (iii) a vanilla agent and (iv) an OE-regularised agent, both equipped with the broom. Figure 5.3 reports the average number of episodes required for convergence across 10 independent training runs for each configuration, with error bars indicating the corresponding standard deviation. As a convergence criterion, the value of the double moving average of the return across episodes was required to be within 0.9 of the optimal return (with window sizes of 100 and 2000). Lower values denote faster convergence. The four bars represent, from left to right: (1) vanilla (picker), (2) OE-regularised (picker), (3) vanilla (broom), and (4) OE-regularised (broom). The broom clearly emerges as an "inefficient" tool: even with empowerment regularisation, the agent required signifi-

[2]The value $h = 3$ was selected as it yielded the most consistent and effective results across simulations for both tools.

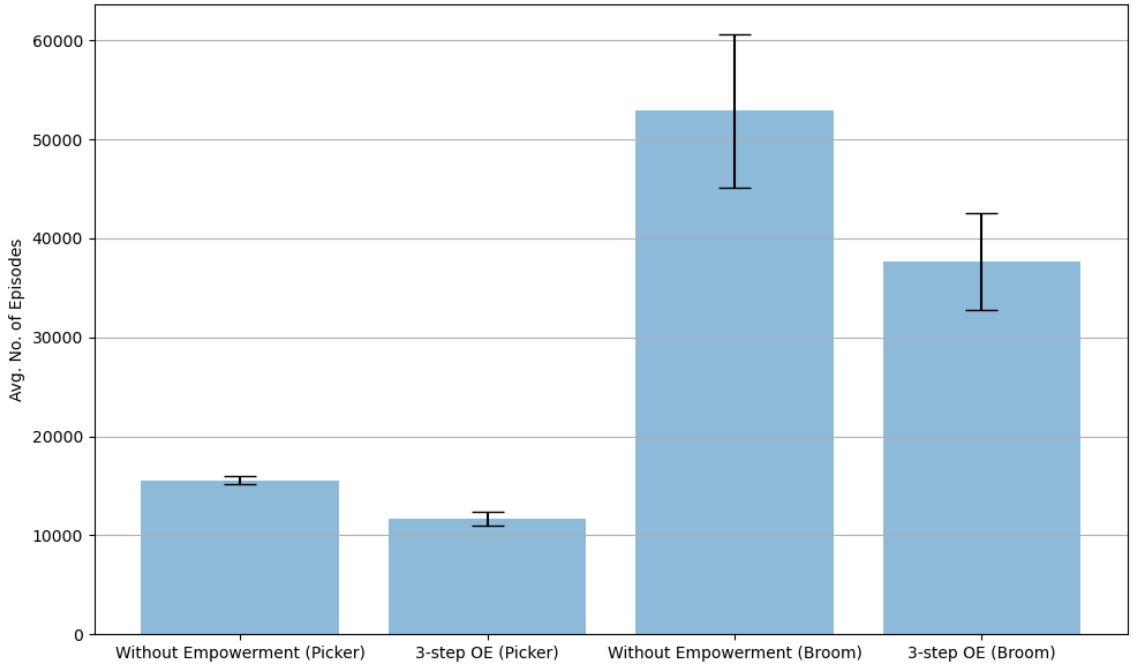


Figure 5.3: Average number of episodes until convergence in Experiment 1. Error bars indicate standard deviation across 10 runs. Lower values denote faster convergence (episodes to reach the optimal return).

cantly more episodes to learn the task compared to the picker case. Specifically, the agent using the picker with $\mathfrak{E}^{3}_{\mathfrak{T}_{\text{picker}}\mathfrak{O}}$ converged in an average of 15,555 episodes (with $\beta = 0.08$)[3], while the broom agent required 37,657 episodes on average. The OE-regularised agents consistently outperformed their vanilla counterparts for both tools, demonstrating that empowerment-guided intrinsic motivation accelerates learning. Figure 5.4 further illustrates this difference by showing the rolling-window (of size 100) average of returns for all four agents across episodes. A black line at the top of the plots indicates the optimal returns of -12 and -10 for the broom and picker cases, respectively.

Discussion

Figure 5.3 and Figure 5.4 show that the intrinsically motivated agent with the fastest convergence is the one using the picker, which also has the largest average object empowerment $\hat{\mathfrak{E}}^{3}_{\mathfrak{T}_{\text{picker}}\mathfrak{O}}$ among the two tools. On the contrary, the regularised agent with the broom took a very long time to converge and, when compared with the picker, its average OE was much smaller (0.52 bits vs. 0.28 bits). The effect of empowerment on the picker is quite

[3]The weighting factor $\beta = 0.08$ was found empirically to provide a balanced trade-off between intrinsic and extrinsic rewards for both tools.

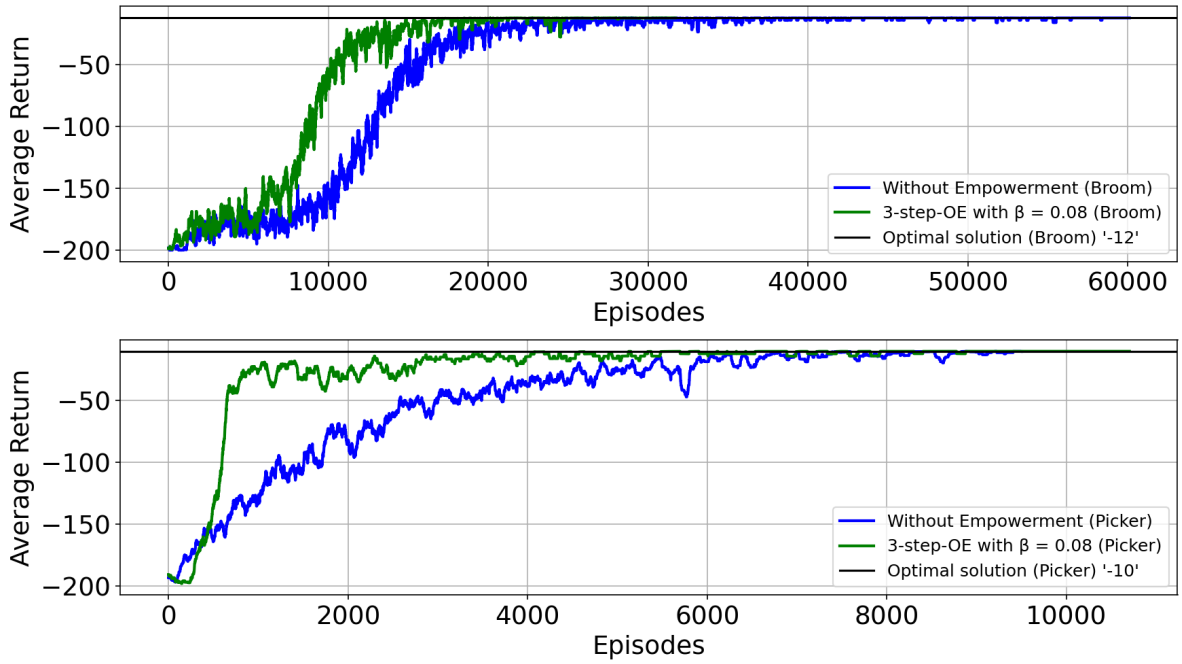


Figure 5.4: The average return received by agent in Experiment 1.

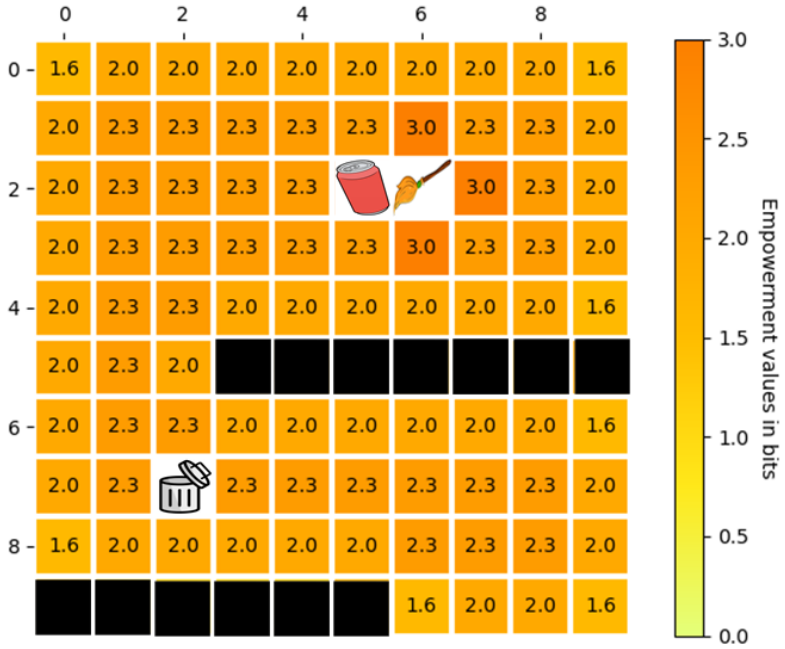
strong, because it dominates by far the other agent since the early episodes. To understand this behaviour, the OE values around the picker and the can (Figure 5.5a), and around the broom and the can (Figure 5.5b), can be examined. Here, the coloring of the states reflects their empowerment, whose values in bits are reported within the corresponding cell and in the color bars. Usually, to use a tool with a larger value of $\hat{\mathfrak{E}}^h_{\mathfrak{T}\mathfrak{O}}$ implies also a larger attraction towards the tool and object positions, which provides an additional boost to the benefits of the empowerment-based regularisation.

In principle, also FOE could be used to compare the picker and broom tools, but FOE has some undesirable properties in this regard. For instance, let us consider the $\mathfrak{E}^1_S$ landscape of Figure 5.6a, where the broom is next to the can. When the broom is replaced with the picker in Figure 5.6b, the resulting landscape remains identical. This occurs because FOE evaluates empowerment over the full environment state $\mathcal{S}$, which includes both tool and object positions. Even if the broom does not influence the can in some transitions, moving the broom itself still produces new overall states, thereby increasing the count of reachable states in $\mathcal{S}$. In contrast, OE focuses exclusively on the tool–object relationship, counting only those transitions that result in changes to the object's state. Consequently, FOE cannot distinguish between the total number of possible tool states (similar for broom and picker) and the subset of those that actually affect the can (smaller for the broom). So, when h is low, $\mathfrak{E}^h_S$ is not able to distinguish between the broom and the

(a) 1-step OE landscape with the can next to the picker.

(b) 1-step OE landscape with the can next to the broom.

Figure 5.5: 1-step OE landscapes with the can next to the picker and the broom in the experiment 1.



(a) 1-step FOE landscape with the can next to the broom.

(b) 1-step FOE landscape with the can next to the picker.

Figure 5.6: 1-step FOE landscapes with the can next to the broom and the picker in the experiment 1.

(a) 1-step FOE landscape with the can far away from the broom.

(b) 1-step FOE landscape with the can far away from the picker.

Figure 5.7: 1-step FOE landscapes with the can far away from the broom and the picker in the experiment 1.

picker (the same holds for the average FOE $\hat{\mathfrak{E}}^1_S$), while these tools have different impacts on the object. Figure 5.5a and Figure 5.5b show that this is not the case for $\mathfrak{E}^1_{\mathfrak{T}\mathfrak{O}}$, which attains a maximum value of 3 bits next to the picker and 1 bit next to the broom. Furthermore, when the can is moved far away from the broom (Figure 5.7a) and the picker (Figure 5.7b), the resulting 1-step FOE values around both tools remain identical to those observed when they are positioned next to the can (Figures 5.6a and 5.6b). This similarity arises because FOE treats each distinct configuration of the environment as a new state, regardless of whether the object's position has changed. Thus, as long as each tool movement leads to a different tool position in $\mathcal{S}$, the number of reachable states remains the same—even if the object remains unaffected. Empowerment in this case reflects the diversity of tool positions rather than the causal influence on the object.For higher horizons h, this effect becomes even more pronounced. The $\mathfrak{E}^h_S$ of states near the tools, and the corresponding average $\hat{\mathfrak{E}}^h_S$, can be even larger for the broom than for the picker. This occurs because the broom allows the agent to release the can and continue moving independently, thereby producing a larger number of distinct environment states as the tool and object occupy separate cells. In contrast, with the picker, the tool and the can remain attached until the end of the episode, reducing the number of possible distinct states in $\mathcal{S}$ and therefore the computed FOE. In other words, the broom's decoupled dynamics artificially inflate FOE values because empowerment counts state diversity, not functional influence. On the contrary, it is observed that the value of $\hat{\mathfrak{E}}^h_{\mathfrak{T}\mathfrak{O}}$ for the picker is always larger than that of the broom, and their discrepancy grows as h becomes higher.

5.2.2 Experiment 2: Comparison of Intrinsic Motivation Mechanisms

The second experiment considers a different environment designed to investigate how different intrinsic motivation mechanisms influence learning behaviour in a sparse-reward tool-use task. In particular, it compares the effect of OE, FOE, CBE, and their combination (OE+CBE) when used as intrinsic regularisers in RL (see Section 5.1.1 for definitions and reward formulation). While FOE and CBE provide exploration incentives based on global controllability and state novelty respectively, OE specifically emphasises states from which the agent can exert causal influence on the object through the tool. This experiment therefore examines whether object-centred intrinsic motivation leads to more effective discovery of tool-mediated interactions compared to broader exploration strategies. Unlike Experiment 1, which involved two tools, here only the picker is included.

Each episode begins from the initial configuration shown in Figure 5.8. In this setup, the agent is initially located in a corridor, while the picker is placed outside it. To solve the task, the agent can either approach the can directly and attempt to push it into the waste bin, or it can first exit the corridor to equip the picker and then use it to bring the can

into the waste bin. For this initial configuration, the latter strategy represents the optimal behaviour, as it allows the agent to complete the task in a minimum of 15 steps, compared to 18 steps required when pushing the can directly without the tool. This experimental design creates a deliberate exploration–exploitation trade-off: the most immediately accessible solution (pushing the can) is sub-optimal, while the globally optimal solution (using the picker) requires the agent to explore further before exploiting the reward. Consequently, this setting provides a suitable testbed for evaluating whether intrinsic motivation signals can guide exploration toward discovering the optimal tool-mediated interaction.

This aligns with the following state-average object empowerment analysis. The average agent's object empowerment without a tool is $\hat{\mathfrak{E}}^{6}_{\mathfrak{A}\mathfrak{O}} = 0.6\,\text{bits}$, which is smaller than the tool's object empowerment $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}\mathfrak{O}} = 1.4\,\text{bits}$.[4] This means that the picker has an impact on the state of the can (i.e., $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}\mathfrak{O}} > 0$), where the condition $\mathfrak{E}^{h}_{\mathfrak{T}\mathfrak{O}} > 0$ (see Section 4.1.5) formalises the assumption that the agent must be able to influence the object's state through the tool for empowerment to be meaningful. This impact is larger than what the agent can achieve unaided by a tool. In other words, there are transformations of the can's state that the agent can only realise by using the picker. The relatively small magnitude of $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}\mathfrak{O}}$ reflects the fact that only when the agent is close enough to the can can it move the latter with the picker. More generally, tools that can act on the object from greater distances exhibit larger values of $\hat{\mathfrak{E}}^{h}_{\mathfrak{T}\mathfrak{O}}$ because their empowerment is distributed over a wider spatial region. When empowerment is highly localised around the object, many states have zero values, lowering the overall state-average. In contrast, when a tool can affect the object from multiple distant positions, more states contribute non-zero values, thereby increasing the average empowerment.

Results

To evaluate the effect of the intrinsic signals defined in Section 5.1.1, five agents are trained using the same A2C learning algorithm and environment configuration: (i) a standard vanilla RL agent, (ii) an agent regularised by FOE, (iii) an agent regularised by OE, (iv) an agent using CBE, and (v) an agent combining OE and CBE. All agents share the same policy architecture and training procedure; only the intrinsic reward formulation differs.

Figure 5.9 presents the average number of episodes needed for each agent to converge towards the optimal solution across 10 independent training runs. It shows that the agent employing OE $\mathfrak{E}^{6}_{\mathfrak{T}\mathfrak{O}}$ $(\beta = 0.1)$[5] converged faster than all other agents, with an average

[4] The value $h = 6$ was selected as it yielded the most consistent and effective results across simulations. A systematic discussion of the effect of the horizon parameter h is provided later in Section 5.3.

[5] The values of h and β were selected empirically as those yielding the most stable and consistent learning performance across experiments. A systematic analysis of their roles is provided in Section 5.3.

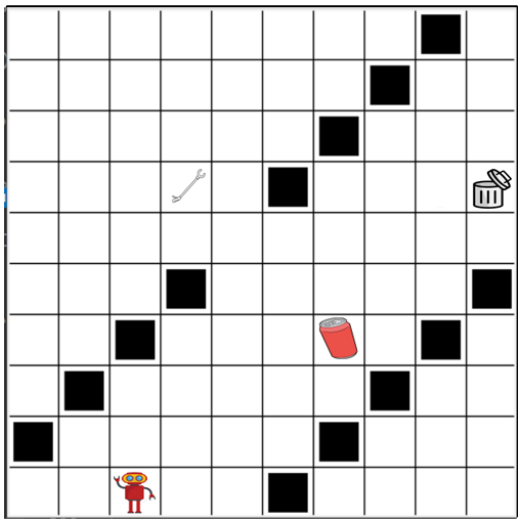


Figure 5.8: Grid-world environment for experiment 2. The picker represents the tool, the can represents the movable object, and the bin represents the goal position. The goal of the task is to move the can onto the cell containing the bin.

number of episodes equal to 11,494 ± 564. The second fastest agent was the one regularised by FOE $\mathfrak{E}_S^5$ ($\beta = 0.09$), where the horizon $h = 5$ and the corresponding β value were selected empirically as those yielding the best performance for FOE, analogous to $h = 6$ and $\beta = 0.1$ chosen for OE, with convergence equal to 12,559 ± 1,406. For CBE, the intrinsic weight $\beta = 0.08$ was selected through systematic empirical tuning analogous to OE and FOE (details omitted for brevity), yielding convergence of 17,692 ± 1,032, while for the combined OE+CBE agent the same individually selected values were used ($\beta_{\text{OE}} = 0.1$, $\beta_{\text{CBE}} = 0.08$) without performing an exhaustive joint parameter search, resulting in 13,514 ± 1,120. The agent using the combined OE+CBE formulation ($\beta_{\text{OE}} = 0.1$, $\beta_{\text{CBE}} = 0.08$) also achieved faster convergence than the novelty-based CBE agent alone, suggesting that combining novelty-driven exploration with object-centred intrinsic guidance improves learning efficiency compared to novelty-based exploration by itself. The agent using only CBE ($\beta = 0.08$) converged substantially faster than the vanilla RL agent, indicating that novelty-driven exploration alone can already alleviate part of the sparse-reward exploration difficulty.

Finally, the standard RL agent with no intrinsic regularisation required the largest number of episodes to converge (37,200 ± 3,400). In addition, the vanilla A2C agent exhibits substantial variability during the early stages of learning, reflecting unstable and inconsistent exploration in the absence of intrinsic guidance. These results confirm that OE-based intrinsic motivation provides more directed and effective exploration by biasing the agent towards states that enable meaningful tool–object interactions. In contrast, FOE promotes broader exploration based on overall controllability of the environment, while

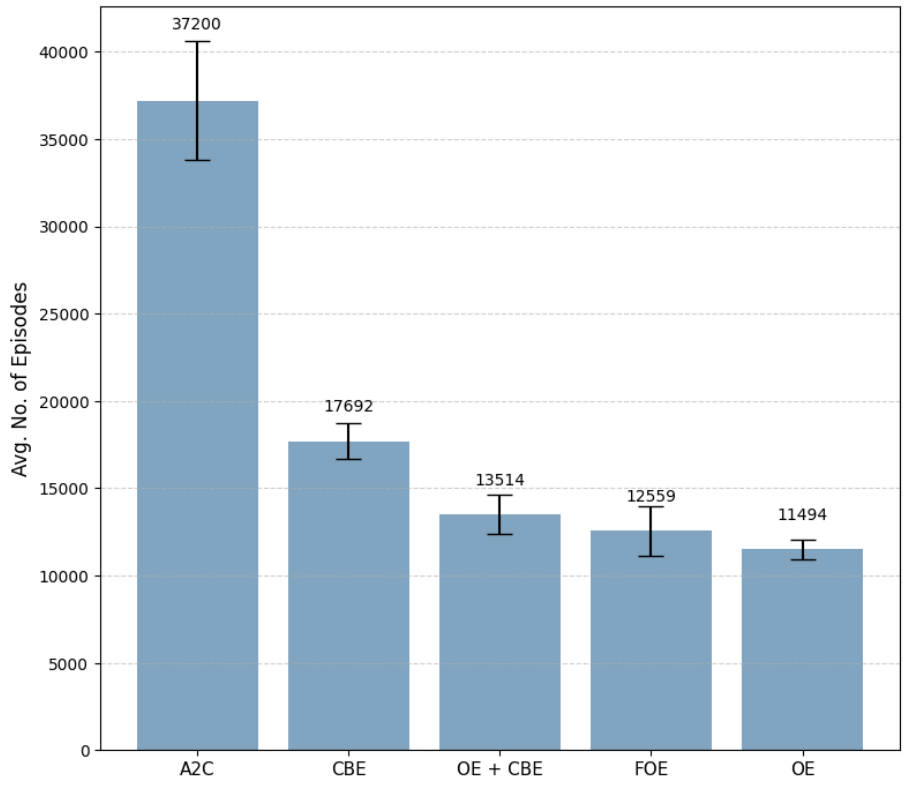


Figure 5.9: Average number of episodes required for convergence in Experiment 2. Error bars indicate standard deviation across 10 runs.

CBE encourages exploration of rarely visited states without explicitly prioritising object-relevant interactions. Figure 5.10 shows the learning curves for the five agents, reporting the average return per episode (rolling window of size 100) across the training runs.

Discussion

The results have shown that the intrinsically motivated agents (FOE, OE, CBE, and OE+CBE) perform better than the standard RL agent, confirming that regularisation methods, in conjunction with an appropriate intrinsic motivation signal, can be useful to contrast the sparsity of reward characterising tasks such as the one considered (i.e., here the agent receives a reward different from -1 only when the task is completed). But not all intrinsic motivations are the same. The results show that FOE and OE have distinct effects on the agent's performance. This difference can be understood by examining how these intrinsic motivations shape agent behaviour in terms of interactions with the tool and the object present in the environment. As FOE measures the number of future states reachable by the agent, the ability to change the state of the picker when equipped increases its value.

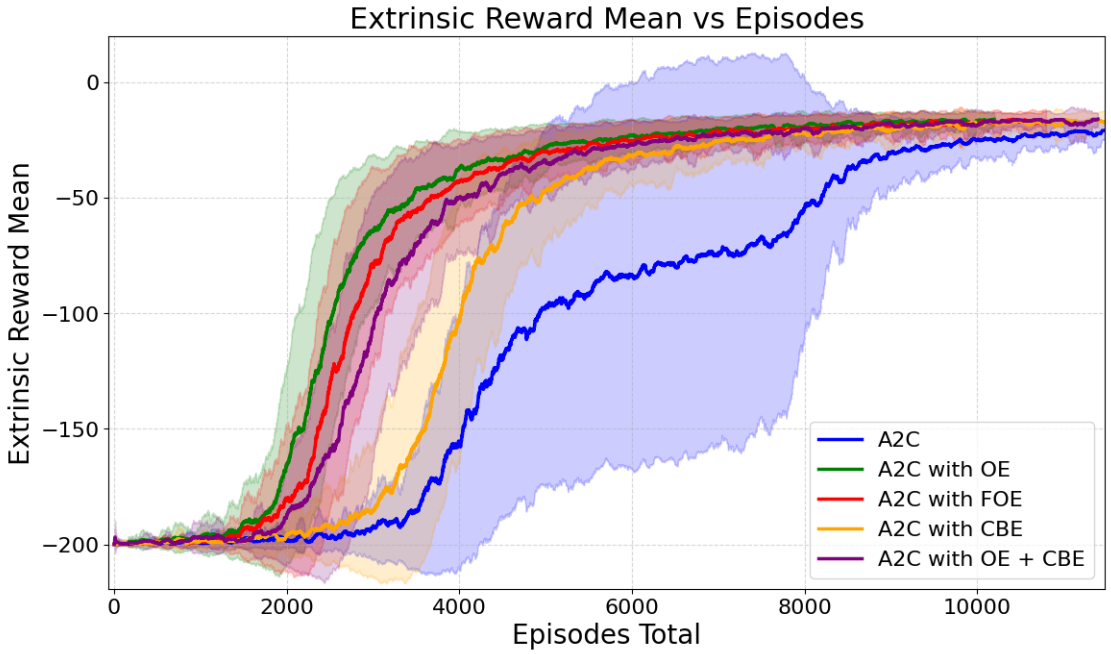


Figure 5.10: The average return obtained by the five agents in Experiment 2.

This is illustrated in Figure 5.11 and Figure 5.12, which report snapshots of empowerment landscapes for fixed tool and object positions. In Figure 5.11a, $\mathfrak{E}_S^1$ attains particularly high values in states surrounding the tool, ranging from 2.6 to 3 bits. Consequently, incorporating $\mathfrak{E}_S^1$ into the reward makes these states attractive, guiding the agent towards the tool and facilitating tool acquisition. Increasing the FOE horizon to $h = 5$ (Figure 5.11b) results in higher empowerment values, with $\mathfrak{E}_S^5$ reaching up to 7 bits near the picker. In this case, high empowerment values are distributed across states surrounding both the picker and the can, since five steps are sufficient either to equip and manipulate the tool or to move the can directly using the agent's actions. As a result, maximising $\mathfrak{E}_S^5$ can also encourage movement towards the can. However, although FOE incentivises the agent to reach and equip the tool, it does not provide additional guidance towards the can once the tool is acquired. This is because FOE captures the total number of reachable future states of the system, including tool dynamics, without distinguishing whether these states meaningfully affect the object. Consequently, after increasing its reachable state space by equipping the tool, the agent receives no further intrinsic signal to prioritise states that bring the tool closer to the can.

In contrast to empowerment-based signals, CBE encourages the agent to visit rarely encountered states by assigning higher intrinsic reward to less frequent state configurations. This produces a broad exploration strategy that improves coverage of the state space without explicitly prioritising tool–object interactions. As a result, CBE accelerates learning

relative to the vanilla agent but remains less effective than the other intrinsic motivation mechanisms considered, as it lacks a mechanism to guide behaviour toward states that are causally relevant for solving the task. This limitation explains its comparatively slower convergence despite improving exploration efficiency.

OE measures the number of positions the agent is able to move the can to. Hence, it is observed that in the grid world the states with the largest $\mathfrak{E}^1_{\mathfrak{IO}}$ of 1 bit are those located in the cells next to the can, whereas all other states have $\mathfrak{E}^1_{\mathfrak{IO}}$ equal to 0 bits (see Figure 5.12a). This occurs solely due to the agent's body actions, as in these states the agent has not yet equipped the tool. However, if the agent would be only influenced by $\mathfrak{E}^1_{\mathfrak{IO}}$, it may go to the can before picking the tool, without executing the optimal solution. Interestingly, when the OE horizon h is sufficiently large, the corresponding action sequences originating from states near the picker encompass not only the steps required for the agent to reach and equip the tool, but also the subsequent actions that enable it to manipulate the can using the equipped picker. In other words, a larger horizon allows OE to capture the extended causal chain linking tool acquisition to object manipulation, which shorter horizons fail to represent. Since the picker enables a large influence on the can's state, in this case $\mathfrak{E}^h_{\mathfrak{IO}}$ can be higher next to the picker than next to the can. This phenomenon is visible in Figure 5.12b, where the maximum value of $\mathfrak{E}^6_{\mathfrak{IO}}$ next to the picker is 4.3 bits, while it is 3.6 bits next to the can. It can be observed that OE with a sufficiently large horizon h not only steers the agent toward the tool, but also, once the tool is equipped, continues to guide behaviour toward the can, because OE will be the largest around the can when the tool is equipped. The combined OE+CBE formulation integrates these two complementary effects: CBE promotes broad exploration of the environment, increasing the likelihood of discovering the tool, while OE subsequently focuses behaviour on object-relevant interactions. This combination results in improved performance compared to CBE alone, as observed in the results, highlighting the benefit of combining general exploration incentives with task-specific intrinsic guidance.

To confirm that the usage of FOE and OE as intrinsic rewards encourages the agent to approach the picker, Figure 5.13 shows the proportion of time during which the picker is equipped by the agent, averaged across 10 runs for each episode. The results show that at the initial stage of learning, when compared with a standard RL agent with no regularisation (orange curve), the agents that employ FOE $\mathfrak{E}^5_S$ and OE $\mathfrak{E}^6_{\mathfrak{IO}}$ as intrinsic rewards (fuchsia and purple curves, respectively) spend more time on average with the tool equipped. Since once the tool is picked this remains equipped until the end of the episode, the plots indicates that the intrinsically motivated agents find the tool earlier than the standard agent. Furthermore, the agent regularised by FOE $\mathfrak{E}^5_S$ equips the tool sooner that the agent regularised by OE $\mathfrak{E}^6_{\mathfrak{IO}}$, likely because FOE's global intrinsic signal initially promotes movement toward any state change, including those involving the tool, more

(a) 1-step FOE.

(b) 5-step FOE.

Figure 5.11: 1- and 5-step FOE in the third experiment.

(a) 1-step OE.

(b) 6-step OE.

Figure 5.12: 1- and 6-step OE in the third experiment.

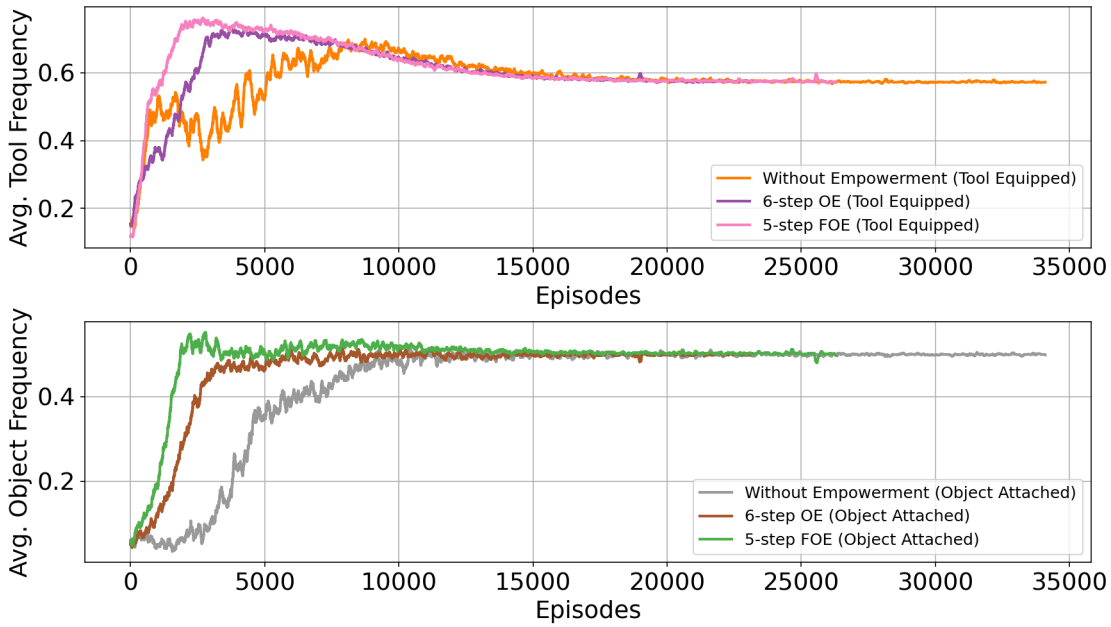


Figure 5.13: The average proportion of time steps during which the agent has the picker equipped (fuchsia, purple, and orange curves) and during which the can is attached to the picker (green, brown, and grey curves) in the third experiment.

strongly than the object-specific guidance provided by OE. Figure 5.13 also reports the average proportion of time, after the tool is equipped, spent by the intrinsically motivated agents while the can is attached to the picker. The figure shows that the agents employing FOE $\mathfrak{E}^5_S$ (green curve) and OE $\mathfrak{E}^6_{\mathfrak{T}\mathfrak{O}}$ (brown curve) spend more time on average with the can attached to the picker than the vanilla agent (grey curve). Since once the can is attached to the picker, it remains attached until the end of the episode, the latter finding shows that the intrinsically motivated agents attach the can before than the vanilla agent. Moreover, the agent motivated by $\mathfrak{E}^5_S$ picks the can before the agent motivated by $\mathfrak{E}^6_{\mathfrak{T}\mathfrak{O}}$. These observations indicate that the combined attractivity of both the tool and the object contributes to the improved learning performance of the agents regularised by $\mathfrak{E}^5_S$ and $\mathfrak{E}^6_{\mathfrak{T}\mathfrak{O}}$, when compared with the standard agent. The superior performance of the OE-regularised agent (Figure 5.9) can be attributed to the fact that FOE tends to prioritise exploration toward regions of high overall empowerment rather than guiding behaviour toward the goal, which typically lies in a region of low empowerment [139].

Across the two experiments, empowerment-based regularisation consistently improved learning over the vanilla A2C baseline, but the *form* of intrinsic motivation mattered. In Experiment 1, tool efficacy translated directly into faster convergence: the picker's higher object empowerment $\hat{\mathfrak{E}}^3_{\mathfrak{T}_{\text{picker}}\mathfrak{O}}$ aligned with markedly better performance than the broom,

illustrating how OE reveals object-relevant affordances and predicts downstream learning speed. In Experiment 2, FOE, OE, and CBE all accelerated learning relative to vanilla RL, yet they induced distinct behavioural biases: FOE tended to keep the agent near the tool and encourage repeated tool interaction, CBE promoted broad exploration of the state space without task-specific focus, whereas OE steered behaviour toward object-relevant state changes, yielding faster convergence and higher asymptotic returns. The combined OE+CBE formulation further demonstrates that integrating general exploration incentives with object-centred guidance can improve performance, suggesting a complementary relationship between different intrinsic motivation mechanisms.

