Cooperative Intent

An Exploration of Computational Learning in a Discrete Preference Space

by

Simon Charles Stanton

BSc Hons (Comp Sci)

School of Information & Communication Technology

College of Sciences and Engineering

Submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

University of Tasmania

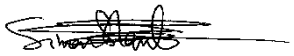
December 2023

for

CMG

Declaration of Originality

This thesis contains no material which has been accepted for a degree or diploma by the University or any other institution, except by way of background information and duly acknowledged in the thesis, and to the best of my knowledge and belief no material previously published or written by another person except where due acknowledgement is made in the text of the thesis, nor does the thesis contain any material that infringes copyright.

Signed: 

Date: 14/12/2023

Statement of Authority of Access

This thesis may be made available for loan and limited copying and communication in accordance with the Copyright Act 1968.

Signed: 

Date: 14/12/2023

Publications Related to this Research

Chapter Five of this thesis, *Representational Equivalence,* includes material that has been published in the paper *Representation-Induced Algorithmic Bias* (Stanton, Dermoudy & Ollington, 2022). Chapter Five is an expansion on the short paper and contains new text, figures, and analysis.

Declaration of Code and Data Availability

This thesis includes code and data from three sets of experiments, referred to in the thesis as Experiment Series One, Two, and Three. The code for each experiment series is available as a release on GitHub. Each release is identified by a DOI and is published under an MIT licence. Code and data availability for each experiment series is listed in Appendix B.5.

This research was supported by
an Australian Government Research Training Program (RTP) Scholarship.

Acknowledgements

There are many people who have contributed to this thesis over the last number of years, and to all of them I extend my thanks and appreciation. Each and every one of you have influenced my progress and helped to shape this thesis into what it is now.

A big special thanks goes to my supervisors, Robert Ollington and Julian Dermoudy. Their ongoing support and guidance over my candidature has been fundamental. It has been a long road and I thank them for their unfailing enthusiasm, sage advice, analytical mindset, and critical eye. A special note of thanks also goes to Vishv Malhotra. From being my Honours supervisor all those years ago, through to providing an academic reference for my candidacy, I am most grateful.

Over the time of my candidature I have been fortunate to receive assistance from many people in the school and also in the wider university community; my thanks goes to them all, but particularly so to the late Kris Purton, for always being ready to lend a hand to sort things out, and for always being you—you are very much missed; to both Nicole and David Herbert, for their advice and willingness to share their time and hard-won knowledge; to Quan Bai for motivating me to think about the narrative; to James Montgomery for his thoughtful consideration on those occasions when obligations clashed; to my lab-mates, especially Louise and Lachlan, for their camaraderie and levity. A big thank you to Elaine Stratford for helping me with my writing. To Karl Goetz, Geli Kourakis, and John Miezitis, for their helpful and effective onboarding to the cluster at Digital Research Services, and likewise, a big thank you to IT Services. A massive thanks to the Research Librarians at Library Services, who helped me source books, articles, and documents (old and new) from dusty stacks and far-flung repositories around the world … thank you *so much*. Many thanks to Lisa Maclean for her advice and assistance, it has been invaluable; likewise, to Paula Andrea Martinez, thank you for your help, at just the right time. To Claire d'Este, who catalysed my return to study in the first place, this thesis really would not have happened without your generosity of spirit so thank you.

For their encouragement and support over this time my thanks and gratitude goes to my friends and neighbours in this little bit of rural Tasmania, particularly Bob and Marie Scales, Greg and Evonne Bender, Sarah and Bonnie, and Andrew and his family; you have all been terrific. To Dr Luz Montes, a very big thank you for everything over the years.

And to my own family, particularly John and Karen, and Al, Merran, Thea and Angus, thanks for being there, for the support you have provided, and for the interest you have shown. It has been wonderful to be able to share it with you.

I would also like to acknowledge the authors, maintainers, and developers of the software and services I have used in creating this thesis, most particularly *Zotero* (Corporation for Digital Scholarship, 2018), *diagrams.net/draw.io* (JGraph, 2021), *Photopea* (Kutskir, 2013), *Caffeine* (Revell, 2019), *Notepad++* (Ho, 2003), *R* and *R Studio* (R Core Team, 2023), the *R* package *ggrain* (Allen et al., 2021), and *Zenodo* (Nowak et al., 2016).

Table of Contents

List of Tables

List of Figures

List of Algorithms

Abstract

This thesis is an exploration of cooperation, game theory, and machine learning. The aim is to develop a method for profiling *cooperative intent* as found in the behaviour of learning algorithms over highly constrained games of cooperation; in order to step towards the future use of *cooperation-as-policy* in the reinforcement learning domain, and contribute to the theory of grounding emergent software systems.

The fundamental question explored is whether an agent can construct an online model—derived from a strictly ordinal $\mathbf{2} \times \mathbf{2}$ preference space—of its behaviour, to bridge numerical reward with the symbolism inherent to descriptive game theory models. Identification of a game model has utility in assessing the current state of a system, the current state of an agent, and, to assessing an agents' *cooperative intent*, i.e., the cooperatively-contextualised reification of an agent's internal state representation, or *state-of-mind congruent to action*. In doing this, the benefits and risks inherent in the anthropomorphisation of machine systems are placed within a mitigatory scope: defining intent, ultimately, as the reification of a trajectory *between states* (a state being found by a mapping from the agent's behaviour to any one of the game models in a strictly ordinal preference space) that an agent may experience, which leads to the hypothesis that in nominating target states, and rewarding agents for reaching those targets, the application (via reflection) of cooperation as a policy instrument in machine learning algorithms may deliver beneficial outcomes to the agent and to the system as a whole.

This thesis develops through three sets of experiments that progressively undergo constraint devolution such that the rules and bounds of *imperfect-* and *incomplete-information* are (in-part) relaxed, while remaining based in Markov formalism. The initial set of experiments conducts a multi-model tournament to calculate a *mutual cooperation rate* metric on the performance of algorithms drawn from the game theory, bandit, and foundational reinforcement learning literature. The second set of experiments examines variance in the behaviour of agents under specific conditions of change and confirms that foundational reinforcement learning algorithms are sensitive to isomorphic changes in input; a finding that guides the third set of experiments with the development of a method for reflective-mapping, i.e., the identification of game models in sequential $\mathbf{2} \times \mathbf{2}$ matrix games; where agents are constrained by *perfect-* and *incomplete-information* visibility. For single agents, this method maps into the strictly ordinal preference space with a one-in-twelve resolution of the correct game model. An external observer, with the relaxed constraint of *complete-information*, offers singleton mapping into the strictly ordinal space. A singleton mapping is the recognition of an agent's actions corresponding to the strategic dynamics of the game model constituting the environment.

These experiments are conducted in the context of concepts of cooperation, as have been developed in biological science, and their use in computational learning. This thesis attempts to place machine cooperation as a cooperative mode that is at once like, and unlike, biological cooperation, framed as the latter is by theories of evolution to which machines need not abide. Therefore, constraints that apply to biological modes of cooperation may not apply in an algorithmic context, as machine behaviour can manifest in unexpected, and seemingly irrational, ways. To gain traction on these problems, a descriptivist approach to game theory is pursued in an ethological manner, in order to empirically assess agent behaviour with respect to cooperative dynamics.

Chapter One

Introduction

A few more moves having been made, no step is certain.

—E. A. Poe[1]

As artificial intelligence (AI) becomes ever more capable and widespread there is clearly growing societal interest and concern about how we and our world will be impacted by a technology that increasingly exhibits apparent emergent agentic autonomy. Much research and development is directed into AI such that investment, both governmental and corporate, has been growing rapidly (Littman et al., 2021)[2]. Notwithstanding this huge growth, the long-term ramifications of AI on society are not clear. Not least of the concerns in this space is the question of whether we will be able to interact with future AI *safely*; indeed, this question is now becoming an issue with currency rather than remaining solely as a theoretical, and distant, possibility.

In response to the broad question of how autonomous agents will integrate into our daily lives this thesis is situated as basic research into understanding the behaviour of emergent software systems. The objective is to develop and evaluate a method for identifying *cooperative intent* from observation of the behaviour of learning algorithms, over highly constrained game theory models.

Within the field of AI Ethics, a prescriptive, or normative, approach promotes the aim of instilling values into machines so that they will be aware of, and be able to reason about, ethical issues (Winfield et al., 2019). A normative approach seeks to prescribe a method of behavioural control onto an autonomous agent (Criado et al., 2011); which, although a practical engineering approach, by its very nature gives rise to entirely valid questions of *whose* norms and *whose* ethics should be instilled, given that people have differing ethical systems (Moor, 2011).

Here, the approach is descriptivist. This work endeavours to be an assistive tool *for* a *future* normative control, by way of developing a method to measure the cooperativeness of an agent with a given externality. To achieve this the domain is reduced to an abstraction in a game theory context, to isolate the core requirement of equipping an agent with the ability to reflect on its own behaviour, and so identify a mapping between that

[1] Edgar Allen Poe, attributed to by Tresch, J. in *The Reason for the Darkness of the Night: Edgar Allan Poe and the Forging of American Science* (2021, p. 90).

[2] Growth in AI research and development is attracting ever more investment, particularly so as industry overtakes academia as the driving force behind AI; respectively, thirty-two to just three notable machine learning models were released by the two sectors in 2022 (Maslej et al., 2023).

behaviour and a *cooperative state*; i.e., a strictly ordinal $\mathbf{2} \times \mathbf{2}$ game model obtained from a discrete preference space derived from work in topological game theory by Robinson and Goforth (2005). Mappings of cooperative dynamics may be of use in techniques for enabling a normative *cooperation-as-policy* paradigm in autonomous agents and learning algorithms, and more generally as a contributory mechanism for the grounding (Searle, 1980; Thill et al., 2014) of emergent software systems.

To achieve this outcome, this research attempts to weave the threads of cooperation, computational game theory, and computational learning together. The remainder of this chapter goes further to the motivation for this research and then outlines the context, scope, and constraints that frame the theoretical basis for the experimental domain.

1.1 AI Now: Bias, Risk and Aspiration

The impact of AI on society is deep and far-reaching. AI has become a part of not just research and academia but very much part of our daily lives, from autonomous vehicles (Fisac et al., 2018; Gonzalez, W., 2022) to robot vacuum cleaners (Stone et al., 2016); data-centre power control systems (Luo et al., 2022), to media production (Herndon, 2021). In recent years, the presence of AI across the cultural landscape has become increasingly pervasive, in no small part due to the Transformer (Vaswani et al., 2017) architecture and subsequent services such as ChatGPT (OpenAI, 2022) and Stable Diffusion (Stability AI, 2022). AI permeates, influences, and increasingly mediates our interactions with others, with organisations, and with machines. Adoption has been rapid, and is often not transparent (Stone et al., 2016; Littman et al., 2021).

Measuring the value of AI in our lives is difficult, and likely to become an ever more intractable undertaking given not just the immense complexity of the interactions of these systems, on society, but also the inherent complexity of the systems themselves. To this we can add the underlying component agents, which have varying degrees of autonomy and are able to exhibit emergent behaviour (Kubí, 2003). For example, individual agents can exhibit behaviour that reveals zero-sum—i.e., self-interested—motivation. Emergent behaviour can result in conflict between the agents themselves (Wagner et al., 1999), and can also result in unpredictable behaviour for the systems they are part of or with which they have articulation or effect (Mossalam et al., 2016). This unpredictability, given articulation with hardware systems as well as other software, can be surprising, even comical: for example, the Roomba "Circle Dance" (Sung et al., 2007, p. 157). However, this unpredictability can also lead to potentially serious outcomes. This is indicated not only by the growth in consideration of the safety of self-driving vehicles (Tafidis et al., 2022) but also by the apparent emergent behaviour exhibited by large language models (LLMs), characterised by some as sentience[3]. However, LLMs can produce extreme and hateful output (Maslej et al., 2023) due to cultural, racial, and prejudicial bias that is evident in the datasets of multi-modal generative AI (Birhane et al., 2023).

Wei et al. (2022) evaluate a range of LLMs including LaMDA, PaLM, and GPT–3 and consider an ability to be emergent if, by scaling the model up along one of the axes of "training compute, model parameters, and dataset size" (2022, p. 8), an ability is exhibited

[3] See Marcus (2022a) for discussion of claims of emergence, and interestingly, his later conversation with Blake Lemoine (Marcus, 2022b).

that was not present in the model at a smaller scale. Although Wei et al. (2022) recognised that scaling induces a phase transition in the capabilities of the model, enabling capabilities not seen in the model at smaller scale, they posit that abilities so exposed by scaling may reveal techniques that can be used in smaller-scale versions of the same model. Wei and colleagues also note that this brings *risk* by virtue of the unpredictability of emergent abilities.

Cave et al. (2019b) write on the risks of ethics in AI and identify several broad classes of possible harm, from the risk of an AI reaching a "morally unacceptable" ethical position (2019b, p. 569), to the difficulty of a machine finding a solution to an ethical issue in a pluralist setting (such that a partial ordering of the ethical responses, or actions, available to a machine is not resolvable to a solution), and further, that attributing agency to machines creates a new class of 'patient' to which moral agency must be ascribed, and lastly, the displacement of responsibility if ethical decisions are undertaken by autonomous agents rather than people. Issues brought up by the concept of a 'moral patient' have also been raised by Bryson (2018), who said that if we create intelligent machines to meet a moral standard, then we must also consider what *moral status* should be given to the machines themselves.

Benjamin writes that the datasets and algorithms that comprise models are not just artefacts that reflect our current society—with all of its cultural, racial, and economic biases intact—but are indicative of a systemic encoding of values that "reproduce and amplify existing prejudices" (2019, p. 50). Birhane et al. (2022) analysed encoded values in 100 high-impact machine learning papers and found that the vast majority (99%) did not consider or address the potential of negative outcomes arising from the work being presented. However, such negative outcomes are readily found. Patton et al. (2017) show bias in the analysis by US law-enforcement of social media channels; and the work by Angwin et al. (2016) on the predictive tool COMPAS, which exposed racial bias emerging from between-group-calibration issues in an algorithmic risk-assessment tool used across the US incarceration industry (Walsh et al., 2019). The COMPAS investigation brought the difficulty of working with protected attributes in datasets into sharp relief at the same time as showing actual downstream effects on already marginalised communities where the tool would "consistently overestimate the risk of Black defendants who didn't reoffend and underestimate the risk of White defendants who did reoffend" (Christian, 2020, p. 68). The COMPAS tool highlights Benjamin's (2019) point that systemic encoding of values reproduces systemic prejudice in new technologies; and, in doing so, is maintaining "persistent correlations between race and disadvantage" (Walsh et al., 2019, p. 177).

Hooker (2021, p. 3) has also written about the "diffusion of responsibility" by machine learning practitioners over negative externalities. Hooker writes that practitioners can constrain the scope of their concerns, and so evade responsibility for algorithmic bias. However, Hooker goes further than stating that bias is emergent from just the dataset being used—postulating that bias is also formulated through the algorithm itself in the numerous large and small decisions that practitioners make when developing their models. In previous work, Hooker et al. (2020) have shown that methods of algorithmic compression and pruning over datasets can lead to fairness issues in model output. Specifically, they found that bias is not distributed evenly across the dataset after pruning but tends to isolate, and penalise, subgroups—which in turn show higher levels of error

in output relative to the change in the model's overall accuracy. Work by Amodei et al. (2016) contributes to the analysis of engineering design in AI, examining several scenarios that can lead to misspecifications in design, for example the maximisation of an objective function ill-suited to the task at hand.

It is clear that AI can cement already present systemic societal issues. A normative ethical agent constructed with a ruleset amenable to one group is unlikely to be amenable to another group, let alone amenable to all members of any group, which may serve only to perpetuate existing biases and prejudices. As well, new dilemmas emerge: Adams (2023, p. 264) writes of the "diversity dilemma", that objectification of machines in anthropomorphic terms tends towards having racially 'white' representations, as portraying machines as other than white invites the association of "robots as slaves".

Popular conceptions in dominant cultural imaginings of AI are heavily influenced by an "anglophone Western portrayal" (Cave & Dihal, 2023, p. 5) despite the rich source of AI narratives that exist, and have existed, throughout human society and history (Cave & Dihal, 2023; Cave, Dihal & Dillon, 2020). For example, Polish writer Stanislaw Lem's rich oeuvre of science writing that leans into the mystical with his conception of "automatic gnosis" (Konior, 2023, p. 104); Soviet narratives involving intelligent machines date to the early 1920s (Pervushin, 2023); Latin American science fiction exploring how technology enables new forms of exploitation (Areco, 2023); and the story of a Chinese automaton dancer from the Warring States period, more than two thousand years ago, remains influential in contemporary Chinese science fiction (Wu, 2023).

Compared to contemporary Western fears of existential risk from AI (Armstrong et al., 2016; Bostrom, 2012; Yudkowsky, 2008), materially different attitudes are found in China (Song, 2021; 2023), South Korea (Kim, 2023), Singapore (Lee & Matthews, 2023), and Japan (Katsuno & White, 2023). In South Korea, a major governmental national strategy places AI as an engine-of-growth into the future, an optimistic outlook motivated by a desire to remain at the leading edge of technological development (Kim, 2023). In China and Japan, the development of AI and robotics is substantially influenced by *relationality*, positioning the integration of AI into society as a net positive, and aspirational, rather than being founded on fears of extinction as a product of the "advancement of cognition beyond human capacities" (Katsuno & White, 2023, p. 299) that so dominates Western discourse. Katsuno and White underscore Japanese priorities in AI development: "even more important in Japan than giving artificial cognition a body, however, is giving it heart." (2023, p. 295).

The relational view, that of AI as *companion*, of AI being "partners to humans" (Katsuno & White, 2023, p. 300), finds resonance in the tradition of *oskabewis*, or "helpers" in ritual Anishinaabe ceremony, where the helper assists the participants in a model of care, which Lewis (2023, p. 213) suggests as a model of "how we might want AI systems to support us".