Beyond comparing the effects of different empowerment formulations, it is also important to examine how intrinsic motivation behaves under different parameter configurations. In particular, the empowerment horizon h controls the temporal depth of the intrinsic signal (thereby reshaping the landscape and its attractors), while the weighting factor β calibrates the balance between intrinsic and extrinsic rewards. The following section systematically analyses the influence of h and β on learning performance and behavioural outcomes for both FOE and OE.

5.3 Influence of Horizon (h) and Weighting Factor (β)

The previous experiments demonstrated that incorporating intrinsic motivation in the form of Tool's Object Empowerment (TOE)[6] and Fully Observable Empowerment (FOE) improves the performance of a baseline RL agent. In this section, the focus shifts to understanding how the choice of the horizon parameter h and the weighting factor β influences learning. Specifically, the analysis investigates how varying h alters the temporal depth of empowerment, and how different values of β balance the contribution of intrinsic and extrinsic rewards. Together, these parameters determine not only the rate of convergence but also the strategies that agents adopt when interacting with tools and objects.

To ensure comparability with the previous results, the same environment introduced in Section 5.2.2 (shown in Figure 5.8) is used here. The agent is trained with the A2C algorithm (see Section 3.1.5), in line with the experimental setup applied throughout this chapter. Throughout this section, the performance comparison is made based on the the average number of episodes needed for each agent to converge towards the optimal solution.

[6]Throughout this section, the term TOE is used interchangeably with OE (Object Empowerment).

5.3.1 h-step Fully Observable Empowerment

FOE provides a natural baseline for studying the influence of the horizon parameter h, because it evaluates the agent's potential causal influence over *all* observable environmental states without the additional compositional constraints of tool–object interactions. Extending the empowerment horizon from one step to multiple steps makes the intrinsic signal dependent on longer action sequences and thus more sensitive to deeper temporal dependencies in the transition dynamics. This introduces a fundamental trade-off: while larger values of h allow the agent to anticipate the long-term consequences of its actions, the resulting intrinsic feedback becomes less directly connected to the agent's immediate situation, as it reflects potential influence over more distant future states. For completeness and to support direct comparison across horizons, the 1-step and 5-step FOE landscapes are reproduced here in the same plotting style as the remaining FOE landscapes, even though their qualitative behaviour was introduced earlier in Section 5.2.2.

Figure 5.14 presents the 1-step FOE landscape, where the maximum empowerment values (3.0 bits) are concentrated in cells adjacent to the tool. Figure 5.15 shows the average number of episodes required to solve the task under 1-step FOE. Relatively higher β (0.15–0.21) accelerates convergence, but excessively large β (0.30–0.40) fails to solve the task within the cutoff. Conversely, very small β (0.14) produces an intrinsic signal too weak to guide the agent toward task-relevant regions of the environment, resulting in inefficient or unproductive exploration.

Figure 5.16 presents the 2-step FOE landscape, which extends further to capture two-step consequences. The maximum empowerment values (5.0 bits) remain near the tool. Figure 5.17 shows performance under 2-step FOE. Best convergence occurs for $\beta \approx 0.08$–0.14; very small β (0.04) under-explores, while larger β values (e.g., 0.20) fail to solve the task.

Figure 5.18 presents the 3-step FOE landscape, where empowerment expands further with maximum values of 6.0 bits near the tool. Figure 5.19 shows performance. Stable convergence occurs for $\beta \approx 0.10$–0.15. Too little intrinsic signal ($\beta = 0.07$) slows learning, while $\beta \gtrsim 0.16$ fails to solve the task.

Figure 5.20 presents the 4-step FOE landscape, with maximum values of 6.5 bits near the tool. Figure 5.21 shows performance, where $\beta \in \{0.07, 0.09, 0.13\}$ yields robust convergence. Very small β (0.04) slows learning, while $\beta \gtrsim 0.14$ destabilises performance.

Figure 5.22 presents the 5-step FOE landscape, where empowerment covers broader regions and captures more distant action chains. The maximum values (7.0 bits) remain near the tool. Figure 5.23 shows performance, with $\beta \approx 0.06$–0.10 yielding best results. Both extremes (too low or too high) harm convergence.

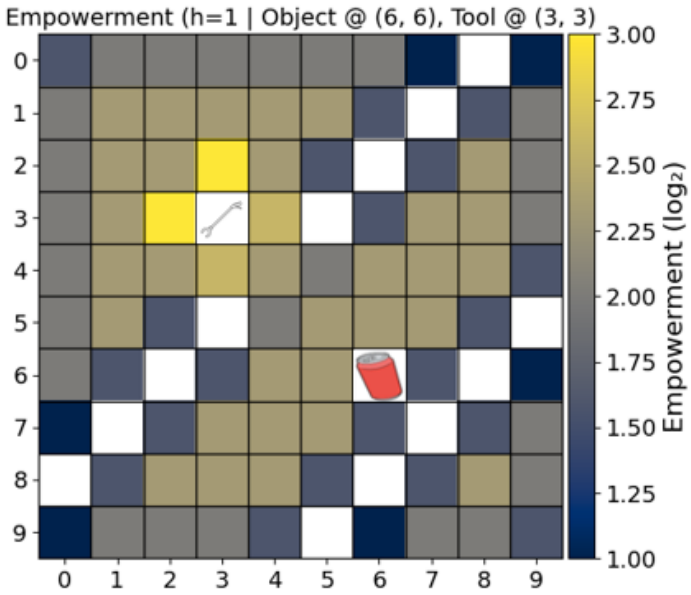


Figure 5.14: 1-step FOE landscape.

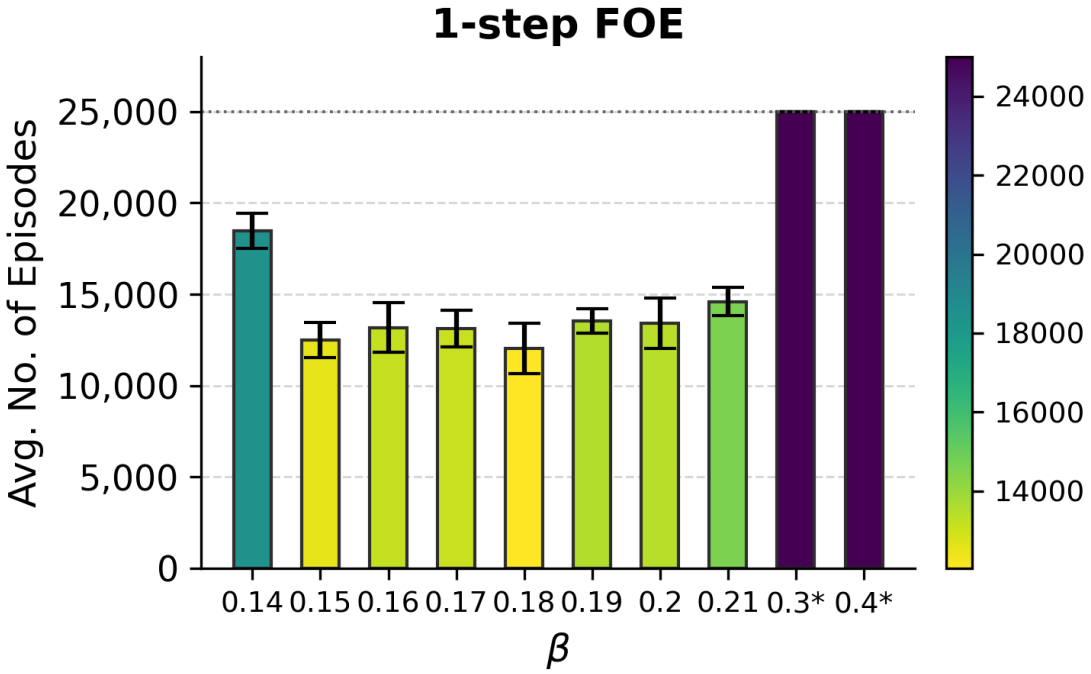


Figure 5.15: 1-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 25,000 episodes.

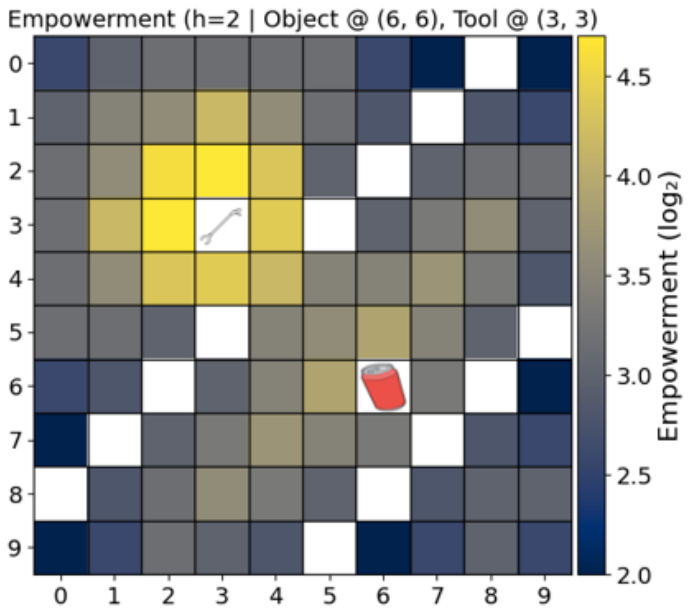


Figure 5.16: 2-step FOE landscape.

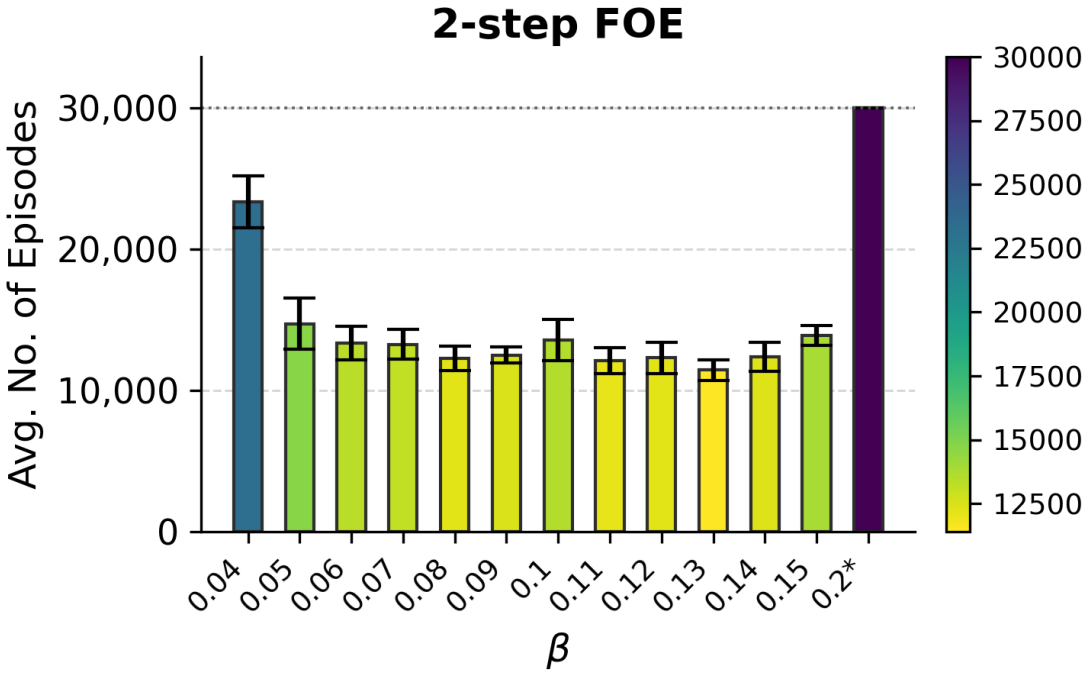


Figure 5.17: 2-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 30,000 episodes.

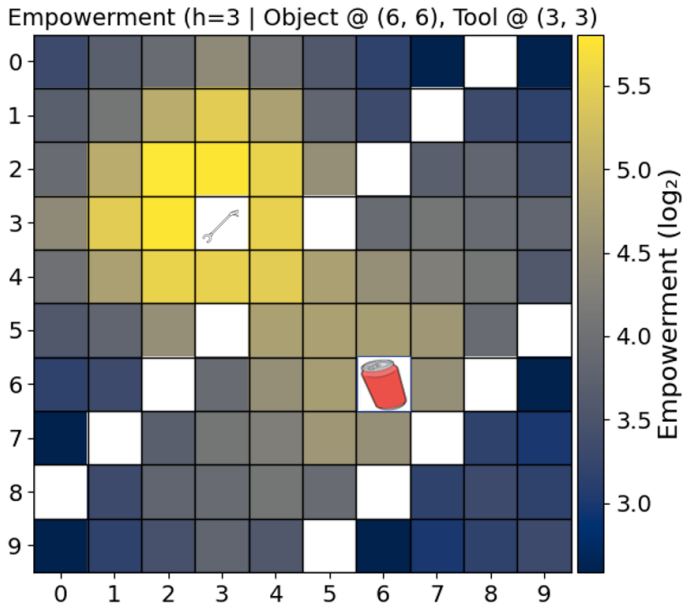


Figure 5.18: 3-step FOE landscape.

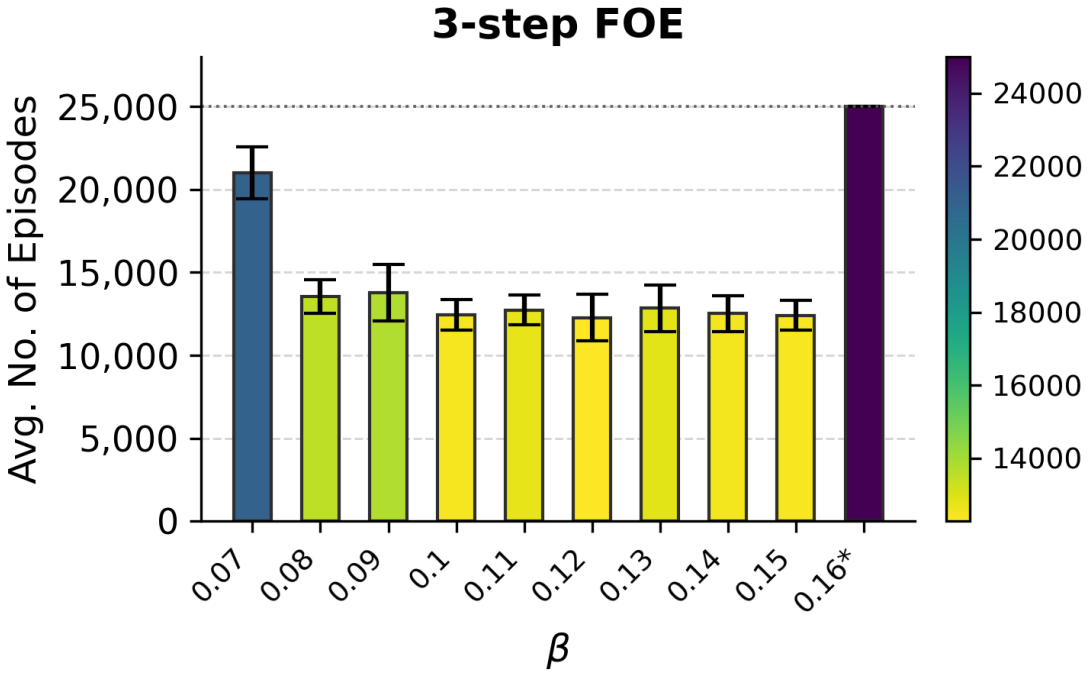


Figure 5.19: 3-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 25,000 episodes.

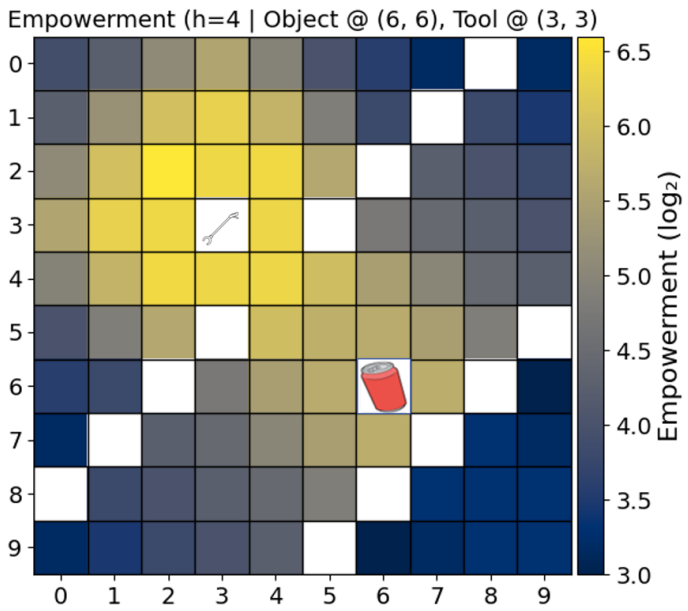


Figure 5.20: 4-step FOE landscape.

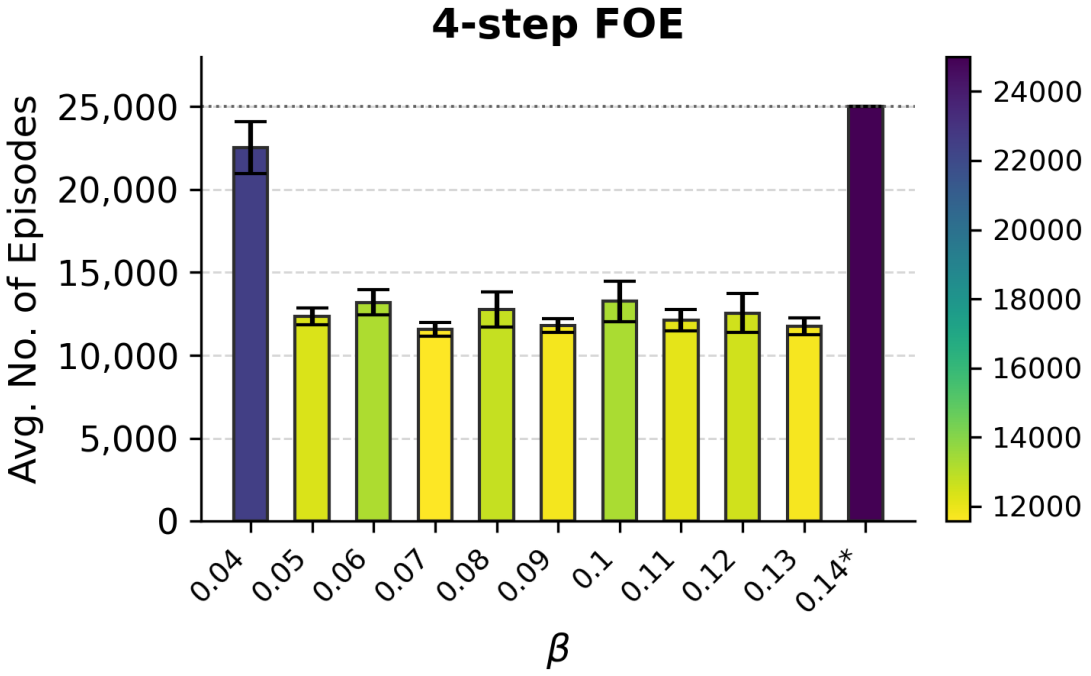


Figure 5.21: 4-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 25,000 episodes.

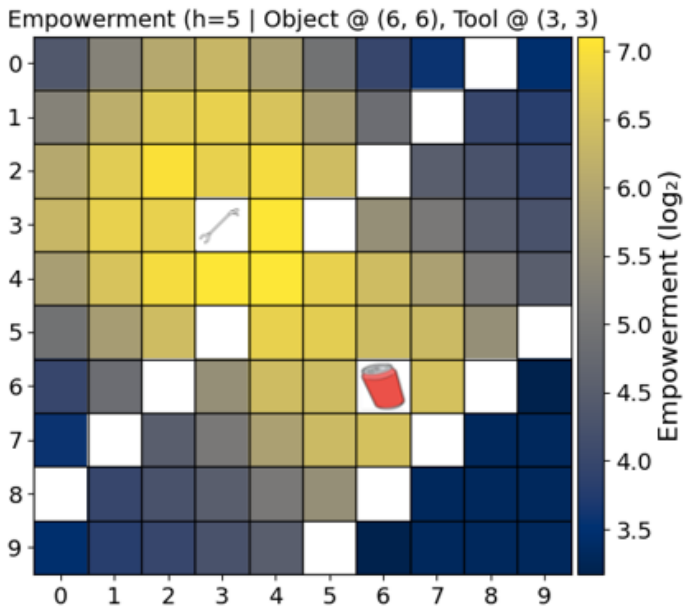


Figure 5.22: 5-step FOE landscape.

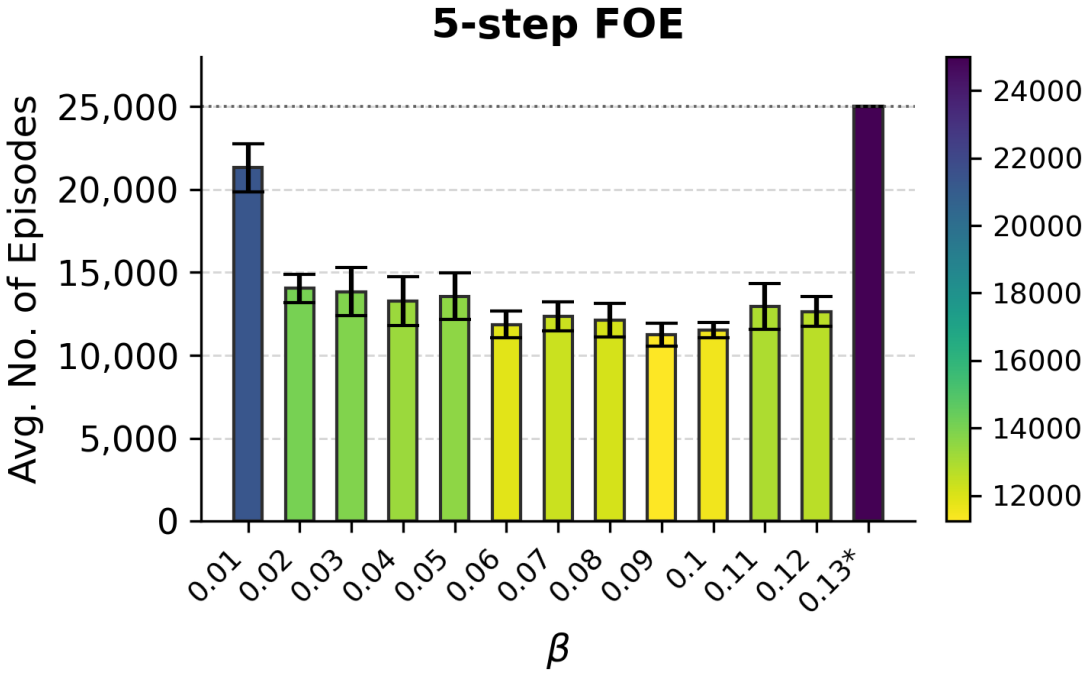


Figure 5.23: 5-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 25,000 episodes.

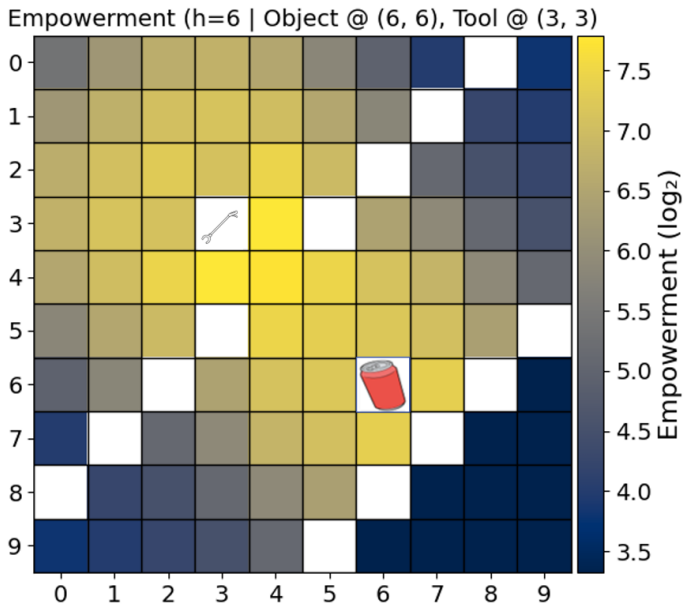


Figure 5.24: 6-step FOE landscape.

Figure 5.24 presents the 6-step FOE landscape, where the maximum values (7.5 bits) are near the tool. Figure 5.25 shows performance, where $\beta = 0.06$ yields the best results. Very small β (0.01) under-explores, while larger values (e.g., $\beta = 0.12$) fail to solve the task.

Figure 5.26 presents the 7-step FOE landscape, where the maximum empowerment values (8.0 bits) remain near the tool. Figure 5.27 shows performance, with $\beta = 0.08$ yielding the best results. Very small β (0.03) under-explores, while $\beta \geq 0.11$ fails to solve the task.

Figure 5.28 presents the 8-step FOE landscape, where empowerment reaches maximum spread, with high values (8.5 bits) across most accessible states. Figure 5.29 shows performance. Stable convergence occurs for $\beta \approx 0.06$–0.09. Too little intrinsic signal ($\beta = 0.009$) slows learning, while $\beta \geq 0.11$ fails to solve the task.

Summary across horizons:

Two consistent patterns emerge. (i) For every h, an intermediate β band yields the fastest learning; both under- and over-weighting empowerment are detrimental. (ii) The *optimal* β systematically *decreases* with h: short horizons require stronger intrinsic shaping to encourage exploration, whereas long horizons accumulate sufficient intrinsic signal and

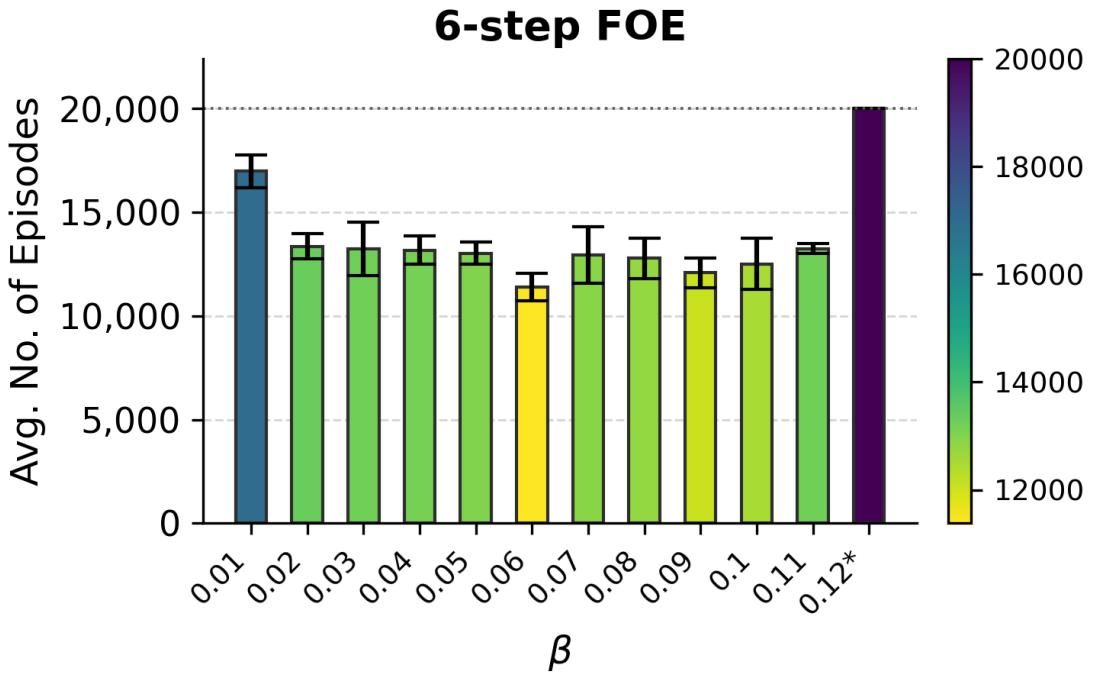


Figure 5.25: 6-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 20,000 episodes.

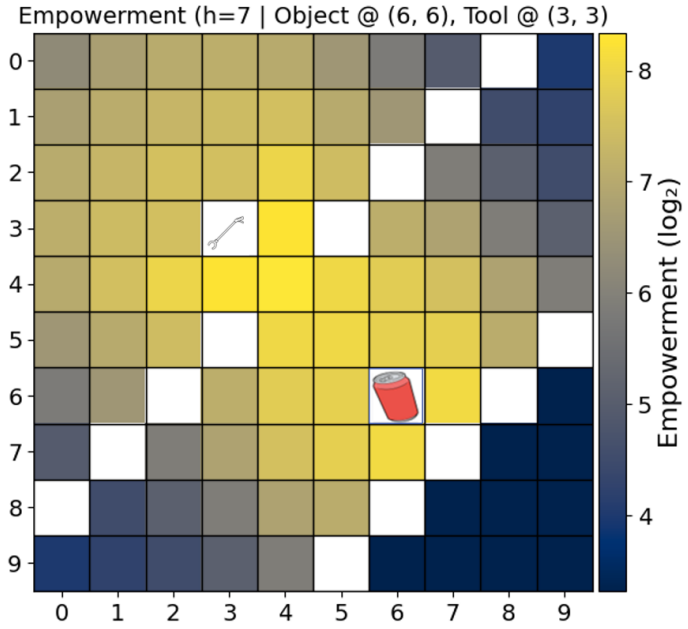


Figure 5.26: 7-step FOE landscape.

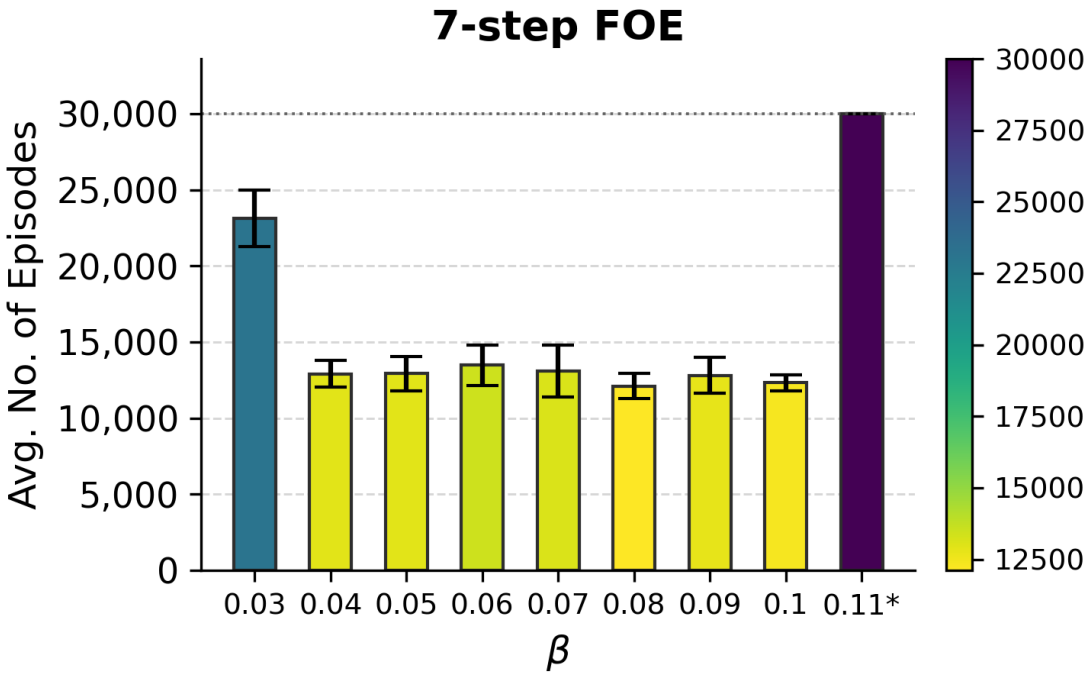


Figure 5.27: 7-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 30,000 episodes.

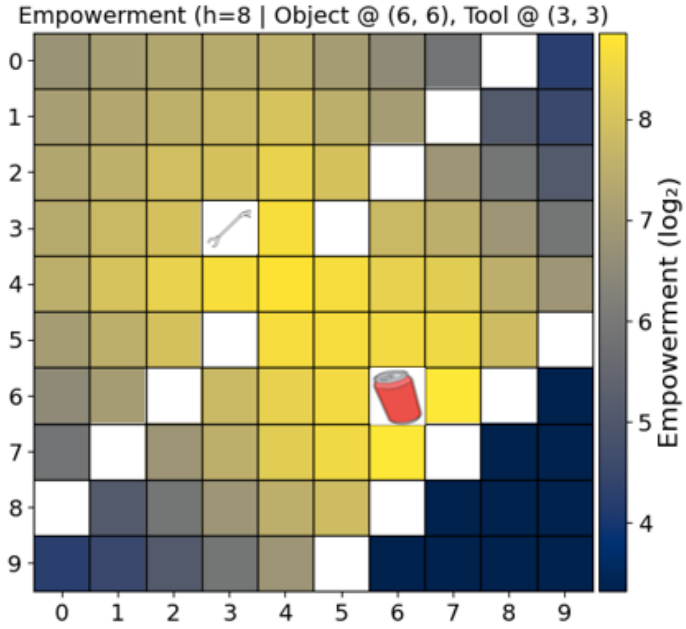


Figure 5.28: 8-step FOE landscape.