However, many uses of AI have motivations that must give us pause. From surveillance capitalism (Zuboff, 2019) to the control of populations and state-controlled censorship (Cave et al., 2019c), to the militarisation of autonomous lethality (Galliott & Wyatt, 2020), transparency over AI systems is vital to informing international and national law and regulation (Narayanan & Kapoor, 2023). How to enforce adherence to normative expectations in a machine intelligence is an open question. To enable a degree of control over AI under a normative code suggests requiring at least three bases: sound

epistemological foundations; reliable measures of the technology; and a means to cease its operation. Further, the incorporation of a diversity of voices in the formulation of a generalised artificial intelligence is arguably necessary for global society to be able to trust, cooperate, and live with AI embodied and agentic in the world. Just as AI may in future be able to autonomously adapt, learn, and grow; *relationality* may allow an acceptance of AI as entities with Bryson's (2018) *moral agency* equipped.

1.1.1 Mechanism and Function

In the same vein that the technical decision processes of developers are mechanisms for inducing bias, algorithmic bias is also introduced into systems by innate factors in human cognition. In their work on how our beliefs can form subjectively (i.e., where uncertainty in data leads to intuitive probability estimation) Tversky and Kahneman (1974, p. 1124) argue that instances of bias—which they equate with "systematic errors"—are the result of heuristics that we apply in our decision-making processes. As an effect on algorithmic bias, innate factors are variable as well as pervasive.

A corollary to the separation of human and machine is found in the work of Herzing (2014) profiling species intelligence. Herzing enumerates the dimensions[4] along which the intelligence of other species have historically been measured and discusses their shortcomings, which are in part due to our inherent bias. Herzing (2014) describes a model for profiling *species intelligence* that places less emphasis on cognition as a signal marker for intelligence.

The identification of the value of appreciating multiple dimensions of intelligence is a theme in the *machine behaviour* work of Rahwan et al. (2019). Machine behaviour, as a field of study, has the aim to regard intelligent machines "not as engineering artefacts, but as a class of actors" (2019, p. 477). They argue for an interdisciplinary research agenda that adopts an ethological approach to ground the study of machine behaviour in Tinbergen's (1963) four dimensions for the study of animal behaviour: *causation*, *survival (or adaptive) value*, *ontogeny*, and *evolution*. Tinbergen wrote that the classification of these dimensions—the "four problems of Biology" (1963, p. 426)—is a pragmatic approach to descriptive work in the study of animal behaviour.

To take a similar pragmatic approach to the behaviour of machines necessitates the enumeration of the dimensions of interest and the consideration of what an observational methodology should aim to measure. In the context of machine behaviour Rahwan et al. (2019) map Tinbergen's four dimensions to *mechanism (causation)*, *function (survival*, or *adaptive*, *value)*, *phylogeny (evolution)*, and *development (ontogeny)*. Of these, *mechanism* and *function* are observations over the current behaviour of a machine, whereas *phylogeny* and *development* are views informed by historicity of development.

Key to this approach, Rahwan et al. stress that observations performed on performance maximising algorithms are "not optimal" to the goal of observing (and thereby *being descriptive* of) machine behaviour overall (2019, p. 479).

Therefore, to investigate recognition of cooperative intent the metrics of interest are not solely the optimal (highest reward, highest rate of mutual cooperation) but also

[4] Herzing's (2014) six dimensions of intelligence: physical measurement, observational and sensing measurements, data-analysis, experimentation, direct interfaces, and accidental interactions.

observation of metrics within an ethological frame of reference: as such, for *function*, the correlation (or not) of an agent's rate of mutual cooperation with that agent's reward (Chapter Four), and, the calculation of variance between aggregated distributions of agent behaviour (Chapter Five); and, for *mechanism*, the *time-to-achieve* of the act of recognition (Chapter Six). This perspective interprets an ethological frame on the dimension of function to be inclusive of *function of*; and, therefore, is a behaviour that can be observed and measured. The *mechanism* dimension is interpreted to be inclusive of process and algorithm, specification, and structure.

1.2 Autonomous Agents and Cooperation

Bias aside, we do not know whether we can necessarily trust agents[5] if we consider that, to some degree, individual agents are "self-interested and … unreliable" (Huynh et al., 2006, p. 120). Another aspect undermining trust is poor design, as this can lead to "unintended and harmful" outcomes (Amodei et al., 2016, p. 21). Soares et al. (2015) theorise that, in models where agents are designed to optimise reward, the agents may actively dissemble any external (i.e., human, or other machine) instruction to alter their current utility function(s) if they are incentivised by the existing utility function(s). In this case, the agent's utility function(s) may direct behaviour that the external actor wishes to attenuate or stop altogether. The end result of these processes would be that the agent can pursue its own goals, considered as an equivalence to *agency* in the terminology of superintelligence (Bostrom, 2012).

However, Bostrom (2012, p. 74) believes super-intelligent AI agents will be so complex they will have "no clear functional analogues" that we would be able to understand. The possibility of superintelligence that Bostrom, and others (Hawking, Tegmark & Russell, 2014; Omohundro, 2007; Totschnig, 2019; Minsky, 1961) describe attracts anthropomorphically-charged language about behaviour, for example terms and phrases like *runaway paperclip maximisers* (Bostrom, 2003) and *hallucinations* (Ji et al., 2023). Brooks (2017) takes a sanguine view of the capabilities and impact of AGI, as well as of the time it is likely to take to develop such technology. Marcus (2022c) generally concurs with this outlook but does not believe human-level AI will be achieved through deep-learning alone, calling for an increased focus on integrating symbolic AI into connectionist models. In reference to sentiment, findings of a survey by Cave et al. (2019a) highlight inequality in access to communication channels by diverse actors, and also a lack of clarity in the informational content in the media, as factors in undercutting awareness and lowering confidence toward all forms of AI, beneficial, or otherwise.

Omohundro's (2007; 2008) use of conceptual anthropomorphisms to build reasoning about agent motivation using human abstractions is in counterpoint to Bostrom's position of 'no analogues', unless we consider that although the agent's mechanism may be unknowable, the *behaviour remains visible*—even if reduced to a 'black-box' that allows knowledge over inputs and outputs only. Davis and Marcus (2021) emphasise that

[5] A distinction is drawn here between autonomous learning agents, aka adaptive agents, which have the ability to learn from experience in real-time, and pre-trained Large Language Models (LLMs). While it is clear we cannot blindly trust LLM output (Liu et al., 2023) it is also clear that we do not yet have a good handle on exactly what LLMs are: whether they be savants (Wei et al., 2023), able to hold Theory-of-Mind (Ullman, 2023), or essentially statistical parrots (Bender et al., 2021) is as yet, unclear.

whatever form(s) AI, and AGI, will take, they need not be predicated on how our own minds work.

Earlier, Turing (1950) wrote that anthropomorphisation is not just inescapable—it can also be *useful*, ideas he pursued over some years (Turing, 1948, 1950, 1951; Copeland et al., 2017); for example, while the processes that machines use to perform computation are very different from what we ourselves do in the act of thought, the term 'thinking' when applied by us, to a machine that is processing instructions and data, can serve a fundamentally beneficial role as a conceptual structure with which we can regard such processes, so that "anthropomorphisation [can] be managed" (Proudfoot, 2011, p. 952). Writing with respect to the role of abstraction in the field *intention recognition* (Han, 2013; also see §1.2.1.2), Heinze observes that the "abstract concepts" that have displayed empirical value when modelling agency are those that "match human behaviours" (2003, p. 2); this observation, from a related field to the work presented in this thesis (see §1.2.1.2), underlines the value anthropomorphisation can offer to modelling abstractions in autonomous agents.

A useful anthropomorphisation that can be applied to the behaviour of autonomous agents is to assess their *cooperativeness*. As a context for the framing of the anthropomorphisation, the concept of cooperation provides a meaningful referent in several ways. Firstly, the concept of cooperation is something tangible to most people, something that provides a basis for trust and for our sense of what is right (or wrong), as our feeling of comfort in a situation can be affected by our appreciation of whether the situation is in our interest, or not.

Secondly, cooperation has a significant research literature, across disciplines (Axelrod & Hamilton, 1981; Nowak, 2006; Raihani, 2021) and over time, going back to Darwin's "special difficulty" (1859, p. 209), which refers to his observations on the eusocial nature of the *Hymenoptera* order; a 'difficulty' now subsumed into evolutionary discourse as "the problem of altruism" (Herbers, 2009, p. 214). The multitude of research in the field of cooperation does more than signify great interest; it attests to the fundamental nature of cooperation not just to humanity, but to all living things.

Thirdly, the fundamentality of cooperation extends to the field of computer science: not least the commonly used abstractions in concurrency and messaging, but also, for example, in resource use, and in the optimisation of multi-agent systems. This fundamentality can also be seen in algorithms for adaptive learning—especially so when individual agent behaviour is coupled in some way to global (aka, system, or environmental) outcomes. These dynamics are therefore of core concern to this thesis, to the extent that it is the fundamentality of the concept of cooperation to computing that motivates this work to build out from cooperative behavioural dynamics via game theory to a measure of cooperative intent.

Exploration of cooperative landscapes in multi-agent systems reveals that the agents in such systems are commonly faced with the dichotomy of optimising agent-specific goals without necessarily having full knowledge of a mutable environment, nor do they necessarily know the goals, or state, of other agents. Given this lack of visibility, agents may then have no option but to optimise for themselves alone—at the expense of, and to the possible detriment of other agents, the system, and, if applicable (e.g., the system is enclosed within a wider context with which it has articulation) to the environment. This suggests that cooperation is a strategy that multi-agent systems may be able to take

advantage of, to improve outcomes overall. The canonical game-theory representation of an environment with dynamics that capture this situation is the $\mathbf{2} \times \mathbf{2}$ Prisoner's Dilemma (Rapoport, 1974; Colman, 1995; Binmore, 2007; Shubik, 1970), a mathematical model where a lack of communication hinders the ability of participating agents to knowingly cooperate, leading each agent to make their only rational choice, which in turn leads to a sub-optimal outcome for all involved. The Prisoner's Dilemma game model has been used extensively as a research tool across the sciences, particularly so in the biological sciences (Nowak & May, 1992; Yang et al., 2019; Masuda et al., 2009; Lewis & Dumbrell, 2013; Boyd & Lorberbaum, 1987) and in the social sciences (Paternotte & Grose, 2013; Kümmerli et al., 2007; Orbell & Dawes, 1993).

The use of game models in machine learning is also well established (Teófilo et al., 2012; Robertson & Watson, 2014; Crandall & Goodrich, 2011; Omidshafiei et al., 2020; Han et al., 2011b). The extensive use of game theory as a modelling device is very much predicated on the field having a fundamental relevance to not just social dynamics, but importantly, to dynamics in natural processes like predator-prey interactions, and evolution (Rubinstein, 1991). The mathematical relationships that game theory is built on do have intuitive analogues to the real world, but, as Rubinstein observes, these mappings do not have to be "isomorphic to reality" (1991, p. 910) *to be useful or accurate*. Rubinstein offers the view that the value of game theory models lies in their applicability to our own perception and understanding of the phenomenon at hand. Indeed, beyond the Prisoner's Dilemma many dynamics that we perceive in the world around us have been modelled and analysed using models and techniques from game-theory. For example, Skyrms (2001, 2003, 2016) explores the Stag Hunt game as an alternative model for cooperation and evolution, while Doebeli and Hauert (2005) evaluate the evolution of cooperation using the Snowdrift game in addition to the Prisoner's Dilemma.

Snowdrift, Prisoner's Dilemma, and Stag Hunt are examples of a class of games known as *social dilemmas*, a type of game where the participants have mixed motives (Macy & Flache, 2002). In an analytical setting of the single-stage (i.e., one-off, or single-shot) game, mixed-motive games can be solved by calculating solution concepts for Nash (1950a, 1951) equilibria. However, Macy and Flache (2002) observe that calculating solution concepts in a repeated game becomes intractable, particularly where the number of participants, or the cardinality of the action-space, becomes large.

The process of comparing specific game models has been generalised by examining the relations between different game models. This has the goal of forming a coherent *model of game models*. Early classification work by Rapoport and Guyer (1966) and Brams (1993) has led to the topological treatment by Robinson and Goforth (2005). The Robinson and Goforth topological space is the set of 144 strictly ordinal $\mathbf{2} \times \mathbf{2}$ matrices. Being ordinal, the payoff values of a game outcome indicate the preference that each participant has for that game outcome. The game models range from both participants obtaining their most-preferred outcome through to both participants obtaining their least-preferred outcome; in addition, the set of all game models in the topology spans across the participant's preferences in between those two extremes. The Robinson and Goforth topology is then a *discrete preference space* model.

Across the topology, each game model displays varying strategic dynamics, leading to the proposition that, as a numerical tool, and as a semantically-intuitive set of strategic

dynamics, the topology may be useful to humans and machines alike, albeit in different ways.

1.2.1 Cooperative Intent

Within the space of cooperation afforded by computational game theory models a further conceptual abstraction can be formed: that of *cooperative intent*. This abstraction is the mapping of the dynamics of an agent's behaviour with the strategic dynamics of a game model in the Robinson and Goforth topological space—a mapping that allows the agent's behaviour to be characterised as 'of a type', and therefore, *conforming* to the game model so mapped. The objective of this thesis is to formulate the identity function over game models in the Robinson and Goforth (2005) topology, using agent behaviour as an input, to obtain a recognition event indicating a specific game model's strategic dynamics have been observed in the behaviour of an agent, or agents. Occurrence of game model recognition is then interpreted as a state of *cooperative intent*, emergent to the agent's fine-grained, individual, actions.

How then, is *intent* being defined, and how does the qualifier *cooperative* limit the scope? The second of these questions is addressed in the adoption and use of a discrete preference space modelled over the strategic dynamics of the game models in the topology[6], to which agent behaviour will be mapped. Correspondence of observed agent behaviour to an element in the discrete preference space that exhibits a verisimilitude with respect to strategic dynamics, will be regarded as signifying a conception of the *cooperative intent* of the agent at a point in time. This is a representation of the internal state of the agent, manifested as (an observable) behaviour. Identifying an agent's internal state as corresponding to a behaviour, for any particular discrete temporal interval, reifies the machine at that point in time—if an agent has knowledge of its behaviour, represented by this measure, identifying the behaviour as a discrete state in a model, and continues with that behaviour, then the agent's *active* policy can be seen to have the intent of the *current* policy. This view upon the definition of *cooperative intent* removes the anthropomorphism of requiring causal motivation, i.e., a reason for one's actions. For a machine, its continuing behaviour is the result of the interplay of algorithmic computation, complexity, bias, and noise. Reifying a dynamic captured from the behaviour the machine exhibits at a point in time is then a measure of that behaviour, an indication to how the machine is behaving at that moment. Correspondence of the behaviour to an element in the Robinson and Goforth (2005) topology creates a discrete state representation of the behaviour in the form of the identified game model. The topology structure describes where a cooperative outcome is located in relation to other cooperative outcomes. For example, if the state maps to a game that describes a dynamic where both participants are obtaining their most preferred outcomes, an assertion can potentially be derived that an agent in this context is *fully cooperating* with the other; further, this can be generalised to an assertion that the agents are engaged in maximum, or optimal, cooperation. A highly reductive metric for assessing cooperative dynamics

[6] Strategic dynamics are modelled on the presence, and kind, of *equilibria*, in game models, see §2.3.1 and §2.3.3.4; and, for locations of multiple equilibria types over all game models in the Robinson and Goforth topological space, see **Table B.7**.

over time may then be obtainable by deriving the frequency of occurrence of any single state, collection of states, or temporal sequence of states, marking a trajectory.

To the first question, *intent* has a variety of meanings[7], two of which are key for the usage here: firstly, *an aim, will, or purpose with which one does an act*, and secondly, a legal sense, *the state of mind necessary for an act*, or more generally, the *state of mind congruent to an act*. The first suggests some element of causal motivation. In humans that might be thought to have been brought about by a rational thought process, or as a consequence of an emotion, or a feeling. In no way is the use of intent in this thesis suggesting that machines develop intent from such sources. Rather, intent is used to capture the machine's computational state at a point in time, as the legal sense would have it, 'the state of mind' to which a behaviour can be attributed. This definition sets *intent*, as used here, as an instance of the non-realist, *instrumentalist* form of *intentionality* (Cave et al., 2019b); where, Cave et al. explain, the realist view of intentionality attributes causality; and the instrumentalist form is any useful "ascription" for "explaining … behaviour", directed toward a machine (2019b, p. 265). While both of these forms actively place causality into the profiling of machine behaviour, identifying intent is primarily concerned with the current state, over time, of an agent; setting this use somewhat apart from an intention-based methodology (Shanahan, 2018). The realist form does not apply to intent as used here, apart from the observation that a state of cooperative intent is formed over a sequence of actions of mutable window length[8], so could well be used as a measure for interpreting trajectories of states of intent through a causal lens.

Consider that there is a small likelihood that a random sequence of actions could occur, that happen to match the pattern for activating one of the game models in the preference space. While this occurrence introduces a type one error, it also demonstrates that the pattern of a strategic dynamic does not necessarily have to occur as the result of *planning*. Here, planning encompasses activities we think of as agency, causality, purpose, and motivation, i.e., the realist form of intentionality.

The method for identifying cooperative intent here is focused on the occurrence of these states, only. It is then, simply, an identity function. Identity functions can be strung together in a history, and predictive paths can be generated in the formation of trees (i.e., directed acyclic graphs, as used in extensive-form game theory as well) over possible futures. For future work, both of these modalities could potentially allow the use of the method here as a measure of the cooperative dynamics in a system.

Clearly, the recognition event—being the observation that the behaviour of the observed entity is conforming to a dynamic as given by a strictly ordinal $\mathbf{2} \times \mathbf{2}$ game model—is *discrete*, so each location in the preference space is separated by an effectively unmapped continuous space, assumed, for reasons of parsimony, to have properties that allow the space between game models to be thought of in linear terms. Accepting this

[7] Definitions of intent, sources:
- American Heritage Dictionary of the English Language, (2023).
- Oxford English Dictionary, OED Online. Oxford University Press (2023).
- Criminal Code Act 1995 (Cth) s5.2 Intention. https://www.legislation.gov.au/
- Collins English Dictionary (2023).
- The Commonwealth Criminal Code: A Guide for Practitioners (2002)

[8] Here, 'mutable window length' refers to a bounded temporal interval i.e., over a given timespan. Chapter Six presents data from observation of the variable length of the interval for successful recognition of a game model.

supposition allows the sequence of recognition events to be modelled as instances of events. A sequence of these events is a trajectory over game models. Such trajectories may be useful in the profiling of cooperative behaviour, through change of the observed strategic dynamic(s), over time. A trajectory is then a path through the graph network of the game models in the Robinson and Goforth (2005) space; as such these paths could be investigated using computational methods for such structures (Bronstein et al., 2021; Rossi et al., 2020; Veličković et al., 2018). Brams' (1994) investigations into how and why games change are an example of working with trajectories through points in the ordinal space; Brams' trajectories were mapped into the Robinson and Goforth (2005) topology by Bruns (2012), in doing so clearly showing that equilibria in a game model can change, via a sequence of minimal changes to game models, resulting in a path. An example of such as change in a game is the path from Prisoner's Dilemma to the Revelation game (see §2.3.3.2 and §2.3.3.5).

1.2.1.1 Reflection

The software engineering design pattern *reflection* (Astrachan et al., 1998; Buschmann et al., 1996) consists of two phases, *observation*, and *change*. In the first phase the pattern undertakes the task of observing the state and behaviour of the entity to which it has visibility. Subsequently, the first phase may undertake further processing on the data so obtained. In the second phase of *reflection* the pattern undertakes to articulate with the mechanisms of which the entity is composed (that the reflection component has visibility, and at least some control, over), to effect change in those mechanisms either through structural means or by parametric change.

Heinze (2003) developed software design patterns (Gamma et al., 1995) for the implementation of *intention recognition* (see 1.2.1.2) in military simulation systems. These design patterns address both conceptual and practical issues from a software architecture point of view; an approach that is mirrored to a small extent in the consideration of *cooperative intent* using the *reflection* design pattern.

The principal output of this research, an algorithm for the observation and identification of cooperative intent in the behaviour of an agent, fulfils the first, interrogative, phase of the reflection design pattern and is presented in Chapter Six. The second phase, not part of this research, is to change the agent itself via a mechanism for modifying policy and structural configuration, using the metrics obtained from the first phase as inputs. Policy implementation is ordinarily achieved by manipulation of the agent's own state, based on some assessment; inputs to the assessment process are the state itself, potentially some history of the state, plus the incoming environment signal(s), such as reward. A model for the experimental evaluation of the second phase of reflection, using cooperative intent, is discussed as future work, in Chapter Seven.

1.2.1.2 Intention Recognition

Intention recognition (Charniak & Goldman, 1993; Armentano & Amandi, 2007; Han, 2013) is a field with a broad scope of methods and objectives, especially when compared to *cooperative intent*. A clear point of difference is that the latter would be entirely subsumed and integrated into an intention recognition model if the two were combined.

That is to say, cooperative intent could be integrated endogenously into agents with probabilistic (Geib & Goldman, 2009), predictive (Bui, 2003), logic-based (Sadri, 2011), or planning-focused (Sohrabi et al., 2016) intention recognition models; given this, cooperative intent could also be used to inform intention recognition's planning objective (Han & Pereira, 2013).

In addition, cooperative intent could be applied exogenously, to a whole multi-agent system, to observe the cooperative dynamics of the members of a set of intention recognition agents. Both of these options, that of integration of cooperative intent to an agent, or to use it exogenously, part of the system or environment, raise interesting questions and scenarios for any future work concerned with placing this research in an intention recognition context.

Intention recognition has been modelled quite abstractly in iterative game models, where agents seek to predict future opponent behaviour, using a Bayesian network for belief formation (Han et al., 2011a; Han, 2013; Han & Pereira, 2013). This work explored the role of intention recognition in processes of cooperation, particularly its evolution, through an evolutionary framework. Recent work has extended the investigation of intention recognition in iterated games with analysis of supervised learning models, including LSTM and Logistic Regression (Di Stefano et al., 2023).

A key difference of intention recognition, to the work in this thesis, is that with cooperative intent the mechanism itself has no scope to form models or create hypotheses; the mechanism's scope is a temporal observation window to a Markov environment in which game model recognition events can occur (see §2.2.2). An act of game model recognition is interpreted as the current cooperative state of the agent. The observation of this event is the sole mechanistic outcome of cooperative intent.

Where Heinze (2003) wrote that intention recognition is an abstract form, cooperative intent is an abstraction again, isolating, from the gamut of situations and application that intention recognition encompasses, properties that apply to the temporal dynamics of cooperation, and nothing else. Context for the scope of this research, that centres the observation of cooperative temporal dynamics, is now addressed, in §1.3.

1.3 Research Scope

The research in this thesis operates under constraints that delineate its scope. Some constraints and parameters vary over the series of experiments; divergences are discussed in-text where they occur.

The rationale for adopting the constraints presented here is to clearly demarcate the space in which the theory of cooperative intent is articulated. The principal constraints on all experiments are:

- On-line
  Agents compute in real-time in non-stationary environments in single-shot experiment instances.
- Imperfect- and incomplete-information (*state information visibility*)
  Agents receive a reward value only; they have no visibility into the environment, the game model, other agent's reward, nor other agent's strategy (policy).

- Infinite horizon
  $\boldsymbol{T} = \infty$. Agents are unaware of how long an episode will last.
- Agent / Environment boundary
  The only information crossing the boundary between an agent and the environment are reward values (game model payoffs) and an action signal.
- No 'talk'
  Any communication that occurs between agents is via their actions only, therefore emergent forms of signalling (Spence, 1974) are permissible. Any communication beyond this is excluded.

1.3.1 On-line

Agents operate *on-line*, i.e., they compute in real-time on data as it is received and make their action-choices per each timestep. Allowing that agents may add this received data to a computational structure, but not explicitly store full histories, aligns the experiments with the Markov property of the next state (and reward) being conditional only on the current state and (combined joint) actions (Ratitch & Precup, 2002; van der Wal, 1980).

In addition, the intention of this research is to direct the research towards autonomy. A non-stationary environment is an environment characterised by change. In a highly reductive game model environment change is essentially only observed in the shifts of policy that agents undergo. By virtue of having a learning policy, agent actions change in frequency relative to game outcomes, unless and until mutual convergence occurs which is complicated by the fact that the other agent's policy is also changing (Crandall et al., 2018b). Weinberg and Rosenschein (2004) examine provably convergent learning agents in game-theoretic non-stationary environments and find that if *all* agents use a provably-convergent learning algorithm, then *none* will actually converge. Their solution is presented as a best-response policy. Here, the issue is not so important, as the focus of this research is observing the *breadth* of agent behaviour as opposed to obtaining optimal performance.

1.3.2 State Information Visibility

In game theory the terms *perfect information* and *complete information* refer to two different concepts. Perfect information does not have an entirely strictly defined meaning, with conjecture in the literature over whether the term should include games of chance. The following definitions for *perfect*, *complete*, and *incomplete* forms of information are derived from a synthesis of Colman (1995) and Binmore (2007). Here, *perfect information* would mean that participants have visibility into the entire history of all participant's actions up to and including the current timestep, giving every participant a full history of all previous game outcomes (states). The term *complete information* refers to the visibility that participants have of the environment and over other participants. This includes payoffs from the game model to all participants. Where the totality of all this information may not be known the game becomes one of *incomplete information*.

The two terms can be exclusive to each other when describing any given environment, for example a game can afford *perfect* information (agents can see and recall all participant actions), but still be *incomplete* (participants do not know what payoff, or

reward, other participants receive). In this research explicit recall over entire histories of actions, by any participant, of *even their own action history*[9] is disallowed, so as to constrain all participants to a Markov formalism (discussed further in §2.2.2). Where an experiment instance relaxes this constraint (and so become non-Markovian) the occurrence is discussed in the accompanying text. It is permissible for agents to process their reward such that they create an internal representation that captures historical information about their action and reward histories. In these cases, the agents store that representation in a data structure such as a trace or Q-tables. Thus, participants are able to create a representation built from their experience of events, as they experience the events; but they are not allowed to retain the full state of the events in an explicit data structure.

The constraint of *incomplete information* does not preclude a participant from inferring another participant's payoff in the most recent timestep, nor what it may have been at a previous timestep. Likewise, the participants are not aware of the type (i.e., family of strategy, or learning algorithm) of participants with which they engage. They may infer this detail if they are constructed with the ability to model the conjecture. The constraint of *incomplete information* therefore precludes the state of *perfect information* from being accessible to any participant, except in the case of a game at a point in time where the number of elapsed timesteps is equal to one. This is because the participant's memory depth allows them to only recall the last timestep, but at $t = 1$ the last timestep also encompasses the entire history, which gives a (fleeting) state of *perfect information*. This constraint also holds by default in the case of agents that enforce a memory depth of zero, which is the case for deterministic algorithms that respond to an action from another participant in the most recent timestep in a fixed, i.e., deterministic, manner.

The constraints on *state information visibility* (*SIV*) presented here enable the construction of the scope for agent and environment interfaces. In addition, these constraints assist clarity in assessing the justification for an explicit relaxation, and furthermore they serve to contextualise findings in each of the experiments presented in Chapters Four, Five, and Six.

1.3.3 Infinite Horizon

From an agent's point of view, the timestep-length of an episode in any experiment instance is unknown, so that agents have no conception of the length of an episode, i.e., for an agent, the time horizon is infinite. This constraint is imposed because agents who are aware of an episode's length can then plan accordingly—when agents are aware they have a finite number of steps in which to interact they can adopt policies to take advantage of *backwards induction*. Backwards induction, formulated as the *Chain-Store Paradox* by Selten (1978), is well-documented in the game theory literature (Colman, 1998) as the ability to derive a course of action upon knowing how many turns are left in an iterated game. This knowledge introduces the possibility for further considerations in an agent's planning and confounds analysis. Further, incorporating backwards induction alters the assessment that an agent will make about the value of cooperating versus defecting. When

[9] This is implemented by configuring the memory depth of the algorithms to one timestep into the past (that is, *memory_depth=1*, see Appendix A.3).

an agent can calculate out to a predetermined horizon the agent is able to use defection as a dominant strategy, and this in turn biases the result (von Neumann and Morgenstern, 1953, p116; Colman, 1998). However, this concern is only relevant if the participants know how many more turns, or timesteps, are left in an episode; so, for this research all experiments are modelled without this information being available to the agents.

1.3.4 Agent / Environment Boundary

The boundary between agents and the environment is a result of applying *incomplete information* to the game model: the only signal supplied directly to an agent is the reward value obtained from the game model at each timestep. Strict application of this constraint does not allow for any other information to cross the boundary except for the action signal from agent to environment.

Relaxing this constraint would allow for a fusion between *incomplete information* and *partial perfect information* by also allowing agents to have visibility into other agent's actions. This relaxation is applied in this thesis for a small set of deterministic algorithms, the algorithm *fictitious play*, and three variants of a *supervised learning bandit*[10] algorithm; but only for one set of experiments that are presented in Chapter Three. All other experiments in this thesis enforce a strict interpretation of this constraint and so conform to a single-reward reinforcement learning computational paradigm. This constraint also forbids communication (in the form of 'talk') between agents, which is discussed next.

1.3.5 No 'Talk'

This constraint precludes any explicit communication between agents except that which is emergent from signalling (Spence, 1974), i.e., by way of their actions. This constraint therefore precludes communication such as *cheap-talk* (Farrell & Rabin, 1996), as well as any other message-passing, between agents or other externalities in the environment.

This research sets out to examine the behaviour of agents in environments via observing an agent's actions *only*. Adding explicit communication not only blurs the distinction between action and reward due to the credit-assignment problem (Minsky, 1961) but adds a layer of complexity in analysing agent motivation, as much cheap-talk is *manipulative* (Crandall et al., 2018a). Cheap-talk can be intimidatory, consist of bluff or bluster, amount to enticement or entrapment, or be straight-up coercion. Understanding what constitutes *cooperation* under these conditions becomes more difficult. Bringing about actions in one agent that conform to another agent's wishes through the use of manipulation is not cooperation in the *volitional* sense, but *coercion*, a circumstance where cooperation is demanded by one party, and given in acquiescence by the other.

A clear case of coercion without signalling is the game theory model commonly known as Ultimatum (Güth, Schmittberger & Schwarze, 1982). This is where one agent provides a choice that is not fair to the other, but the options range from accepting an unfair proposition and receiving a low-value outcome, versus the option of not accepting the

[10] These three are: 1) *Supervised Learning, Direct*; 2) *Supervised Learning, Learning Automata, Linear-Reward-Inaction*; and 3) *Supervised Learning, Learning Automata, Linear-Reward-Penalty* (Sutton & Barto, 1998; Sutton & Barto, 2018).

proposition and so receiving nothing. This is not really a choice at all. But accepting the first option can be construed as *cooperating*. The Ultimatum game demonstrates that these dynamics *do* exist even when *only actions* are considered, i.e., there is no cheap-talk or other signalling (besides actions). Similarly, Press and Dyson (2012) illustrated an algorithm with a coercive dynamic by introducing the *zero-determinant* family of strategies, which are able to induce exact behaviour in opponents and thereby control reward distribution. Coercive behaviour can also be observed in canonical game theory strategies with descriptive titles such as *Bully*, or *Tit-for-Tat* (Axelrod, 1984; Crandall, 2015; Rapoport et al., 2015). Therefore, it is not at all the case that modelling *only actions* prevents the opportunity for manipulation, or indeed any other behaviour, to arise.

1.4 Thesis Structure and Outline of Experiments

The current chapter has discussed the motivation for this research, has made an argument for the research question, and delineated the conceptual scope of the work by specifying important context and constraints. Next, in Chapter Two, these concerns are investigated further by exploring literature in the research domain found at the intersection of cooperation, computational learning, and game theory. Chapter Three presents a dual of a reduced form of the Robinson and Goforth topology[11] for use in the experiments conducted in Chapters Four, Five, and Six. These three chapters account, respectively, for three sets of separate experiments, listed below. These experiments follow a process of epistemic iteration[12] (Chang, 2004). Accordingly, they create a chain-of-referents, where each series of experiments extends from the last. The motivation for undertaking each series is given in its chapter, as such this list serves as a quick overview:

- Experiment Series One (Chapter Four) extracts a descriptive measure of cooperation from a multi-model tournament, following a model given by Crandall et al., (2018a; 2018b).
- Experiment Series Two (Chapter Five) is an investigation into the equivalence of agent behaviour under positive linear transformations of the agent's observation function, under the constraint of social dilemma inequalities holding for the modelled domain.
- Experiment Series Three (Chapter Six) first presents, then evaluates, an algorithm for *game model recognition*. This algorithm enables a behavioural dynamic to be identified as corresponding to a game in the Robinson and Goforth topology.

[11] The topology itself is a dual of the "familiar matrix representation" in game theory that shifts analysis "from the space of strategies to the space of payoffs" (Robinson & Goforth, 2005 p. 9).

[12] "I propose a brand of coherentism buttressed by the method of 'epistemic iteration'. In epistemic iteration we start by adopting an existing system of knowledge, with some respect for it but without any firm assurance that it is correct; on the basis of that initially affirmed system, we launch inquiries that result in the refinement and even correction of the original system. It is this self-correcting process that justifies (retrospectively) successful courses of development in science, not any assurance by reference to some indubitable foundation." (Chang, 2004, p. 6). This approach, coupled with the notion of the "referent" (La Tour, 1999, p. 310) form the basis, and probable extent, of any explicit positioning within philosophy of science that this thesis attempts.

Chapter Seven summarises the research and concludes the thesis. Additional supplementary material is located in Appendices A and B. Appendix A provides an overview of the agent model software framework, gives the default values used by the algorithms in each experiment series, tabulates energy use calculations, and also discusses some aspects arising from implementation in this domain. Appendix B contains availability information for the code and data used in this thesis, a graph adjacency list for the principal data structure that is derived in the thesis, and supplemental data in the way of tables and figures for each experiment series.

Chapter Two

Computing Cooperation

I wouldn't worry too much about the computer.

—Mission Control, *2001: A Space Odyssey*[13]

The discipline of biology has sought to understand cooperation using game theory, as has the field of computing and other branches and sub-disciplines of mathematics. This chapter aims to illustrate how problems in the domains of cooperation, computational learning, and computational game theory are intertwined. Attention is first turned to a historical exploration of the study of cooperative processes in biology, particularly as used in understanding evolution (§2.1). Following this, the chapter turns to an overview of computational methods commonly used in this domain to provide agents with the ability to adapt and learn (§2.2). The last section (§2.3) first examines the history and use of the workhorse of computational game theory: the normal form representation of the iterated Prisoner's Dilemma. This then leads to a discussion of the class of games known as the *social dilemmas*. In conclusion, the chapter summarises typological, taxonomical, and topological classifications of $\mathbf{2} \times \mathbf{2}$ game models.

2.1 Concepts of Cooperation

What we mean when we talk about *cooperation* can be conveyed in simple phrases, for example "a collective activity", "common effort", or "working together". But behind the apparent simplicity of these phrases are paradoxes (Lewis & Dumbrell, 2013), apparent contradictions (Wang & Guo, 2019), rules (Nowak, 2006), "puzzles" (Kreps et al., 1982 p. 246), extortion (Stewart & Plotkin, 2012), and conflict (Shubik, 1970; Rapoport & Chammah, 1965) that illustrate that this concept represents much more than can be boiled down to a reassuring maxim. Raihani (2021, p. 2) writes that cooperation, "sewn into the fabric of our lives", is at the heart of human society; that by "forming coalitions … our fortunes depend to a greater extent on the company we keep than on physical prowess." (2021, pp. 187–188).