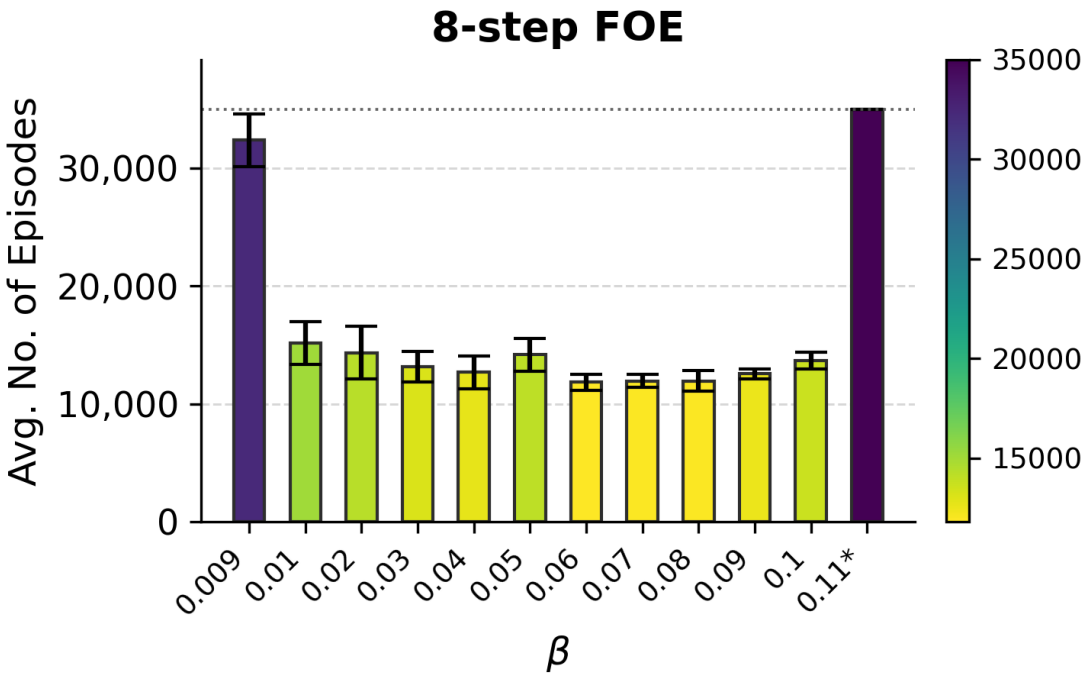


Figure 5.29: 8-step FOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 35,000 episodes.

therefore demand a smaller weight to avoid overpowering the extrinsic objective. For example, $\beta = 0.18$ produced the best performance under 1-step FOE, but failed entirely for horizons $h \geq 3$. Conversely, $\beta = 0.14$ underperformed at $h = 1$, but became one of the best-performing values at $h = 3$. This illustrates that h (temporal depth) and β (relative strength) must be tuned jointly for effective learning. Table 5.2 provides the best β values found that yielded the fastest convergence for each horizon h under FOE.

Table 5.1: β values that yielded the fastest convergence for each horizon h under FOE.

h=1	h=2	h=3	h=4	h=5	h=6	h=7	h=8
0.18	0.13	0.12	0.07	0.09	0.06	0.08	0.06

5.3.2 h-step Tool's Object Empowerment

Tool–Object Empowerment (TOE) evaluates the agent's potential causal influence specifically on the state of the object, mediated by the tool's dynamics, rather than on the entire environment as in FOE. This compositional formulation reshapes the empowerment landscape and, consequently, the way intrinsic rewards shape learning. While both FOE and TOE can encourage the discovery and use of tools, TOE does so by explicitly linking

empowerment to the agent's capacity to manipulate the object through the tool, thereby aligning intrinsic motivation more directly with task-relevant affordances. This section examines how the horizon h and the weighting factor β interact under TOE. For completeness and to support direct comparison across horizons, the 1-step and 6-step TOE landscapes are reproduced here in the same plotting style as the remaining TOE landscapes, even though their qualitative behaviour was introduced earlier in Section 5.2.2.

Figure 5.30 and Figure 5.32 show the 1 and 2-step TOE landscapes, respectively. In contrast to FOE, where empowerment expands uniformly with horizon, TOE concentrates high values in states where the agent can directly affect the object. At $h = 1$, the states with the largest empowerment value (1.0 bits) are those adjacent to the object, while all other states yield 0. As the horizon increases, the agent can interact with the object from greater distances. However, under 1- and 2-step TOE, the agent may move directly toward the object before equipping the tool, thereby failing to follow the optimal solution.

The learning consequences of using TOE as a regulariser are shown in Figure 5.31 and Figure 5.33, which report the number of episodes required to solve the task for varying β across horizons $h = 1$ and 2. Compared with the vanilla RL baseline, no marked improvements in convergence were observed at these short horizons. This is not merely due to the small magnitude of empowerment at $h = 1$ and 2, but rather because the corresponding intrinsic signal is highly localised around the object and therefore provides limited guidance toward states from which the object can actually be manipulated. At $h = 1$, the agent converged only for intermediate β values (0.1–0.3), while very large β (≥ 0.4) prevented task completion within the cutoff. With 2-step TOE, the agent failed to solve the task even for $\beta \geq 0.3$.

Since, under 1- and 2-step TOE, the agent may move directly toward the object before equipping the tool, the horizon has been increased to $h = 6$. When the TOE horizon is sufficiently long (e.g., $h \geq 6$), action sequences originating near the tool naturally include both the steps required to equip the tool and the subsequent moves needed to manipulate the object. As a result, the empowerment landscape now exhibits its highest values in the vicinity of the tool, because these states are the starting points from which the object can be influenced through the tool (see Figure 5.34). Once the tool is equipped, however, the intrinsic drive shifts toward the object, as the agent's influence on the object's state becomes maximised through tool use. This sequential transition, from tool-directed to object-directed attraction, captures how TOE structures intrinsic motivation across stages of interaction.

The learning consequences of TOE are shown in Figure 5.35, which report the number of episodes required to solve the task for varying β under horizon $h = 6$. With $h = 6$, convergence was accelerated for β in the range (0.08–0.13), while excessively large β ($\geq$

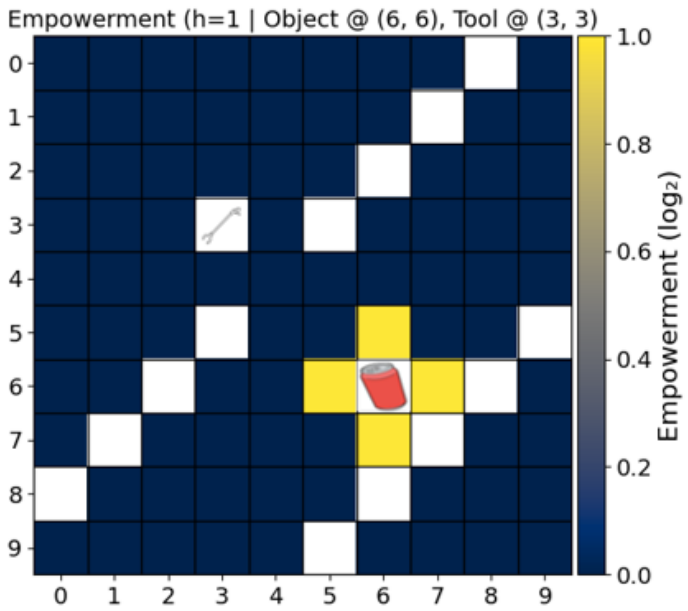


Figure 5.30: 1-step TOE landscape.

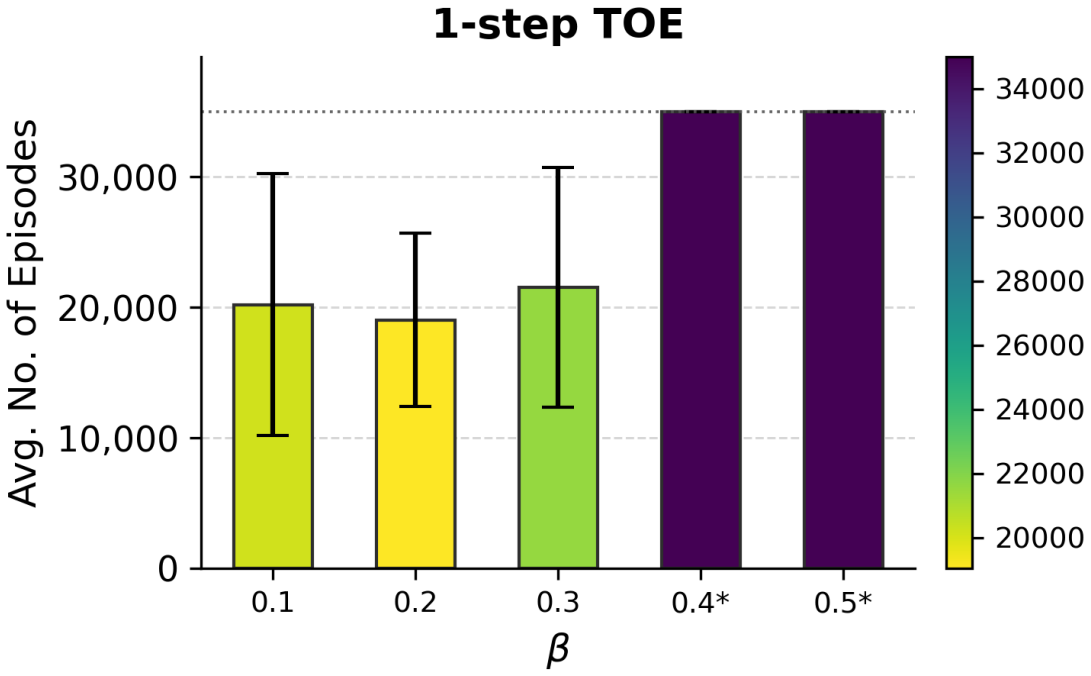


Figure 5.31: 1-step TOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 25,000 episodes.

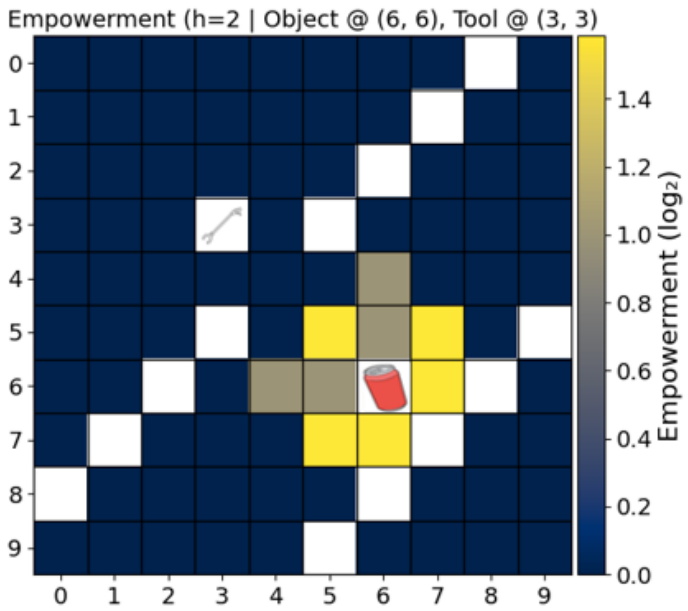


Figure 5.32: 2-step TOE landscape.

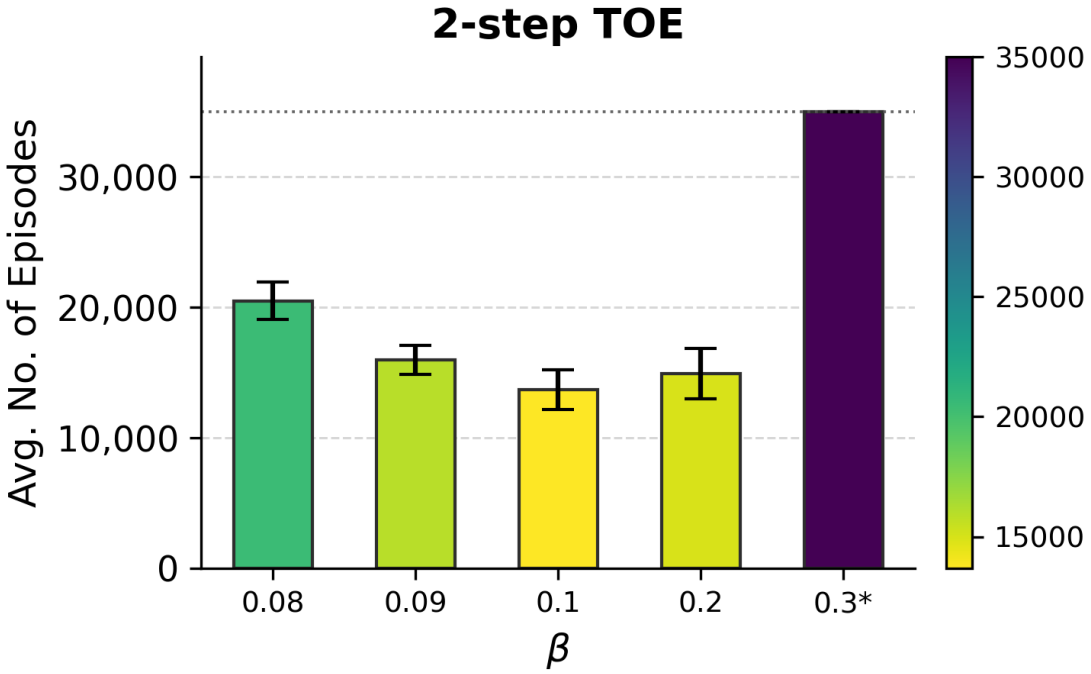


Figure 5.33: 2-step TOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 30,000 episodes.

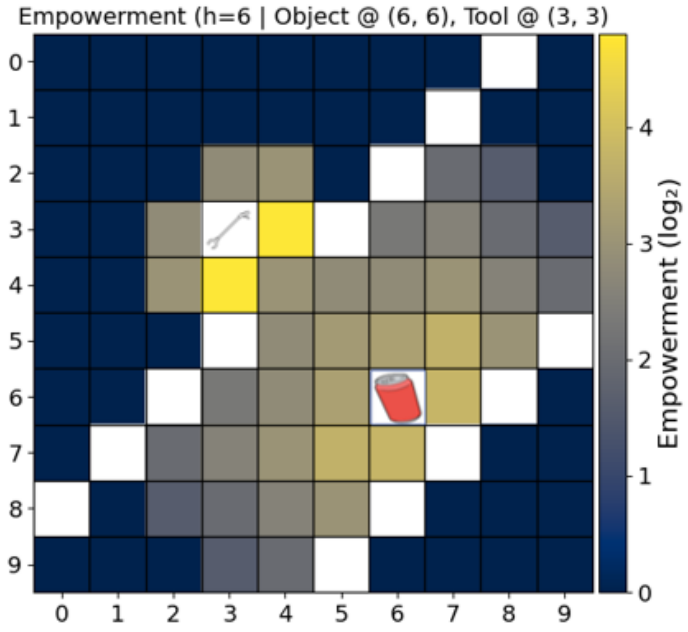


Figure 5.34: 6-step TOE landscape.

0.17) again hindered learning.

Figure 5.36 presents the 7-step TOE landscape, where the maximum empowerment values (5.0 bits) remain near the tool. Figure 5.37 reports the number of episodes required to solve the task for varying β under horizon $h = 7$. With $h = 7$, convergence was accelerated for β in the range (0.11–0.13), while excessively large β (≥ 0.17) again hindered learning.

Similar trend has been observed in 8-step TOE. Figure 5.38 presents the 8-step TOE landscape, where the maximum empowerment values (≈ 5.5 bits) remain near the tool. Figure 5.39 shows performance. Stable convergence occurs for $\beta \approx 0.07$–0.12. Too little intrinsic signal ($\beta = 0.02$) slows learning, while $\beta \geq 0.17$ fails to solve the task.

Summary across horizons:

TOE produces more localised and affordance-specific empowerment landscapes compared to FOE. At short horizons ($h = 1$ and 2), TOE focuses the agent directly on the object, sometimes bypassing the tool altogether and therefore failing to produce effective solutions. At longer horizons ($h \geq 6$), however, TOE captures extended action sequences that include equipping and using the tool, with empowerment peaks emerging near the tool and then only guiding the agent toward effective tool-object interaction. Notably, TOE exhibits a

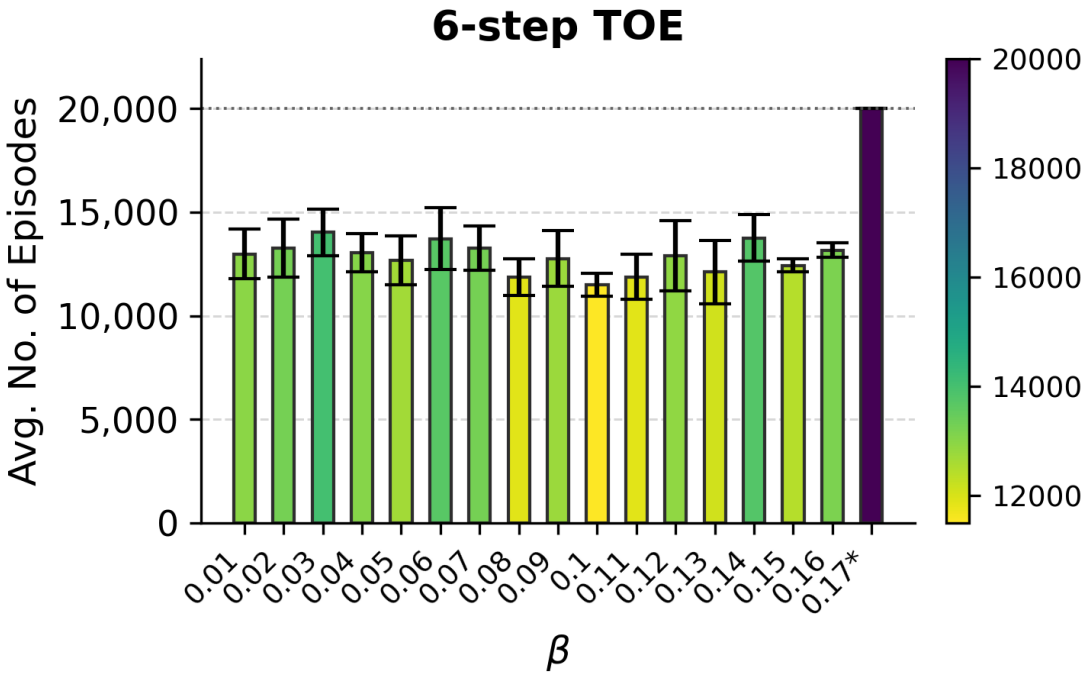


Figure 5.35: 6-step TOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 20,000 episodes.

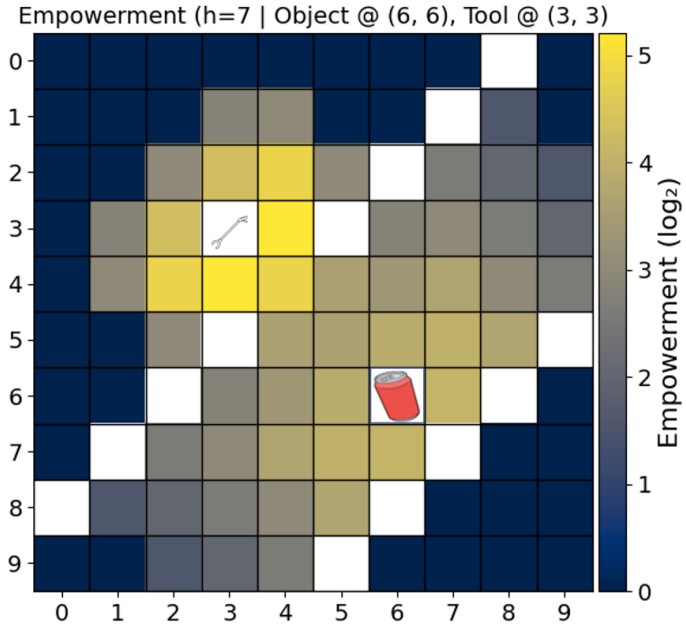


Figure 5.36: 7-step TOE landscape.

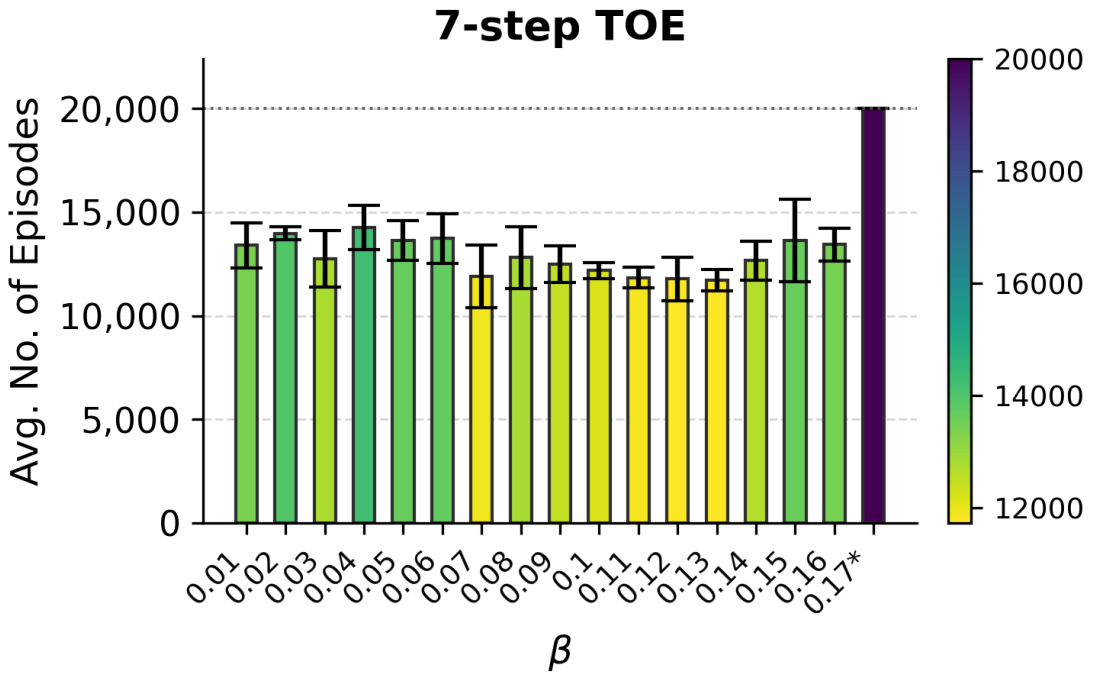


Figure 5.37: 7-step TOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 20,000 episodes.

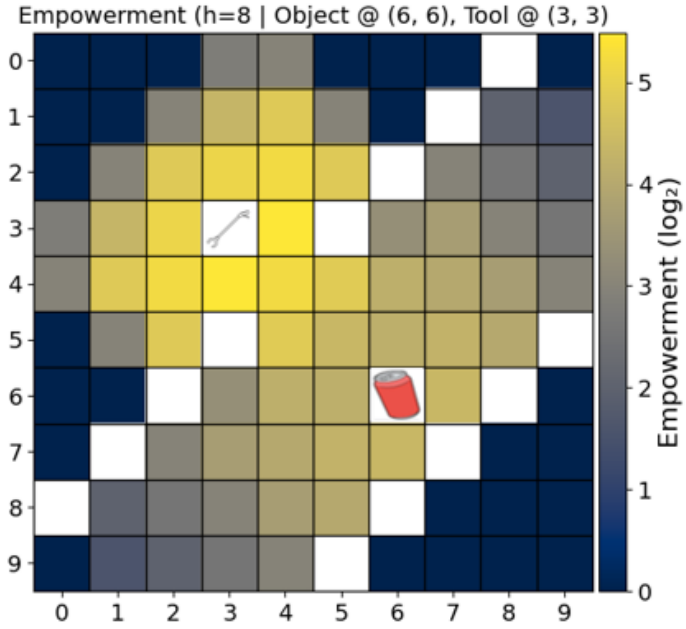


Figure 5.38: 8-step TOE landscape.

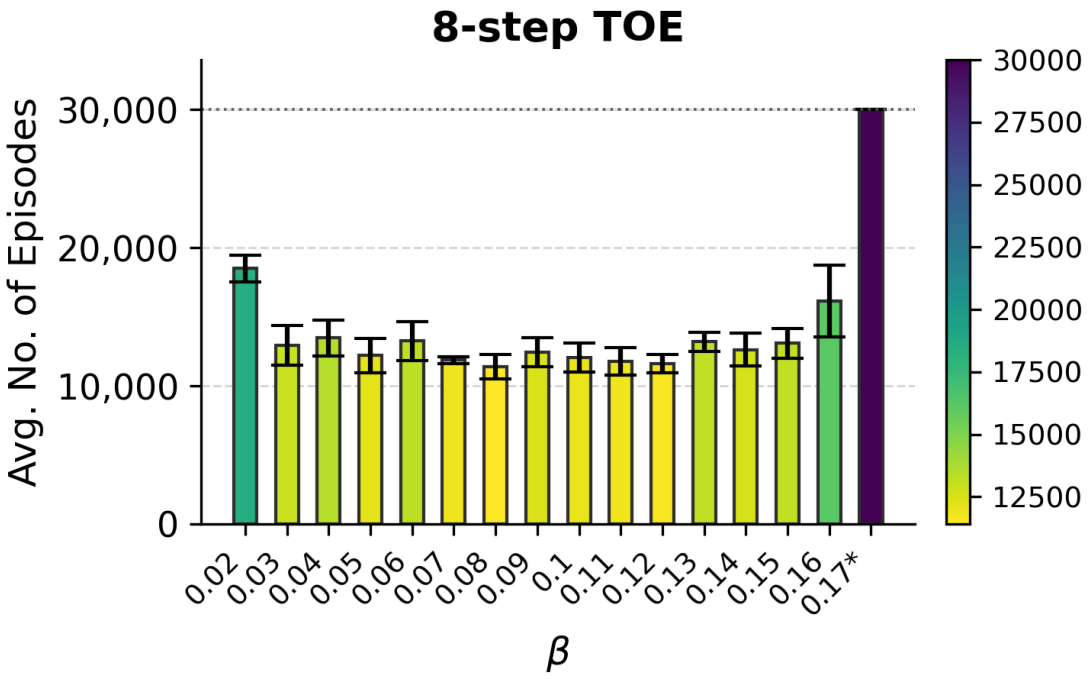


Figure 5.39: 8-step TOE: episodes to solve vs. β. Lower values indicate faster convergence. $*$ denotes runs that did not solve within the cutoff of 30,000 episodes.

more stable range of β values across horizons, suggesting that object-centric formulations of empowerment mitigate the sensitivity to regularisation that characterises FOE. As a result, TOE not only provides more targeted intrinsic signals but also yields more robust learning dynamics when h and β are tuned jointly.

Table 5.2: β values that yielded the fastest convergence for each horizon h under TOE.

h=1	h=2	h=6	h=7	h=8
0.2	0.1	0.1	0.13	0.08

5.3.3 Behavioural Consequences of FOE and TOE

Beyond aggregate performance metrics, it is instructive to analyse how different intrinsic motivation signals shape the agent's spatial behaviour. State visitation heatmaps and representative trajectories highlight how FOE and TOE induce distinct exploration biases depending on the choice of β. For clarity, the following qualitative analyses illustrate representative high-β conditions for both empowerment formulations. The selected (h, β) pairs correspond to those that produced the most characteristic behaviours under strong intrinsic motivation. Similar qualitative patterns were consistently observed across the

other horizons, as detailed below.

High β under TOE

Figure 5.40 shows the visitation heatmap when TOE is combined with a high weighting factor $\beta = 0.4$ at $h = 1$. The agent concentrates almost exclusively around the *object*, rarely exploring other states. Unlike FOE, which spreads influence around all states, TOE explicitly privileges the object's affordances. A corresponding example trajectory in Figure 5.41 confirms this: the agent approaches the object and remains nearby, without necessarily engaging with the tool itself.

A similar object-centric behaviour was observed for other low-horizon configurations (e.g., $h = 2, \beta = 0.3$), where empowerment peaks are concentrated near the object. However, for higher horizons, the behaviour changes qualitatively. For instance, at $h = 6$ (with $\beta = 0.17$), as shown in Figures 5.42 and 5.43, the agent first moves toward and acquires the tool, and subsequently interacts with the object, remaining in its vicinity after tool-mediated interaction. Similar behaviour was observed for $h = 7$ and $h = 8$ (also with $\beta = 0.17$), indicating a consistent shift in strategy at larger horizons. This reflects the change in the empowerment landscape, where high-value states become associated with tool-mediated influence rather than immediate proximity to the object.

Overall, these results demonstrate that excessive intrinsic weighting biases exploration toward regions of high empowerment, but the spatial focus of this bias depends strongly on the planning horizon.

High β under FOE

Figure 5.44 depicts the heatmap when FOE is combined with a high $\beta = 0.3$ at $h = 1$. Unlike TOE, the agent stays primarily near the tool, reflecting that FOE rewards potential influence over all states rather than specifically over the object. The example trajectory in Figure 5.45 confirms this, showing persistent movement around the tool location and repeated engagement with the tool, even when such behaviour is not directly relevant to the task. In contrast to TOE, where the behavioural focus shifts with increasing horizon, the qualitative behaviour under FOE remains largely consistent across different horizons. Similar behaviours were observed for $(h = 2, \beta = 0.2)$, $(h = 3, \beta = 0.16)$, $(h = 4, \beta = 0.14)$, $(h = 5, \beta = 0.13)$, $(h = 6, \beta = 0.12)$, and $\beta = 0.11$ for $h = 7$, and 8, confirming that across horizons, large β values under FOE consistently drive the agent to prioritise tool-centric exploration.

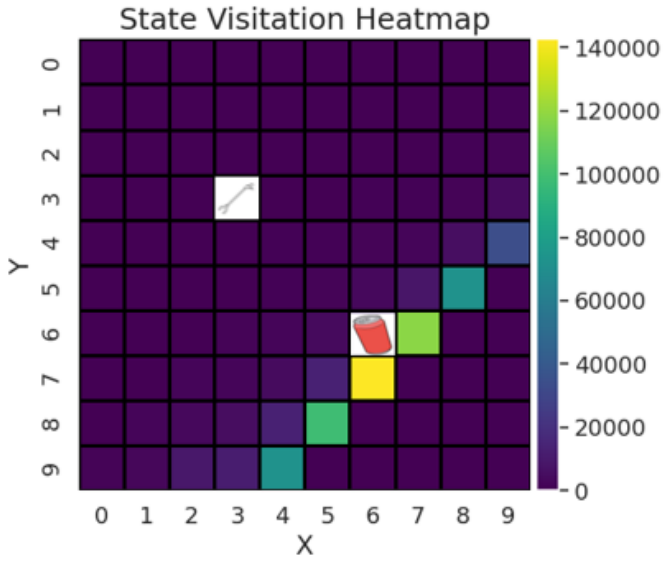


Figure 5.40: State visitation heatmap under high $\beta = 0.4$ with 1-step TOE. The agent remains concentrated around the object, ignoring the tool.

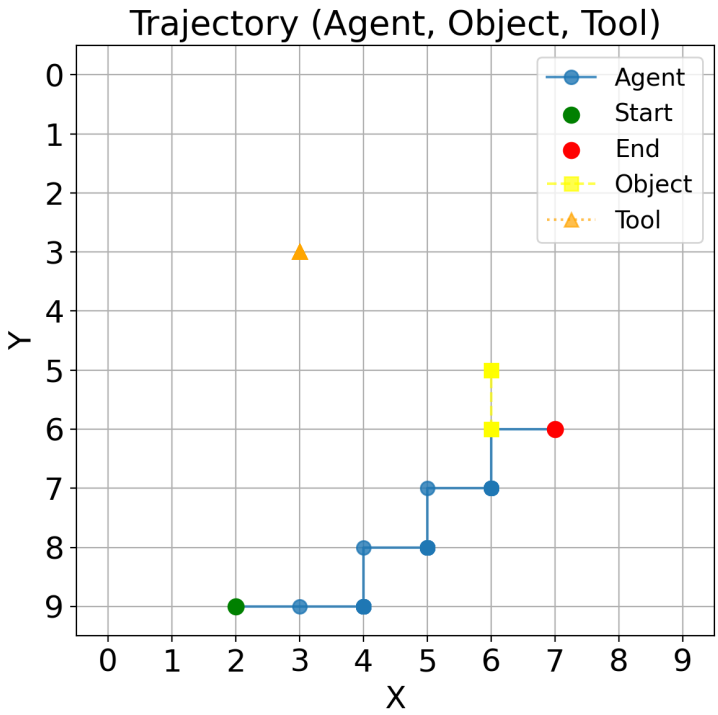


Figure 5.41: An example of trajectory under high $\beta = 0.4$ with 1-step TOE. The agent approaches the object and stays nearby, without engaging with the tool.

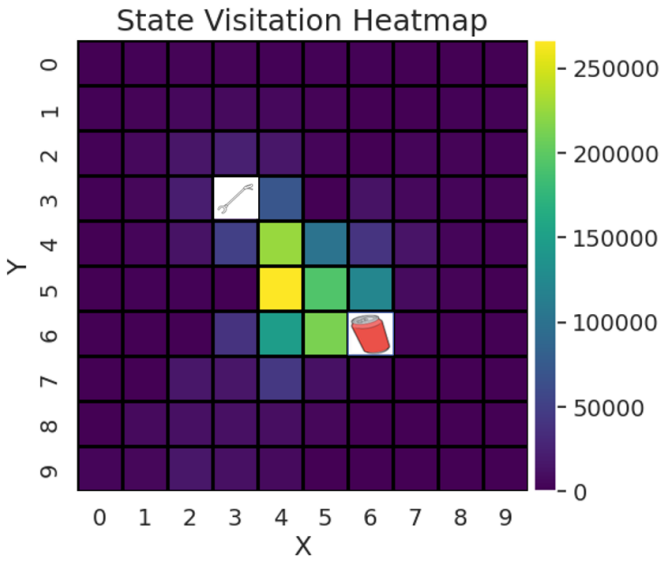


Figure 5.42: State visitation heatmap under high $\beta = 0.17$ with 6-step TOE. In contrast to low horizons, the agent's behaviour shifts toward the tool before interacting with the object.