Cooperation, in game theory models, can be measured as the outcome of the joint actions that result in a mutually cooperative solution for a game model (Crandall, 2018a). However, the process to result in 'cooperation', in any scenario except deterministic automata, is stochastic. At any point in the temporal history of the process there will be

[13] Directed by Kubrick (1968), screenplay by Kubrick and Clarke (1968).

measurable levels of 'cooperation' and 'non-cooperation'. In zero-sum games the distinction is clearly drawn but in general-sum games the dynamics of cooperation are a representation of an $\boldsymbol{n}$-dimensional, continuous space, discretised. Nevertheless, at each discrete timestep in a model, there will be a quotient (inclusive of zero) of the rate of mutual cooperation (*MCR*) over time[14].

2.1.1 Definitions

Definitions of cooperation[15] include those from, firstly, *The Oxford English Dictionary* (OED) (2023)[16]: "The action of co-operating, i.e., of working together towards the same end, purpose, or effect; joint operation." The OED gives the etymology of the word, from the Latin *cooperātiō*, as the "*action of cooperating, working together*."

Secondly, there is the definition for cooperation in the *Merriam-Webster*[17] (2023) dictionary: "The actions of someone who is being helpful by doing what is wanted or asked for: common effort … association of persons for common benefit".

From these definitions it follows that cooperation is a public good; that cooperation involves people working in concert to achieve a goal beyond any of them individually; and that an individual may expect to receive personal benefit by associating with others in a team. The wording also points to one of the less obvious facets of cooperation: 'wanted or asked for' can indicate coercion—inasmuch as the one doing the cooperating may not be receiving benefit from the interaction. The benefit may well be going to the one asking for cooperation. Even though activity is aligned, benefit is not necessarily so. In a similar vein, the phrases 'cooperating with police' and 'an offer you can't refuse' are euphemisms applied to situations where an individual may not have any choice *but by virtue of the act* is assessed as having 'cooperated'. This dynamic is at play in many situations where the act of cooperation is not freely given, not without fear of reprisal, retribution, or other consequence. In game theory, *zero-determinant* strategies are able to induce *varying* levels of cooperation in other participants (Press & Dyson, 2012), capturing the ultimatum dynamic and its power imbalance with precision. Cooperation is not necessarily an all or nothing concept—it has gradations inherent to its nature. Cooperation can stand-in for conflict, compromise, and coercion; as well it can be a signifier of positive group outcomes.

2.1.2 Cooperative Modalities in Evolutionary Science

Darwin (1859) observed eusociality in members of the *Hymenoptera*, that is, all ants and many of the species of bees and wasps. Eusociality is primarily characterised by the partition of reproductive labour in a population (Herbers, 2009). But, as individuals of the species specialise into non-reproductive roles they can lose, or never gain, the opportunity to perform breeding roles, raising the question of what is the evolutionary selection

[14] In the experiments in this thesis, the method used for determining an agent's *MCR* in a discrete game model is derived from Crandall et al. (2018a), with additional contributions from Robinson and Goforth (2005), and Bruns (2012), see §4.1.3.

[15] Both being valid, the spelling *cooperation* is used in this thesis in preference to *co-operation*.

[16] Oxford English Dictionary: https://www.oed.com/view/Entry/41037.

[17] Merriam–Webster: https://www.merriam-webster.com/dictionary/cooperation.

pressure that affords eusociality a favourable niche? Eusociality of this form, i.e., cooperative breeding, is found in other species, including naked molerats, meerkats, pied-babblers, and termites (Raihani, 2021). Darwin considered his inability to resolve his understanding of the dynamics at work in eusocial species as "one special difficulty, which at first appeared to me insuperable, and actually fatal to my whole theory." (1859, p. 236). The question of how members of species that aid other members of their group, to their own apparent detriment, would have evolved under natural selection is not entirely accepted as having been adequately explained despite an array of theories, including Darwin's own, eventual, arrival at the view that in these instances the group itself (or colony, in the case of the *Hymenoptera*) was the unit of selection, not the individual (Nowak et al., 2010). However, opinions vary; the dominant contemporary view says that the issue between the competing theories that attempt to explain the phenomenon is not resolved. Others contend that the entire question is no longer significant (Ratnieks, 2011). Herbers (2009) elaborated on the discourse in a letter to the Royal Society celebrating Darwin's 200th birthday[18] by stating that the question has long since become ingrained in the life sciences, generalised as the concept of altruism, "the evolution of a trait disadvantageous to its bearer but advantageous to another." (2009, p. 214).

2.1.2.1 Kin-selection

Laland and Brown (2011), writing on the history of evolutionary science and its practitioners, regard this specific evolutionary puzzle as not receiving satisfactory treatment until 1964, when Hamilton (1964a, 1964b) published on *kinship* and *inclusive fitness*. Laland and Brown observe that Hamilton's theory provided a model for understanding altruistic behaviour in related individuals, essentially boiling down to a sacrifice on the part of an individual in "helping closely related kin to reproduce" (2011, p. 53). Hamilton's theory, *kin-selection*, did not so much as run counter to Fisher's (1999)[19] earlier assertion that group-selection, i.e., a "benefit to the species", would not be a viable mechanism (Hamilton, 1963, p. 354); as show under what familial circumstances a selection pressure may contribute to the increased frequency of altruistic genes in (closely-related) populations. Hamilton, building on the earlier work of Haldane, formed a model that predicts the emergence of altruistic behaviour when genetic relatedness overcomes the ratio of the cost incurred, by the one who sacrifices, to the benefit afforded to the population (Laland and Brown, 2011, p. 53). Hamilton (1964a) wrote that this mechanism would place limits on intra-population competitive behaviour. Trivers (1974) concurred that kin-selection can result in conflict, giving an example of offspring that compete for food from parents. However, Nowak et al. (2010) put the view that inclusive fitness is an unnecessary theory. They reasoned that eusocial behaviours can be explained by natural selection alone. In response, Abbot et al. (2011)[20], argued

[18] Charles Darwin's 200th birthday, and also the 150th anniversary of the publication of 'On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life' (1859).

[19] Hamilton cites Fisher's 1958 (Fisher, 1958) paperback edition of the original 1930 (Fisher, 1930) work. The 1999 edition is an omnibus collection, 'A Complete Variorum Edition' (Fisher, 1999) published by Oxford University Press.

[20] There were 137 alphabetically listed co-signatories in this response to Nowak et al.'s (2010) position.

that inclusive fitness is as general a theory as natural selection and has demonstrable power to make falsifiable predictions. They rejected natural selection exceptionalism, and thereby also rejected the rejection of inclusive fitness. Ratnieks (2011) wrote that both Dugatkin and Wilson had posited that Darwin himself had already solved the 'riddle'. Wilson (1975) asserted that Darwin originated the concept of kin-selection, while Dugatkin (1997) wrote that Darwin had described a theory for inclusive fitness. While Ratnieks (2011, p. 489) concludes that the 'special difficulty' is "no longer considered a major evolutionary problem" he also concedes that it was the work of latter-day theorists such as Smith, Williams, and Hamilton that brought "altruism to the forefront by showing that it was a real evolutionary paradox".

2.1.2.2 Reciprocal Altruism

Trivers (1971) proposed *Reciprocal Altruism* as a model of behaviour where one individual acts to provide benefit to another without any relationship such as kinship or group membership being required. The question Trivers sought to answer was *why* any individual would act to help another if there was no benefit, particularly so if that act were detrimental to their own self. Trivers argued that long-term benefits of altruism would surface for the altruistic actor as such behaviours are (given certain conditions) favoured by natural selection. In Trivers' view, unrelated individuals who repeatedly interact would find that initial acts of altruism would be reciprocated, and eventually the initial cost of altruism could be out-weighed by long-term benefits. A secondary effect due to reciprocal altruism is the likelihood of individuals cheating—not reciprocating—and thereby accruing benefits from the altruism of others without experiencing the cost(s) to themselves. Laland and Brown (2011) stated that under this view reciprocal altruism would be more likely where the frequency of interaction was high enough to overcome the memory effect—that is, forgetting that the other party had not reciprocated. Trivers (1971) wrote that he identified repeated, symmetrical, and reciprocal situations as analogous to the Prisoner's Dilemma. Trivers focus was on the analysis of reciprocal altruism in humans: identifying conditions for altruistic behaviour and examining modalities for such behaviour: subtle cheating, the use of moralistic aggression to punish cheaters, network size, and cheating recognition. Trivers also noted that altruistic behaviours can be thought of as being calculated, but also just as behaviour without causal motivation (1971).

Dugatkin and Mesterton-Gibbons (1996) consider the iterated Prisoner's Dilemma as too weak a model. They asserted that there appear to be other evolutionary processes at work in promoting cooperation between unrelated individuals besides reciprocal altruism, namely "trait group selection [and] by-product mutualism" (1996, p. 19). They regard the iterated Prisoner's Dilemma as only having value to the consideration of the genesis of reciprocity, meaning that, as a theory, it is inadequate to account for other modalities of cooperation. Instead, Dugatkin and Mesterton-Gibbons (1996) proposed instead the *Cooperator's Dilemma*, a variant of Prisoner's Dilemma, for exploring alternative cooperative modalities.

However, reciprocal altruism is conceived of as having two modalities: indirect, and direct (Nowak & Sigmund, 1998). Thus, *direct reciprocity* is where a single act of altruism is performed in an exchange of some kind between two individuals. In contrast,

indirect reciprocity operates sans any exchange from the original receiver to the original giver, i.e., an altruistic act will not be reciprocated by the recipient in the initial exchange. Indirect reciprocity relies on reputational effects working in the original altruists' favour, influencing others in turn to act with altruistic intent towards the instigator in a virtuous karmic cycle. So, it is not necessary to have an exchange with the same individual(s) that an act of altruism was bestowed upon in the first place. Reputational effects also arise from bad behaviour—those who cheat will become known, and others who know of this will be disinclined to interact favourably with the cheater. Nowak and Sigmund (1998) introduced indirect reciprocity as a new framework that promoted the idea that cooperation can be beneficial to an initial altruist due to reputational influence. In later work, Nowak and Sigmund (2005) suggest that indirect reciprocity relies on all parties having a theory of mind; that while kin-selection relies[21] on genetic relatedness, indirect reciprocity would rely on a theory of mind (2005). Roberts (2008) examined these two modes of reciprocity (direct, and indirect) in communities and identified that previous models did not account for situations where the modes are blended. Roberts found that the potential to re-meet an individual, with whom indirect reciprocity had previously occurred, was not considered in previous work. Thus, the potential for direct reciprocity occurring after an act of indirect reciprocity (and vice versa) was being excluded from models. Roberts (2008) adopts an island model (confining the population) and analysed the prevalence of both direct and indirect reciprocity under various conditions, confirming Nowak and Sigmund's theory that indirect reciprocity rests upon the assumption that individuals never re-meet.

2.1.2.3 Group-selection

The cooperative modality *group-selection* proposes that natural selection can operate at the level of the group, such that some animals will make a sacrifice for the good of the group (Wynne-Edwards 1963; Laland and Brown, 2011). However, the theory was quite controversial. A critical attack by Maynard Smith (1964) attacked Wynne-Edwards' work on multiple grounds. Wynne-Edwards' (1964) letter in response was published along with Maynard Smiths' and acknowledged that they had little common ground for constructive dialogue. Williams (1966) criticised group selection sharply, arguing that the effects attributed to group selection were already adequately explained by natural selection (Laland and Brown, 2011). In the face of such criticism, and in clear contrast to the general acceptance of both inclusive fitness and reciprocal altruism, group selection became less accepted as a viable theory. However, it has been re-invigorated in part due to affinity with the cooperative modality *multi-level selection* that essentially rejected Hamilton's theory outright (Nowak et al., 2010). Among Nowak et al.'s (2010) claims for the dismissal of inclusive fitness, and thereby kin-selection, were the predicate of a causal link between haplodiploidy and eusociality, the failure of the theory when applied to diplodiploid sex determination, the apparent scarcity of eusociality in evolution, and the negative effects of relatedness through a lack of genetic diversity. Nowak (2006) derived game theory representations of what he identified as the five dominant mechanisms for the evolution of cooperation: inclusive fitness, group selection, direct

[21] "… provided Hamilton's rule holds…" (Nowak and Sigmund, 2005, p. 11).

reciprocity, indirect reciprocity, and network reciprocity (a graph-based model). Nowak (2006) considers cooperation to be as fundamental to evolution as natural selection.

2.1.2.4 Gene-level-selection

The view of evolution that centred the gene as the unit of agency in evolutionary science began in the 1930s and continues to this day as the Modern Synthesis (Bradley, 2020). In this view, agency at the level of the organism—as Darwin thought of it—was dismissed, in favour of a "molecular understanding of evolution" (2020, p. 93).

In the evolutionary schools that declare the centrality of the gene, firstly Sociobiology, then subsequently, evolutionary psychology (specifically, the Santa Barbara school), and then, through Dawkins' claims to the primacy of the gene (Bradley, 2020; Dawkins, 1976) the view crystallised that the agency of altruism is extinguished, reduced to an effect manifested in a lowering of the "survival prospects *of the genes* of the 'altruist'" and a "raising of the prospects *of the genes* of the … beneficiary" (Bradley, 2020, pp. 94–95), to the extent that, under this framing, "altruistic acts cease to be observable phenomena" (2020, p. 95). The phrase 'raise the prospects' refers to the gene's likelihood of reproducing into the next generation of the organism. The directedness of the control is explained by evolutionary psychology through methods akin to, if not actual, programming; i.e., the gene programs and controls the organism (Bradley, 2020).

Raihani (2021, p. 17) explains that the 'selfish' in *selfish-gene* refers to genes being "self-interested … each [having] a singular overriding" goal, to reproduce into the next generation, rather than referring to a gene having agency. The primacy of gene-level-selection in biology is apparent with the application of attribution of agency in gene-level selection, for example, the gene as the unit of measurement of relatedness, *against which* the *worth* of altruistic acts is calculated. Bradley (2020, p. 99) relates how West-Eberhard developed theory for the development of the phenotype as "the fulcrum of evolution". West-Eberhard's (2003) model of phenotype plasticity accounted for phenotypic variations and related these to gene-frequency variation, showing that not all organism behaviour or morphology is due to the effects of an agentic gene (Bradley, 2020). Powers (2010, p. 44) states that genes are "indirectly selected through the success of the phenotypes that they create", which allows the view that holding genes as the sole agents of replication does not preclude other mechanisms of selection (individual, and group) from acting upon a genotype; rather the gene is the expression of the change in gene frequencies as a result of, for example, group selection pressures (2010). These views, along with *niche construction theory* (Odling-Smee et al., 2003), where organisms "can actively create their own selection pressures" (Powers, p. 12), are evolutionary models that acknowledge varied pressures and mechanisms in evolutionary processes, in contrast to the free-will-less organism promulgated by the selfish-gene and evolutionary psychology (Bradley, 2020).

2.1.2.5 Institutional Cooperation

While not directly an evolutionary cooperative modality, the role of cooperation in group formation (institutions) in human society can be viewed as a selection pressure over a population, which would have considerable effects on the health and well-being of populations.

Eriksson and Strimling (2012) describe laboratory experiments that explored outcomes in public good games, under varying conditions, designed to model a community. Firstly, Eriksson and Strimling (2012, p. 1) acknowledge the "robust finding" from across disciplines that *people do not conform to the behavioural expectations of rational agency*. As many have found (Colman, 1998; Bowling & Veloso, 2001; Sally, 1995; Conlisk, 1996; Frohock, 1996; Kreps et al., 1982; Guyer & Rapoport, 1972), the game theory ideal that people will function as rational actors, i.e., *homo economicus*[22], is not supported as a finding in empirical research. Guyer and Rapoport phrase the rational solution in this dichotomy as a "unique normative solution" (1972, p. 413). Henrich et al. (2001, p. 73) observed that normative solutions are undermined by "large, consistent deviations", which they attributed to a tendency for people to act in service to fairness. This led them to ask if the observed variance is shaped by local factors, as opposed to being observable in any setting.

Eriksson and Strimling (2012) found that, where experimental studies do not select participants for a like-minded willingness to cooperate but instead select such participants that they reflect a heterogeneity of types, levels of cooperation often decline. They define the *easy problem of cooperation* as when all members of a group are of a cooperative type, and the *hard problem of cooperation* as when all members are predisposed to non-cooperation (i.e., defectors). They noted that most groups consist of individuals with tendencies in between these extremes. An important tenet is that they do not necessarily consider the non-cooperative type as a paragon of the rational actor. Instead, they consider that choices are made as outcomes of a variety of information sources including processes, social situations, and preferences, all of which can change rapidly over time, or vary substantially by situation or context. In other words, they allowed that cooperation is conditional and can vary in frequency. The public good game that Eriksson and Strimling (2012) used in their experiments with people models institutions following Ostrom's (2015) definition[23]. Eriksson and Strimling (2012) speculate that an institution modelled to this definition would exhibit increased levels of cooperation due to the rules allowing a variety of solutions towards cooperation. As a function of societies, Eriksson and Strimling extrapolate from Rothstein's (2000) summary that *beliefs* and *belief systems* are a result of specific, and localised, factors; not a 'law' of human nature. Eriksson and Strimling prescribe their institutional policies to be either *weak* (policy strictures accept low levels of cooperative contribution) or *strong* (demand higher levels of cooperation). Alternatively, the policies could be allowed to *evolve* by actors voting on the rules and strictures of the areas of institutional concern: *operational rules*, *monitoring*,

[22] The theory of *homo economicus* (https://en.wikipedia.org/wiki/Homo_economicus) says that a rational agent is expected to be purely self-interested and to always choose the option that leads to the maximum utility for themselves. Whether people *should* act in this way is subject to normative prescription. The expectation that decision-making will be performed rationally is founded in utility theory, specifically *expected utility* (von Neumann & Morgenstern, 1944, p. 9). Agents encompass a spectrum in regard to being cooperative: from the pure free-rider, or non-cooperative type, who will choose the defect option every time, through to the cooperative type whose first thought is to commit acts of altruism, even at their own immediate expense.

[23] In Eriksson and Strimling's words: "In the context of common-pool resource problems, Ostrom's precise definition of institutions are 'the sets of working rules that are used to determine who is eligible to make decisions in some arena, what actions are allowed or constrained, what aggregation rules will be used, what procedures must be followed, what information must or must not be provided, and what payoffs will be assigned to individuals dependent on their action' (Ostrom, 2015)." (2012, p. 2).

rewards, *punishments*, and *change of rules* (Eriksson and Strimling, 2012). They found that under a weak institution the actor's types (cooperative, or non-cooperative) determined levels of cooperation; under a strong institution the types mattered little; and under an evolving institution the influence of the ratio of the types on the institutional character was marked.

In the case of a weak institution, Eriksson and Strimling (2012) found that groups dominated by, or homogenously composed of, non-cooperative types achieve very low levels of cooperation; but with a strong institution the same groups achieve high levels of cooperation. Further, the same non-cooperative groups did not evolve to a strong institution when afforded the capability, demonstrating that groups of rational actors do not behave as *homo economicus* would demand, that is, to take the opportunities that always lead to the highest payoff.

In contrast to Eriksson and Strimling's investigations of Ostrom's work in laboratory settings, Pitt (2003, 2017, 2021), with colleagues (Pitt et al., 2011; Petruzzi et al., 2017; Pitt et al., 2020; Diaconescu and Pitt, 2017; Scott and Pitt, 2023), has developed a body of work that examines the programmatic formalisation of Ostrom's (2015 [1990]) institutions in a computational logic domain of adaptive agents, collectivised in multi-agent systems, acting as systems of systems, in the context of evolving and self-governing "electronic institutions" (Pitt, 2021, p. 165). Pitt's work empirically analyses Ostrom's eight principles for "the evolution of enduring self-governing institutions" (Pitt, 2021, p. 176), leading to a range of findings, for example, the finding that electronic institutions, in the context of a "linear public good game" (p. 165), can be "stable and enduring" (p. 190). Pitt underscores this finding by observing that concerns aimed at Ostrom's principles—as not being scalable—are themselves cast into question by empirical analysis of sustainable Minecraft servers (Frey and Sumner, 2019); which in turn informs the debate "on whether, for institutions at scale, coercion is required" (Pitt, 2021, p. 177) in order to endure.

In addition, Pitt's work has investigated the analysis of a suite of metrics for the evaluation of processes in electronic institutions, such as the process of evaluating and determining procedural justice (Pitt et al., 2013). Much of Pitt's work is directed towards application of Ostrom's principles in order to evaluate their utility for application to autonomous agents in a variety of settings. Pitt sees the application of electronic institutions in multi-agent systems for the management of distributed energy networks as a practical outcome of his work, and a model for future sustainable resource management (2021).

The wider field in which Pitt's work is situated, that of computational analysis of agent learning, behaviour, and dynamics, is discussed next, in §2.2.

2.2 Computational Learning

Given a deterministic and stationary environment, agents can use dynamic programming (DP) algorithms to determine optimal policies. Achieving optimality, however, is dependent upon the agents having a correct and whole model of the environment (Sutton et al., 1992). When this is not the case, if the model of the environment is incomplete, or it has errors, agents estimate the model using the method *indirect adaptive control*. This

method is costly as the model has to be progressively updated, which necessitates recomputing the policy at every timestep (Sutton et al., 1992). An alternative and less costly method is *direct adaptive control* which views the environment as a Markov decision problem of (unknown) state transition probabilities (Sutton et al., 1992). From this control theory starting point, they describe model-free reinforcement learning (RL) methods as incremental DP algorithms that use experience of the environment. As such, RL methods are a "computationally simple, direct approach" (1992, p. 19) to solving non-linear control (i.e., complex policy) problems. The RL algorithm *Q-Learning* is model-free, that is, it does not estimate a system model (1992). Rather, *Q-Learning* will estimate the *return* for performing actions in any given state. The return is the discounted sum of future expected rewards (1992) for an action in a state. As the algorithm is calculating returns to choose actions, rather than using a model of the environment, *Q-Learning* is classed as a *direct* and *adaptive* control method. This class of methods aligns with the constraint outlined in §1.3.1, i.e., to be *on-line*.

While they are an improvement over algorithms based on indirect adaptive control methods, RL methods in non-stationary environments are still expensive. By definition, non-stationary environments change, so any estimation of return will likely be based on incomplete data. For example, an algorithm may not be aware of things occurring in the environment over which they have no visibility. Adopting a Markov formalism, sans explicit model-building, reduces computational complexity by imposing a memory depth constraint on agents to mitigate the 'curse of dimensionality' (Papadimitriou & Tsitsiklis, 1987; Wong et al., 2023; Bellman, 1957; Sutton & Barto, 2018). Markov models also fit the constraint that *agents have incomplete information*, as prescribed in §1.3.2; and in addition, the on-line constraint given in §1.3.1: *to compute on data as it is received*. Markov application to iterated $\mathbf{2} \times \mathbf{2}$ stage games is described in §2.2.2.

An agent's optimal policy is one that maximises its own reward—likewise for all other agents in the same world space. However, this self-interest may come at a cost. An agent may find that what is optimal for itself in the short-term may not be so in the long-term (Watkins, 1989). If the world is stochastic, i.e., the transition probabilities between environment states are non-stationary, then calculating an optimal policy *now* may not hold *later*. Attempting to learn optimal policies in a complex world is an uncertain endeavour. There may be many possible paths through the strategic space to an optimal solution, of which only some may be of net benefit. Exploration of possible paths will necessitate some quantum of inefficient learning, as "it is usually necessary to make mistakes in order to learn" (Watkins, 1998, p. 9).

Watkins (1989) explores the relationship between *optimal learning* and *learning of efficient strategies* to highlight their function and their contribution to adaptation and learning of behaviours. The first, optimal learning, informs behaviour by learning how to learn so that the process of learning improves. The second, learning of efficient strategies, is characterised by short-term objectives. A maximally efficient strategy is an optimal method for achieving an intermediate criterion, while optimal learning is the learning of the methods to initiate, actuate, evaluate, and then exploit strategies, contingent to circumstance. The tension between optimal learning and its constituent strategies is analogous to the dichotomy found at the heart of cooperation. The challenge the agent faces is to align short term strategies to an overall good. Short term payoffs obtainable through strategies have the potential to be myopic, blinding an agent, to its own detriment,

if it pursues an immediate reward rather than an overall objective that may result in increased reward. Represented as a matrix of two options for two participants this dynamic is reducible to the Prisoner's Dilemma, where the 'other' participant is the entire world, i.e., the environment. This *all-against-one* class of games is referred to in game theory as a *game against nature* (Colman, 1995). In computational learning game theory literature Lanctot et al.'s (2017) *independent learners* ignore the existence of other agents in an approach that blends game-theory, deep learning and reinforcement learning in non-stationary environments. Agents that are aware of other agents and share full knowledge of the world were modelled by Claus and Boutilier (1998) as *Joint Action Learners* (JAL), in a cooperative model of *Q-Learning* agents. The conception of *JAL* draws on Brown's (1951) algorithm *Fictitious Play*, in which an iterative process tracks an agent's expectation of the other agent's actions based on observed previous actions. *Fictitious Play* can be thought of as 'I will play my best response to your most frequent action', or in more colloquial terms: *do the best I know, for what you always do*. Brown describes this method as "choosing at each turn a pure strategy which is optimal against the cumulated history of the opponent's plays to date." (1951, p. 374). *Adaptive Play* was introduced by Young (1993) as a variant of *Fictitious Play*. Where *Fictitious Play* retains observability over the entire history of the actions of others, *Adaptive Play* retains only a limited window over the recent past; the memory of other agents' actions being collapsed to a finite window which in effect allows agents to forget previous interactions with other agents after a certain time. The previous interactions will at some point no longer be explicitly taken into consideration when evaluating future actions. Young shows that a stochastically stable equilibrium *is* possible, as *adaptive play* agents will converge on a subset of Nash Equilibria states.

2.2.1 Learning from a Dilemma

Wooldridge (2012a) argues that game theory and the Prisoner's Dilemma are well founded as a mathematical model for the study of cooperation in the interactions of agents as problems of this type genuinely reflect real-world situations—for example, the dynamic of *mutually assured destruction* reduces to a zero-sum security dilemma (Lindley, 2001). Jervis (1978) examines security dilemmas (spiral models) as instances of both Stag Hunt and Prisoner's Dilemma games. Applying game theory to the dynamics of "self-interested agents" reduces a given situation to an "abstract mathematical model of a multiagent decision-making setting" (Wooldridge, 2012a, p. 76). The utility of these abstract mathematical models is obtained through both *descriptivist* and *normative* interpretations—the two predominant ways in which game theory is thought about and used in modelling behaviour or situations (Wooldridge, 2012b). Descriptivist game theory is undertaken through the observation of behaviour and the capturing of metrics, while normative game theory seeks to provide instructional directives: given a situation, what is the optimal course of action? When played as a single-stage game, the canonical representation of a decision-making situation, the Prisoner's Dilemma, suggests that both participants will choose the action that is sub-optimal (Poundstone, 1993). Shubik (1970) shows that even though there is a Pareto-optimal reward surface in the Prisoner's Dilemma, the dominant strategy, mutual defection, is not in the Pareto-optimal set. In other words, both players could do better, but when faced with the uncertainty of what

the other players' decision will be, each may (and likely will) choose to defect (and thus, do worse). When iterated, the challenge becomes to see past the short-term payoff available from defection and trust the other participant, to gain the increased payoff available from mutual cooperation.

2.2.2 Imperfect Information Markov Games

Markov games (van der Wal, 1980; Littman, 1994), also known as stochastic games (Shapley, 1953; Owen, 1982), extend MDPs in several ways: firstly, the single-agent MDP is now cast in a multiagent context, and secondly, state determination is by the joint actions of the participants (Bowling & Veloso, 2001). In addition, cycles can occur in state transitions, so infinite games are possible (Owen, 1982). In effect, Markov games are a superset of MDPs (Laurent et al., 2011). Markov games have been used extensively with reinforcement learning (Littman, 1994; Wang & Sandholm, 2002; Busoniu et al., 2008; Laurent et al., 2011). Importantly, Laurent et al. (2011) note that a non-stationary environment remains Markovian if the next transition depends only on the current state and on the next set of actions from participants, i.e., that the transition function is not history dependent (2011).

The constraint given in §1.3.2 regarding *observability of state* is imposed on an environment by both the specification of *perfect-*, or *imperfect-information*, and by the specification of *complete-*, or *incomplete-information*. In an on-line model the observation function is most obviously affected as a participant's observation space is reduced from visibility over all participants actions in a *perfect-information* game to visibility over only one's own actions.

Leibo et al. (2017) give a treatment for a *partially observable markov decision process* (POMDP) Markov game $\boldsymbol{M}$ for two participants, $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, which provides for *imperfect-information* in each participant's observation function $\boldsymbol{O}_i$. Where a Markov game $\boldsymbol{M}$ for two participants is a set of states $\boldsymbol{S}$ and an observation function $\boldsymbol{O}$:

$$\boldsymbol{O}: \boldsymbol{S}\ \boldsymbol{x}\ \{\mathbf{1},\mathbf{2}\} \rightarrow \boldsymbol{R}^{\boldsymbol{d}}$$

A set of actions $\boldsymbol{A}$ (restricted to either *cooperate,* $\boldsymbol{C}$**,** or *defect,* $\boldsymbol{D}$) for each participant is combined with state $\boldsymbol{S}$ to obtain a transition function $\boldsymbol{T}$, which gives a discrete transition probability distribution Δ over $\boldsymbol{S}$ and a reward function for each participant $\boldsymbol{A_i}$:

$$\boldsymbol{A_1} \text{ and } \boldsymbol{A_2}\ ;\ \boldsymbol{A_1} = \boldsymbol{A_2} = \{\boldsymbol{C},\boldsymbol{D}\}$$

$$\boldsymbol{T}: \boldsymbol{S}\ \boldsymbol{x}\ \boldsymbol{A_1}\ \boldsymbol{x}\ \boldsymbol{A_2} \rightarrow \Delta(\boldsymbol{S})$$

$$\boldsymbol{r_i}: \boldsymbol{S}\ \boldsymbol{x}\ \boldsymbol{A_1}\ \boldsymbol{x}\ \boldsymbol{A_2} \rightarrow \boldsymbol{R}$$

Each participant has an observation space $\boldsymbol{O_i} \in \boldsymbol{O}$; participant $\boldsymbol{i}$ selects action $\boldsymbol{A_i}$ by policy $\boldsymbol{\pi_i}$:

$$\boldsymbol{O_i} = \{\ \boldsymbol{O_i} \mid \boldsymbol{s} \in \boldsymbol{S}, \boldsymbol{O_i} = \boldsymbol{O}\ (\boldsymbol{s},\boldsymbol{i})\ \}$$

$$\boldsymbol{\pi_i}: \boldsymbol{O_i} \rightarrow \Delta(\boldsymbol{A_i})$$

An extension by Leibo et al. (2017) sets the social dilemma inequalities into Markovian terms. This extension is covered in §2.3.2.

2.2.3 Practical Considerations for Adaptive Agents

There are many issues that arise in implementing computational learning agents, from the theoretical to the practical (for example, the application of constraints, or the handling of overflow in *softargmax* functions to allow the resulting action-selection to still be regarded as a *valid* action-selection).

This section discusses some observations and some issues that arose during the implementation of the algorithms featured in the experiments presented in this thesis. These issues are detailed here as, in retrospect, they have some theoretical interest, and secondly, while these issues are essentially orthogonal to the experiments presented in Chapters Four, Five, and Six; they do, however, inform the experimental context.

2.2.3.1 The Problem of Now

Each algorithm studied in this thesis engages in a sequence of operations at each timestep that comprise that algorithm's internal processing. An algorithm receives an input from the environment (the reward), performs some computation and updates its internal state, then calculates and emits an action-selection. For some algorithms, the literature is clear that the above sequence holds. However, for a small number of algorithms the literature is less clear, and it can be interpreted from text descriptions and pseudocode that the sequence of operations is that the algorithm receives an input, algorithm calculates and emits an action-selection, algorithm performs some computation and updates its internal state.

Another way of putting this issue is that it manifests in the use of a state variable in an algorithm's action-selection mechanism such that an algorithm would draw its next action from its internal representation of the *prior state of the world* ($\boldsymbol{t}$), which is one step behind its *current understanding of the world* ($\boldsymbol{t} + \mathbf{1}$). This results in action-selection such that at time $\boldsymbol{t} + \mathbf{1}$ (i.e., *now*) the processing of the incoming reward signal had been completed and internal state had been updated, but the action for the next timestep was being drawn using the state at timestep $\boldsymbol{t}$ (i.e., the immediate past). Not only does this break a strict Markov interpretation, but also, due to the lack of clarity, the potential arises for the sequence of operations that agents perform to be applied inconsistently. Sutton and Barto (1998; 2018) use a softmax action-selection technique in the first edition, and continue its use in the second edition, of *Reinforcement Learning: An Introduction*. However, their presentation is not explicit with regard to the sequence of steps in action-selection when using the softmax (specifically, *softargmax*[24]) technique in the A*ctor/Critic* algorithm, but their treatment of *bandit* algorithms (those that use softmax action-selection) does follow an explicit procedure:

1. receive incoming reward,
2. process incoming reward and update internal state, and
3. determine action-selection using the current, now updated, state.

[24] The function *softargmax* differs from *softmax* in that *softargmax* uses an exponential shift-invariant procedure to mitigate against computational overflow in the calculation of the probability distribution from which actions are drawn.

Szepesvári (2010) is explicit regarding the sequence of updates to state and subsequent action-selection in *Actor/Critic*. This sequence of operations conforms to that given for Sutton and Barto's *Bandits*.

The conclusion arrived at in implementing these algorithms is to standardise the sequence for all algorithms, that is, to create a global constraint that all processing of incoming signal must be completed *before* an action is selected from the *current state*. The interval for the agent to incorporate the reward signal is bounded by, at the earliest, the reward signal being received, and at the latest, the action being conveyed back to the environment. An agent may asynchronously process internal state outside of this interval without contravening this sequence if it maintains conformity to Markov constraints. Two *Actor/Critic* variants[25] are studied in this thesis in addition to *Actor/Critic*. Both of these implementations conform to the explicit sequence of operations that draws action-selection from the current timestep.

2.2.3.2 Algorithmic Generalisation

A vast array of algorithms and strategies have emerged from the game theory literature, in many cases they target specific $\mathbf{2 \times 2}$ matrix games. This can be seen not just in the growth in the number of strategies submitted to Axelrod's (1980a, 1980b) two tournaments (growing from fourteen entrants in the first competition to sixty-two entrants in the second competition) but also in the subsequent proliferation of strategies in the game theory literature that reference the dominant algorithms that emerged from these competitions (Rapoport et al., 2015; Hofstadter, 1983). For example, the *win-stay-lose-shift* (WSLS) strategies' actions, dependent upon the outcome of the last timestep, have a semantic clarity (Nowak & Sigmund, 1993). As its name suggests *WSLS* is geared towards a game-theoretic dynamic. Thus, there is a distinction that can be drawn between algorithms that are specifically designed to reason in some manner on the Prisoner's Dilemma matrix game, and algorithms that are more generalised in their learning capability. Crandall et al. (2018a) observe that algorithms specifically designed for the Prisoner's Dilemma exhibit poor performance across alternative game models, i.e., their ability to generalise to new environments appears to be limited.