Optimal β and h

This example refers to the TOE case, illustrating the agent's behaviour under the optimal combination of parameters ($h = 6, \beta = 0.1$). Equivalent analyses for FOE exhibited analogous qualitative trends at their respective optimal settings (e.g., ($h = 5, \beta = 0.09$)). When β and h are tuned appropriately, the resulting visitation heatmap (Figure 5.46) shows balanced coverage of task-relevant regions without over-concentration on a single entity. The corresponding example trajectory (Figure 5.47) demonstrates successful interaction with the tool and object, leading to efficient task completion.

Vanilla RL (no empowerment)

Finally, Figure 5.48 shows the heatmap of a vanilla RL agent without empowerment. The agent explores large parts of the environment, including irrelevant states, reflecting inefficient exploration. This contrasts sharply with the focused behaviour induced by FOE and TOE.

These qualitative patterns illustrate clear behavioural differences between the intrinsic

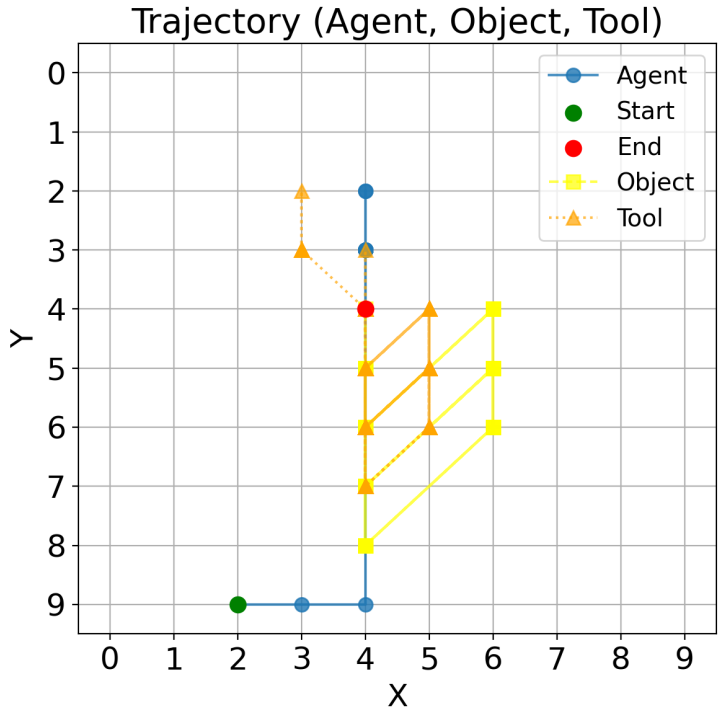


Figure 5.43: An example of trajectory under high $\beta = 0.17$ with 6-step TOE. The agent first acquires the tool and then interacts with the object, remaining in its vicinity thereafter.

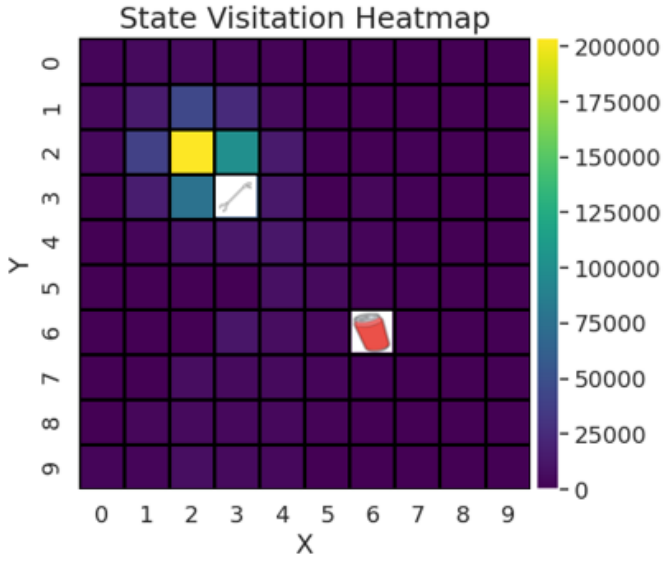


Figure 5.44: State visitation heatmap under high $\beta = 0.3$ with 1-step FOE. The agent remains near the tool and persistently engages in tool use.

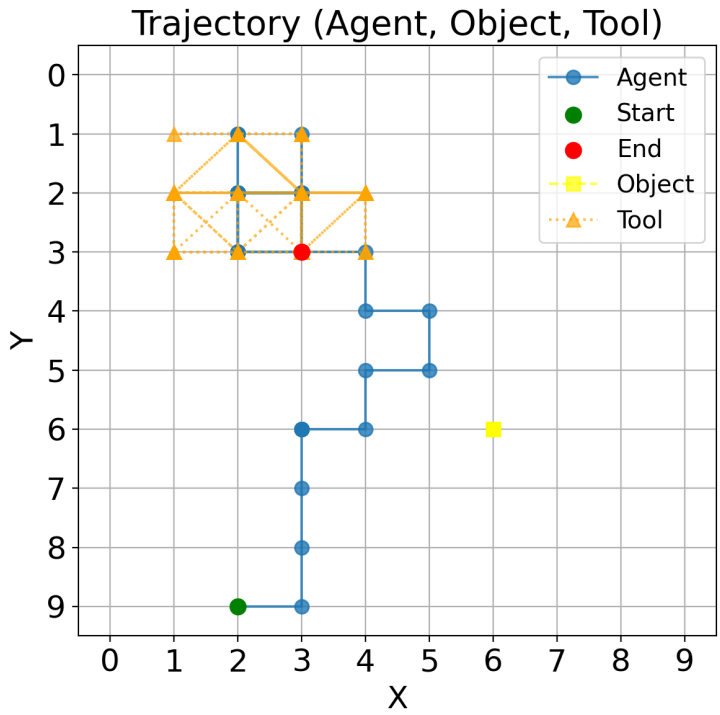


Figure 5.45: An example of trajectory under high $\beta = 0.3$ with 1-step FOE. The agent predominantly stays around the tool location and repeatedly interacts with the tool.

signals. With large beta, FOE biases the agent to remain near the tool and persistently engage in tool use, even when such behaviour is not directly task-relevant. By contrast, TOE directs the agent toward affordance-relevant regions, resulting in behaviour concentrated around the object while neglecting the tool itself. When h and β are tuned optimally, the agent achieves balanced, task-driven exploration by effectively using the tool to interact with the object. In comparison, the vanilla RL agent without empowerment explores large portions of the environment indiscriminately, including irrelevant states, reflecting inefficient exploration.

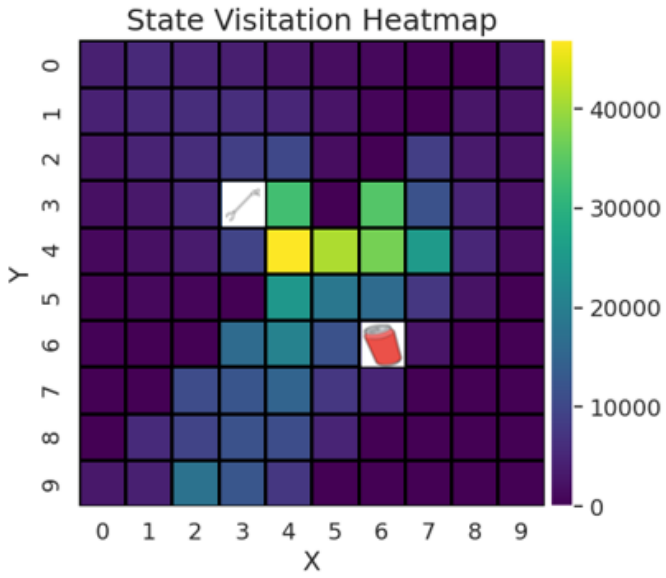


Figure 5.46: Visitation heatmap under optimal parameters ($h = 6$, $\beta = 0.1$) with TOE. The agent balances exploration across task-relevant regions, focusing on both tool and object.

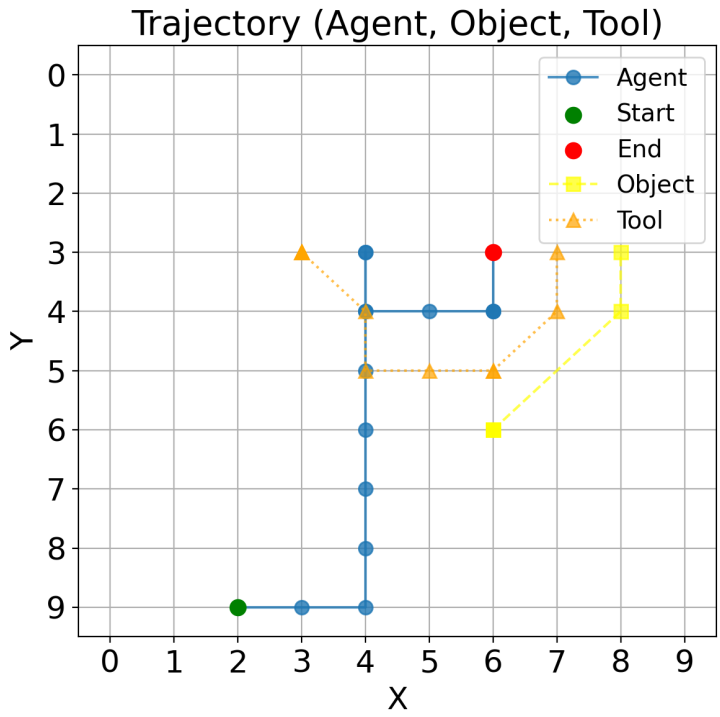


Figure 5.47: An example of trajectory under optimal parameters ($h = 6$, $\beta = 0.1$) with TOE. The agent effectively uses the tool to interact with the object, enabling task completion.

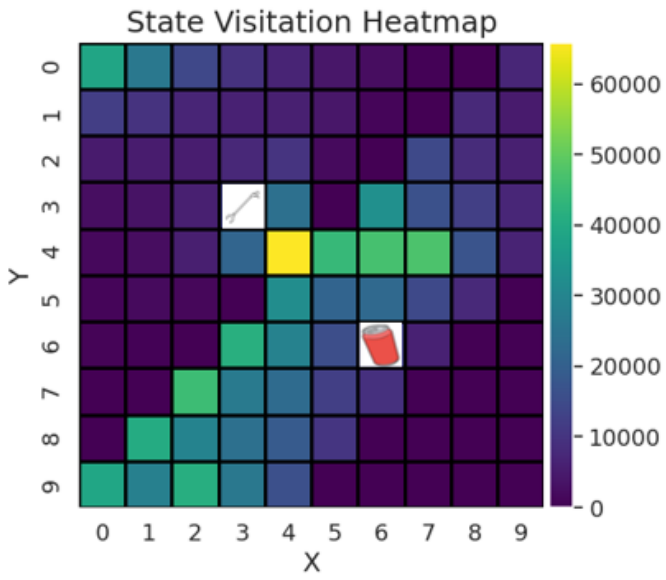


Figure 5.48: Visitation heatmap for a vanilla RL agent. The agent explores large parts of the environment, including many irrelevant states, reflecting inefficient exploration.

5.4 Summary

This chapter empirically evaluated the proposed framework for learning tool-object interactions under sparse rewards. The reward regularisation used the intrinsic motivation empowerment, instantiated either as Fully Observable Empowerment (FOE) or as Tool's Object Empowerment (TOE). Two experiments highlighted complementary insights. In *Tools Comparison*, differences in the object empowerment of the picker and the broom aligned with learning speed: higher $\hat{\mathfrak{E}}^{h}_{\mathfrak{IO}}$ (picker) translated into faster convergence than a tool with lower empowerment (broom), even for vanilla agents. In the *comparison between FOE and TOE*, both intrinsic signals improved over a baseline, but shaped behaviour differently: FOE tended to bias policies toward tool-centric regions and repeated tool engagement, whereas TOE concentrated behaviour around affordance-relevant object states and yielded faster convergence and higher asymptotic returns.

A systematic study of horizon h and weight β clarified how intrinsic signals should be parameterised. Increasing h expands the temporal reach of empowerment: under FOE, attraction concentrates near the tool and extends toward the object at larger horizons; under TOE, short horizons (h=1–2) over-focus on the object and may bypass the tool, whereas longer horizons ($h \geq 6$) naturally capture tool's equip-and-use sequences, often

yielding empowerment peaks near the tool. Across signals, intermediate β values produced the best learning performance; overly small values under-shaped exploration, while overly large values caused the extrinsic objective to be dominated. TOE exhibited a more stable β range across horizons, suggesting reduced sensitivity to regularisation compared to FOE. Behavioural analyses (heatmaps and trajectories) corroborated these effects: high-β FOE induced persistent tool-proximal activity, high-β TOE induced strong object-centricity, and properly tuned (h, β) produced balanced, task-directed exploration; by contrast, a vanilla agent explored widely, including irrelevant states.

To the best of my knowledge, this is the first work to use Object Empowerment as an intrinsic reward regulariser in RL. Prior studies typically restrict to FOE or aggregate empowerment measures, whereas here TOE is shown to provide a more affordance-sensitive drive, capable of steering the agent toward effective tool-mediated object manipulation.

While the results demonstrate that object empowerment can effectively guide exploration and accelerate the acquisition of tool-use behaviours, several considerations remain. The experiments in this chapter focus on environments containing a single relevant tool–object relationship, where the agent only needs to discover how a particular tool influences a particular object. In more complex environments containing multiple tools and multiple objects, the agent must additionally determine which tool is most suitable for influencing a given object. Addressing this challenge requires extending the empowerment framework beyond single-object settings.

Overall, the chapter demonstrates that empowerment-based regularisation can reliably accelerate the acquisition of tool-use behaviours in RL, provided that the temporal horizon and weighting are matched to the structure of tool-mediated affordances. The next chapter builds on these results to address *learning tool selection* in multi-tool, multi-object settings, extending the formalism to multi-object empowerment and introducing a matrix-based selection mechanism that prioritises tools with the greatest actionable influence on task-relevant objects.

Chapter 6

Learning Tool Selection

This chapter addresses research question RQ4, which investigates how empowerment can be generalised to environments containing multiple tools and objects in order to enable systematic tool selection. Building on the object empowerment framework introduced in Chapter 4 and its integration into RL explored in Chapter 5, this chapter extends the formulation to multi-tool, multi-object environments by introducing the concept of *multi-object empowerment*. This extension enables the construction of a tool–object empowerment matrix that captures the average influence that each tool can exert on each object. The resulting representation provides a principled basis for selecting the tool with the greatest potential influence on a target object. Parts of the material presented in this chapter correspond to the contributions reported in publications C2, W1, W2, and W3, where the multi-object empowerment formulation and tool selection mechanism were first introduced.

The previous chapter demonstrated how empowerment, when restricted to tool–object interactions, can guide agents toward meaningful affordances and facilitate effective tool use. However, the scenarios considered so far were limited to a single tool and a single object, where the agent's challenge was primarily to discover and exploit the affordance structure of that tool.

In more realistic environments, agents are confronted with multiple tools and multiple objects, each with potentially different affordances. For instance, a robotic agent may have access to a hammer, a wrench, and a screwdriver, each suited to a different subset of manipulable objects. Some tools may be perfectly suited for specific objects, others only partially effective, and some completely irrelevant. In such settings, an additional layer of decision-making is required: *tool selection*. An agent must not only learn how to use a tool, but also which tool is most appropriate for interacting with given objects.

The ability to autonomously select among alternative tools is central to adaptive and

flexible behaviour in both animals and artificial agents. From a robotics perspective, tool selection supports generalisation across tasks and environments, enabling a single system to reuse existing tools in novel contexts. In the context of intrinsically motivated RL, this problem is particularly challenging because extrinsic rewards may be delayed or sparse, requiring the agent to rely on internal signals to identify promising tool-object interactions.

To formalise this problem, the framework of tool-object empowerment from Chapter 5 is extended to multi-tool, multi-object environments. The extension introduces the concept of *multi-object empowerment*, which quantifies how a tool's potential influence extends over sets of objects, and defines the *tool-object empowerment matrix*, a compact representation capturing the average empowerment that each tool exerts on each object. From this representation, a *tool selection mechanism* is derived that allows an agent to systematically identify the tool most capable of exerting control over a target object.

While the mathematical structure of empowerment remains unchanged, this chapter focuses on its integration and operationalisation in more complex environments. The main contribution lies not in the formalism itself, but in how empowerment computations are organised and used to support decision-making at a higher level of abstraction, namely, the *selection* of the appropriate tool. This chapter shows how the proposed tool selection mechanism can be integrated into RL agents, providing an intrinsic drive that biases exploration toward the tool with the most impact on the object relevant to the given task in multi-tool settings.

6.1 Tool-Learning Framework

The tool-learning framework developed in this chapter extends the single-tool, single-object formulation of Chapter 5 to multi-tool, multi-object settings. It formalises how agents, tools, and objects jointly define the environment's state and action spaces, and how empowerment can be specialised to quantify the influence of specific tools over a subset of objects.

6.1.1 State Space

The environment now consists of an agent, a set of n tools $\mathfrak{T} = \{\mathfrak{T}_1, \mathfrak{T}_2, \ldots, \mathfrak{T}_n\}$, and a set of m objects $\mathfrak{O} = \{\mathfrak{O}_1, \mathfrak{O}_2, \ldots, \mathfrak{O}_m\}$. Each entity's space contributes to the overall state space, defined as:

$$\mathcal{S} \coloneqq \mathcal{S}^{\mathfrak{A}} \times \left(\prod_{j=1}^{n} \mathcal{S}^{\mathfrak{T}_j} \right) \times \left(\prod_{i=1}^{m} \mathcal{S}^{\mathfrak{O}_i} \right) \times \mathcal{S}^{\mathfrak{W}}, \tag{6.1.1}$$

where $\mathcal{S}^{\mathfrak{A}}$ is the agent's state space, $\mathcal{S}^{\mathfrak{T}_j}$ is the state space of the j-th tool (e.g., its position or equipped status), $\mathcal{S}^{\mathfrak{O}_i}$ is the state space of the i-th object (e.g., its location or condition), and $\mathcal{S}^{\mathfrak{W}}$ captures static elements of the environment.

6.1.2 Action Space

The overall action space $\mathcal{A}$ can be again partitioned according to whether actions involve tool use but this time including the actions of multiple tools and objects:

- $\mathcal{A}^{\mathfrak{A}} \subseteq \mathcal{A}$: actions executed directly by the agent, independently of tools.
- $\mathcal{A}^{\mathfrak{T}_j} \subseteq \mathcal{A}$: actions corresponding to the use of tool $\mathfrak{T}_j$, for $j = 1, \ldots, n$.
- $\mathcal{A}^{\mathfrak{A}\mathfrak{T}_j} \coloneqq \mathcal{A}^{\mathfrak{A}} \cup \mathcal{A}^{\mathfrak{T}_j}$: the combined set of agent actions and those specific to tool $\mathfrak{T}_j$, for $j = 1, \ldots, n$.

For simplicity of notation, $\mathcal{A}^{\mathfrak{A}\mathfrak{T}_j}$ is denoted as $\mathcal{A}^{\mathfrak{T}_j}$ in the following.

6.1.3 Multi-Object Empowerment

To model scenarios requiring the manipulation of multiple objects simultaneously with the same tool, the concept of *multi-object empowerment* is introduced. Let $\mathfrak{D} = \{\mathfrak{D}_1, \ldots, \mathfrak{D}_q\} \subseteq \mathfrak{O}$ be a subset of objects. The h-step multi-object empowerment from tool $\mathfrak{T}_j$ to $\mathfrak{D}$ is defined as:

$$\mathfrak{E}^h_{\mathfrak{T}_j\mathfrak{D}}(s) \coloneqq \max_{P(a^{\mathfrak{T}_j^h}|s)} I(O^{\mathfrak{D}_1}_{t+h}, \ldots, O^{\mathfrak{D}_q}_{t+h}; \mathcal{A}_t^{\mathfrak{T}_j^h} \mid S_t = s) \tag{6.1.2}$$

This extension allows empowerment to capture tool–object dependencies and could be used in a variety of scenarios, such as those where successful task completion depends on coordinated effects across several objects, or more generally whenever the agent's influence must be assessed over multiple interacting entities within the environment.

6.1.4 Tool Selection Mechanism

In multi-tool, multi-object environments, empowerment offers a principled means of selecting the most effective tool for interacting with each object. To encode all tool–object relationships, the *tool–object empowerment matrix* is defined as:

$$\mathbb{T}[j, i] = \hat{\mathfrak{E}}^h_{\mathfrak{T}_j\mathfrak{O}_i}, \quad j = 1, \ldots, n,\ i = 1, \ldots, m, \tag{6.1.3}$$

where $\hat{\mathfrak{E}}^h_{\mathfrak{T}_j\mathfrak{O}_i}$ denotes the state-averaged empowerment of tool $\mathfrak{T}_j$ on object $\mathfrak{O}_i$. Here, the dependence on the horizon h is omitted from the matrix $\mathbb{T}$ and is implied by context, since h is fixed during the computation.

Table 6.1: Tool–object empowerment matrix $\mathbb{T}$ showing the state-averaged empowerment $\hat{\mathfrak{E}}^h_{\mathfrak{T}_j\mathfrak{O}_i}$ for each tool–object pair. Values indicate the degree of influence each tool has over each object, and i^* denotes the object of interest (i.e., task-relevant (target) object).

	$\mathfrak{O}_1$	$\cdots$	$\mathfrak{O}_{i^*}$	$\cdots$	$\mathfrak{O}_m$
$\mathfrak{T}_1$	$\hat{\mathfrak{E}}^h_{\mathfrak{T}_1\mathfrak{O}_1}$	$\cdots$	$\hat{\mathfrak{E}}^h_{\mathfrak{T}_1\mathfrak{O}_{i^*}}$	$\cdots$	$\hat{\mathfrak{E}}^h_{\mathfrak{T}_1\mathfrak{O}_m}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathfrak{T}_{j^*}$	$\hat{\mathfrak{E}}^h_{\mathfrak{T}_{j^*}\mathfrak{O}_1}$	$\cdots$	$\hat{\mathfrak{E}}^h_{\mathfrak{T}_{j^*}\mathfrak{O}_{i^*}}$	$\cdots$	$\hat{\mathfrak{E}}^h_{\mathfrak{T}_{j^*}\mathfrak{O}_m}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathfrak{T}_n$	$\hat{\mathfrak{E}}^h_{\mathfrak{T}_n\mathfrak{O}_1}$	$\cdots$	$\hat{\mathfrak{E}}^h_{\mathfrak{T}_n\mathfrak{O}_{i^*}}$	$\cdots$	$\hat{\mathfrak{E}}^h_{\mathfrak{T}_n\mathfrak{O}_m}$

The relationships between all tools and objects can be represented compactly in a matrix form, as shown in Table 6.1. Here, the term *object of interest* $\mathfrak{O}_{i^*}$ refers to the specific object that is relevant to the current task. For instance, the object that must be manipulated, moved, or transformed to complete the episode objective. Tools with non-zero empowerment over an object are considered candidates for interacting with it, while those with zero empowerment across all objects are not considered tools, as they do not exert any influence on any of the available objects. For interacting an object of interest $\mathfrak{O}_{i^*}$, the most effective tool is:

$$\mathfrak{T}_{j^*} \coloneqq \arg\max_j \ \hat{\mathfrak{E}}^h_{\mathfrak{T}_j\mathfrak{O}_{i^*}}. \tag{6.1.4}$$

Equation (6.1.4) defines a *tool selection mechanism* that enables agents to automatically identify which tool is most effective for influencing a given object $\mathfrak{O}_{i^*}$. Without requiring prior knowledge, the tool selected by this criterion has, on average, the highest likelihood of producing meaningful changes in the state of the object of interest. Here, "on average" refers to the state-average object empowerment (Equation (4.1.7)), computed across all possible states and object configurations. It reflects the expected empowerment of each tool over the distribution of reachable states, rather than a specific instance. In this sense, it captures the typical influence that each tool can exert across the environment. The term "meaningful" denotes those state changes in which the tool produces an actual causal effect on the object's dynamics, as opposed to mere positional variations without interaction. It should be noted, however, that this mechanism does not always guarantee task relevance. It is possible to construct tasks in which the tool with the highest object empowerment is not the one required to solve the task. In such cases, empowerment remains a task-independent measure of potential control, rather than a direct indicator of task-specific usefulness. In the experiments presented in the following sections, this mechanism was applied by using the corresponding object empowerment value $\mathfrak{E}^h_{\mathfrak{T}_{j^*}\mathfrak{O}_{i^*}}$ from the selected

tool $\mathfrak{T}_{j^*}$ as the intrinsic reward within the regularised reward function (Equation (5.1.1)). This guided the agent to explore and use the tool that was most relevant to the current task involving the target object $\mathfrak{O}_{i^*}$, effectively biasing exploration toward the tool that could exert the strongest influence on that object.

The framework extends object empowerment to multi-tool, multi-object environments by formalising the state and action spaces and introducing multi-object empowerment. The state-average object empowerment values obtained for each tool–object pair are used to construct the tool–object empowerment matrix, from which the tool expected to be most effective on average for a given object can be identified. The resulting selection mechanism provides agents with a principled intrinsic drive, supporting efficient exploration and targeted tool use in complex environments, and serves as the basis for the empirical analysis presented later in this chapter.

6.2 Experiments

Experiments are conducted in MiniHack environments [137], introduced earlier in Chapter 4 (see Section 4.4 for details on states, actions, and MiniHack game mechanics). These grid-based environments support complex interactions between agents, tools, and objects.

As a reminder, the state space encodes the agent's location, tool positions and states (equipped and hidden flags), and object positions and states (hidden and destruction flags). The action space consists of movement actions and tool-use actions, the latter available only when a tool is equipped. Tool use follows MiniHack's three-step procedure: *apply*, *choose*, and *direction*.

In the experiments, these components are embedded into episodic MDPs with sparse rewards: the agent receives +1 only upon achieving the task objective, and 0 otherwise. Rewards are delivered through in-game messages, which signal when a goal condition has been satisfied. For example, in a tree-destruction task, the message "*You cut down the tree.*" corresponds to successful completion. The environment dynamics are deterministic, allowing object empowerment $\mathfrak{E}^h_{\mathfrak{T}_j\mathfrak{O}_i}$ to be computed via Equation (4.1.6). Learning is performed with PPO [121] (see Section 3.1.5, Chapter 3), using the RLlib implementation [127].

6.2.1 Experiment 1: Tool Selection in a Single-Object Task

The first environment (Figure 6.1) contains two manipulable objects (a tree and a wall) and four available tools (an axe, a pickaxe, a tin opener, and a key). Two separate tasks are considered: chopping the tree or destroying the wall, with the corresponding task-relevant

Figure 6.1: Initial state of the environment of Experiment 1. Black cells represent unobserved areas hidden from the current agent's field of view.

object denoted $\mathfrak{O}_{i^*}$.

In this environment, the tools and objects are positioned symmetrically with respect to the agent's initial location (see Figure 6.1). This design choice avoids introducing any spatial bias that could favour a particular tool or object, ensuring that the agent's learning behaviour emerges purely from interaction dynamics rather than from the environment's geometry. This convention is maintained consistently across all experiments for the rest of the thesis.

The corresponding tool–object empowerment matrix is reported in Table 6.2. Only the axe exhibits non-zero empowerment toward the tree ($\hat{\mathfrak{E}}^h_{\mathfrak{T}_{\text{axe}}\mathfrak{O}_{\text{tree}^*}} = 4.233 \times 10^{-8}$ bits), which represents a true non-zero value computed from the averaged empowerment across all valid states in the environment. The small numerical magnitude arises from the very large state space of the underlying MDP (over 70,000 states). This large number of states results from the combinatorial combination of tool and object configurations, each with three possible states, and from the overall sparsity of the empowerment landscape. Although small in absolute terms, this non-zero value indicates that the axe has a measurable causal influence on the tree, corresponding to its ability to chop it down. Similarly, only the pickaxe influences the wall ($\hat{\mathfrak{E}}^h_{\mathfrak{T}_{\text{pickaxe}}\mathfrak{O}_{\text{wall}^*}} = 4.233 \times 10^{-8}$ bits), reflecting its ability to break walls. The tin opener and key have no measurable effect on any object ($\hat{\mathfrak{E}}^h_{\mathfrak{T}_{\text{tinop}}\mathfrak{O}_{\text{tree}^*}} = 0$ bits, $\hat{\mathfrak{E}}^h_{\mathfrak{T}_{\text{key}}\mathfrak{O}_{\text{wall}^*}} = 0$ bits), so they should not be considered "tools" in this environment. According to the selection mechanism of Equation (6.1.4), the axe $\mathfrak{T}_{\text{axe}^*}$ is selected for $\mathfrak{O}_{\text{tree}^*}$ and the pickaxe $\mathfrak{T}_{\text{pickaxe}^*}$ for $\mathfrak{O}_{\text{wall}^*}$. In the subsequent experiments, the corresponding object empowerment value of the selected tool is employed as the intrinsic regulariser in the agent's reward function.

Table 6.2: State-averaged tool-to-object empowerment $\hat{\mathfrak{E}}^{3}_{\mathfrak{T}_j \mathfrak{O}_i}$ in bits for each tool–object combination of Experiment 1. All object empowerment values are computed with the corresponding tool in the equipped state.

	$\mathfrak{O}_{\text{tree*}}$	$\mathfrak{O}_{\text{wall*}}$
$\mathfrak{T}_{\text{axe*}}$	4.233×10^{-8}	0
$\mathfrak{T}_{\text{pickaxe*}}$	0	4.233×10^{-8}
$\mathfrak{T}_{\text{tinop}}$	0	0
$\mathfrak{T}_{\text{key}}$	0	0

To illustrate the spatial distribution of object empowerment, landscapes are computed before and after the relevant tool is equipped (see Figures 6.2 and 6.3 for the axe, and Figures 6.4 and 6.5 for the pickaxe).

Tree-Chopping Case

Figure 6.2 reports the unequipped tool landscape of $\mathfrak{E}^{8}_{\mathfrak{T}_{\text{axe*}} \mathfrak{O}_{\text{tree*}}}$. Empowerment is non-zero only at the axe's location, since eight steps suffice for the agent to pick it up, reach the tree, and attempt chopping. At that location, empowerment is 1 bit, reflecting the binary choice between chopping the tree or leaving it intact. Once the axe is equipped, the landscape of $\mathfrak{E}^{3}_{\mathfrak{T}_{\text{axe*}} \mathfrak{O}_{\text{tree*}}}$ (Figure 6.3) shows non-zero values in cells adjacent to the tree, indicating that three steps are sufficient to execute the tool-use sequence ("apply" $\rightarrow$ "choose" $\rightarrow$ "direction") and chop the tree.

Wall-Destruction Case

A similar pattern is observed for wall destruction. When the pickaxe is unequipped, the landscape of $\mathfrak{E}^{8}_{\mathfrak{T}_{\text{pickaxe*}} \mathfrak{O}_{\text{wall*}}}$ (Figure 6.4) exhibits a non-zero peak at the pickaxe's location, since the agent can equip it, reach the wall, and attempt destruction within eight steps. Once equipped, the 3-step landscape $\mathfrak{E}^{3}_{\mathfrak{T}_{\text{pickaxe*}} \mathfrak{O}_{\text{wall*}}}$ (Figure 6.5) shows empowerment concentrated in cells adjacent to the wall, reflecting the localized action of destroying it.

To assess learning under sparse rewards, performance is compared between a standard PPO agent and one regularised with the selected tool–object empowerment. In the tree-chopping task, the unequipped empowerment landscape $\mathfrak{E}^{8}_{\mathfrak{T}_{\text{axe*}} \mathfrak{O}_{\text{tree*}}}$ (Figure 6.2) served as intrinsic reward before the axe was equipped, while the equipped landscape $\mathfrak{E}^{3}_{\mathfrak{T}_{\text{axe*}} \mathfrak{O}_{\text{tree*}}}$ (Figure 6.3) was used afterward. By switching horizons in this way, the object empowerment signal maintains consistent guidance: first directing the agent toward the axe,

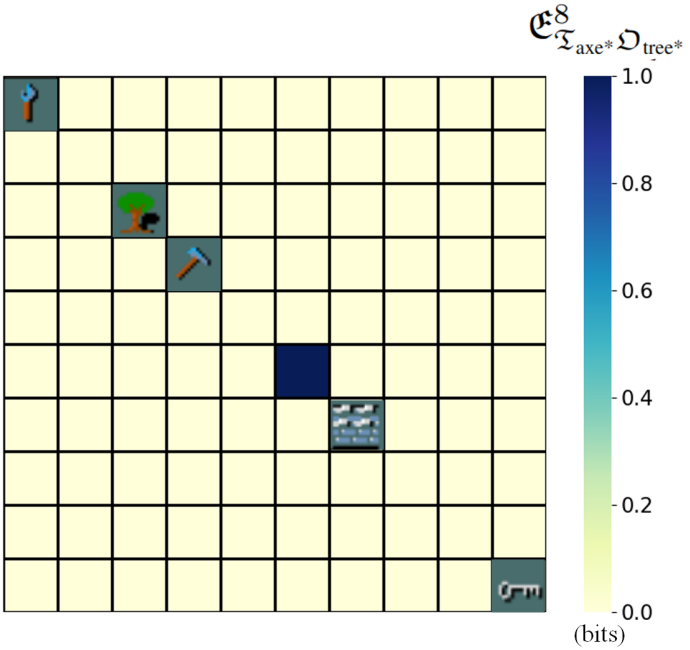


Figure 6.2: 8-step axe-to-tree empowerment $\mathfrak{E}^{8}_{\mathfrak{T}_{\text{axe*}}\mathfrak{O}_{\text{tree*}}}$ landscape for all possible agent locations (in bits), when the axe is not equipped.