This thesis investigates the behaviour of algorithms using $\mathbf{2 \times 2}$ games as a vehicle for representing the environment, as they afford a measure of tractability, and this assists the process of inference. The aim is not to necessarily find, or develop, an algorithm that shows optimal performance for a given $\mathbf{2 \times 2}$ game at the expense of all other considerations, but to use generalised algorithms in the context of $\mathbf{2 \times 2}$ games. This consideration leads to a focus in this thesis on learning algorithms, specifically *foundational Reinforcement Learning* (*fRL***)** algorithms, rather than game theory algorithms or strategies. This consideration does not allow the inclusion of RL algorithms that require substantial prior training. Algorithms that require long lead-ins in single-shot runs to achieve better-than-average performance, i.e., 'training-on-the-fly', are also excluded[26]. Ultimately this is reflective of the direction of the research overall: that is, to

[25] The two variants are *Actor/Critic with Eligibility Traces*, and *Actor/Critic with Replacing Traces*. The *Actor/Critic* algorithms are used in Chapter Four and Chapter Five.

[26] Crandall et al. (2018a) report that the variant of *DQN* they used, *Deep-Q*, showed relatively poor performance.

make generalisation and reproducibility a matter of *a priori* interest rather than engage with specialised algorithms (of any paradigm) that either necessarily have specific goals for the achievement of highly-specific objectives in the $\mathbf{2} \times \mathbf{2}$ game; or, have reference architectures and implementations that are either closed, only sketched, or that contradict each other, as is sometimes the case with recent deep reinforcement learning algorithms. It may be that the separation of concerns here is illusory, at least partly, and that any game theory strategy for a specific $\mathbf{2} \times \mathbf{2}$ gameform would generalise well enough in environments representing larger problems, that the question becomes moot.

Regardless, the issue of generalisation shifts the focus away from game-theoretic algorithms to generalised learning algorithms. In support of this approach to the selection of algorithms for experimental work in game theory, Merhej et al. (2021) make a similar argument for choosing reinforcement learning algorithms over game-theoretic algorithms in their investigation of *intention recognition* (see §1.2.1.2) in a public goods game.

2.2.3.3 Discontinuities & Lock-In

A discontinuity in computational learning is a sudden change of behaviour in an algorithm. Where an algorithm has exhibited a series of actions that can be regarded as having a character (e.g., playing the 'defect' action, and only 'defect', for a considerable number of timesteps) and then abruptly changing that behaviour such that the series of actions can be regarded as having a new character. The new character may be to alternate cooperate and defect, rather than to only defect, or to only cooperate when previously it was only defecting. The agent's character emerges from a process governed by its learning algorithm operating on its representation of internal state. A discontinuity may arise where the values in the internal state experience a tipping point in the stochastic probability of choosing one action over another. Alternatively, a discontinuity may arise via the flow-on effects of having taken a low-probability action. For that matter discontinuities may arise from any action taken at the right time if the conditions at that moment allow some movement in the reward landscape to a new region.

Where the internal state is such that a certain sequence of actions is significantly more probable than any alternative, and the action-selection method induces an exploration action out of character with the current pattern of behaviour then that single exploration action may not be enough to induce a change in the behavioural character unless the internal state is somehow otherwise predisposed to a change. For example, if visualised as a trajectory of steps in a 3D surface where the agent has followed a character that can be thought of as flowing down an incline to a saddle point, an action that is not in keeping with the current pattern might have the effect of shifting the vector so that the agent's trajectory heads down (or up) one or the other slopes or inclines of the saddle point rather than maintaining the previous vector towards the saddle point.

A discontinuity that is induced via a tipping point in the internal state representation may occur where the values of the various parts of the algorithm combine in a way to alter the trajectory of some part of the internal state representation such that a new character emerges. Once an internal state variable approaches a value that is either very large or very small then it is unlikely that a discontinuity will occur. It is highly likely that the algorithm will remain 'locked in' to the current behavioural pattern. In these cases, it appears unlikely that even an out-of-character exploration action will bring about

any wholesale shift to a new character given the very large or very small probabilities involved. In such cases an agent experiences *lock-in*. In the experiments in this thesis, agents have only two actions so they do not have so many opportunities to climb out of minima as might be available in an environment where the agent has a larger action space with which to interact with, and alter, the environment. Having a larger action space allows an agent to explore more, and in more ways find more paths in the landscape. For the relatively simple environments that are $\mathbf{2} \times \mathbf{2}$ games, trajectories to, and once in, lock-in are quite stark, and offer little opportunity to escape.

There is synchrony in these observations with the description Sutton and Barto (1998) give of the difficulty that *bandits* have in binary-action domains such as a $\mathbf{2} \times \mathbf{2}$ matrix; see §4.1.2.2.

2.2.4 Algorithm Groups

The discussion of computational learning in game theory has, to now, had a broad focus concentrating on theory-driven scoping constraints for the intended experimental domain. This section applies these constraints to the selection of algorithms for the series of experiments in this thesis.

The requirement of being *on-line* (§1.3.1) excludes algorithmic solutions that require the solving of equilibriums, game values, or solution concepts; as such methods are generally too computationally expensive. While effectively tractable in $\mathbf{2} \times \mathbf{2}$ game environments these methods scale poorly (Nash, 1951; Nash, 1953; Binmore et al., 1986). This constraint also applies to algorithms that *depend on training*. As single-shot experiment instances, each algorithm starts with a blank slate.

The algorithms used in this thesis are of three types: *game-theoretic*; *binary bandits*; and *foundational Reinforcement Learning* methods (*fRL***)**. Not all of the algorithms are strictly learning algorithms: some are naïve game-theoretic strategies; others are deterministic finite automata; some are both. The game-theoretic group consists of a selection of the well-known, e.g., *Tit-for-Tat*, the practical, e.g., *Random*, and also the ubiquitous (Fudenberg & Levine, 1998; Vrieze & Tijs, 1982; Perrin et al., 2020; Heinrich, Lanctot & Silver 2015; Ma et al., 2017) *Fictitious Play* (Brown, 1951; Robinson, 1951). *Fictitious Play* is a special case in regard to its categorisation as it is at once game-theoretic, belief-based, and is also a learning algorithm, plus it can be a pure strategy (dependent on externalities) as *Fictitious Play* does not require[27] any internal stochasticity. *Fictitious Play* is used in Chapter Four only, and is grouped with game-theoretic strategies, in part due to its game theory lineage but more generally for its pervasiveness, apparent simplicity, and for its utility (Fudenberg & Levine, 1998)[28]. The 'best-response' mechanic of *Fictitious Play* requires visibility into other participant's actions, i.e., a *perfect information* view. This casts the affected experimental instances

[27] Contrarily, this algorithm has been extended to incorporate stochasticity; for an example see *Stochastic Fictitious Play* (Hofbauer and Sandholm, 2002; Crandall et al., 2018b).

[28] The reasoning for changing scope for *Fictitious Play*, as well as a small number of other algorithms, is discussed further in §4.1.2. Of the 37 algorithms used in Chapter Four there are three supervised learning *binary bandit* methods and two static game-theoretic strategies that also require relaxation of the observation function. As with *Fictitious Play* these five algorithms are used in Chapter Four only.

from POMDP to MDP, in other words, to a Markov game where each participant's observation function $\boldsymbol{O_i}$ is equivalent to the game's observation function $\boldsymbol{O}$.

A *binary bandit* is a simplified form of a *k-armed bandit* (Sutton & Barto, 1998; 2018), where the algorithm's action draws a reward from a probability distribution distinct to each action. As they are distinct, the bandit cannot use associative learning to transfer knowledge of rewards obtained from one action to infer anything about the rewards that may be obtained by a different action. The bandit class of algorithms highlight the general exploration/exploitation problem, but not without "strong assumptions about stationarity … that are either violated or impossible to verify" (Sutton & Barto, 2018, p. 27).

The *fRL* algorithms are drawn from the literature (Sutton 1998, 2018; Szepesvári, 2010) and predominantly adopt a tabular approach to discretisation of state; in addition, there are some methods using linear function approximation. Beyond the groupings already mentioned, the algorithms examined in this thesis can be grouped according to their functional components. Among the learning algorithms a distinction can be made between those that implement tabular data structures (for processing internal state) and those that use gradient methods. Similarly, there are several distinct branches of learning algorithms, for example, the *Actor/Critic* family and *Q-Learning* and its variants.

Contemporary reinforcement learning algorithms, such as *Proximal Policy Optimisation* (Schulman et al., 2017), *Deep-Q-Network* (Mnih et al., 2013), or *Soft-Actor-Critic* (Haarnoja et al., 2018), were considered in early phases of this research but were rejected due to their generally high requirements for training that would challenge the on-line constraint (see §1.3.1**)**, as evidenced by Crandall et al.'s experience with a *Deep-Q-Network* variant, *Deep-Q*, which exhibited "slow runtime" (2018b, p. 36), and also, for example, by Haarnoja et al.'s (2018, p. 1) finding that, with *Soft-Actor-Critic*, "relatively simple tasks can require millions of steps of data collection".

This suggests that the tractability of the *fRL* algorithms, in contrast to contemporary RL algorithms, is of practical benefit in the experiments in this research for reasons of parsimony. Integrating contemporary RL algorithms to this research stream is a goal for future work, where the conception of focusing on agentic behaviour in more complex domains, with sparse rewards, would better suit the capabilities of contemporary RL algorithms than do $\mathbf{2} \times \mathbf{2}$ game matrices.

Each algorithm group is discussed further in §4.1.2.

2.3 Computational Game Theory

In the previous section mathematical representations of fundamental evolutionary processes and social mechanisms were cast through the lens of cooperation. Game theory, specifically the normal-form representation of the Prisoner's Dilemma—either using the single stage game to explore the core dynamic, or in iterated forms to explore the emergent properties of temporal strategies—has been used extensively in the study of fundamental evolutionary processes. The use of game theory as a research tool is the focus of this section.

For autonomous agents to 'work together' each agent in a group must optimise their intermediate (short-term) processes in a way that results in optimal results for the collective as well. Collective processes arise over a longer-term than the timestep scale

that agents experience. Flack (2012, p. 1802) describes this process in reference to time-scales and the dynamics of social systems: "as an interaction history accumulates the coarse-grained representations consolidate", and further, that as they "consolidate around stable values … slowly become stable predictors of the … system's future state" (Flack, 2012, p. 1807). A *coarse-grain consolidation* approach to the process of agglomeration that Watkins' (1989) describes, regarding *optimal learning* (see above, §2.2) could potentially be of use to the study of individual/collective problems.

Regardless of the mechanism(s), an agent's strategy will benefit the group if the agent is aligned in some way to both long-term and short-term objectives. Once again, the difficulty in achieving this alignment is captured as a strategic dynamic by the game theory model Prisoner's Dilemma.

Poundstone (1993), writing on the origins of the mathematical theory of games, relates how von Neumann (1928a) formed a view that the mathematics of the card game Poker cannot be captured in a model using probability alone—modelling the tactics that people employ demanded a more comprehensive treatment. Generalising from Poker, von Neumann saw that the body of theory "could be applied to economics, politics, foreign policy, and other diverse spheres" (Poundstone , 1993, p. 6). The book *Theory of Games and Economic Behavior* (von Neumann & Morgenstern, 1944) introduced a game as a situation of concurrent choice-selection based in conflict, where the outcome is a result of all the choices in a determinable, and pre-specified, manner (Poundstone, 1993). This framing allowed von Neumann and Morgenstern (1944) to map from games of strategy to the dynamics of behaviour in economics. They defined *utility* as a linear function to derive a numerical valuation for the expectation of gain over a set of probable events. Their treatment assumes complete-information: "all subjects…are completely informed." (1944, p. 30). von Neumann and Morgenstern stated that their goal was to find mathematically complete principles for defining a policy of rational behaviour. They define *zero-sum* games as those interactions of "two participants where the sum of all payments is zero." (1944, p. 34). Previously, in *Zur Theorie der Gesellschaftsspiele*[29], von Neumann (1928a), exploring rational courses of action for players whose interests are opposed (in general-sum games), proved the minimax theorem for zero-sum games (Poundstone, 1993; Kuhn et al., 1996). An equilibrium solution $\boldsymbol{M}$ (von Neumann 1928a, p. 303; von Neumann, Trans. Bargmann, S., 1928b, p. 22) is:

$$\boldsymbol{Max_x\, Min_y\, g(x,y) = Min_y\, Max_x\, g(x,y) = M}$$

where the maximum of the minimum payoff $\boldsymbol{g}$ from the strategy pair $(\boldsymbol{x},\boldsymbol{y})$ (corresponding to the strategies for player $\boldsymbol{S_1}$ and player $\boldsymbol{S_2}$, respectively) is equal to the payoff $\boldsymbol{g}$ of the minimum of the maximum possible.

$\boldsymbol{M}$ is a saddle point which von Neumann (1928b, p. 30) thought of as being a "locus" over which players $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$ would compete: $\boldsymbol{S_1}$ attempting to maximise $\boldsymbol{g}$ and $\boldsymbol{S_2}$ attempting to minimise $\boldsymbol{g}$. Over 'steps' of time, i.e., repeated interactions, this dynamic would resemble a "tug-of-war" (1928b, p. 21). For player $\boldsymbol{S_2}$ this is a *minimax* problem, for player $\boldsymbol{S_1}$, a *maximin* problem (Bhattacharya, 2021).

---

[29] *Zur Theorie der Gesellschaftsspiele*, more well-known as 'The Theory of Parlour Games' translates also as 'The Theory of Games' but is titled 'On the Theory of Games of Strategy' in the 1959 translation (von Neumann, J., & Bargmann, S., 1928; 1959).

Finding an equilibrium solution was solved by Nash (1950a, 1950b), in generalising the von Neumann and Morgenstern equilibria $\boldsymbol{M}$ from two-person zero-sum games to $\boldsymbol{n}$-person, finite, non-cooperative games having at least one equilibrium point, remarking that the set of such points is "the set of all pairs ... of good strategies" (Nash, 1951, p. 286). Solving for Nash Equilibria (NEs) and Nash Bargaining Solutions (NBSs) is potentially computationally expensive for complex games (Nash, 1951; Nash, 1953; Binmore et al., 1986).

2.3.1 Prisoner's Dilemma

The origin of Prisoner's Dilemma is generally attributed to two scientists (Merrill Flood, and Melvin Dresher) who produced a "simple, baffling game" (Poundstone, 1993, p. 8); which, Poundstone relates, their colleague Albert Tucker wrote up in a short memo as *A Two-Person Dilemma*. It is perhaps an apocryphal story. The account by Kuhn et al. (1996) of the origin of *A Two-Person Dilemma* omits Flood and Dresher and attributes the game to Tucker alone. However, Kuhn et al. (1996) do acknowledge the pair's experimental work with Prisoner's Dilemma, which was published in 1959.

Nevertheless, the basic formulation of the Prisoner's Dilemma is given as a situation where two prisoners, charged "with a joint violation of the law, are held separately by the police" (Tucker, 1950; Tucker, 1983, p. 228) and are each given the choice to either confess or remain silent. Each are told that if one confesses and the other does not, the former will receive a reward of one unit and the other will suffer a penalty of two units; they are also told that if both confess each will receive a penalty of one unit. Both prisoners have the expectation that if neither confesses, they will go free; neither receiving a fine or reward. This scenario translates to the matrix depicted in **Figure 2.1a)**.

Prisoner's Dilemma evolved to have varying payoffs and semantics: *defect* (or *hawk*) for *confess*; and *cooperate (*or *dove*) for *silence* (Binmore, 2007). A typical variation with cooperate-defect nomenclature is shown in **Figure 2.1b)**. The exact values found in studies of Prisoner's Dilemma vary substantially and have been a subject of research in themselves. For example, Rapoport and Chammah (1965) examined a variety of linear transformations to the Prisoner's Dilemma payoff values; similarly, Crandall and Goodrich (2005) work with a variation of Stag Hunt that has both negative and positive payoffs (like Prisoner's Dilemma, Stag Hunt is a 'dilemma' game, and is discussed in §2.3.3.4).

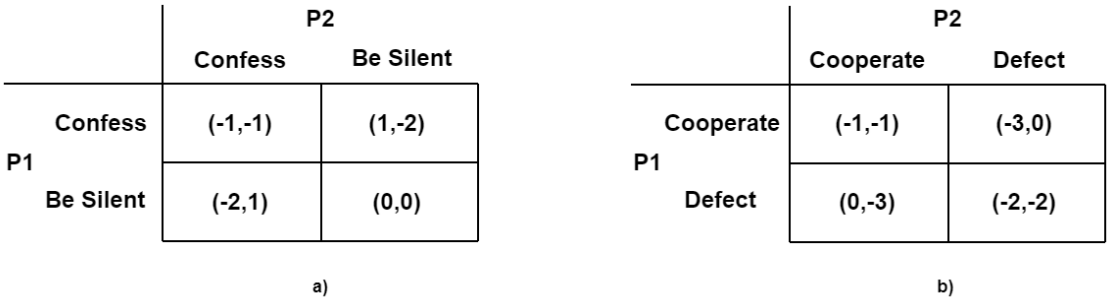


Figure 2.1: Two Prisoner's Dilemma Normal form matrices. **a)** Payoff matrix for the *Two-Person Dilemma* as given in Tucker's (1950) memo. **b)** Payoff matrix for a common variant of the canonical Prisoner's Dilemma, with negative payoffs. **P1** and **P2** refer to participants one and two. The first digit in the tuple in each cell corresponds to the payoff for participant one, the second digit to participant two.