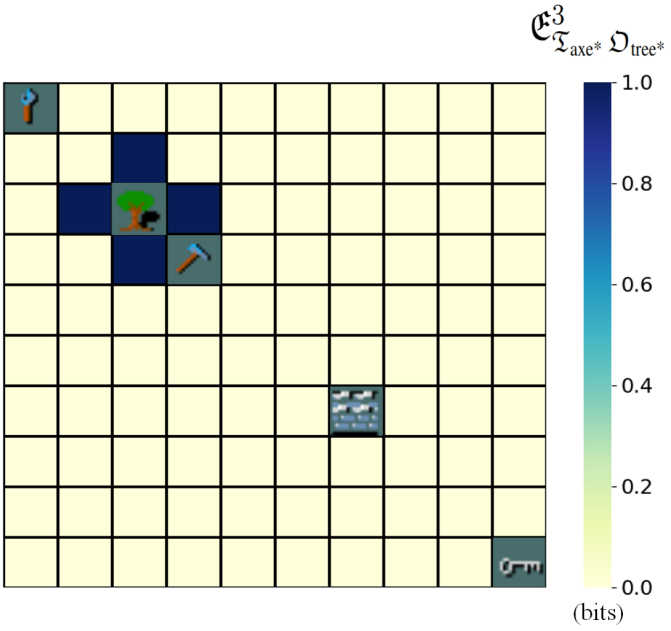


Figure 6.3: 3-step axe-to-tree empowerment $\mathfrak{E}^{3}_{\mathfrak{T}_{\text{axe*}}\mathfrak{O}_{\text{tree*}}}$ landscape for all possible agent locations (in bits), when the axe is equipped.

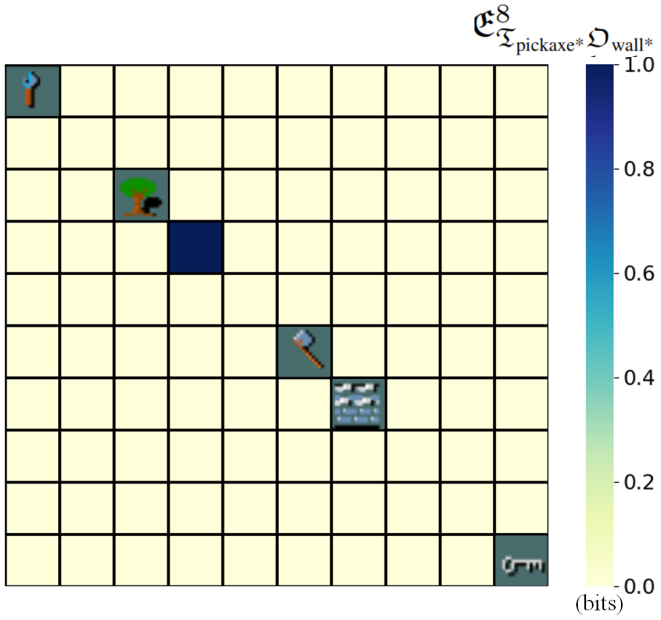


Figure 6.4: 8-step pickaxe-to-wall empowerment $\mathfrak{E}^{8}_{\mathfrak{T}_{\text{pickaxe}*}\mathfrak{O}_{\text{wall}*}}$ landscape for all possible agent locations (in bits), when the pickaxe is not equipped.

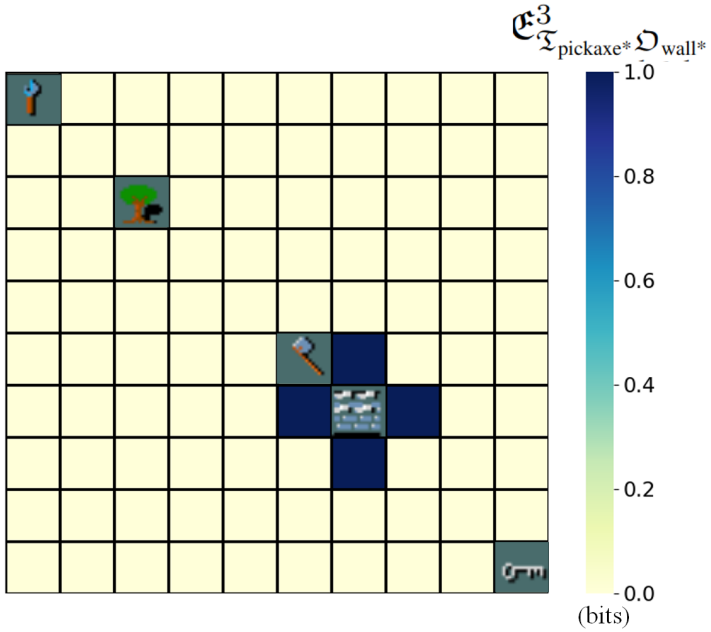


Figure 6.5: 3-step pickaxe-to-wall empowerment $\mathfrak{E}^{3}_{\mathfrak{T}_{\text{pickaxe}*}\mathfrak{O}_{\text{wall}*}}$ landscape for all possible agent locations (in bits), when the pickaxe is equipped.

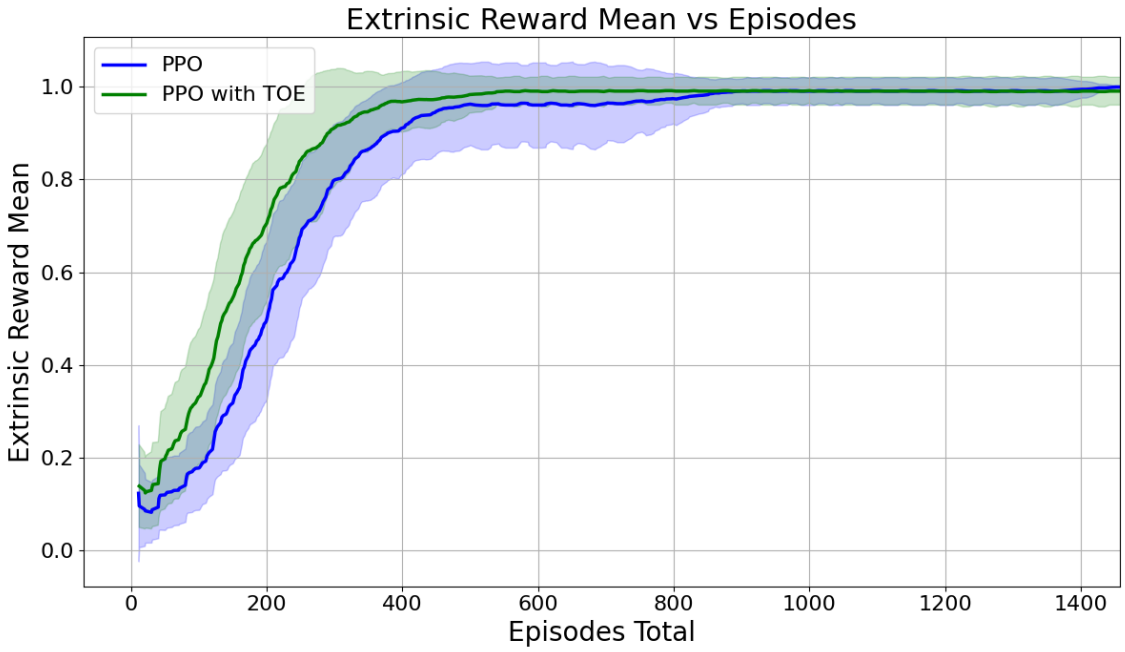


Figure 6.6: Learning performance in Experiment 1. The agent using axe-to-tree empowerment $\mathfrak{E}^{h}_{\mathfrak{T}_{\text{axe}*}\mathfrak{O}_{\text{tree}*}}$ as a regularizer ($\beta = 0.0009$, green) converges faster than standard PPO (blue). Shaded regions indicate standard deviation across 10 runs.

and subsequently driving it toward the task-relevant object (i.e., tree), resulting in faster and more stable convergence. Figure 6.6 reports the average cumulative reward for the tree destruction task, averaged across 10 independent runs. The empowerment-augmented agent converges more rapidly to optimal performance compared to the baseline PPO agent. Equivalent results were observed in the wall-destruction task, confirming the generality of the mechanism. For brevity, only the learning curve for the tree-chopping task is reported.

6.2.2 Experiment 2: Tool Selection in a Multi-Object Task

The second environment (Figure 6.7) presents a more challenging setting in which the agent must destroy two distinct objects: a boulder and a door. Each object yields a reward of +1 when destroyed, making the maximum cumulative return equal to 2. The environment also contains four tools: a wand, an axe, a tin opener, and a katana. The wand can destroy both the boulder and the door, the axe is capable of destroying only the door, while the tin opener and katana serve as distractors with no effect on either object. In this case, static walls act as barriers that constrain agent movement but do not serve as manipulable objects.

Table 6.3 reports the state-averaged tool-to-object empowerment values. For the boulder, the wand achieves $\hat{\mathfrak{E}}^{h}_{\mathfrak{T}_{\text{wand}}\mathfrak{O}_{\text{bould}*}} = 5.292 \times 10^{-7}$ bits, while the axe, tin opener, and

Figure 6.7: Initial state of the environment of Experiment 2. Black cells represent unobserved areas hidden from the current agent's field of view.

katana achieve 0 bits. For the door, the wand achieves $\hat{\mathfrak{E}}^{h}_{\mathfrak{T}_{\text{wand}}\mathfrak{O}_{\text{door*}}} = 6.138 \times 10^{-7}$ bits, the axe achieves 3.281×10^{-7} bits, and the tin opener and katana again have no effect. When considering the combined boulder–door target set, the wand yields the highest empowerment $\hat{\mathfrak{E}}^{h}_{\mathfrak{T}_{\text{wand}}\mathfrak{O}_{\text{bould*}}\mathfrak{O}_{\text{door*}}} = 9.564 \times 10^{-7}$ bits, followed by the axe with 3.281×10^{-7} bits, while the distractor tools remain at 0 bits. This combined value is not a sum of the individual empowerments but rather the state-averaged empowerment computed when both the boulder and the door are simultaneously present in the environment, reflecting the wand's overall influence on both objects within the same configuration. According to the tool-selection mechanism of Equation (6.1.4), the wand $\mathfrak{T}_{\text{wand*}}$ is identified as the most effective tool, and its boulder–door empowerment is used as the intrinsic reward for RL.

Table 6.3: State-averaged tool-to-object empowerment $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}_j\mathfrak{O}_i}$ in bits for each tool–object combination of Experiment 2. The last column reports the multi-object empowerment $\hat{\mathfrak{E}}^{h}_{\mathfrak{T}_j\,\mathfrak{O}_{\text{bould*}}\mathfrak{O}_{\text{door*}}}$ values. All object empowerment values are computed with the corresponding tool in the equipped state.

	$\mathfrak{O}_{\text{bould*}}$	$\mathfrak{O}_{\text{door*}}$	$\mathfrak{O}_{\text{bould*}}\mathfrak{O}_{\text{door*}}$
$\mathfrak{T}_{\text{wand*}}$	5.292×10^{-7}	6.138×10^{-7}	9.564×10^{-7}
$\mathfrak{T}_{\text{axe}}$	0	3.281×10^{-7}	3.281×10^{-7}
$\mathfrak{T}_{\text{tinop}}$	0	0	0
$\mathfrak{T}_{\text{kata}}$	0	0	0

The axe exhibits a more localized empowerment profile over the door compared to the

wand, whose influence extends across both the door and boulder areas. The unequipped 6-step landscape (Figure 6.8) peaks at the axe's location, indicating that the agent must first collect the tool before it can influence the door. Once equipped (Figure 6.9), the empowerment values reflect the axe's potential to destroy the door within the 6-step horizon. The maximum value of 1.0 bit corresponds to the binary choice between destroying the door or leaving it intact. Although the axe has no effect on the boulder, nonzero empowerment values also appear in grid locations beyond the door's immediate vicinity. These values capture states from which the agent can still reach the door and use the axe within six steps, even if these locations are spatially closer to the boulder. Importantly, however, the axe landscapes do not identify a single best position for using the tool; instead, they simply indicate whether the agent is within range to approach and act on the door. As a consequence, such landscapes provide weaker spatial guidance when used as intrinsic beacons, as they lack a distinct attractor or gradient that could continuously direct the agent toward the optimal interaction point.

In contrast, the wand landscapes reveal both long-range influence and distinct peaks of empowerment. When unequipped, the 5-step landscape (Figure 6.10) peaks at the wand's location (1.58 bits), reflecting that five steps suffice to acquire the tool and begin affecting the task objects. Once equipped (Figure 6.11), the 6-step empowerment landscape expands across a wide region of the grid, consistent with MiniHack mechanics that allow the wand to strike along orthogonal directions. Crucially, the landscape exhibits peaks of 2.0 bits at specific positions from which the agent can destroy both the boulder and the door within the horizon. Intermediate values (e.g., 1.58 and 1.0 bits) correspond to positions from which only one of the objects can be affected. Unlike the axe, therefore, the wand landscapes identify clear "optimal" agent positions for exerting maximal influence over multiple objects simultaneously.

To evaluate task performance, learning is compared between PPO and PPO regularised with wand-to-boulder–door empowerment. Before the wand is equipped, the agent uses the 5-step empowerment landscape $\mathfrak{E}^5_{\mathfrak{T}_{\text{wand}^*}\mathfrak{O}_{\text{bould}^*}\mathfrak{O}_{\text{door}^*}}$ (Figure 6.10) as intrinsic reward, which acts as a beacon guiding exploration toward the wand's location. Once the wand is equipped, the 6-step empowerment landscape $\mathfrak{E}^6_{\mathfrak{T}_{\text{wand}^*}\mathfrak{O}_{\text{bould}^*}\mathfrak{O}_{\text{door}^*}}$ (Figure 6.11) is used, capturing the tool's extended range of influence over both objects.

Figure 6.12 reports the mean cumulative reward per episode, averaged across 10 independent runs. The empowerment-augmented agent converges faster and achieves higher final performance compared to baseline PPO. In contrast, the standard PPO agent often plateaus at suboptimal returns, corresponding to policies that succeed in destroying only one object. By switching horizons based on whether the tool is equipped, the empowerment signal provides consistent intrinsic guidance: first attracting the agent toward the

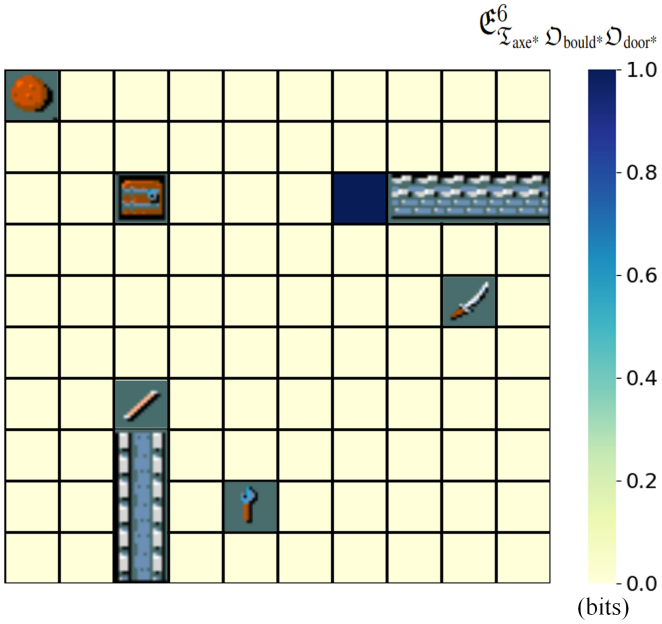


Figure 6.8: 6-step axe-to-boulder-door empowerment landscape when the axe is not equipped. Empowerment peaks at the axe's location.

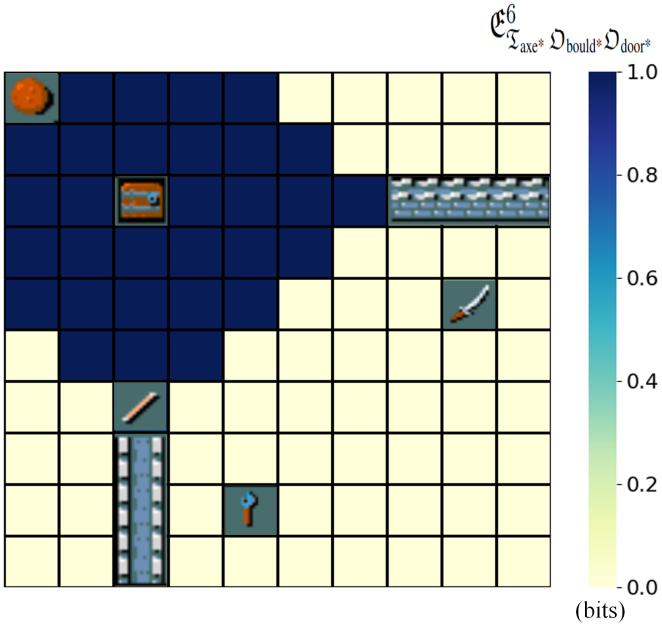


Figure 6.9: 6-step axe-to-boulder–door empowerment landscape when the axe is equipped. Empowerment reflects the axe's ability to influence the door, with nonzero values marking states from which the agent can reach and destroy the door within six steps. These values appear both near the door and in other regions of the grid, indicating feasible access within the horizon but without highlighting a single optimal strike position.

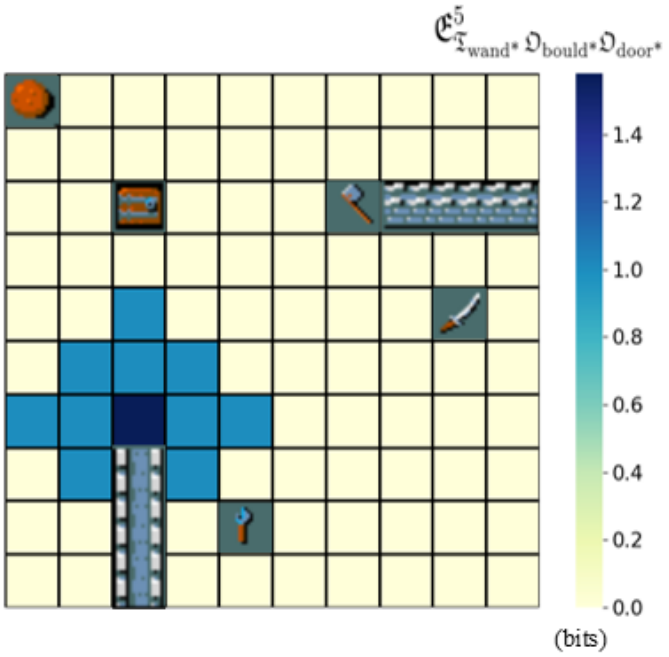


Figure 6.10: 5-step wand-to-boulder-door empowerment landscape when the wand is not equipped.

wand, and subsequently driving it toward both task-relevant objects. This results in more reliable solutions that satisfy both task sub-goals.

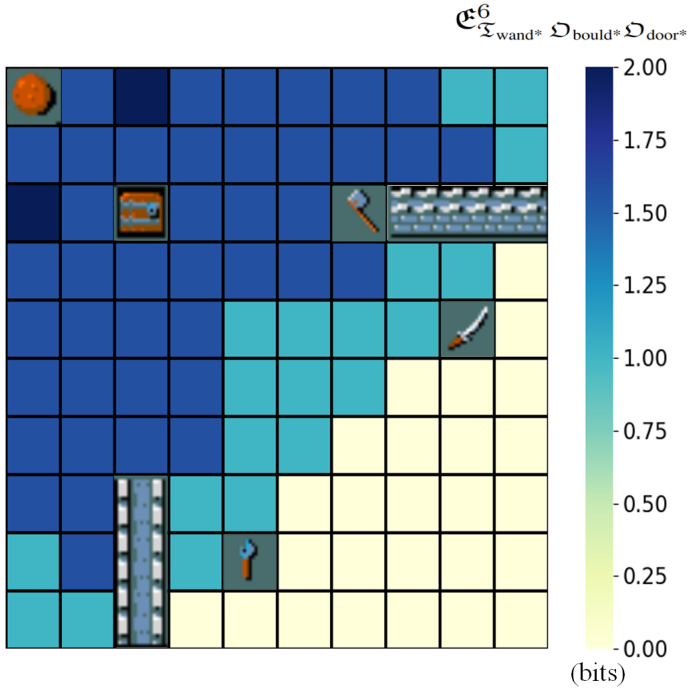


Figure 6.11: 6-step wand-to-boulder–door empowerment landscape when the wand is equipped. Empowerment spans a wide area of the grid, with distinct peaks of 2.0 bits identifying optimal positions from which the agent can destroy both objects within six steps. Intermediate values indicate states where only one of the two objects can be influenced, reflecting the wand's long-range effect.

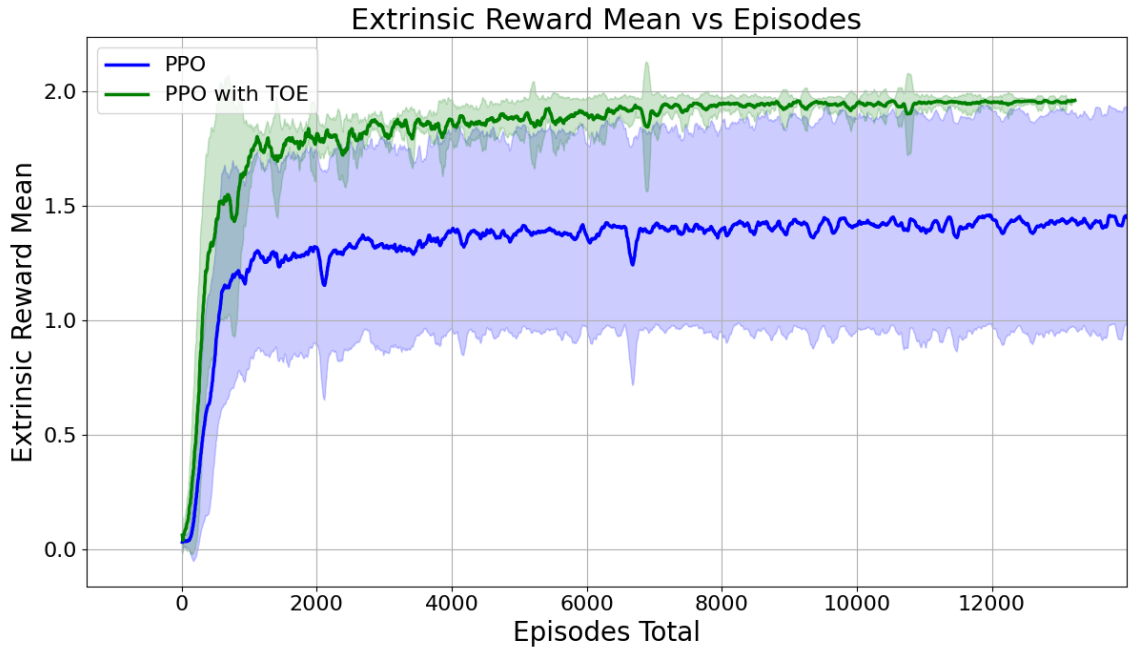


Figure 6.12: Learning performance in Experiment 2. The agent using wand-to-boulder–door empowerment as a regulariser ($\beta = 0.0009$, green) converges faster and more reliably than standard PPO (blue). Shaded regions represent standard deviation across 10 runs.

6.3 Summary

This chapter introduced a novel extension of object empowerment by formalising *multi-object empowerment* and applying it to multi-tool, multi-object environments. Building on the foundation of object empowerment, the framework was generalised to quantify the influence of tools across multiple task-relevant objects simultaneously. This extension enabled the systematic construction of the tool–object empowerment matrix, from which a principled tool selection mechanism was derived. The mechanism offers a new way of modelling tool use, allowing agents to automatically identify and prioritise the tool most capable of influencing the object(s) of interest.

The chapter further demonstrated the practical utility of this framework in MiniHack environments. By embedding object empowerment into RL through regularisation, experiments showed that object empowerment-guided tool selection provides a powerful intrinsic drive, enabling agents to overcome sparse rewards and discover meaningful tool–object interactions more reliably than standard baselines. Together, these results establish multi-object empowerment and the associated selection mechanism as key contributions toward computational models of tool use and selection.

While the proposed framework enables agents to identify which tool is most capa-

ble of influencing a given object, tool selection alone does not fully capture the diversity of tool properties that may affect interaction dynamics. In realistic environments, tools may differ not only in the magnitude of their influence, but also in how that influence unfolds during interaction. Understanding such differences requires analysing additional characteristics that describe how tools behave across states and over time. These considerations motivate the need for a complementary perspective on tool modelling that goes beyond selection based solely on empowerment values. The next chapter builds on these foundations by moving beyond interaction dynamics to address the problem of *tool characterisation*. Specifically, it develops methods for classifying and comparing tools based on their empowerment signatures, thereby providing a complementary perspective on how agents can understand not only which tool to use, but also what makes tools distinct and interchangeable in complex environments.

Chapter 7

Characterisation of Tools

This chapter addresses research question RQ5, which investigates how tools can be systematically characterised beyond their immediate influence on objects. Building on the object empowerment framework developed in Chapter 4 and the tool selection mechanism introduced in Chapter 6, this chapter introduces three complementary dimensions of tool characterisation: *persistence*, *latency*, and *reliability*. Together, these dimensions describe how long a tool remains effective, how quickly it can produce its effect, and how consistently it performs its intended function. Parts of the material presented in this chapter correspond to the persistence analysis reported in publications C2, W1, W2, and W3, while the discussion of latency and reliability further extends these ideas within the broader framework of the thesis.

This chapter explores how tools can be systematically *characterised* within RL environments, extending the investigation beyond a simple measure of their influence on the state of a given object. This chapter asks the following question: *what are other key properties of tools that make them useful or challenging for an agent to employ?*

Three fundamental dimensions of tool characterisation are considered: *persistence*, *latency*, and *reliability*. Together, these dimensions capture how long a tool remains useful, how quickly it can be applied to achieve its effect, and how consistently it performs reliably its intended function. Understanding these characteristics is crucial for designing agents that can adapt their behaviour to tools with different temporal or stochastic properties.

This chapter adopts the same *tool-learning framework* introduced in Chapter 6, including: the state decomposition into agent, tool, object, and world components; the distinction between agent-only and tool-mediated actions; and the use of object empowerment-based reward regularisation. The following section builds on this framework to formally define these three characterisation measures.

7.1 Characterisation Measures

The proposed characterisation of tools is approached through three tools' properties, each capturing a different aspect of a tool's affordances and practical utility:

7.1.1 Persistence of Tools

Persistence refers to the extent to which a tool affords *repeated* or *continuous* influence over the state of its target object. Conceptually, it captures whether the interaction enabled by the tool is reversible (persistence) or irreversible (no persistence). For example, a key affords persistence because the door can be opened and re-closed multiple times, while an axe affords no persistence since destroying the door permanently removes any opportunity for further interaction.

A formal definition of persistence can be given as:

Define the set $Z^h_{\mathfrak{T}_j\mathfrak{O}_i}$ of states that has non-zero tool to object empowerment $\mathfrak{E}^h_{\mathfrak{T}_j\mathfrak{O}_i}$:

$$Z^h_{\mathfrak{T}_j\mathfrak{O}_i} = \{s \in \mathcal{S} \mid \mathfrak{E}^h_{\mathfrak{T}_j\mathfrak{O}_i}(s) \neq 0\},$$

Fix a set of initial states $S_0 \subseteq \mathcal{S}$. A tool $\mathfrak{T}_j$ is *inevitably h-step persistent* in S_0 with respect to object $\mathfrak{O}_i$ if:

1. $$P(Z \mid s, a^{\mathfrak{T}_j}) = 1 \qquad \forall\, s \in Z,\ a^{\mathfrak{T}_j} \in \mathcal{A}^{\mathfrak{T}_j}.$$

2. for every trajectory $(s_t, a_t^{\mathfrak{T}_j})_{t\geq 0}$ with $s_0 \in S_0$, there exists a finite T such that

$$s_t \in Z^h_{\mathfrak{T}_j\mathfrak{O}_i}, \qquad \forall\, t \geq T.$$

Similarly, a tool $\mathfrak{T}_j$ is *possibly h-step persistent* in S_0 with respect to object $\mathfrak{O}_i$ if:

1. $$P(Z \mid s, a^{\mathfrak{T}_j}) = 1 \qquad \forall\, s \in Z,\ a^{\mathfrak{T}_j} \in \mathcal{A}^{\mathfrak{T}_j}.$$

2. if there exists a trajectory $(s_t, a_t^{\mathfrak{T}_j})_{t\geq 0}$ with $s_0 \in S_0$ and a finite T such that

$$s_t \in Z^h_{\mathfrak{T}_j\mathfrak{O}_i}, \qquad \forall\, t \geq T.$$

Note that in finite-state MDPs, these definitions correspond to reachability (i.e., every maximal path enters and remains in the set) and closure (i.e., there exists at least one feasible path entering and remaining in the set) properties of the support graph.

Intuitively, persistence captures whether a tool continues to offer utility for interacting with a specific object over time. In the case of *inevitable* h-step persistence, the tool's influence is guaranteed: starting from any state, every possible trajectory of the agent's interaction with the environment will eventually reach and remain in states where the tool has non-zero empowerment over the object. This means that, regardless of the agent's actions, the tool is always capable of affecting the object once such a region of influence is reached. In contrast, *possible* h-step persistence indicates that such a region is reachable and invariant under tool-related actions, but only along some possible trajectories, not necessarily all.

In this chapter, persistence is examined qualitatively by comparing the temporal evolution of empowerment in key–door and axe–door interactions. The experimental results and their implications for tool-use learning are discussed in Section 7.2.1.

7.1.2 Latency of Tools

Latency characterises the number of steps required for a tool to begin exerting influence over a target object. Conceptually, it captures how *quickly* a tool affords a meaningful change in the object's state once the agent starts acting. Tools with low latency enable fast interaction (e.g., a remote controller that can affect distant objects immediately), whereas tools with high latency require longer action sequences before producing an effect (e.g., an axe that requires approaching the object first).

Formally, the *state latency* of a tool $\mathfrak{T}_j$ on object $\mathfrak{O}_i$ from state s is defined as:

$$\mathcal{L}_{\mathfrak{T}_j,\mathfrak{O}_i}(s) \coloneqq \min\{h \mid \mathfrak{E}^h_{\mathfrak{T}_j,\mathfrak{O}_i}(s) > 0\}, \tag{7.1.1}$$

i.e., the minimum horizon length h for which object empowerment becomes non-zero.

Averaging over all states $s \in \mathcal{S}$ yields the *state-averaged latency*:

$$\hat{\mathcal{L}}_{\mathfrak{T}_j,\mathfrak{O}_i} = \frac{1}{|\mathcal{S}|} \sum_{s\in\mathcal{S}} \mathcal{L}_{\mathfrak{T}_j,\mathfrak{O}_i}(s). \tag{7.1.2}$$

These values are arranged into the *tool–object latency matrix*:

$$\mathbb{L}[j,i] = \hat{\mathcal{L}}_{\mathfrak{T}_j,\mathfrak{O}_i}, \quad j = 1,\dots,n,\ i = 1,\dots,m,$$

which takes the form:

For a given object of interest $\mathfrak{O}_{i^*}$, the most responsive tool is the one with minimum latency:

$$\mathfrak{T}_{j^*} \coloneqq \arg\min_j \ \hat{\mathcal{L}}_{\mathfrak{T}_j,\mathfrak{O}_{i^*}}. \tag{7.1.3}$$

Table 7.1: Tool–object latency matrix $\mathbb{L}$ showing the state-averaged latency $\hat{\mathcal{L}}_{\mathfrak{T}_j,\mathfrak{O}_i}$ for each tool–object pair. Values indicate the state averaged minimum number of steps required for each tool to begin influencing each object, where i^* denotes the object of interest (i.e., task-relevant (target) object).

	$\mathfrak{O}_1$	$\cdots$	$\mathfrak{O}_{i^*}$	$\cdots$	$\mathfrak{O}_m$
$\mathfrak{T}_1$	$\hat{\mathcal{L}}_{\mathfrak{T}_1\mathfrak{O}_1}$	$\cdots$	$\hat{\mathcal{L}}_{\mathfrak{T}_1\mathfrak{O}_{i^*}}$	$\cdots$	$\hat{\mathcal{L}}_{\mathfrak{T}_1\mathfrak{O}_m}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathfrak{T}_{j^*}$	$\hat{\mathcal{L}}_{\mathfrak{T}_{j^*}\mathfrak{O}_1}$	$\cdots$	$\hat{\mathcal{L}}_{\mathfrak{T}_{j^*}\mathfrak{O}_{i^*}}$	$\cdots$	$\hat{\mathcal{L}}_{\mathfrak{T}_{j^*}\mathfrak{O}_m}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathfrak{T}_n$	$\hat{\mathcal{L}}_{\mathfrak{T}_n\mathfrak{O}_1}$	$\cdots$	$\hat{\mathcal{L}}_{\mathfrak{T}_n\mathfrak{O}_{i^*}}$	$\cdots$	$\hat{\mathcal{L}}_{\mathfrak{T}_n\mathfrak{O}_m}$

Equation (7.1.3) defines a latency-based selection mechanism that prioritises tools whose effects manifest most quickly. In subsequent experiments (Section 7.2.2), the empowerment value $\mathfrak{E}^h_{\mathfrak{T}_{j^*},\mathfrak{O}_{i^*}}$ of the selected tool is used as the intrinsic reward in the regularised reward function, analogous to the approach used in Chapter 6.

This latency measure provides a complementary perspective to object empowerment magnitude: while object empowerment quantifies *how much* a tool can affect an object, latency indicates *how soon* such an effect becomes possible. In subsequent experiments (Section 7.2.2), latency-based tool selection is shown to favour tools that can rapidly influence the target object, which is especially beneficial in time-constrained environments.

7.1.3 Reliability of Tools

Reliability characterises how consistently a tool influences an object when its actuation becomes stochastic. It measures the extent to which a tool's causal relationship with its target object remains stable under uncertainty. For instance, a well-functioning screwdriver reliably tightens a screw every time it is used, whereas a worn-out or defective screwdriver may slip or strip the screw head that makes it unreliable. Similarly, a reliable tool in RL environment consistently produces the intended effect on the target object when actuated. In this context, a reliable tool exhibits high and consistent empowerment because its causal influence on the object is stable and predictable. Conversely, an unreliable tool does not have controllability on the target object as stochasticity disrupts its intended effects.

Formally, object empowerment of a tool–object pair is low under a non-deterministic transition model, where the environment dynamics incorporate probabilistic deviations in tool actuation. In this context, the tool–object empowerment matrix $\mathbb{T}[j,i]$ (see Equation

(6.1.3)) encodes the empowerment of each tool under a specified level of actuation noise. Importantly, this noise is applied specifically to the actuation of tool $\mathfrak{T}_j$ with respect to object $\mathfrak{O}_i$, reflecting how uncertainty in the tool's action affects its ability to influence the target object. Since reliability is inversely related to noise, tools with higher empowerment under noisy conditions are interpreted as more reliable, as they retain greater influence on the target object despite perturbations. Conversely, tools with low empowerment under noise are deemed less reliable, as stochastic effects diminish their ability to affect the object predictably. Higher empowerment values under noisy conditions indicate that the causal influence of the tool on the object is minimally affected by stochasticity. However, this measure reflects only the resulting effect of the noise on empowerment. It does not imply that the tool actively compensates for uncertainty. Rather, it quantifies how much influence a tool *retains* under a given level of actuation noise, serving as an intrinsic indicator of robustness to randomness in the environment.

In subsequent experiments (Section 7.2.3), the empowerment value $\mathfrak{E}^{h}_{\mathfrak{T}_{j^*},\mathfrak{O}_{i^*}}$ of the selected tool is used as the intrinsic reward in the regularised reward function, analogous to the approach used in Chapter 6.

7.2 Experiments

This section empirically evaluates the tool characterisation framework introduced in the previous section. Experiments are designed to assess how persistence, latency, and reliability of tools influence agent learning and exploration behaviour.

For persistence and latency, the MiniHack platform [137] is used. Learning is performed using Proximal Policy Optimisation (PPO) [121] via RLlib [127]. State and action spaces are defined as in Chapter 6. Tool use in MiniHack follows the standard three-step interaction protocol (*apply–choose–direction*), ensuring that all tool actions involve an explicit selection and orientation phase consistent with prior experiments. Different reward schemes are used depending on the characterisation measure under study:

- **Persistence:** sparse positive reward (+1) upon successful task completion and 0 otherwise, highlighting how persistent vs. non-persistent tools affect downstream task success.
- **Latency:** sparse penalty (−1) for each time step until the task is completed, encouraging faster discovery and use of low-latency tools.

Experiments on **reliability** are conducted in the simpler, custom grid-world environment introduced in Chapters 4 and 5. This environment allows fine-grained control over

stochasticity in tool dynamics by varying noise levels in the transition model, enabling systematic evaluation of robustness to uncertainty. Learning in this setting is also performed using PPO, implemented via the Stable Baselines3 framework [128]. A step-penalty reward scheme (-1 per step until task completion) is used.

Performance in all settings is compared between a baseline PPO agent and one regularised with object empowerment. The following subsections describe the individual experimental setups and results for each tool characterisation dimension.

7.2.1 Experiment 1: Persistence of Tools

This experiment investigates a scenario where tools enable task completion indirectly, by modifying intermediate objects whose manipulation is a precondition for interacting with the goal object. The environment (Figure 7.1) is a 9 x 9 grid-world based on MiniHack that contains two tools: an axe and a key, and two objects: a locked door and a movable boulder. The agent's task is to push the boulder onto a designated goal location (highlighted in blue). The boulder can be moved directly by the agent without requiring any tool, but it is initially inaccessible, enclosed by walls and a locked door. To reach and push the boulder, the agent must first clear the door, either by opening the latter with the key or destroying it with the axe.

Figure 7.1: Initial state of the "persistence" experiment. The agent must either open the door with the key or destroy it with the axe to access and push the boulder onto the blue goal location.

Both the axe and the key therefore provide *instrumental affordances*: they do not act directly on the goal object but instead modify an intermediate object (the door), enabling the agent to eventually interact with the boulder. As a result, the boulder empowerment

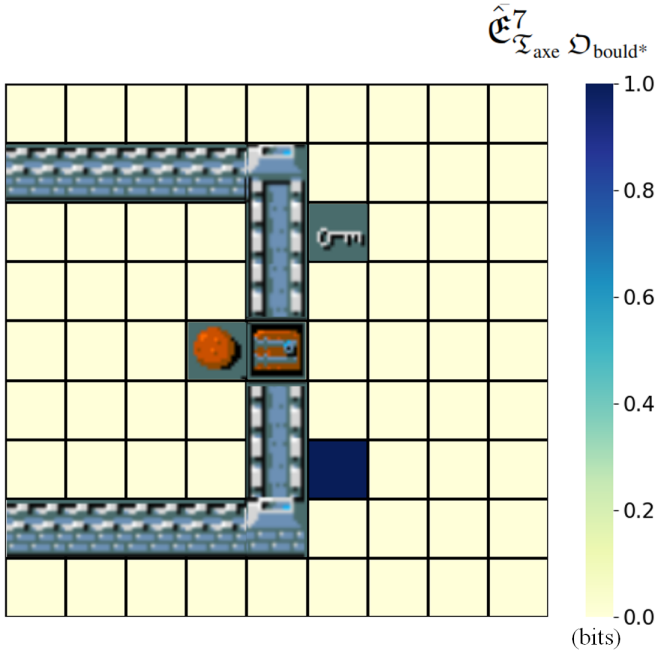


Figure 7.2: 7-step axe-to-boulder empowerment $\hat{\mathfrak{E}}^{7}_{\mathfrak{T}_{\text{axe}}\,\mathfrak{O}_{\text{bould*}}}$ landscape when the axe is unequipped. Empowerment peaks at the axe's location, since seven steps suffice to equip it, destroy the door, and reach the boulder.

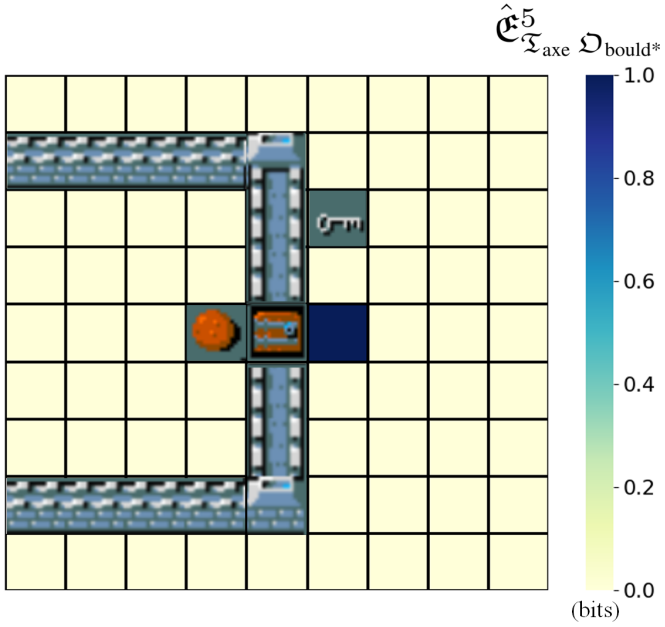


Figure 7.3: 5-step axe-to-boulder empowerment $\hat{\mathfrak{E}}^{5}_{\mathfrak{T}_{\text{axe}}\,\mathfrak{O}_{\text{bould*}}}$ landscape when the axe is equipped. Empowerment values appear in states from which the agent can clear the door and approach the boulder within the horizon.

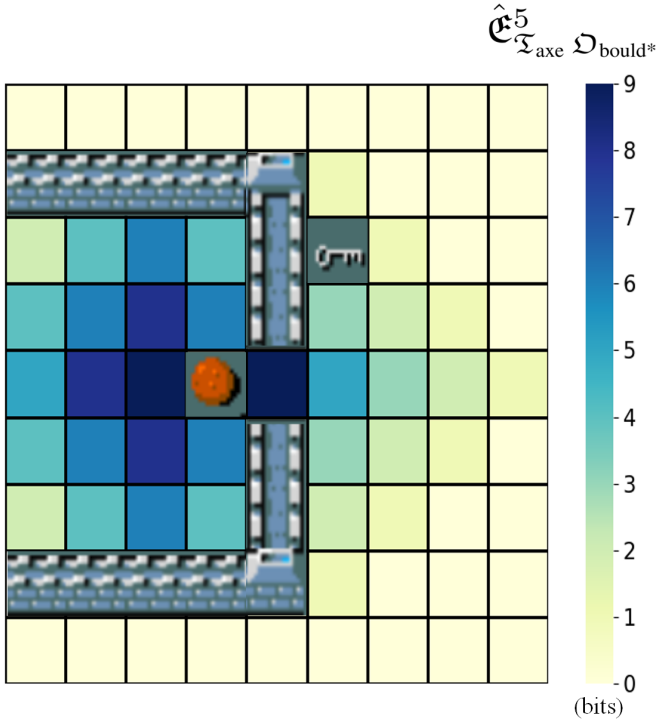


Figure 7.4: 5-step boulder empowerment landscape after the door has been cleared. The empowerment now acts as a gradient that guides the agent towards the boulder.

induced by either tool is non-zero only for horizons $h \geq 7$, reflecting the multi-step causal chain: equip the tool, change the state of the door, and finally push the boulder.

Figure 7.2 shows the 7-step axe-to-boulder empowerment landscape when the axe is not equipped. Non-zero empowerment appears only at the axe's location, since seven steps suffice to collect it, clear the door, and reach the boulder. When the axe is equipped, the 5-step landscape (Figure 7.3) highlights regions from which the agent can destroy the door and subsequently reach the boulder within the horizon. Once the door has been cleared, the boulder empowerment landscape (Figure 7.4) becomes much richer: because the boulder can be pushed multiple times in different directions, its empowerment increases with proximity and h, effectively forming a gradient towards the boulder that guides exploration.

Although both tools act in the same way in the present scenario, simply enabling access to the room so that the agent can reach and push the boulder, their distinction becomes apparent in more temporally extended situations. For instance, if an external threat (e.g., a monster) were present outside the room, the agent would benefit from the key's reversibility: it could open the door, enter the room, and then reclose it to protect itself. The axe, by contrast, would leave the door permanently open once destroyed, removing this possibility of securing the environment. This highlights how persistence captures an important property of tools, one that becomes relevant in temporally extended

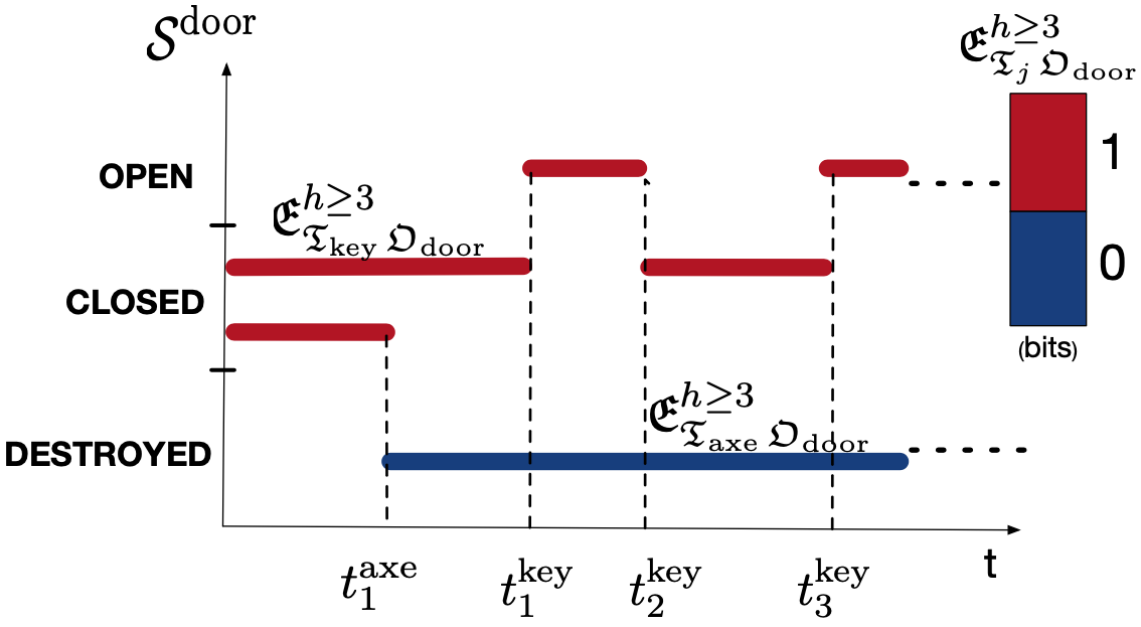


Figure 7.5: Temporal evolution of the door state and its associated object empowerment $\mathfrak{E}^{h\geq 3}_{\mathfrak{T}_j,\mathfrak{O}_{\text{door}}}$ for the key (red) and the axe (blue). The vertical axis enumerates the possible semantic states of the door (*open*, *closed*, *destroyed*), and $\mathcal{S}^{\text{door}}$ denotes the subset of the full state space in which the door still exists as a manipulable object. The horizontal axis shows discrete time steps, where t_1^{axe} marks the moment the agent uses the axe and irreversibly destroys the door, while $t_1^{\text{key}}, t_2^{\text{key}}, t_3^{\text{key}}$ correspond to repeated openings and closings of the door using the key. The colour bar on the right encodes the value of object empowerment in bits. For the key (red bars), empowerment remains at 1 bit for all $h \geq 3$, because the agent can always choose between two distinct future door states (open or closed) and can repeat this interaction indefinitely. For the axe (blue bar), empowerment is non-zero only once: after destruction there is no alternative future state of the door, so empowerment collapses to 0 bits and remains there permanently. Thus, the figure illustrates the difference between reversible (persistent) and irreversible (non-persistent) tool–object interactions.

tasks.

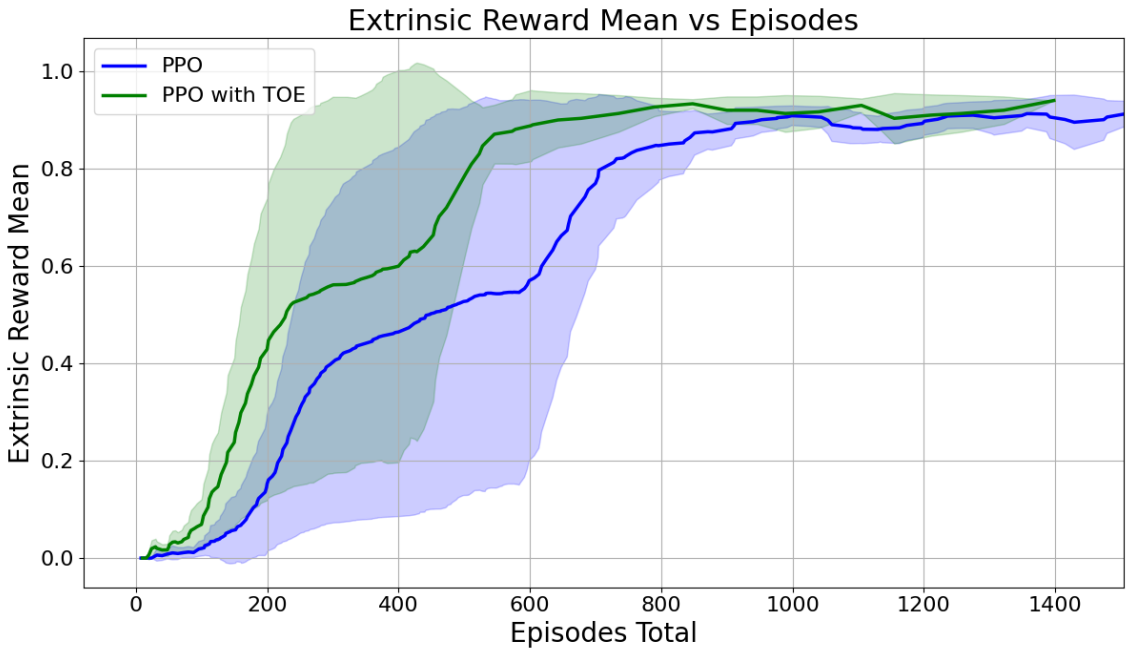


Figure 7.6: Learning curve for the persistence experiment. The empowerment-regularised agent (green) learns faster and achieves higher performance than standard PPO (blue). Shaded regions represent standard deviation over 10 runs.

This contrast can be formalised using the definition of h-step persistence introduced earlier. Although both tools yield the same instantaneous empowerment value $\mathfrak{E}^3_{\mathfrak{T}_j\mathfrak{O}_{\text{door}}} = 1$ bit when the agent stands in front of the door (the door is either cleared or not), their temporal behaviour differs once the set $Z^h_{\mathfrak{T}_j\mathfrak{O}_{\text{door}}} = \{s \in \mathcal{S} \mid \mathfrak{E}^h_{\mathfrak{T}_j\mathfrak{O}_{\text{door}}}(s) \neq 0\}$ is considered. For the key, every key-mediated action keeps the agent inside this set, and from any initial state in S_0 the agent can eventually reach and remain in states where the door can still be opened or closed. According to the formal criteria, the key is therefore *inevitably h-step persistent* with respect to the door: the interaction is reversible, and empowerment remains non-zero indefinitely. For the axe, however, once the door is destroyed, the resulting state leaves $Z^h_{\mathfrak{T}_j\mathfrak{O}_{\text{door}}}$ permanently, since no action sequence can restore the door or produce further transitions. The closure condition of persistence is violated, meaning the axe is *not h-step persistent*: it enables only a single irreversible effect, after which empowerment collapses to zero. This behaviour is reflected in Figure 7.5: key-to-door empowerment stays constant over time, whereas axe-to-door empowerment drops to zero immediately after destruction, differentiating persistent (reversible) from non-persistent (irreversible) tools.

Supplementary RL Evaluation

Although the focus of this section remains on analysing the temporal persistence of tool–object empowerment (as shown in Figure 7.5), it is informative to examine how object empowerment influences policy learning in the same environment setup. The following RL experiment serves this complementary purpose. While not intended to measure persistence directly, the setup highlights how empowerment can support multi-step decision making in tasks involving sequential tool use and object interactions.

Learning performance is reported in Figure 7.6. The agent trained with object empowerment regularisation (using $h = 7$ before equipping the axe and $h = 5$ afterward) converges faster and attains higher asymptotic returns compared to the PPO baseline. This improvement demonstrates how object empowerment can encourage policies that account for multi-step dependencies, such as clearing intermediate obstacles to eventually reach the goal object.

7.2.2 Experiment 2: Latency of Tools

This experiment examines the concept of *latency* by comparing tools that differ in the number of steps required for their influence to manifest on the target object. The environment (Figure 7.7) is a 10 x 10 grid-world based on MiniHack that contains two tools: a wand and a pickaxe, and one target object, a boulder. The agent (bottom-right corner) must destroy the boulder (top-left corner) while navigating around static obstacles in the form of iron bars, which act as impassable walls that cannot be destroyed or modified by any tool. To ensure a fair comparison between tools, the environment is designed such that when the agent exits the area enclosed by iron bars, both the pickaxe and the wand are positioned at equal distances from it. This prevents any spatial bias from influencing tool selection.

Both tools can destroy the boulder, but they differ in how quickly their effects can be realised. The *pickaxe* requires physical proximity: the agent must reach a cell adjacent to the boulder before acting. By contrast, the *wand* affords remote interaction. Once equipped, it can destroy the boulder from any position aligned orthogonally with it (i.e., in the same row or column). This enables faster causal impact and results in a much lower latency. The reward structure penalises prolonged trajectories by assigning a reward of 0 when the boulder is destroyed and -1 for every time step otherwise, encouraging the agent to identify and exploit tools that afford faster causal impact.

Table 7.2 reports the state-averaged latency values for each tool–object pair, computed according to Equation (7.1.2). The wand exhibits substantially lower latency ($\hat{\mathcal{L}}_{\mathfrak{T}_{\text{wand}}\mathfrak{O}_{\text{bould*}}} = 0.0582$) than the pickaxe ($\hat{\mathcal{L}}_{\mathfrak{T}_{\text{pickaxe}}\mathfrak{O}_{\text{bould*}}} = 0.1100$), confirming that the wand enables faster

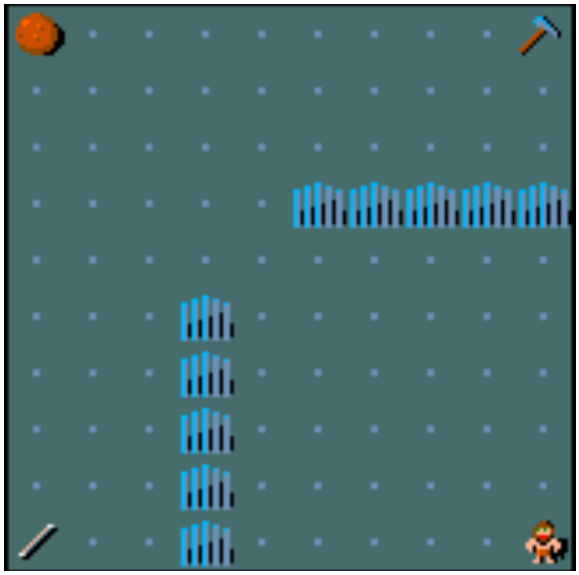


Figure 7.7: Initial state of the "latency" experiment. The agent (bottom right) must destroy the boulder (top left) using either the wand (bottom left) or the pickaxe (top right). Blue-grey bars represent static obstacles that restrict movement but cannot be destroyed.

causal influence on the target object. This quantitative difference is consistent with the spatial latency patterns described below. Although this difference is intuitively clear, it is now made objective through the proposed latency measure

Table 7.2: State-averaged latency $\hat{L}_{\mathfrak{T}_j\mathfrak{O}_i}$ for each tool–object pair in the latency experiment. Lower values indicate faster causal influence.

	$\mathfrak{O}_{\text{bould*}}$
$\mathfrak{T}_{\text{wand*}}$	0.0582
$\mathfrak{T}_{\text{pickaxe*}}$	0.1100

The latency landscapes of both tools are shown in Figures 7.8 and 7.9. The wand exhibits consistently low latency values across most of the grid, never exceeding 12, reflecting its ability to act from a distance once equipped. In contrast, the pickaxe displays a steep spatial gradient in latency, increasing with distance from the boulder and reaching a maximum of 20 at the opposite corner. This highlights its reliance on close physical contact before any effect can occur, making it a high-latency tool.

This difference is also evident in the corresponding object-empowerment landscapes (Figures 7.10–7.11). When the wand is unequipped (Figure 7.10), empowerment remains concentrated at the wand's location for low horizons h, as the agent must first reach

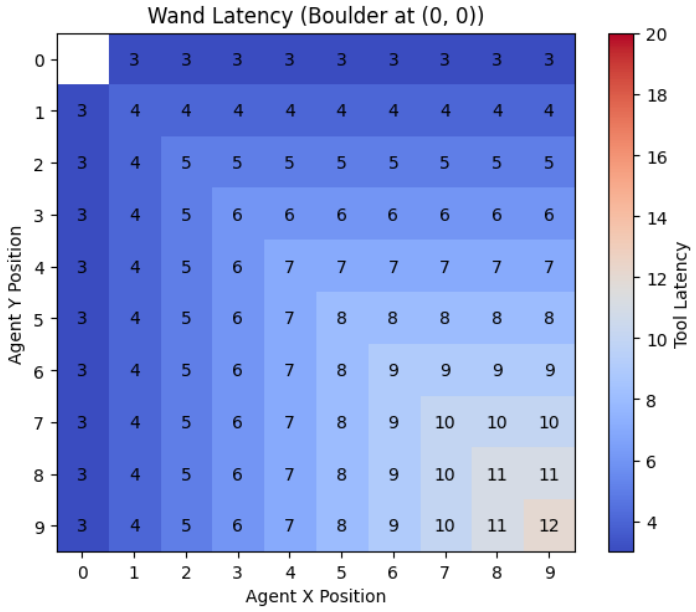


Figure 7.8: Wand latency landscape for the boulder at $(0, 0)$. Latency grows with Manhattan distance, since the wand can only act when the agent is aligned in the same row or column as the boulder (diagonal positions do not allow activation). The maximum latency of 12 arises from the farthest starting state $(9, 9)$, which requires 9+3 steps: 9 movement steps to reach a valid orthogonal cell and 3 additional steps to apply the wand. Lower latency (blue) corresponds to states from which causal influence can be exerted more quickly.

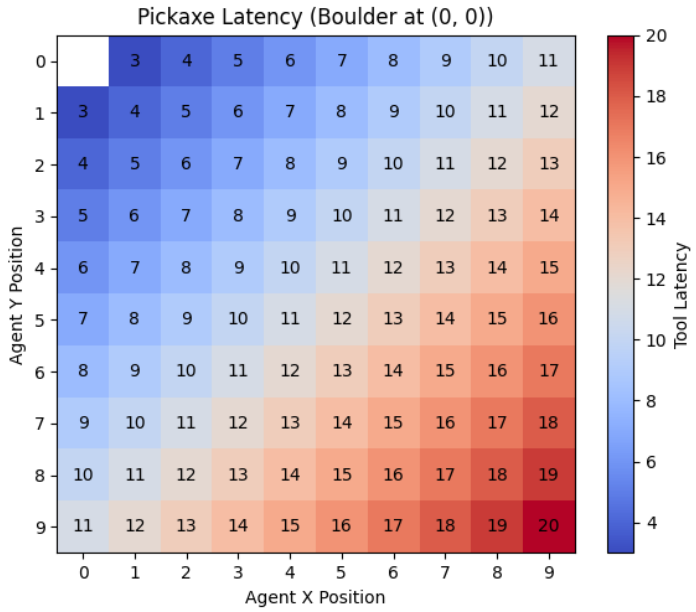


Figure 7.9: Pickaxe latency landscape for the boulder at $(0, 0)$. Latency grows with spatial distance, reaching a maximum of 20 at the bottom-right corner. The pickaxe requires the agent to approach the boulder directly, resulting in high latency in distant regions.

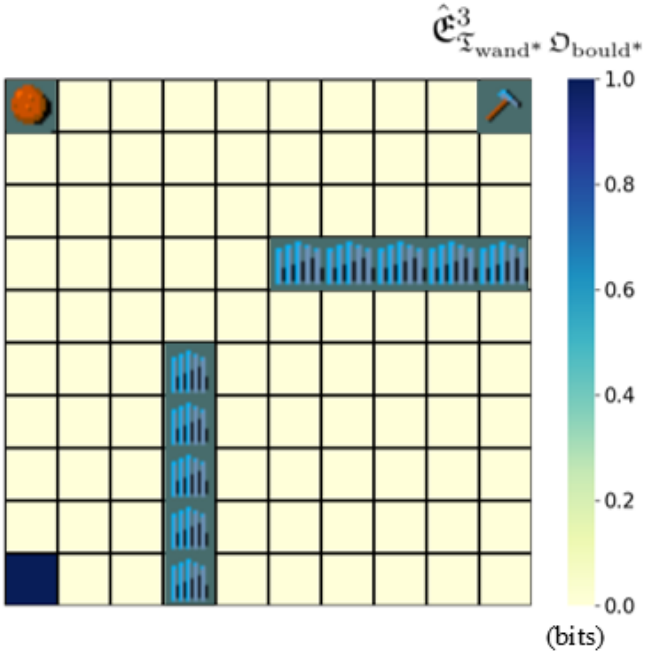


Figure 7.10: 3-step wand-to-boulder empowerment $\hat{\mathfrak{E}}^3_{\mathfrak{T}_{\text{wand}}\,\mathfrak{O}_{\text{bould}^*}}$ landscape (in bits) for all possible agent locations when the wand is not equipped.

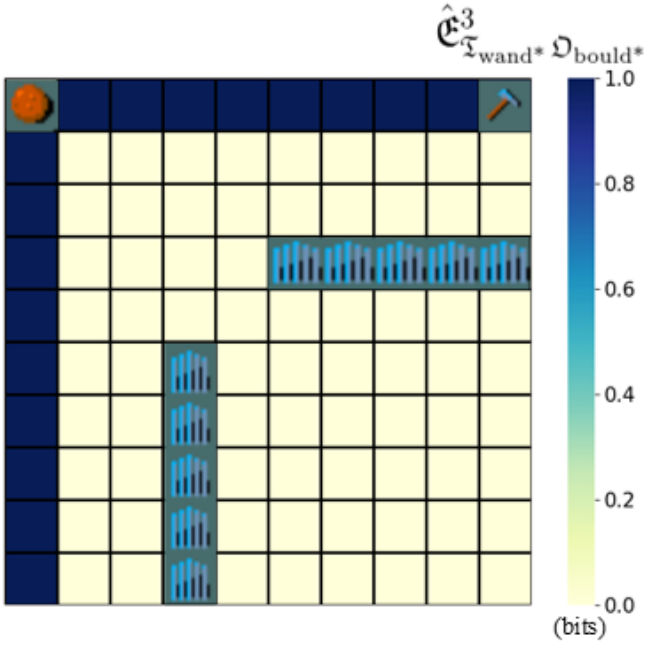


Figure 7.11: 3-step wand-to-boulder empowerment $\hat{\mathfrak{E}}^3_{\mathfrak{T}_{\text{wand}}\,\mathfrak{O}_{\text{bould}^*}}$ landscape (in bits) for all possible agent locations when the wand is equipped.

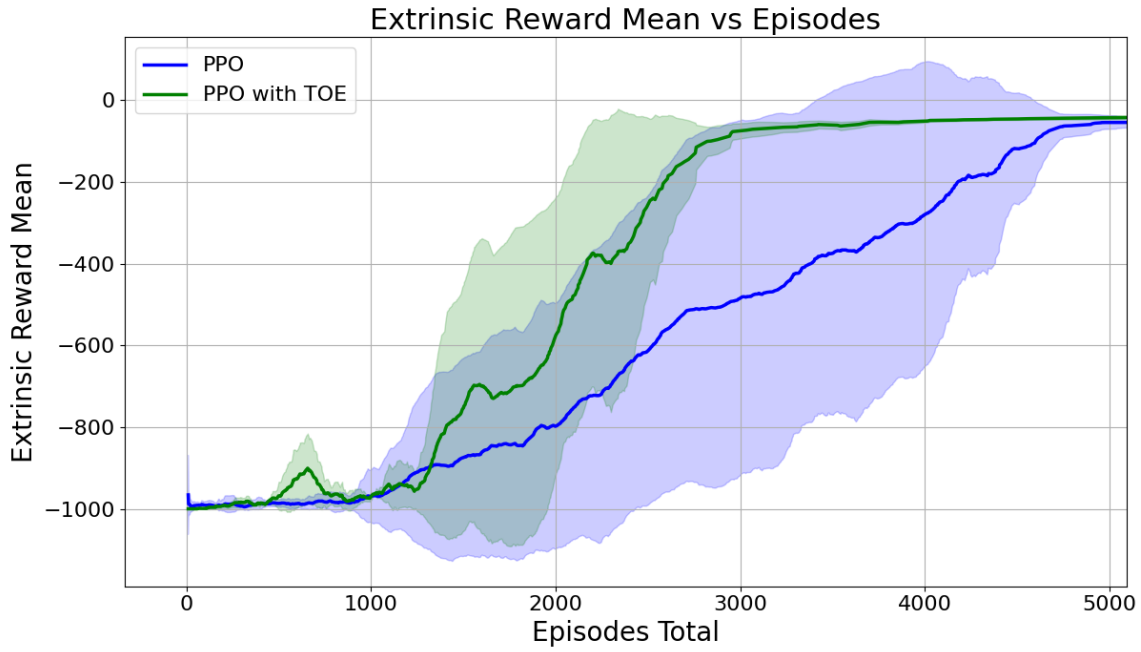


Figure 7.12: Learning curve for the latency experiment. The empowerment-regularised agent (green) learns faster and achieves higher performance than standard PPO (blue). Shaded regions represent standard deviation over 10 runs.

and equip it before acting on the boulder. Once equipped (Figure 7.11), empowerment values appear in all positions orthogonal to the boulder, indicating that the agent can influence it from any of these directions without further movement. This spatial expansion of empowerment after tool acquisition directly illustrates the link between low latency and high accessibility of causal influence.

According to the latency-based selection mechanism defined in Equation (7.1.3), the wand $\mathfrak{T}_{j^*} = \mathfrak{T}_{\text{wand}}$ is identified as the most responsive tool for interacting with the boulder, as it achieves the minimum average latency. In the experiment, the empowerment value $\mathfrak{E}^h_{\mathfrak{T}_{j^*},\mathfrak{O}_{i^*}}$ associated with the selected tool is used as an intrinsic reward in the regularised return function, thereby promoting policies that favour tools with rapid causal efficacy. In a task where minimising the number of steps is crucial, such a low-latency tool is particularly advantageous, as it enables the agent to complete the objective with minimal time steps.

Learning performance is compared between PPO and PPO regularised with wand-to-boulder empowerment. Before the wand is equipped, the agent uses the 3-step empowerment landscape $\mathfrak{E}^3_{\mathfrak{T}_{\text{wand}^*}\mathfrak{O}_{\text{bould}^*}}$ (Figure 7.10) as intrinsic reward, which acts as a beacon guiding exploration toward the wand. Once equipped, the 3-step empowerment landscape $\mathfrak{E}^3_{\mathfrak{T}_{\text{wand}^*}\mathfrak{O}_{\text{bould}^*}}$ (Figure 7.11) is used, capturing the tool's extended range of influence. Figure 7.12 reports the mean cumulative reward per episode, averaged across ten independent

runs. The empowerment-regularised agent converges faster and achieves higher asymptotic performance than the PPO baseline, demonstrating how the latency-based selection mechanism supports more temporally efficient tool use.

7.2.3 Experiment 3: Reliability of Tools

This experiment examines *reliability* by assessing how noise in a tool's dynamics reduces the agent's effective control over objects (i.e. object empowerment) and how this, in turn, affects task learning. The environment is a 13×13 grid world that includes an agent (robot), a movable object (can), walls (black cells) that restrict movement, a goal location (waste bin), and three tools of the same type (pickers). These tools differ *only* in their actuation reliability, which is controlled by a noise parameter $\theta \in [0, 1]$.