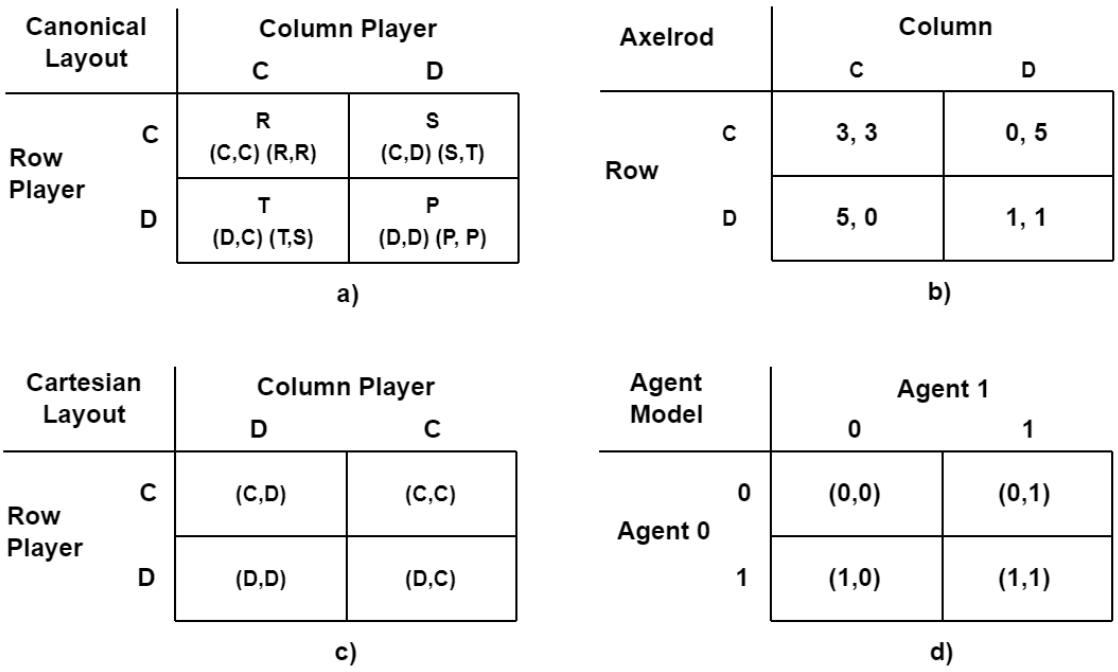


Figure 2.2: Generalised forms of the Prisoner's Dilemma. In **a)** the labels **R**, **S**, **T**, **P** indicate the game outcome in the top row of each cell, obtained from the joint play of actions **C** (cooperate) or **D** (defect) as a shared outcome **CC** (**R**, **R**), **CD** (**R**, **T**), **DC** (**T**, **S**), **DD** (**P**, **P**). In **b)**, the layout and payoffs used in Axelrod tournaments (Axelrod & Hamilton, 1981; Hofstadter, 1983; Axelrod, 1984). In **c)**; Cartesian payoff vector ordering (**CD**, **CC**, **DD**, **DC**) by Robinson and Goforth (2005), adopted by Bruns (2010), and continued with by Crandall et al. (2018a; 2018b). In **d)** the normal form matrix, sans semantic labels, generalised to indicate player and game outcome by agent index, and cell index, respectively. Joint actions **(0,0)**, **(0,1)**, **(1,0)**, **(1,1)** correspond to semantic outcomes **CC**, **CD**, **DC**, **DD**, and likewise to shared outcomes **R**, **S**, **T**, **P**.

The Prisoner's Dilemma can be abstracted to a generalised form, where ***T*** is *Temptation* (to defect), ***R*** is *Reward* (for cooperating), ***P*** is *Punishment* (penalty for defecting), and ***S*** is *Sucker* (result of cooperating when the other defects) as shown in **Figure 2.2a)**. Prisoner's Dilemma (and other social dilemmas) can be defined by inequalities over these generalised outcomes (see §2.3.2). The novel generalised normal-form matrix shown in **Figure 2.2d)** elides the familiar semantic labels from both actions and outcomes, utilising cell index to locate and identify the game outcome. The generalised normal-form is a novel representation useful for programmatic and descriptive methods as it removes semantics from action labels, abstracting the action-space to two generic options (*cooperate* ⇒ 0, *defect* ⇒ 1). The generalised normal-form is used in Chapters Four, Five, and Six.

2.3.1.1 Axelrod's Single-Model Tournament

In 1979, Axelrod invited game theorists to submit strategies for playing Prisoner's Dilemma in a computerised tournament program with the intent to pair every strategy against every other strategy in an iterated match of 200 rounds (Hofstadter, 1983). Axelrod (1980a, 1980b) reformulated the Prisoner's Dilemma to have non-negative payoffs, as shown in **Figure 2.2b)**.

The goal of the competition was to determine which strategy would score the highest total payoff. The invitation prescribed rules for each submission:

- That each player would be able to respond to the previous action of another player (encoded as $\boldsymbol{C}$, cooperate, or $\boldsymbol{D}$, defect);
- That each player could remember interactions;
- Each player should respond with an action of either $\boldsymbol{C}$ or $\boldsymbol{D}$;
- The response need not be deterministic.

The result of this tournament of fifteen[30] strategies was a win for the simplest submitted, Rapoport's *Tit-for-Tat* (TFT). *TFT* cooperates on the first move, then, for every subsequent move it responds with the action most recently played *against it*. Axelrod subsequently ran entries from the first tournament in simulated replays and found that a variant of *TFT* named *Tit-for-Two-Tats* (TFTT) showed surprising success, so much so it would have won in a hypothetical 'replay' of the first tournament (Hofstadter, 1983). The approach of this variant was to be more forgiving, thus *TFTT* would tolerate two defections before replying in kind.

In Axelrod's second tournament, however, simple reciprocity was again the most successful strategy with *TFT* gaining the highest aggregate payoff (Axelrod, 1980b). There were sixty-three (63) entries submitted to the second tournament. The results of the first tournament were made available to all competitors, including the results of Axelrod's replay simulations. As Hofstadter conjectures, given the success of *TFTT* in the replays, if it was entered into the second tournament it might be expected to win. However, the variant of *TFTT* submitted by Smith ranked 24th. *TFT* again placed first. Hofstadter (1983) accounts for *TFTT*'s lack of success by recognising that the second tournament had changed—the new community of entries created different opportunities and potential for the evolution of cooperation. Hofstadter summarised the second tournament by drawing the conclusion that no strategy could be optimal in all environments; and that *TFT* had "the advantage of being able to get along well with a great variety of strategies" (Hofstadter, 1983, p. 9). Axelrod (1984) came to a number of conclusions after the second tournament and identified four properties that help to make a strategy successful:

- avoidance of unnecessary conflict by cooperating as long as the other player does;
- an ability to be provoked, in the face of an uncalled-for defection by the other;
- forgiveness after responding to a provocation; and
- clarity of behaviour so that the other player can recognize and adapt to the pattern of actions.

Axelrod (1984) concluded that for stable cooperation to emerge there are conditions which must be met:

- someone must be willing to reciprocate a cooperative act with a likewise cooperative act;
- individuals do not have to be rational;
- individuals do not need to communicate;
- individuals do not need to assume trust with other players;
- altruism is not a requirement;
- the environment must indefinitely extend temporally; and,

[30] There were fourteen strategies submitted plus the *Random* strategy (that played **C** or **D** with equal probability) which was added by Axelrod (Rapoport et al., 2015).

- that no central authority is required.

Axelrod (1984) considered that these findings altered the accepted view that the foundations of cooperation are built on trust—that they are in fact built on the durability of relationships.

Rapoport et al. (2015) raised issue with the way the tournaments were run. They observed that the total score obtained by a participant strategy was, firstly, an aggregate of all payoffs in all rounds in every round-robin match; and secondly, this was the sole determinant of a strategy's ultimate rank placement. Under these rules, it was conceivable that a strategy could score consistently low but also win the most matches. Rapoport et al. likened this to a football team being crowned champions at the end of a season on the "number of goals they scored" (2015, p. 2), and on this basis questioned whether the Axelrod tournaments were indeed measuring cooperation. In the first tournament, when ranked by number of wins, *TFT* came last (Rapoport et al., 2015).

Another issue they raise is that *TFT* has not done nearly as well in any subsequent tournaments performed by other researchers as it did in the Axelrod tournaments. Rapoport et al. (2015) highlight the role of the payoff values in Axelrod's Prisoner's Dilemma, shown in **Figure 2.2b)**, as creating conditions for certain behaviours to be promoted over others, with the imputation that the values used by Axelrod assisted *TFT*. To assess their concerns, Rapoport et al. (2015) recreated the first Axelrod tournament (as a two-stage round-robin tournament) and analysed the results on two criteria: total wins and total score. They hypothesised that these two metrics would be positively correlated but report that they found no evidence to support this supposition.

Rapoport et al.'s (2015) contention that the payoffs themselves (i.e., the actual values, their quantum, and their range) can affect the behaviour of strategies (such that the original Axelrod formulation benefited *TFT* in some way) is raised as an observed effect but was not pursued to any further detail. This topic is explored further in this thesis with an examination of behavioural equivalence under isomorphic transformations of a game model, in Chapter Five.

2.3.1.2 Related Work

Maynard Smith (1982) developed computational approaches for the analysis of cooperation and stability in evolutionary game theory models. Evolutionarily stable strategies are defined as a population with an 'uninvadable' finite set of pure strategies (Maynard Smith, J., 1982). To *invade* means a newly introduced strategy becomes dominant in an environment of existing agents with alternate strategies. Yao (1996) defined a pure strategy in evolutionary game theory as a complete strategy set where all possible moves are pre-determined for any situation that may arise. Boyd and Lorberbaum (1987) had earlier shown that there exist *no* finite set of pure strategies in the $\boldsymbol{n}$-person iterated Prisoner's Dilemma that are evolutionarily stable. However, Boyd (1989) later demonstrated that the opposite can hold (the existence of a finite set of pure strategies can occur) if mistakes are allowed; a mistake being defined as when an individual intends to act one way (either $\boldsymbol{C}$, or $\boldsymbol{D}$), but actually acts in the other[31]. Yao

[31] Mechanisms for achieving 'mistakes' vary but usually rely on the application of noise somewhere in an agent's process of receiving input, processing signals, and/or transmitting action-selection.

(1996) then extended this work on two-person games and confirmed Boyd's findings: there exist no finite set of pure strategies in the ***n***-person iterated prisoner's dilemma that are evolutionarily stable, unless mistakes are allowed.

Billard (1996) modelled co-evolutionary dynamics of reward using learning automata in stochastic Prisoner's Dilemma models. Billard was concerned with oscillations in player strategies towards cooperation and coordination in an environment that incorporated delays in information channels, primarily modelled using delay differential equations. The use of delay was intended to model the physical distribution of participants in a real environment, and also to model latency impacts on the effect of actions or decisions in predator-prey interactions, on variables in food supply chains, in blood production systems, and on incentives for restraining polluters. Billard (1996) showed that delayed information significantly affected stability, such that momentary oscillations could become persistent.

Sandholm and Crites (1996) applied reinforcement learning to the iterated Prisoner's Dilemma in order to study how learning agents fare under various conditions against an unknown opponent. The varying conditions they studied included the length of contextual history retained, the type of memory used, and the pattern of exploration schedules, i.e., policies. Sandholm and Crites describe their work as fitting into a lineage of *multi-agent reinforcement learning* (MARL) research from the 1950s to the 1990s. This lineage incorporated the work of Tsetlin (1973), on collective behaviour of learning automata; Barto (1985), and Barto and Anandan's (1985) application of associative learning automata on non-sequential tasks; Barto et al. (1983), Sutton (1988), and Watkins (1989) studied sequential tasks in a single-agent context; Watkins' (1989) work in *MARL*, and the development of *Q-Learning*; Sen et al.'s (1994) work in block-pushing; Tan's (1993) grid-world predator/prey interactions using MARL; Bradtke (1993) who applied MARL to the dynamics of steel beams; and Crites and Barto's (1998) proposal to apply MARL to elevator (lift) dispatch control systems.

Sandholm and Crites identified that almost all of the problems previously studied had agents that received "totally positively correlated payoffs (team problems) or totally negatively correlated payoffs (zero-sum games)" (1996, p. 150). The learning agents that Sandholm and Crites implemented used *Q-Learning*. The memory types employed were either lookup tables using restricted history windows, or recurrent neural networks. Each agent played either against other learning agents, or against a deterministic player that used a pure strategy, such as *TFT*. Overall, Sandholm and Crites (1996) found that agents who had longer history windows, used lookup tables, and had longer exploration schedules performed better than agents that did not use such methods. They also found that learning agents learnt to play optimally against the *TFT* player regardless of *how* the agent's Q-values were stored—however they also observed that when playing against a pure strategy agent the model was effectively a single-agent in that the *TFT* player was an indistinguishable part of the environment, therefore the problem had reduced to a stationary environment. Learning agents that played other learning agents experienced a considerably more complex non-stationary environment, as the opponent agent is also learning and also varying its exploration policy. The results of their experiments with two learners found that convergence proofs of *Q-Learning* no longer held and that due to the non-stationary environment the agents struggled to achieve sustained mutual cooperation (Sandholm & Crites, 1996).

Grim (1996) explored a stochastic, spatial, variant of Prisoner's Dilemma in an evolutionary context using cellular automata strategies to investigate the dynamics of populations. In a spatialised form the Prisoner's Dilemma is a two-dimensional space where "a cell in a grid competes against neighbours and adopts the strategy of a neighbour with a higher score" (1996, p. 10). The use of a spatialised Prisoner's Dilemma provided a model that reduces to a matrix game while allowing a larger action-space. Among the strategies were several variants of *TFT* including *generous TFT* (gTFT). Where *TFT* always defects in response to defection, *gTFT* would forgive 1/3 of the time and cooperate in response to a defection. However, this latter strategy was not favoured in the next generation. In contrast, a strategy that was twice as generous in forgiving as *gTFT*, but also a third as likely to defect in response to a cooperative act, performed better overall, and better than *TFT*, under stochastic environment conditions.

Macy and Flache approached evolutionary computational models with the view that "evolution operates on the global distribution of strategies within a given population" (2002, p. 7235). They viewed that learning, in contrast, operates on a local distribution of strategy interaction between individual members of the population. Their focus was on agents learning emergent strategies, which they called *cognitive game theory*. Macy and Flache (2002) extended Rapoport and Chammah's (1965) work of analysing social dilemmas as Markov models using the Bush-Mosteller learning model (Bush & Mosteller, 1951). Macy and Flache (2002) added neural networks to their agents but found difficulty with an exponential increase in the complexity of coordination as the strategy space grew. The Bush-Mosteller learning model extended stochastic learning automata by using reward to update action probabilities (Sutton and Barto, 2018). Macy and Flache also found that agents could sometimes find their way from being in situations characterised by the dynamic of the Prisoner's Dilemma into more cooperative contexts via a random walk—a process they called *stochastic collusion* (2002).

Work in *adaptive game theory* by Erev and Roth (1998) is conceptually adjacent to cognitive game theory in using reinforcement learning and Bush-Mosteller models. Adaptive game theory relaxes the rationality assumption in that there is no expectation that the participants consider all possible strategies, and further, to not assume that the participants were "expected-utility maximizers" (1998, p. 875). Macy and Flache reflected on this in comparison with their own use of Bush-Mosteller models, identifying that both differ from analytical game-theoretic solutions by seeking the existence of a cooperative equilibrium that is *not* equivalent to a Nash equilibrium. Macy and Flache (2002) question whether this type of solution concept would generalise away from the Prisoner's Dilemma to other domains, as earlier positive results were possibly artefacts, or due to assumptions in the model.

Several lines of work in the area of *intention recognition* (see §1.2.1.2) have drawn on the contextual advantages that game theory models provide via their reductive environments that are highly constrained. Hilbe et al. (2017) and Schmid et al. (2021) both investigate models of reciprocity in an evolutionary paradigm, seeking normative principles for application to more effective planning.

An example of a modern, alternate representation to normal- and extensive-form game theory models is found in the work of Fang (2016), who has developed a range of models, techniques and approaches and applied them to practical issues, such as the monitoring and protection of wildlife in danger of poaching (Fang et al., 2017; Bondi et al., 2019),

and to the effective scheduling of coast guard patrols (Fang et al. 2013). One novel method in Fang's work is *compact representation*, a technique that can, among other uses, cast problems as network-flow graphs, where the probabilities of actions are the weights on edges; and each identified target is represented by a node. This formulation produces a graph model for analysing deception, enabling inference over the transition matrix and the construction of new, optimised, patrol schedules (Fang et al., 2013). This approach also mixes one participant's discrete strategy-space with the other participant's continuous strategy-space (Fang et al., 2013; Xu et al., 2020; Wang et al., 2019). Fang works with *Deep Reinforcement Learning* (DRL) extensively and places her recent work in AI as a social good (Shi et al., 2020). Fang's ongoing research is described as "integrating machine learning with game theory" (2021, p. 23) in the areas of artificial intelligence and multi-agent systems.

Leibo et al. (2017) explored iterated $\mathbf{2} \times \mathbf{2}$ games using the deep reinforcement learning algorithm *Deep-Q-Network* (DQN). Leibo et al. studied the Wolfpack and Gathering games, aiming to take a descriptivist view in asking the question "what social effects emerge when each agent uses a particular learning rule?" (2017, p. 3) while also attempting to measure the proportion of states that a game had been in where the dynamic corresponded to one of a small set of social dilemmas. Each agent is modelled independently of the other and so regards the other as part of the environment, which they present as a type of *bounded rationality* in that agents do not reason, or attempt to model, each other. In their evaluations of their experimental runs, they attempted to measure the correlation of dynamics in the games Wolfpack and Gathering, to the dynamics of the social dilemmas Prisoner's Dilemma, Stag Hunt, and Chicken. By identifying a relevant behavioural characteristic in a game (for example, lone-wolf behaviour in Wolfpack) as (in this example) a *defection dynamic*, they set out to quantify the observation of the dynamic as a metric. To achieve this, they aggregated the observations such that the values obtained combined to form payoff matrices corresponding to social dilemma game models through regressing the observations to the game value (Shapley, 1952). Some social dilemma games indeed emerged as components of the larger Wolfpack or Gathering games. However, it is unclear from Leibo et al.'s presentation what the ratio of *identified dilemma games* to *non-identified games* was in their experiments, with indications that many aggregated values did not map (i.e., regress to a game value) to a social dilemma.

This work by Leibo et al. (2017) echoes much earlier work in a series of studies by Rapoport and Chammah (1965), which investigated human subjects engaged in playing Prisoner's Dilemma. Rapoport and Chammah were exploring a fundamental proposition: that a particular dynamic in a social situation (i.e., a game model) would either *induce* or *require* a pattern of play from participants. Rapoport and Chammah applied a variety of methods (Markov Chain Models, Equilibrium Models, Stochastic Learning Models, and Classical Dynamic Models) to profiling patterns in the participants' behaviour in Prisoner's Dilemma, however they were not ultimately successful in their endeavour to identify a strategic dynamic 'signature' in patterns of human play, and it appears the question remains open.