Formally, θ specifies the probability that the tool's intended effect is *not* applied during interaction:

$$\theta = \Pr(\text{tool actuation fails}).$$

Thus, a fully reliable tool has $\theta = 0$ (its effect is always successful), while higher values of θ increase the likelihood of tool failure. In this setup, the green picker is fully reliable ($\theta = 0$), the blue picker is moderately unreliable ($\theta = 0.15$), and the red picker is highly unreliable ($\theta = 0.9$). The agent receives a reward of -1 per time step and 0 when the can is successfully placed into the waste bin. To ensure a fair comparison between tools, the environment layout is designed such that, when the agent exits the starting corridor, all three pickers are positioned at equal distances from that location, preventing spatial bias in tool selection. As in Chapters 4 and 5, the agent may use any picker to move the can or push it directly with its body; all other transition dynamics remain unchanged. Notably, at the start of each episode, the can blocks the only exit of the corridor, meaning the agent must first push the can out of the way before it can access any of the pickers. This setup enforces early interaction with the environment's dynamics before tool selection becomes possible.

Reliability is quantified using the object empowerment computed under stochastic tool dynamics. State-averaged values $\hat{\mathfrak{E}}^h_{\mathfrak{T}_j, \mathfrak{O}_i}(\theta)$ populate the tool–object empowerment matrix $\mathbb{T}[j, i]$ (see Equation (6.1.3)).

Table 7.3 reports the state-averaged empowerment values for each tool–object pair under different noise levels θ. The green picker ($\hat{\mathfrak{E}}^h_{\mathfrak{T}_{\text{green\_pick}}\mathfrak{O}_{\text{can}}}(\theta{=}0.0) = 0.7011$) achieves the highest empowerment, reflecting fully deterministic control over the can. As stochasticity increases, empowerment values gradually decline: the blue picker ($\hat{\mathfrak{E}}^h_{\mathfrak{T}_{\text{blue\_pick}}\mathfrak{O}_{\text{can}}}(\theta{=}0.15) = 0.6818$) retains moderate empowerment, while the red picker ($\hat{\mathfrak{E}}^h_{\mathfrak{T}_{\text{red\_pick}}\mathfrak{O}_{\text{can}}}(\theta{=}0.9) = 0.6518$) shows the lowest empowerment. While the absolute decrease (from 0.70 to 0.65) may ap-

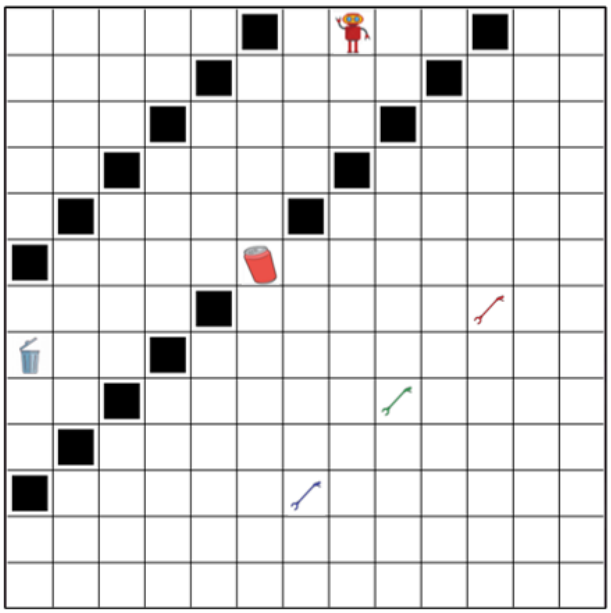


Figure 7.13: Initial state of the reliability experiment. The can must be brought into the bin. Three pickers are available and differ only by their noise level θ: green (0), blue (0.15), red (0.9).

pear relatively small, this is expected: noise is injected *only* during the picker's actuation phase (e.g., during rotation), whereas all other state transitions remain unaffected. Since empowerment is averaged over all states in the environment, and many of these do not involve tool use at all, the cumulative effect of actuation noise is naturally attenuated. Nevertheless, the monotonic decrease in empowerment with increasing θ clearly captures the relationship between unreliability and loss of causal control.

Table 7.3: State-averaged empowerment $\hat{\mathfrak{E}}^{h}_{\mathfrak{T}_j\mathfrak{O}_i}(\theta)$ (in bits) for each tool–object pair under different noise levels θ. Lower values indicate reduced reliability due to stochastic tool dynamics.

Tool	Noise level θ	$\hat{\mathfrak{E}}^{h}_{\mathfrak{T}_j\mathfrak{O}_{\text{can}}}(\theta)$
$\mathfrak{T}_{\text{green\_pick}}$	0.0	0.7011
$\mathfrak{T}_{\text{blue\_pick}}$	0.15	0.6818
$\mathfrak{T}_{\text{red\_pick}}$	0.9	0.6518

While state-averaged empowerment captures overall robustness, spatial landscapes reveal where and how reliability influences the agent's potential to affect the object. To capture the full causal chain from tool acquisition to object manipulation, the comparison across tools was conducted with a horizon of h=6. For shorter horizons (e.g., h=1), em-

powerment is highly localised (i.e., 1.0 bit in the cell adjacent to the can and 0 elsewhere), since the agent cannot yet reach, equip, and use the tool within such a limited temporal window. Thus, a longer horizon is needed, which allows the influence of each tool to manifest fully, reflecting the multi-step sequence in which the agent first picks up a tool and subsequently employs it to affect the can.

The resulting 6-step empowerment landscapes for the three tools are shown in Figures 7.14–7.16. To better visualise where the agent can causally influence the object, these landscapes include red outlines highlighting the regions where tool-mediated actions directly affect the can's state. For the fully reliable green picker (θ=0.0; Figure 7.14), empowerment peaks at 4.86 bits in the cells directly adjacent to the tool, forming a broad and symmetric field that extends toward the can. This indicates that the agent can consistently acquire and use the green picker to act on the can from a wide range of starting positions, reflecting strong and deterministic causal control.

For the moderately unreliable blue picker (θ=0.15; Figure 7.15), the empowerment field weakens and becomes more fragmented. The maximum empowerment value (3.81 bits) now appears both near the can and next to the blue picker, suggesting that the agent's influence remains effective but less stable, as uncertainty occasionally disrupts successful tool use.

In contrast, the highly unreliable red picker (θ=0.9; Figure 7.16) shows a pronounced collapse of empowerment around the tool itself. Here, the highest values (3.81 bits) occur only near the can, while the cells adjacent to the red picker exhibit reduced empowerment (2.40 bits). This pattern indicates that, under high stochasticity, the agent cannot reliably use the tool to influence the can. As a result, empowerment values no longer reflect mediated control via the tool, but rather arise from the agent's ability to affect the can through direct body contact when in close proximity. Together, these landscapes demonstrate how increasing noise progressively suppresses empowerment both spatially and in magnitude, offering a clear visual representation of how reliability governs causal reach. Intuitively, higher noise reduces both the magnitude and spatial extent of empowerment peaks; reliable tools retain broad, high-value regions, whereas unreliable tools exhibit sparse, low-valued landscapes.

Figure 7.17 presents the 1-step classical empowerment landscape. Unlike object empowerment, which quantifies the agent's potential influence on a specific object, classical empowerment reflects the agent's overall capacity to influence its environment. Interestingly, the highest empowerment values (3.00 bits) occur near the most reliable (green) picker, while the second-highest values (2.70 bits) appear near the moderately reliable (blue) picker. In contrast, cells adjacent to the highly unreliable (red) picker exhibit empowerment values around 2.33 bits, only marginally higher than the most of the environment (2.32 bits). This alignment between classical and object empowerment supports the

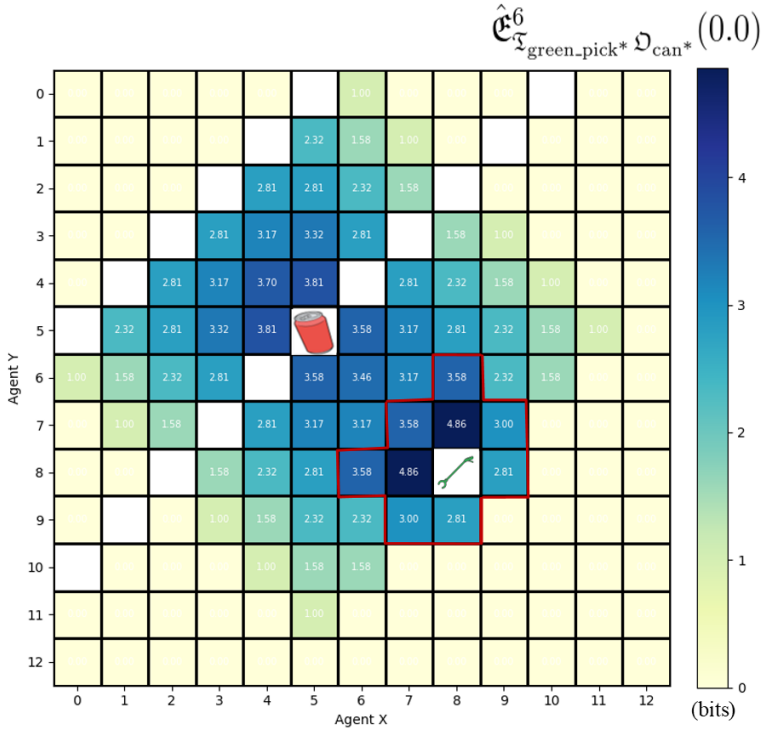


Figure 7.14: 6-step green_picker-to-can empowerment landscape $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}_{\text{green\_pick}}\mathfrak{O}_{\text{can}^*}}(\theta{=}0.0)$. Empowerment peaks (4.86 bits) near the fully reliable (green) picker and extends broadly toward the can, reflecting stable and deterministic tool dynamics.

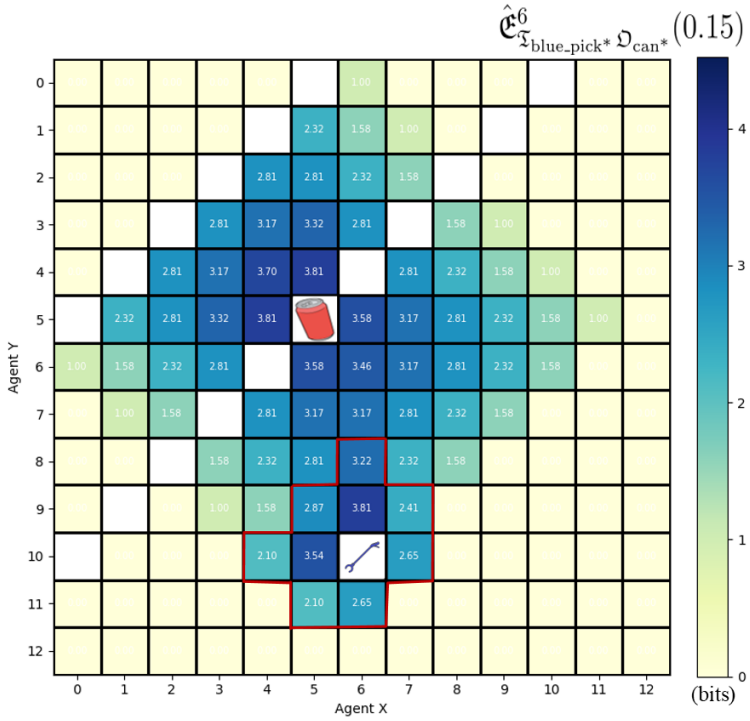


Figure 7.15: 6-step blue_picker-to-can empowerment landscape $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}_{\text{blue\_pick}}\mathfrak{O}_{\text{can}^*}}(\theta{=}0.15)$. Peak empowerment (3.81 bits) occurs both near the can and adjacent to the moderately reliable (blue) picker, reflecting partially degraded but functional causal control.

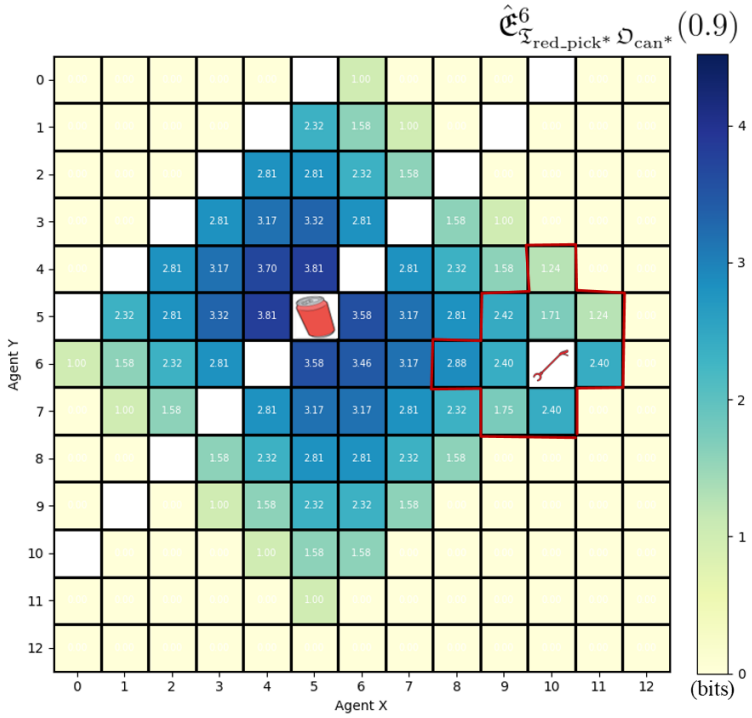


Figure 7.16: 6-step red_picker-to-can empowerment landscape $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}_{\text{red\_pick}}\mathfrak{O}_{\text{can}^*}}(\theta{=}0.9)$. Maximum empowerment (3.81 bits) is limited to the can's vicinity, while values near the unreliable (red) picker drop to 2.40 bits, indicating unstable tool influence under high noise.

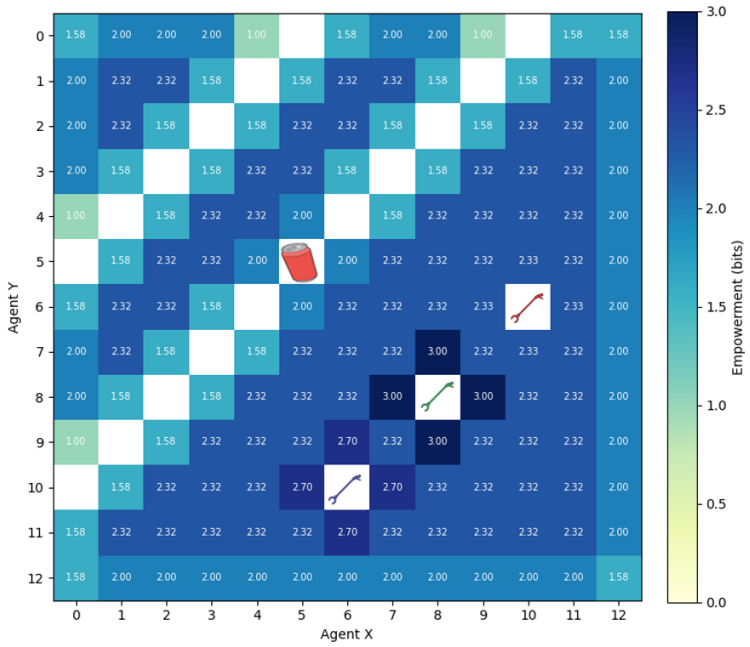


Figure 7.17: 1-step classical empowerment landscape. The highest empowerment values (3.00 bits) occur near the most reliable (green) picker, followed by 2.70 bits near the moderately reliable (blue) picker, and 2.33 bits near the unreliable (red) picker. Notably, the empowerment around the unreliable picker is only marginally higher than the most of the environment (2.32 bits). This pattern mirrors the reliability hierarchy observed in object empowerment.

interpretation that reliability, and thus predictability of action outcomes, is a key factor determining an agent's causal control. While classical empowerment also reveals which tools are more reliable in general, object empowerment refines this insight by quantifying reliability specifically in terms of the tool's capacity to influence the task-relevant object.

According to the selection mechanism defined in Equation (6.1.4), the tool with the highest state-averaged empowerment under noise is selected as the most reliable for acting on the object. In this case, the green picker $\mathfrak{T}_{j^*} = \mathfrak{T}_{\text{green\_pick}}$ achieves the maximum empowerment $\hat{\mathfrak{E}}^{h}_{\mathfrak{T}_{\text{green\_pick}}\mathfrak{O}_{\text{can}}}(\theta{=}0.0) = 0.7011$, indicating the most stable causal control over the can. In the experiment, the object empowerment value $\mathfrak{E}^{h}_{\mathfrak{T}_{j^*},\mathfrak{O}_{i^*}}$ associated with the selected tool is used as an intrinsic reward in the regularised return function, guiding the agent toward interactions that favour reliable tools with consistent effects.

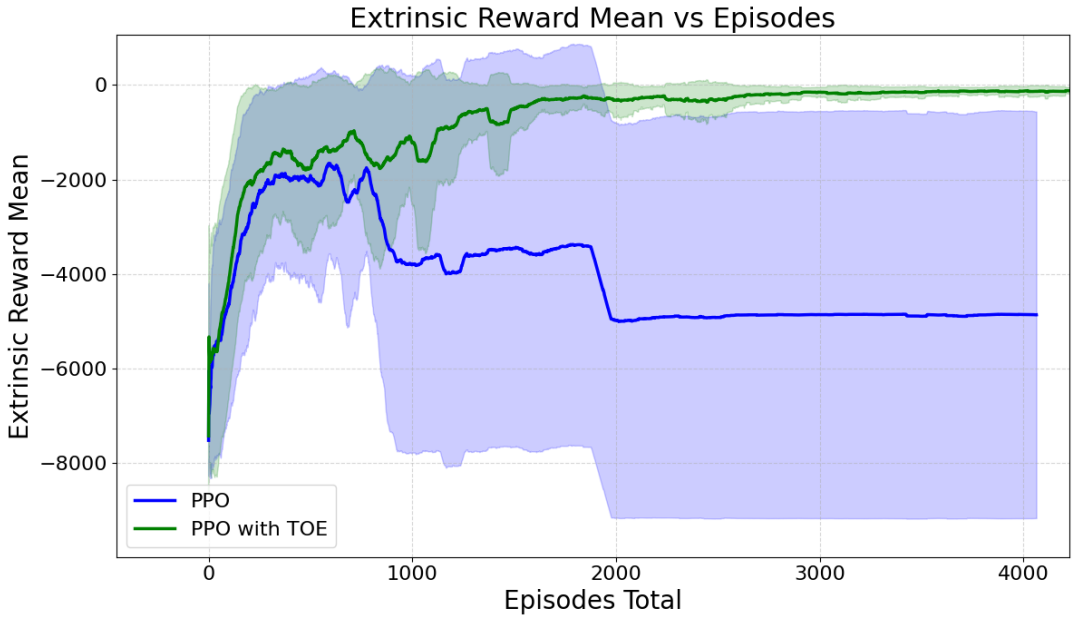


Figure 7.18: Learning performance comparison between PPO and PPO regularised with the 6-step green_picker-to-can empowerment $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}_{\text{green\_pick}}\mathfrak{O}_{\text{can}}}$ ($\beta = 0.004$). Curves show the mean extrinsic reward over training episodes. Shaded regions represent standard deviation over 10 runs.

Figure 7.18 compares the learning performance of standard PPO with PPO regularised with the 6-step green_picker-to-can empowerment $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}_{\text{green\_pick}}\mathfrak{O}_{\text{can}}}$. The regularised agent achieves faster convergence and higher asymptotic performance. This shows that object empowerment-guided exploration helps stabilise learning by biasing the policy toward consistent and reliable interactions. In contrast, some of the unregularised PPO runs suffer from *catastrophic forgetting*. After initially learning to solve the task, these agents later fail to retain or recover the optimal behaviour, which leads to a pronounced drop in the average

return. This instability arises because the unregularised agent overfits to specific successful action sequences that are highly sensitive to the stochasticity of the tool dynamics (e.g., unreliable picker actuation). When these brittle sequences fail due to noise, the agent receives contradictory feedback and gradually unlearns the previously successful behaviour. By incorporating object empowerment as an auxiliary intrinsic signal, the agent maintains a more coherent behavioural structure and avoids such regressions. This demonstrates that reliability-aware object empowerment improves both robustness and learning stability in tool-mediated control tasks.

7.3 Summary

This chapter presented a complementary and novel perspective on tool-use by *characterising* tools along three empowerment-based dimensions: *persistence*, *latency*, and *reliability*. While previous empowerment studies have focused primarily on action selection or agent-centric exploration, this chapter is the first to formalise how empowerment can be used to *describe, compare, and classify* tools themselves. Within the same tool-learning framework as Chapter 6, three empowerment-based measures were introduced. Persistence was defined using the set of states $Z^h_{\mathfrak{T}_j\mathfrak{O}_i}$ in which a tool–object pair has non-zero empowerment, and expressed in terms of *inevitable* and *possible* h-step persistence, depending on whether all or some trajectories eventually reach and remain in these controllable states. This formalism captures whether the tool's influence on an object is reversible and can be sustained over time. Latency was formalised as the first horizon at which object empowerment becomes non-zero, yielding a tool–object *latency matrix* and a latency-based selection rule. Finally, reliability was characterised as the empowerment that a tool retains when stochasticity is introduced specifically into its actuation with respect to a target object. Higher empowerment under noise indicates higher reliability. Together, these measures constitute a unified language for describing how long a tool remains useful, how quickly its effects manifest, and how robustly it maintains influence under uncertainty.

Empirical results demonstrated how these characterisations translate into practical learning benefits. In MiniHack, persistence analysis distinguished key–door (reversible) from axe–door (irreversible) interactions, and empowerment regularisation accelerated learning on tasks requiring intermediate object manipulation. Latency experiments showed that a distance-acting tool (wand) achieves markedly lower state-averaged latency than a contact tool (pickaxe), and that using the wand's empowerment as an intrinsic signal improved sample efficiency and final returns. In the custom grid world considered here, empowerment decreased monotonically with increasing noise, both in state-averaged empowerment and in spatial landscapes; the empowerment-based tool selection favoured the most dependable tool, and its empowerment served as a stable intrinsic guide. Across settings, embedding

the selected tool's empowerment into the regularised return consistently promoted faster convergence and more targeted exploration.

Each characterisation was paired with a specific RL evaluation, showing how empowerment-based structure can shape learning: persistence revealed how reversible tools can sustain long-term utility and accelerate convergence; latency demonstrated how tools with faster causal impact can improve sample efficiency; reliability emphasised the stabilising role of empowerment under uncertainty, helping mitigate catastrophic forgetting in noisy domains.

In summary, this chapter extends object empowerment-driven tool-use beyond selection to a principled *characterisation* framework. By quantifying temporal continuity (persistence), time-to-effect (latency), and robustness to noise (reliability), it introduces a novel computational foundation for comparing and categorising tools through empowerment structure alone. These formulations equip agents with actionable signals for preferring reversible interactions, prioritising rapid causal impact, and remaining effective under stochastic dynamics, without any task-specific shaping beyond the empowerment regulariser. The next chapter concludes the thesis by integrating the findings across all proposed frameworks and discussing their broader implications for intrinsic motivation, tool-use, and autonomous behaviour.

Chapter 8

Discussion and Conclusions

8.1 Overview

This thesis has developed a unified framework for understanding and modelling *tool use* through the lens of *empowerment*, an information-theoretic measure of control that quantifies how an agent's actions influence its environment. The central objective was to bridge the conceptual gap between intrinsic motivation and goal-directed interaction by formalising how agents can discover, select, and characterise tools based on their causal influence over task-relevant objects.

The investigation followed a structured progression. Beginning with the foundational concept of empowerment as a measure of control, the work extended it to *object empowerment*, explicitly conditioning the empowerment channel on the state of manipulable objects. This formed the basis for quantifying how agent actions propagate through tools to affect environmental entities. The framework was then expanded in successive chapters to model increasingly complex levels of tool use: from single-object interactions, to learning multi-object dependencies, to selecting between multiple available tools, and finally, to characterising tools along distinct causal and temporal dimensions. The overall progression can be summarised as:

$$\begin{aligned}\text{Empowerment} &\rightarrow \text{Object Empowerment} \rightarrow \text{Learning Tool–Object Interactions} \\ &\rightarrow \text{Tool Selection} \rightarrow \text{Tool Characterisation.}\end{aligned}$$

Each stage introduced a new conceptual and computational layer: from the ability to measure influence, to understanding which tools afford that influence, to reasoning about why certain tools are more effective, reliable, or persistent than others. Together, these developments advance the broader goal of constructing intrinsically motivated agents

capable of autonomous and interpretable tool use.

During the course of this research, several methodological challenges were encountered. A central difficulty lies in enabling agents to discover meaningful tool–object interactions in sparse-reward environments, where useful behaviours may not be immediately reinforced. Another challenge concerns the computational complexity of empowerment calculations, which increases rapidly with planning horizon and state-space size. These considerations motivated the use of controlled grid-world environments for analysing empowerment landscapes and guided the progressive development of the framework from simple agent–object interactions to multi-object tool selection and tool characterisation.

An important open question concerns how the proposed framework would extend to settings in which the agent must learn the environment dynamics from scratch. While the current formulation assumes access to, or reliable estimation of, the transition structure required to compute empowerment, prior work has demonstrated that empowerment can be successfully estimated and used within RL settings where transition dynamics are initially unknown [21, 89, 140]. In such cases, empowerment is learned alongside the agent's model of the environment through interaction. However, this introduces an important practical consideration: during the early stages of learning, when the agent has limited experience, the estimated empowerment signal may be unreliable and effectively act as noise. This suggests that, in the context of object empowerment, it may be beneficial to delay or gradually incorporate the intrinsic signal until sufficient experience has been gathered to produce stable estimates.

A related consideration concerns the distinction between tools and objects, which in this work is explicitly defined as part of the environment. More broadly, this distinction is not sharply defined in the literature, where the role of an entity often depends on its functional use rather than its intrinsic properties. While this assumption enables a clear analysis of tool–object interactions, it may not hold in more general settings where entities can take on different functional roles depending on context. Within the proposed framework, however, this distinction can be operationalised in terms of causal influence. By computing object empowerment between entities, it becomes possible to identify objects that exert non-zero influence over others. In this sense, an entity can be interpreted as a tool if its associated object empowerment with respect to another object exceeds zero, indicating that it mediates a causal effect. This provides a principled mechanism not only for identifying tools but also for quantifying the extent of their influence within the environment. More generally, the proposed framework relies on a structured representation of the environment, including predefined object identities and their relation to tools. While this design enables precise analysis of causal influence and tool selection, it also introduces a dependence on prior knowledge about the environment. Future work could therefore investigate how such

structure might be learned or inferred from interaction, allowing agents to autonomously discover not only how to use tools, but also which entities function as tools in the first place.

Taken together, the developments presented across chapters form a coherent and progressive research trajectory. Rather than listing contributions individually, the following section synthesises these findings into an integrated perspective that links theoretical insights, methodological design, and empirical evidence into a unified account of empowerment-driven tool use.

8.2 Integration of Findings

The findings across all chapters collectively contribute to a unified account of *empowerment-driven tool use*. At its core, empowerment functions as a bridge between information theory and embodied action, allowing agents to discover causal regularities in their environment without requiring task-specific rewards. The successive extensions developed throughout the thesis illustrate how this principle can be systematically expanded to capture higher levels of abstraction in agent–environment coupling.

The introduction of *object empowerment* (Chapter 4) established the foundation for reasoning about manipulable entities by isolating the influence of agent actions on specific environmental objects. This formulation provided a measurable link between intrinsic motivation and object-directed control. Building on this, the *learning tool–object interactions* framework (Chapter 5) showed how empowerment can guide the agent not only to interact with objects, but to recognise the intermediate role of tools as mediators of influence. This was a key conceptual step: the agent transitions from directly manipulating objects to understanding that certain entities (tools) extend its own causal reach.

The subsequent chapters built progressively on this insight. In Chapter 6, empowerment was extended to multiple tools and objects simultaneously, leading to the formulation of the *tool–object empowerment matrix*. This matrix enabled quantitative comparison between alternative tools and formalised the process of tool selection as an empowerment maximisation problem. Empirical validation in MiniHack environments confirmed that empowerment-regularised agents autonomously learned to identify and prioritise the most effective tool for a given object, even under sparse reward conditions.

Finally, Chapter 7 advanced the framework from selection to *characterisation*, introducing new dimensions that describe how tools behave over time and under uncertainty. Persistence captured reversible versus irreversible influence, latency measured temporal efficiency, and reliability quantified robustness under stochastic dynamics. These measures not only enriched the understanding of tool-use dynamics but also provided actionable

intrinsic signals that could be embedded into the agent's learning process.

Taken together, the results demonstrate that empowerment provides a principled basis for modelling tool-use across multiple levels of abstraction: from discovering objects of influence, to selecting appropriate tools, to characterising the properties that make tools effective or reliable. Thus, the framework offers a coherent and extensible foundation for studying intrinsically motivated behaviour, causal reasoning, and autonomous skill acquisition within RL.

8.3 Limitations and Future Work

While the frameworks and experiments presented throughout this thesis establish a coherent foundation for empowerment-driven tool use, several limitations remain that open avenues for future investigation. These limitations concern both theoretical assumptions and practical constraints, as well as broader opportunities for extending the current work to more general and realistic domains.

8.3.1 Learning Empowerment from Interaction

The thesis assumed access to a known or explicitly simulated transition model for computing empowerment and its object-conditioned variants. While this assumption facilitated precise and interpretable analysis, it constrains the framework to environments where dynamics are fully specified or accurately approximated. A natural next step is to develop methods for *learning object empowerment directly from interaction*, allowing agents to infer which tools afford control over which objects and under what conditions, without requiring model access. Such learned empowerment estimators would align the framework with model-free RL and could enable scalable deployment in complex or partially observed domains.

8.3.2 Generalisation Across Tools, Goals, and Environments

Another central limitation concerns generalisation. Although the current experiments systematically compared tools and tasks within a given environment, they did not explicitly address how empowerment-based representations transfer across different environments, goals, or tool sets. Future work should therefore investigate *how empowerment generalises across tasks and domains*, particularly in settings where agents must reason about previously unseen tools or affordances. Bridging this gap may involve hierarchical or meta-learning approaches that abstract empowerment patterns across multiple levels of control.

8.3.3 Comparison with Other Intrinsic Motivations

Beyond empowerment, the intrinsic motivation literature includes several alternative formulations such as curiosity [7], novelty [8], and RND [77]. This thesis focused exclusively on object empowerment as an intrinsic motivation, which extends the classical empowerment formulation by quantifying the degree of control an agent can exert specifically over objects within its environment. This provides a more targeted and interpretable account of tool–object interactions, linking intrinsic motivation directly to task-relevant causal influence.

Earlier results in this thesis (see Chapter 5) also compared object empowerment with classical empowerment in similar tool-use RL settings, showing that classical empowerment can, in some cases, also support effective tool-use learning. In addition, Chapter 5 includes a preliminary comparison with count-based exploration, a widely used novelty-driven intrinsic motivation mechanism. The results indicate that while count-based exploration improves performance relative to standard RL by encouraging broader state-space exploration, it lacks an explicit mechanism to prioritise object-relevant interactions, in contrast to object empowerment, which directly encodes causal influence over task-relevant entities. While these results provide an initial comparison between empowerment-based and novelty-based intrinsic motivations, it would be informative to extend this analysis to a broader range of approaches, such as curiosity-driven exploration and RND. Furthermore, the inclusion of the combined OE+CBE formulation suggests that different intrinsic motivations may play complementary roles, where general exploration incentives and object-centred guidance can be integrated within a unified framework. Exploring such combinations more systematically represents a promising direction for future research, particularly in understanding how different intrinsic signals interact to shape exploration, learning efficiency, and task-specific behaviour. Such a comparison would help determine whether the observed improvements arise specifically from the object-empowerment formulation or represent a more general benefit of intrinsic reward shaping. Conducting a systematic comparison between object empowerment and these alternative intrinsic motivations constitutes a promising direction for future research, offering insights into whether different intrinsic drives converge or diverge in their treatment of causal affordances.

8.3.4 Computational Complexity

Empowerment estimation is computationally intensive, particularly when factoring large state and action spaces or extending to long horizons. Despite the use of parallelisation and multi-step optimisation, scalability remains a practical bottleneck. In discrete deterministic environments, the computation of object empowerment as formulated in Equation (4.1.6)

scales exponentially with the planning horizon. To address this, Monte Carlo sampling methods [96] and UCT-like pruning strategies [141] have been proposed to reduce the complexity of exact empowerment computation in such settings. In addition, the variational estimations [21, 89, 140] also provide efficient approximations for large or high-dimensional discrete environments and can be readily applied to the computation of object empowerment.

8.3.5 Extension to Continuous and Robotic Domains

The current framework operates in discrete grid-world settings, which, while simplified, were deliberately chosen for their transparency and interpretability. In such environments, every element of the causal chain between the agent, tool, and object is explicitly observable, making it possible to visualise empowerment as structured, interpretable landscapes. This design choice allowed clear illustration of how control propagates through the environment, and how factors such as temporal horizon, stochasticity, or tool reliability modulate empowerment. In contrast, real-world or continuous environments often obscure these causal dependencies due to sensor noise, high-dimensional state spaces, and complex dynamics.

Nevertheless, the principles developed throughout this thesis generalise naturally to stochastic, partially observable, and continuous domains. In stochastic discrete environments, the definition of object empowerment in Equation (4.1.4) remains fully valid; only the underlying state transition probabilities become non-deterministic. For partially observable settings, empowerment has been extended such that the receiver variable of the actuation channel corresponds to the agent's observations $\mathcal{O}$ rather than the state $\mathcal{S}$ [10, 97]. This formulation measures the amount of influence that the agent can *perceive*, in the case of object empowerment, quantifying how much change in the object the agent can detect through its sensory feedback.

In continuous domains, empowerment has been successfully approximated using Gaussian-channel-based methods with known [96, 142] and unknown [23] dynamics, as well as through variational approximations [21, 89, 140]. These approaches have enabled applications of empowerment to complex robotic systems [105, 106, 143] and can be directly employed to estimate object empowerment in high-dimensional control spaces. Applying object empowerment to tool-use robotics therefore represents a promising direction for future work, where persistence, latency, and reliability could be studied under realistic physical and perceptual constraints. Such an extension would bridge the current theoretical framework with embodied implementations, advancing empowerment-based reasoning toward adaptive, sensorimotor intelligence in real-world robotic systems.

8.3.6 Extensions of Tool Characterisation Framework

The characterisation of tools through latency, persistence, and reliability in this thesis introduced new intrinsic dimensions for analysing tool-use but also raised further questions for future work. Each dimension offers more than a descriptive utility. They suggest principled strategies for time-sensitive, uncertainty-aware decision-making in autonomous agents.

Latency exposes the time cost associated with tool actuation. In time-critical tasks, even an optimal tool may be discarded if its effect manifests too slowly. Extending empowerment-based selection to reason about time constraints opens the possibility of agents that learn to balance short-term delays with long-term utility.

Persistence, in contrast, captures how long a tool remains useful once activated. Persistent tools offer sustained influence, enabling policies that leverage repeated utility over time. This suggests a dual optimisation problem: agents could favour tools that deliver enduring control in extended interactions, even if they incur high upfront latency. Formulating this trade-off introduces a temporal dimension to tool selection, where long-term value (persistence) and short-term cost (latency) must be balanced within intrinsic or extrinsic objectives.

Reliability further complements this view by quantifying robustness under uncertainty. As discussed in Chapter 7, an unreliable tool may occasionally succeed, but its stochastic nature forces the agent to invest additional effort (e.g., repeated attempts) to compensate for unpredictable outcomes. Integrating reliability more deeply into empowerment-based reasoning would enable agents to anticipate and mitigate failure modes, instead of reacting to stochasticity post hoc. Future work could explore how reliability estimates evolve with experience and how they inform temporal planning in noisy environments.

Together, these three dimensions motivate a broader research direction: designing agents that evaluate tools along multiple, interacting dimensions during autonomous planning. This opens questions such as: How should agents choose between tools that act quickly but unreliably, or slowly but consistently? How can such characterisations be embedded into hierarchical or meta-learning systems to support fast adaptation? And how do these properties interact in continuous, embodied domains where perception and physics introduce additional costs?

Thus, the proposed framework is not just descriptive, it lays the foundation for agents that reason about when, how long, and how reliably tools act. By integrating these dimensions into intrinsic motivation and decision-making, future systems may progress toward robust, context-aware tool use where effectiveness is assessed through both outcome and process.

In summary, these limitations highlight clear paths for extending the present work toward more autonomous, generalisable, and scalable models of empowerment-driven behaviour. Addressing them would strengthen the theoretical and practical impact of empowerment as a unifying principle for intrinsic motivation, tool use, and adaptive control.

8.4 Conclusions

This thesis has presented a unified, information-theoretic account of tool use grounded in the concept of *empowerment*, the causal influence an agent can exert on its environment through its actions. By extending empowerment from a general measure of control to a structured framework for object and tool interaction, the work bridges intrinsic motivation, affordance learning, and autonomous behaviour. Through successive developments, empowerment evolved from a theoretical construct into a functional mechanism that guides agents toward meaningful and interpretable interactions in complex environments.

This work was guided by the central hypothesis that classical empowerment can be extended to model tool use by explicitly capturing the agent's causal influence over task-relevant objects, and that such a formulation can support the discovery, selection, and characterisation of tools in RL. The research questions posed in Chapter 1 were addressed progressively throughout the thesis. In particular, RQ1 concerned how empowerment can be extended to capture object-specific control, which was addressed through the formulation of object empowerment in Chapter 4. RQ2 examined whether object-centred empowerment can serve as an intrinsic signal for discovering functional tool–object interactions, which was demonstrated in Chapter 5. RQ3 investigated how object-empowerment-based intrinsic rewards influence RL dynamics in sparse-reward environments, which was addressed within the same experimental framework through analysis of learning behaviour and convergence properties (Chapter 5).RQ4 focused on how empowerment can be generalised to settings with multiple tools and objects, addressed through the tool selection framework and the formulation of the tool–object empowerment matrix in Chapter 6. Finally, RQ5 examined whether empowerment can provide interpretable dimensions for characterising tools, leading to the introduction of persistence, latency, and reliability in Chapter 7. Together, these results provide a coherent answer to the central research questions by showing how empowerment can be systematically extended from a measure of control to a framework for modelling tool use.

At its foundation, empowerment was revisited as a causal and information-theoretic quantity formalising an agent's potential to influence future states. Building on this, *object empowerment* was introduced to isolate the agent's influence over specific task-relevant entities, providing a finer-grained measure of controllability in structured environments

and establishing a foundation for modelling tool use through empowerment.

The framework was then extended along multiple dimensions. First, *Learning Tool–Object Interactions* demonstrated that empowerment-regularised RL enables agents to autonomously discover functional dependencies between tools and objects, overcoming sparse rewards without explicit supervision. Second, the *Tool Selection* framework introduced multi-object empowerment and the tool–object empowerment matrix, allowing agents to identify the most effective tool for influencing a given object based solely on intrinsic control structure. Finally, *Tool Characterisation* advanced the descriptive scope of the framework, introducing persistence, latency, and reliability as interpretable dimensions for comparing tools in terms of temporal continuity, time-to-effect, and robustness under uncertainty. In particular, the integration of object empowerment with these characterisation measures highlights how causal influence can be analysed not only in terms of magnitude, but also in terms of temporal and stochastic properties. Taken together, object empowerment, persistence, latency, and reliability provide a unified framework for evaluating tools, enabling agents to reason not only about how much control a tool affords, but also how quickly, how consistently, and for how long that control can be exercised. Together, these developments transformed empowerment from a single scalar measure into a comprehensive analytical framework for understanding and generating adaptive, tool-mediated behaviour.

Empirical results across MiniHack and custom grid-world environments validated these theoretical extensions. Object empowerment provided dense, causally meaningful feedback that improved exploration and skill acquisition; multi-object empowerment and tool selection supported systematic decision-making in multi-tool, multi-goal settings; and the characterisation metrics revealed how distinct aspects of tool dynamics shape control and learning efficiency. Across all experiments, empowerment-based regularisation consistently yielded faster convergence, more targeted exploration, and more interpretable behaviour than standard baselines, which demonstrates the viability of empowerment as both a theoretical principle and a practical substrate for intrinsic motivation.

Overall, this thesis contributes a coherent progression from *measuring control* to *using control*, culminating in a framework that enables agents not only to quantify their causal influence but to exploit it for structured, autonomous tool use. By integrating empowerment with RL, it bridges causal reasoning with adaptive behaviour, offering a computational account of how agents can discover, select, and evaluate tools in dynamic environments. These results position empowerment as a general principle for understanding the emergence of purposeful, self-organising behaviour, one that scales naturally from simple grid worlds to embodied, interactive systems.

In the context of recent developments in intrinsic motivation and RL, this work contributes a structured, object-centred perspective that complements existing approaches

based on novelty and related exploration mechanisms. In contrast to such methods, which primarily encourage broad state-space exploration, the proposed framework emphasises causal influence over task-relevant objects, providing a more directed and interpretable basis for autonomous behaviour. At the same time, this formulation relies on several simplifying assumptions, including access to, or reliable estimation of, environment dynamics and predefined object–tool representations. These limitations highlight important directions for future work, particularly in extending empowerment-based formulations to settings where such structure must be learned from interaction and applied in more complex or less structured environments.

The thesis thus closes with a central insight: empowerment, when applied to the domain of tool use, provides more than an intrinsic drive. It offers a language for constructing agents that act not merely to survive or explore, but to *understand and shape* their own possibilities for influence.

Bibliography

[1] A. S. Klyubin, D. Polani, and C. L. Nehaniv, *All else being equal be empowered*, in *European Conference on Artificial Life*, pp. 744–753, Springer, 2005.

[2] R. S. Amant and A. B. Wood, *Tool use for autonomous agents.*, in *AAAI*, pp. 184–189, 2005.

[3] S. L. Washburn, *Speculations on the interrelations of the history of tools and biological evolution*, *Human Biology* **31** (1959), no. 1 21–31.

[4] G. Baldassarre, *What are intrinsic motivations? a biological perspective*, in *2011 IEEE international conference on development and learning (ICDL)*, vol. 2, pp. 1–8, IEEE, 2011.

[5] G. Baldassarre, T. Stafford, M. Mirolli, P. Redgrave, R. M. Ryan, and A. Barto, *Intrinsic motivations and open-ended development in animals, humans, and robots: an overview*, *Frontiers in psychology* **5** (2014) 985.

[6] A. Aubret, L. Matignon, and S. Hassas, *An information-theoretic perspective on intrinsic motivation in reinforcement learning: A survey*, *Entropy* **25** (2023), no. 2 327.

[7] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, *Curiosity-driven exploration by self-supervised prediction*, in *International conference on machine learning*, pp. 2778–2787, PMLR, 2017.

[8] A. Barto, M. Mirolli, and G. Baldassarre, *Novelty or surprise?*, *Frontiers in psychology* **4** (2013) 907.

[9] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel, *Vime: Variational information maximizing exploration*, *Advances in neural information processing systems* **29** (2016).

[10] A. S. Klyubin, D. Polani, and C. L. Nehaniv, *Empowerment: A universal agent-centric measure of control*, in *2005 ieee congress on evolutionary computation*, vol. 1, pp. 128–135, IEEE, 2005.

[11] A. Barto and S. R. Sutton, *Reinforcement learning: an introduction*. The MIT Press, 2018.

[12] K. J. Åström and R. Murray, *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2021.

[13] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: theory and practice*. Elsevier, 2004.

[14] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar, *Bayesian reinforcement learning: A survey*, *Foundations and Trends® in Machine Learning* **8** (2015), no. 5-6 359–483.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., *Human-level control through deep reinforcement learning*, *nature* **518** (2015), no. 7540 529–533.

[16] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., *Mastering the game of go with deep neural networks and tree search*, *nature* **529** (2016), no. 7587 484–489.

[17] N. Chentanez, A. Barto, and S. Singh, *Intrinsically motivated reinforcement learning*, *Advances in neural information processing systems* **17** (2004).

[18] S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg, *Intrinsically motivated reinforcement learning: An evolutionary perspective*, *IEEE Transactions on Autonomous Mental Development* **2** (2010), no. 2 70–82.

[19] D. Biro, M. Haslam, and C. Rutz, *Tool use as adaptation*, 2013.

[20] A. G. Barto, *Intrinsic motivation and reinforcement learning*, in *Intrinsically motivated learning in natural and artificial systems*, pp. 17–47. Springer, 2012.

[21] S. Mohamed and D. Jimenez Rezende, *Variational information maximisation for intrinsically motivated reinforcement learning*, *Advances in neural information processing systems* **28** (2015).

[22] H. Bharadhwaj, M. Babaeizadeh, D. Erhan, and S. Levine, *Information prioritization through empowerment in visual model-based rl*, *arXiv preprint arXiv:2204.08585* (2022).

[23] R. Zhao, P. Abbeel, and S. Tiomkin, *Efficient online estimation of empowerment for reinforcement learning*, *arXiv preprint arXiv:2007.07356* (2020).

[24] P.-Y. Oudeyer and F. Kaplan, *What is intrinsic motivation? a typology of computational approaches*, *Frontiers in neurorobotics* **1** (2007) 108.

[25] R. Zhao, Y. Gao, P. Abbeel, V. Tresp, and W. Xu, *Mutual information state intrinsic control*, *arXiv preprint arXiv:2103.08107* (2021).

[26] R. W. Shumaker, K. R. Walkup, and B. B. Beck, *Animal tool behavior: the use and manufacture of tools by animals*. JHU Press, 2024.

[27] R. St Amant and T. E. Horton, *Revisiting the definition of animal tool use*, *Animal Behaviour* **75** (2008), no. 4 1199–1208.

[28] C. Boesch and H. Boesch, *Tool use and tool making in wild chimpanzees*, *Folia primatologica* **54** (1990), no. 1-2 86–99.

[29] J. Goodall, *The chimpanzees of gombe: patterns of behaviour*, *Harvard University Press* (1986).

[30] C. Sousa, *Use of leaves for drinking water*, *The chimpanzees of Bossou and Nimba* (2011) 85–96.

[31] D. M. Fragaszy, D. Biro, Y. Eshchar, T. Humle, P. Izar, B. Resende, and E. Visalberghi, *The fourth dimension of tool use: temporally enduring artefacts aid primates learning to use tools*, *Philosophical Transactions of the Royal Society B: Biological Sciences* **368** (2013), no. 1630 20120410.

[32] L. V. Luncz, A. Tan, M. Haslam, L. Kulik, T. Proffitt, S. Malaivijitnond, and M. Gumert, *Resource depletion through primate stone technology*, *Elife* **6** (2017) e23647.

[33] G. R. Hunt, *Manufacture and use of hook-tools by new caledonian crows*, *Nature* **379** (1996), no. 6562 249–251.

[34] A. H. Taylor, G. R. Hunt, J. C. Holzhaider, and R. D. Gray, *Spontaneous metatool use by new caledonian crows*, *Current Biology* **17** (2007), no. 17 1504–1507.

[35] J. Chappell and A. Kacelnik, *Tool selectivity in a non-primate, the new caledonian crow (corvus moneduloides)*, *Animal cognition* **5** (2002) 71–78.

[36] A. M. Auersperg, B. Szabo, A. M. Von Bayern, and A. Kacelnik, *Spontaneous innovation in tool manufacture and use in a goffin's cockatoo*, *Current Biology* **22** (2012), no. 21 R903–R904.

[37] J. F. Walsh, J. Grunewald, and B. Grunewald, *Green-backed herons (butorides striatus) possibly using a lure and using apparent bait*, *Journal of ornithology* **126** (1985), no. 4 439–442.

[38] K. Hall and G. B. Schaller, *Tool-using behavior of the california sea otter*, *Journal of Mammalogy* **45** (1964), no. 2 287–298.

[39] M. Krützen, J. Mann, M. R. Heithaus, R. C. Connor, L. Bejder, and W. B. Sherwin, *Cultural transmission of tool use in bottlenose dolphins*, *Proceedings of the National Academy of Sciences* **102** (2005), no. 25 8939–8943.

[40] J. Mann, B. L. Sargeant, J. J. Watson-Capps, Q. A. Gibson, M. R. Heithaus, R. C. Connor, and E. Patterson, *Why do dolphins carry sponges?*, *PloS one* **3** (2008), no. 12 e3868.

[41] J. D. Pierce Jr, *A review of tool use in insects*, *Florida Entomologist* (1986) 95–104.

[42] J. H. Fellers and G. M. Fellers, *Tool use in a social insect and its implications for competitive interactions*, *Science* **192** (1976), no. 4234 70–72.

[43] T. Tanaka and Y. Ono, *The tool use by foragers of aphaenogaster famelica.*, *Japanese Journal of Ecology* **28** (1978).

[44] M. H. Möglich and G. D. Alpert, *Stone dropping by conomyrma bicolor (hymenoptera: Formicidae): a new technique of interference competition*, *Behavioral Ecology and sociobiology* (1979) 105–113.

[45] A. H. Taylor and R. D. Gray, *Is there a link between the crafting of tools and the evolution of cognition?*, *Wiley Interdisciplinary Reviews: Cognitive Science* **5** (2014), no. 6 693–703.

[46] S. Semaw, P. Renne, J. W. Harris, C. S. Feibel, R. L. Bernor, N. Fesseha, and K. Mowbray, *2.5-million-year-old stone tools from gona, ethiopia*, *Nature* **385** (1997), no. 6614 333–336.

[47] D. Stout, N. Toth, K. Schick, and T. Chaminade, *Neural correlates of early stone age toolmaking: technology, language and cognition in human evolution*, *Philosophical Transactions of the Royal Society B: Biological Sciences* **363** (2008), no. 1499 1939–1949.

[48] T. Wynn, *Archaeology and cognitive evolution*, *Behavioral and brain sciences* **25** (2002), no. 3 389–402.

[49] D. Stout and T. Chaminade, *Stone tools, language and the brain in human evolution*, *Philosophical Transactions of the Royal Society B: Biological Sciences* **367** (2012), no. 1585 75–87.

[50] C. Tennie, J. Call, and M. Tomasello, *Ratcheting up the ratchet: on the evolution of cumulative culture*, *Philosophical Transactions of the Royal Society B: Biological Sciences* **364** (2009), no. 1528 2405–2415.

[51] S. H. Ambrose, *Paleolithic technology and human evolution*, *Science* **291** (2001), no. 5509 1748–1753.

[52] J. J. Gibson, *The theory of affordances*, *Hilldale, USA* **1** (1977), no. 2 67–82.

[53] A. Stoytchev, *Behavior-grounded representation of tool affordances*, in *Proceedings of the 2005 ieee international conference on robotics and automation*, pp. 3060–3065, IEEE, 2005.

[54] J. Sinapov and A. Stoytchev, *Learning and generalization of behavior-grounded tool affordances*, in *2007 IEEE 6th International Conference on Development and Learning*, pp. 19–24, IEEE, 2007.

[55] R. Jain and T. Inamura, *Learning of tool affordances for autonomous tool manipulation*, in *2011 IEEE/SICE international symposium on system integration (SII)*, pp. 814–819, IEEE, 2011.

[56] A. Gonçalves, J. Abrantes, G. Saponaro, L. Jamone, and A. Bernardino, *Learning intermediate object affordances: Towards the development of a tool concept*, in *4th international conference on development and learning and on epigenetic robotics*, pp. 482–488, IEEE, 2014.

[57] A. Gonçalves, G. Saponaro, L. Jamone, and A. Bernardino, *Learning visual affordances of objects and tools through autonomous robot exploration*, in *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 128–133, IEEE, 2014.

[58] N. Saito, K. Kim, S. Murata, T. Ogata, and S. Sugano, *Tool-use model considering tool selection by a robot using deep learning*, in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pp. 270–276, IEEE, 2018.

[59] A. Z. Ren, B. Govil, T.-Y. Yang, K. R. Narasimhan, and A. Majumdar, *Leveraging language for accelerated learning of tool manipulation*, in *Conference on Robot Learning*, pp. 1531–1541, PMLR, 2023.

[60] J. Brawer, M. Qin, and B. Scassellati, *A causal approach to tool affordance learning*, in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 8394–8399, IEEE, 2020.

[61] K. Khetarpal, Z. Ahmed, G. Comanici, D. Abel, and D. Precup, *What can i do here? a theory of affordances in reinforcement learning*, in *International Conference on Machine Learning*, pp. 5243–5253, PMLR, 2020.

[62] Y.-C. Liao, K. Todi, A. Acharya, A. Keurulainen, A. Howes, and A. Oulasvirta, *Rediscovering affordance: A reinforcement learning perspective*, in *Proceedings of the 2022 CHI conference on human factors in computing systems*, pp. 1–15, 2022.

[63] M. Chevalier-Boisvert, D. Bahdanau, S. Lahlou, L. Willems, C. Saharia, T. H. Nguyen, and Y. Bengio, *Babyai: A platform to study the sample efficiency of grounded language learning*, *arXiv preprint arXiv:1810.08272* (2018).

[64] M. Chevalier-Boisvert, B. Dai, M. Towers, R. Perez-Vicente, L. Willems, S. Lahlou, S. Pal, P. S. Castro, and J. Terry, *Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks*, *Advances in Neural Information Processing Systems* **36** (2023) 73383–73394.

[65] Z. Liu, S. Tian, M. Guo, C. K. Liu, and J. Wu, *Learning to design and use tools for robotic manipulation*, *arXiv preprint arXiv:2311.00754* (2023).

[66] W. D. Johnston, *The evolution of tools and implements*, *The American Midland Naturalist* **8** (1922), no. 2 49–60.

[67] C. Healey, *Maring classification of cutting tools*, *The Journal of the Polynesian Society* **87** (1978), no. 3 215–229.

[68] W. H. Oswalt, *An Anthropological Analysis of Food-Getting Technology*. Wiley, 1976.

[69] M. Collard, B. Buchanan, M. J. O'Brien, and J. Scholnick, *Risk, mobility or population size? drivers of technological richness among contact-period western north american hunter–gatherers*, *Philosophical Transactions of the Royal Society B: Biological Sciences* **368** (2013), no. 1630 20120412.

[70] J. Sinapov and A. Stoytchev, *Detecting the functional similarities between tools using a hierarchical representation of outcomes*, in *2008 7th IEEE International Conference on Development and Learning*, pp. 91–96, IEEE, 2008.

[71] D. E. Berlyne, *Conflict, arousal, and curiosity.*, .

[72] J. Schmidhuber, *A possibility for implementing curiosity and boredom in model-building neural controllers*, in *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pp. 222–227, 1991.

[73] C. Zhou, T. Machado, and C. Harteveld, *Cautious curiosity: A novel approach to a human-like gameplay agent*, in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 19, pp. 370–379, 2023.

[74] E. Mikhaylova and I. Makarov, *Curiosity-driven exploration in vizdoom*, in *2022 IEEE 20th Jubilee International Symposium on Intelligent Systems and Informatics (SISY)*, pp. 000065–000070, IEEE, 2022.

[75] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos, *Unifying count-based exploration and intrinsic motivation*, *Advances in neural information processing systems* **29** (2016).

[76] H. Tang, R. Houthooft, D. Foote, A. Stooke, O. Xi Chen, Y. Duan, J. Schulman, F. DeTurck, and P. Abbeel, *# exploration: A study of count-based exploration for deep reinforcement learning*, *Advances in neural information processing systems* **30** (2017).

[77] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, *Exploration by random network distillation*, *arXiv preprint arXiv:1810.12894* (2018).

[78] H. Liu and P. Abbeel, *Behavior from the void: Unsupervised active pre-training*, *Advances in Neural Information Processing Systems* **34** (2021) 18459–18473.

[79] Y. Seo, L. Chen, J. Shin, H. Lee, P. Abbeel, and K. Lee, *State entropy maximization with random encoders for efficient exploration*, in *International Conference on Machine Learning*, pp. 9443–9454, PMLR, 2021.

[80] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto, *Reinforcement learning with prototypical representations*, in *International Conference on Machine Learning*, pp. 11920–11931, PMLR, 2021.

[81] R. Y. Tao, V. François-Lavet, and J. Pineau, *Novelty search in representational space for sample efficient exploration*, *Advances in Neural Information Processing Systems* **33** (2020) 8114–8126.

[82] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, *Intrinsic motivation systems for autonomous mental development*, *IEEE transactions on evolutionary computation* **11** (2007), no. 2 265–286.

[83] S. Forestier and P.-Y. Oudeyer, *A unified model of speech and tool use early development*, in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 39, 2017.

[84] A. Baranes and P.-Y. Oudeyer, *Active learning of inverse models with intrinsically motivated goal exploration in robots*, *Robotics and Autonomous Systems* **61** (2013), no. 1 49–73.

[85] V. G. Santucci, G. Baldassarre, and M. Mirolli, *Grail: a goal-discovering robotic architecture for intrinsically-motivated learning*, *IEEE Transactions on Cognitive and Developmental Systems* **8** (2016), no. 3 214–231.

[86] V. G. Santucci, D. Montella, and G. Baldassarre, *C-grail: Autonomous reinforcement learning of multiple and context-dependent goals*, *IEEE Transactions on Cognitive and Developmental Systems* **15** (2022), no. 1 210–222.

[87] A. Romero, G. Baldassarre, R. J. Duro, and V. G. Santucci, *H-grail: A robotic motivational architecture to tackle open-ended learning challenges*, *IEEE Transactions on Cognitive and Developmental Systems* (2025).

[88] K. Rakelly, A. Gupta, C. Florensa, and S. Levine, *Which mutual-information representation learning objectives are sufficient for control?*, *Advances in Neural Information Processing Systems* **34** (2021) 26345–26357.

[89] K. Gregor, D. J. Rezende, and D. Wierstra, *Variational intrinsic control*, *arXiv preprint arXiv:1611.07507* (2016).

[90] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine, *Diversity is all you need: Learning skills without a reward function*, *arXiv preprint arXiv:1802.06070* (2018).

[91] A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman, *Dynamics-aware unsupervised discovery of skills*, *arXiv preprint arXiv:1907.01657* (2019).

[92] J. Achiam, H. Edwards, D. Amodei, and P. Abbeel, *Variational option discovery algorithms*, *arXiv preprint arXiv:1807.10299* (2018).

[93] J. Zhang, H. Yu, and W. Xu, *Hierarchical reinforcement learning by discovering intrinsic options*, *arXiv preprint arXiv:2101.06521* (2021).

[94] H. Kim, J. Kim, Y. Jeong, S. Levine, and H. O. Song, *Emi: Exploration with mutual information*, *arXiv preprint arXiv:1810.01176* (2018).

[95] T. Jung, D. Polani, and P. Stone, *Empowerment for continuous agent—environment systems*, *Adaptive Behavior* **19** (2011), no. 1 16–39.

[96] C. Salge, C. Glackin, and D. Polani, *Approximation of empowerment in the continuous domain*, *Advances in Complex Systems* **16** (2013), no. 02n03 1250079.

[97] C. Salge, C. Glackin, and D. Polani, *Empowerment–an introduction*, *Guided Self-Organization: Inception* (2014) 67–114.

[98] J. Choi, A. Sharma, H. Lee, S. Levine, and S. S. Gu, *Variational empowerment as representation learning for goal-based reinforcement learning*, *arXiv preprint arXiv:2106.01404* (2021).

[99] A. Dahmani, A. Lidayan, and A. Gopnik, *Empowerment and causal learning*, in *Intrinsically-Motivated and Open-Ended Learning Workshop@ NeurIPS2024*.

[100] H. Cao, F. Feng, M. Fang, S. Dong, T. Yang, J. Huo, and Y. Gao, *Towards empowerment gain through causal structure learning in model-based reinforcement learning*, in *The Thirteenth International Conference on Learning Representations*, 2025.

[101] Y. Du, S. Tiomkin, E. Kiciman, D. Polani, P. Abbeel, and A. Dragan, *Ave: Assistance via empowerment*, *Advances in Neural Information Processing Systems* **33** (2020) 4560–4571.

[102] V. Myers, E. Ellis, S. Levine, B. Eysenbach, and A. Dragan, *Learning to assist humans without inferring rewards*, *Advances in Neural Information Processing Systems* **37** (2024) 71540–71567.

[103] F. Massari, M. Biehl, L. Meeden, and R. Kanai, *Experimental evidence that empowerment may drive exploration in sparse-reward environments*, in *2021 IEEE International Conference on Development and Learning (ICDL)*, pp. 1–6, IEEE, 2021.

[104] A. Levy, S. Rammohan, A. Allievi, S. Niekum, and G. Konidaris, *Hierarchical empowerment: Towards tractable empowerment-based skill learning*, *arXiv preprint arXiv:2307.02728* (2023).

[105] S. Dai, W. Xu, A. Hofmann, and B. Williams, *An empowerment-based solution to robotic manipulation tasks with sparse rewards*, *Autonomous Robots* **47** (2023), no. 5 617–633.

[106] H. Cao, F. Feng, J. Huo, and Y. Gao, *Causal action empowerment for efficient reinforcement learning in embodied agents*, *Science China Information Sciences* **68** (2025), no. 5 150201.

[107] T. van der Heiden, C. Salge, E. Gavves, and H. van Hoof, *Robust multi-agent reinforcement learning with social empowerment for coordination and communication*, *arXiv preprint arXiv:2012.08255* (2020).

[108] T. Van Der Heiden, F. Mirus, and H. Van Hoof, *Social navigation with human empowerment driven deep reinforcement learning*, in *International Conference on Artificial Neural Networks*, pp. 395–407, Springer, 2020.

[109] A. Latyshev and A. Panov, *Skill learning with empowerment in reinforcement learning*, *Pattern Recognition and Image Analysis* **34** (2024), no. 3 535–542.

[110] A. Lidayan, Y. Du, E. Kosoy, M. Rufova, P. Abbeel, and A. Gopnik, *Intrinsically-motivated humans and agents in open-world exploration*, *arXiv preprint arXiv:2503.23631* (2025).

[111] K. Seepanomwan, V. G. Santucci, and G. Baldassarre, *Intrinsically motivated discovered outcomes boost user's goals achievement in a humanoid robot*, in *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pp. 178–183, IEEE, 2017.

[112] K. Seepanomwan, D. Caligiore, K. J. O'Regan, and G. Baldassarre, *Intrinsic motivations and planning to explain tool-use development: A study with a simulated robot model*, *IEEE Transactions on Cognitive and Developmental Systems* **14** (2020), no. 1 75–89.

[113] S. Forestier, R. Portelas, Y. Mollard, and P.-Y. Oudeyer, *Intrinsically motivated goal exploration processes with automatic curriculum learning*, *Journal of Machine Learning Research* **23** (2022), no. 152 1–41.

[114] R. S. Sutton, D. Precup, and S. Singh, *Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning*, *Artificial intelligence* **112** (1999), no. 1-2 181–211.

[115] T. G. Dietterich, *Hierarchical reinforcement learning with the maxq value function decomposition*, *Journal of artificial intelligence research* **13** (2000) 227–303.

[116] P.-L. Bacon, J. Harb, and D. Precup, *The option-critic architecture*, in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, 2017.

[117] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu, *Feudal networks for hierarchical reinforcement learning*, in *International conference on machine learning*, pp. 3540–3549, PMLR, 2017.

[118] O. Nachum, S. S. Gu, H. Lee, and S. Levine, *Data-efficient hierarchical reinforcement learning*, *Advances in neural information processing systems* **31** (2018).

[119] R. J. Williams, *Simple statistical gradient-following algorithms for connectionist reinforcement learning*, *Machine learning* **8** (1992), no. 3 229–256.

[120] V. Konda and J. Tsitsiklis, *Actor-critic algorithms*, *Advances in neural information processing systems* **12** (1999).

[121] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal policy optimization algorithms*, *arXiv preprint arXiv:1707.06347* (2017).

[122] M. L. Puterman, *Markov decision processes*, *Handbooks in operations research and management science* **2** (1990) 331–434.

[123] L. Graesser and W. L. Keng, *Foundations of deep reinforcement learning: theory and practice in Python*. Addison-Wesley Professional, 2019.

[124] R. S. Sutton, *Learning to predict by the methods of temporal differences*, *Machine learning* **3** (1988), no. 1 9–44.

[125] K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman, *Quantifying generalization in reinforcement learning*, in *International conference on machine learning*, pp. 1282–1289, PMLR, 2019.

[126] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, *High-dimensional continuous control using generalized advantage estimation*, *arXiv preprint arXiv:1506.02438* (2015).

[127] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. Gonzalez, M. Jordan, and I. Stoica, *Rllib: Abstractions for distributed reinforcement learning*, in *International conference on machine learning*, pp. 3053–3062, PMLR, 2018.

[128] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, *Stable-baselines3: Reliable reinforcement learning implementations*, *Journal of machine learning research* **22** (2021), no. 268 1–8.

[129] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, *Asynchronous methods for deep reinforcement learning*, in *International conference on machine learning*, pp. 1928–1937, PmLR, 2016.

[130] T. M. Cover and J. A. Thomas, *Information theory and statistics*, *Elements of information theory* **1** (1991), no. 1 279–335.

[131] C. E. Shannon, *A mathematical theory of communication*, *The Bell system technical journal* **27** (1948), no. 3 379–423.

[132] A. S. Klyubin, D. Polani, and C. L. Nehaniv, *Organization of the information flow in the perception-action loop of evolved agents*, in *Proceedings. 2004 NASA/DoD Conference on Evolvable Hardware, 2004.*, pp. 177–180, IEEE, 2004.

[133] J. Pearl, *From bayesian networks to causal networks*, in *Mathematical models for handling partial knowledge in artificial intelligence*, pp. 157–182. Springer, 1995.

[134] R. Blahut, *Computation of channel capacity and rate-distortion functions*, *IEEE transactions on Information Theory* **18** (1972), no. 4 460–473.

[135] S. Arimoto, *An algorithm for computing the capacity of arbitrary discrete memoryless channels*, *IEEE Transactions on Information Theory* **18** (1972), no. 1 14–20.

[136] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, *Mutual information neural estimation*, in *International conference on machine learning*, pp. 531–540, PMLR, 2018.

[137] M. Samvelyan, R. Kirk, V. Kurin, J. Parker-Holder, M. Jiang, E. Hambro, F. Petroni, H. Küttler, E. Grefenstette, and T. Rocktäschel, *Minihack the planet: A sandbox for open-ended reinforcement learning research*, *arXiv preprint arXiv:2109.13202* (2021).

[138] H. Küttler, N. Nardelli, A. Miller, R. Raileanu, M. Selvatici, E. Grefenstette, and T. Rocktäschel, *The nethack learning environment*, *Advances in Neural Information Processing Systems* **33** (2020) 7671–7684.

[139] N. C. Volpi and D. Polani, *Goal-directed empowerment: combining intrinsic motivation and task-oriented behavior*, *IEEE Transactions on Cognitive and Developmental Systems* **15** (2020), no. 2 361–372.

[140] M. Karl, P. Becker-Ehmck, M. Soelch, D. Benbouzid, P. van der Smagt, and J. Bayer, *Unsupervised real-time control through variational empowerment*, in *The International Symposium of Robotics Research*, pp. 158–173, Springer, 2019.

[141] C. Salge, C. Guckelsberger, R. Canaan, and T. Mahlmann, *Accelerating empowerment computation with uct tree search*, in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8, IEEE, 2018.

[142] S. Tiomkin, I. Nemenman, D. Polani, and N. Tishby, *Intrinsic motivation in dynamical control systems*, *PRX Life* **2** (2024), no. 3 033009.

[143] N. C. Volpi, D. De Palma, D. Polani, and G. Indiveri, *Computation of empowerment for an autonomous underwater vehicle*, *IFAC-PapersOnLine* **49** (2016), no. 15 81–87.