Rapoport and Chammah's (1965) theory that a dynamic, encoded into a model, would be identifiable by an analysis of the behaviour of participants, is investigated in Chapter Six.

2.3.2 Social Dilemmas

Prisoner's Dilemma is recognised as a *social dilemma* (Macy & Flache, 2002; Ashlock & Kim, 2008). A social dilemma is a construct that distils an archetypal decision-making situation to a distinct choice between two options. In colloquial terms we can think of the consideration of this choice as 'on the one hand, *this*; and on the other hand, *that*'. Social dilemmas have the characteristic that an inferior outcome is often the dominant strategy for both participants (Robinson & Goforth, 2005). With any two participants, each with two options and a preference between them, the dynamic can be captured mathematically in a simple matrix game representation, such as the Prisoner's Dilemma. But there are other social dilemmas that can also be represented in matrix form.

The values in the matrix can map directly to a unit of utility, i.e., monetary amounts, or units of a commodity; or alternatively the values may be a scalar that represents an *ordinal preference*, such that a larger value indicates a preference for that option over a preference with a smaller value. Preferences indicated by ordinal values generally lose any meaning associated with a countable difference *between* the values, so that the strict scalar inequality $\boldsymbol{x > y}$ becomes the strict preference $\boldsymbol{x \succ y}$, regardless of the value $\boldsymbol{|y - x|}$. Thus, ties in preferences are not allowed (i.e., the 'strictly' in 'strictly ordinal').

But not all matrix games are social dilemmas. To be regarded as such they must satisfy a set of inequalities defined over the values obtained from the four game outcomes $\boldsymbol{T}$, $\boldsymbol{R}$, $\boldsymbol{P}$, and $\boldsymbol{S}$[32] that are assigned as payoffs for each outcome, for each participant, from the participants perspective. **Figure 2.3** shows the semantic outcome labelling notation for each cell of a $\mathbf{2 \times 2}$ matrix.

		Column	
		C	D
Row	C	R (C,C) (R,R)	S (C,D) (S,T)
	D	T (D,C) (T,S)	P (D,D) (P, P)

Figure 2.3: Semantic labels attributed to outcomes in the Prisoner's Dilemma. **R** (Reward), **S** (Sucker), **T** (Temptation), and **P** (Punishment) single-letter labels are outcomes from the perspective of the Row player. Labels in parentheses are the (Row, Column) ordered pair where **D** signifies Defect and **C** signifies Cooperate.

The notation for the inequalities below is derived from the forms used by Macy and Flache (2002), Ashlock and Kim (2008), and Leibo et al. (2017):

$$\boldsymbol{R > P} \quad (1)$$

$$\boldsymbol{R > S} \quad (2)$$

$$\boldsymbol{2R > T + S} \quad (3)$$

$$\boldsymbol{T > R} \quad (4.1)$$

$$\boldsymbol{P > S} \quad (4.2)$$

[32] Where $\boldsymbol{T}$ stands for Temptation, $\boldsymbol{R}$ for Reward, $\boldsymbol{P}$ for Punishment and $\boldsymbol{S}$ for Sucker. See §2.3.1.

(1) Reward has a higher payoff than punishment; players prefer *mutual cooperation* over mutual defection;

(2) Mutual reward payoff is greater than unilateral defection; both players prefer *mutual cooperation* to the alternative of one being exploited;

(3) Mutual reward payoff is greater than combined payoff of unilateral defection and exploitation; ensures *mutual cooperation* where probability of either outcome is equal;[33]

(4.1) Both players prefer to *defect* over choosing to cooperate; unilateral exploitation is preferred over mutual cooperation, characterised as *greed*;

(4.2) or *fear*: both players prefer to defect rather than being unilaterally exploited.

Macy and Flache (2002) defined social dilemmas as mixed-motive two-person games where each participant has a choice between two options—to cooperate, or defect. Macy and Flache refer to each participant's choice at any given decision point as being made between the interests of the collective on the one hand and self-interest on the other. In the case of $\mathbf{2 \times 2}$ games, the welfare of both players is synonymous with the 'collective', or group, interest. Like Prisoner's Dilemma, other social dilemmas have descriptive names in common parlance; two of these games are Chicken and Stag Hunt. Chicken models the 'who blinks first!?' dynamic, where participants have a choice to crash headlong into each other, or back down, i.e., swerve, or back down, as a response to a confrontational situation. Stag Hunt models the dynamics of cooperative hunting. Skyrms (2001; 2003) wrote extensively on this game, exploring its role in evolution as well as its dynamics as applied to social situations.

The difference between these two game models and Prisoner's Dilemma is seen in the ordering of the inequalities. For Prisoner's Dilemma they are ordered $\boldsymbol{T \succ R \succ P \succ S}$; for Chicken they are ordered $\boldsymbol{T \succ R \succ S \succ P}$ and in Stag Hunt they are ordered $\boldsymbol{R \succ T \succ P \succ S}$[34]. In Chicken the $\boldsymbol{S}$ outcome is preferred over the $\boldsymbol{P}$ outcome, indicating that the participants would prefer to be exploited by the other than for them to both defect; i.e., one participant would prefer to swerve and allow the other to carry straight-on (and so, 'win'), rather than they both carry straight-on and crash headlong.

In Stag Hunt the preferences are ordered such that the reward outcome is preferable to the temptation outcome, which Macy and Flache (2002) explain as fear being dominant over greed. For Chicken, this is inverted: the greed is dominant over fear. Prisoner's Dilemma is the case where both fear *and* greed are in-play. Leibo et al. (2017) applied the social dilemma inequalities to the Markov Game formalism introduced in §2.2.2. Where $\boldsymbol{\pi^C}$ and $\boldsymbol{\pi^D}$ are cooperating and defecting policies, and the payoff is given as $\boldsymbol{V_i}$, the inequalities can be obtained by:

$$\boldsymbol{R(s) := V_1\ \pi^{C}, \pi^{C}\ (s)\ =\ V_2\ \pi^{C}, \pi^{C}\ (s)}$$

$$\boldsymbol{P(s) := V_1\ \pi^{D}, \pi^{D}\ (s)\ =\ V_2\ \pi^{D}, \pi^{D}\ (s)}$$

[33] Inequality (3) prevents participants alternating between $\boldsymbol{C}$ and $\boldsymbol{D}$ actions (from turn to turn) in the iterated game in preference to adopting sustained mutual cooperation (Ashlock and Kim, 2008; Han et al., 2011a).

[34] Note these *preference* relations are ordered strictly (as opposed to an ordering allowing ties: $\boldsymbol{T \succcurlyeq R \succcurlyeq P \succcurlyeq S}$). For scalar relations, the inequalities are, for Prisoner's Dilemma $\boldsymbol{T > R > P > S}$; for Chicken $\boldsymbol{T > R > S > P}$; and Stag Hunt $\boldsymbol{R > T > P > S}$.

$$\boldsymbol{S}(\boldsymbol{s}) := \boldsymbol{V}_1\ \boldsymbol{\pi}^{\mathrm{C}}, \boldsymbol{\pi}^{\mathrm{D}}\ (\boldsymbol{s}) \ = \ \boldsymbol{V}_2\ \boldsymbol{\pi}^{\mathrm{D}}, \boldsymbol{\pi}^{\mathrm{C}}\ (\boldsymbol{s})$$

$$\boldsymbol{T}(\boldsymbol{s}) := \boldsymbol{V}_1\ \boldsymbol{\pi}^{\mathrm{D}}, \boldsymbol{\pi}^{\mathrm{C}}\ (\boldsymbol{s}) \ = \ \boldsymbol{V}_2\ \boldsymbol{\pi}^{\mathrm{C}}, \boldsymbol{\pi}^{\mathrm{D}}\ (\boldsymbol{s}).$$

Robinson and Goforth (2005) take a more liberal view of what defines a social dilemma by including any game model that possesses an inferior outcome, or outcomes, strongly or weakly dominant over other available strategy choices.

2.3.3 Taxonomies, Typologies, and Topologies

This section first discusses the typology initially developed by Rapoport and Guyer (1966), the taxonomy developed by Brams (1993), and then several other classification systems including those developed by Walliser (1988), and Kilgour and Fraser (1988). These classification systems of $\mathbf{2 \times 2}$ games lead to discussion of an explicitly group-theoretic topology defined by Robinson and Goforth (2005), and in addition, a variation of the Robinson and Goforth topology formulated by Perlo-Freeman (2006).

The extension of game theory modelling into alternative games is founded on the realisation that as Prisoner's Dilemma is an example of a common and identifiable social dynamic; so other social dynamics may also be modelled in similar fashion (Axelrod, 1984; Boyd, 1988; Kümmerli et al., 2007; Leibo et al., 2017; Rapoport & Chammah, 1965). A social dynamic *does not* have to be a strictly defined dilemma to have a valid representation as a $\mathbf{2 \times 2}$ matrix. The $\mathbf{2 \times 2}$ game model that is derived by calculating the preferences over two options that the participants have in an interaction is built by combining each participant's preferences for each outcome of the game. For a $\mathbf{2 \times 2}$ matrix this means there are four outcomes possible. The preferences so derived can be represented by scalar reward values, or as ordinal preferences indicated by a rank ordering[35]. Not all derived game models have as complex a set of dynamics as does Prisoner's Dilemma. For example, a dynamic can be modelled where the dominant strategy is equally rewarding for both participants, as is the case for the two games shown in **Figure 2.4**. In these game representations the names **g316** and **g311** are the Robinson and Goforth (2005) identifiers, the construction and meaning of which is given in §2.3.3.4.

g316

		Column C	Column D
Row	C	4, 4	2, 2
	D	3, 3	1, 1

a)

g311

		Column C	Column D
Row	C	4, 4	2, 3
	D	3, 2	1, 1

b)

Figure 2.4: Ordinal representations of two game models. In **a)** game model **g316**, and in **b)** game model **g311,** in the Robinson and Goforth topology (2005). Note that the placement of **g311** to the right of **g316** also depicts their topological 'neighbour' relationship.

[35] Regarding the properties of the ordinal numeration rank schema applied in this thesis is that rank is an inverted queue—preference four (4) is preferred to preference three (3), transitively down to the floor of the range of the preference schema which in $\mathbf{2 \times 2}$ gameforms is equal to one (1). Further, for each participant all four values must be locatable in the gameform; and for any one participant, no values may be duplicated.

It is unlikely that any participant, except through stochastic exploration, would choose to deviate from a dominant strategy (Colman, 1998; Busch & Reinhardt, 1993) in what Rapoport and Guyer refer to as "non-conflict" game (1966, p. 18). All the same, given that these are valid game theory models, and also given that other interesting dynamics (e.g., the social dilemmas) can also be modelled as **2 × 2** game models, researchers have sought to categorise "trivial" (Rapoport & Guyer, 1966, p. 21) games along with their more interesting brethren to understand how they relate to each other. A key step in understanding these relations is to group sets of games by their affinity to each other, as can be readily seen in two variants of the Alibi game, **g412** and **g413**, which have very similar ordinal game models as shown in **Figure 2.5** (Robinson & Goforth, 2005).

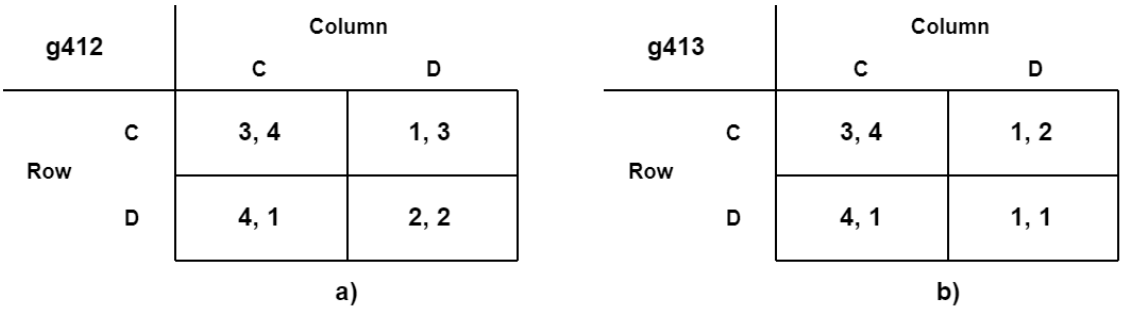


Figure 2.5: Ordinal representations of Robinson and Goforth (2005) games **g412** and **g413**.

2.3.3.1 A Typology of 2 × 2 Games

Rapoport and Guyer (1966) divided the set of 78 strictly ordinal **2 × 2** games into three classes. The first class held the 21 games in which both participants have a dominant strategy, the second class comprised the 36 games where only one participant has a dominant strategy, and the third class consisted of the remaining 21 games that offer no dominant strategy for either participant. Thus, each game is classified according to its *strategic type*, thus forming a typology. Each game in the set of 78 games is transmutable to an equivalent game through interchanging the rows, the columns, or the players. However, none of the 78 games is able to be transmuted into any one of the other 77 games. The third class of games is further subdivided by number of equilibrium outcomes (two, or none). In addition to these properties, Rapoport and Guyer (1966) further categorised games by the number and type of equilibria that each game possesses. This results in ten categories whose members are drawn from across the three principal classes. The only games that were not placed into categories were those in the third class that have no dominant strategy. The taxonomy was further developed by Rapoport, Guyer and Gordon (1976), writing at a time when all previous experimental work was possible only with human subjects. Due to this, they spend some time discussing the effects of normalising payoffs on test subject responses and on other factors that influence research design, subject behaviour, and interpretation of results. Rapoport, Guyer and Gordon (1976) stress the need to be careful with definitions. For example, the meaning—and effect on behaviour—of cooperation in Chicken (**g122**) is different from its meaning and effect on behaviour in Prisoner's Dilemma (**g111**); each are shown in **Figure 2.6**, in *order graph* **a)**, **c)**, and matrix **b), d)**, forms.

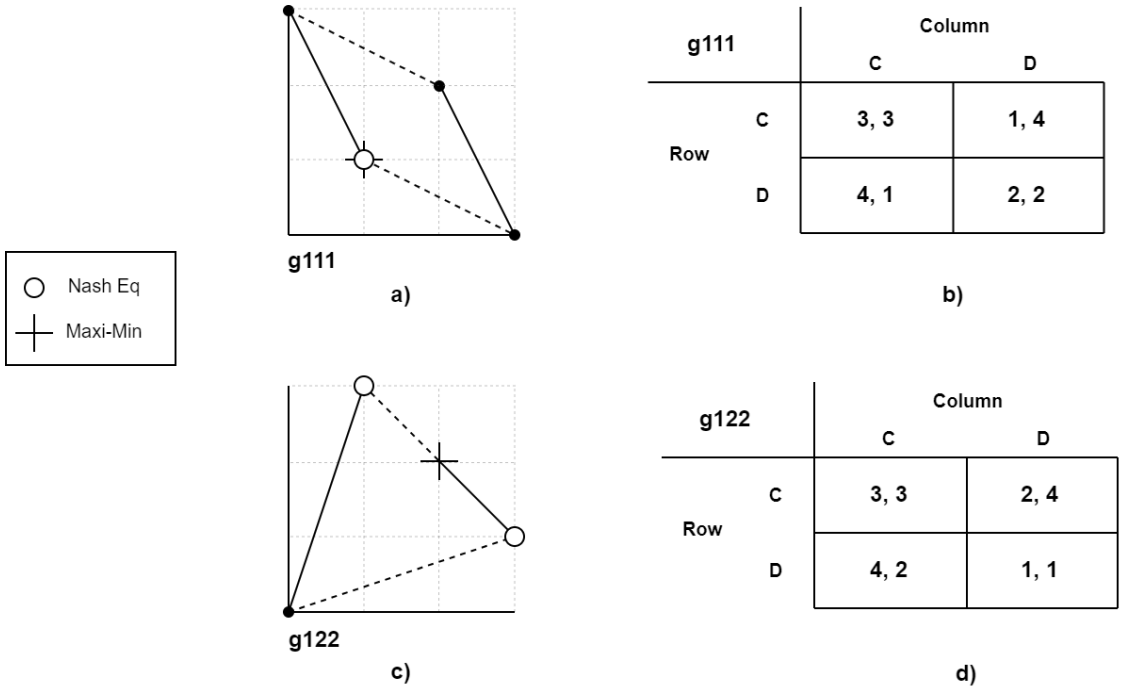


Figure 2.6: Two representations of Prisoner's Dilemma (**g111**) and Chicken (**g122**). The order graph in **a)** and the matrix in **b)** are equivalent representations of **g111**. As are **c)** and **d)** for game model **g122**. Solid lines between two outcomes signifies the *inducement correspondence* for the row participant, the dashed lines signify the column participants' inducement correspondence. See §2.3.3.4. Note the payoff layouts are in canonical, not cartesian, ordering.

Robinson and Goforth's (2005) *order graph* representation visually depicts payoffs for outcomes for both participant's joint actions. In Chicken, the cooperation dynamic has two Nash equilibria (NE) plus a "non-equilibrium outcome" (Rapoport & Guyer, 1966, p. 24) while Prisoner's Dilemma has a "strongly stable deficient equilibrium" (1966, p. 26). Robinson and Goforth specify Chicken as having two NEs and a maxi-min saddle point, at distinct locations, and Prisoner's Dilemma as having a single NE at the same location as the maxi-min point (see §2.3 for a fuller description of maxi-min saddle points).

Many of these examples reveal that cooperation is both a "perfectly measurable statistic" (Rapoport, Guyer and Gordon, 1976, p.105) and also a concept open to context and interpretation—per game, per strategy, per participant; a dichotomy that motivates researchers to find unifying theories of the strategic dynamic relationships found in game models.

2.3.3.2 Brams' Dynamic Game Theory

Brams (1993) developed a taxonomy that arose from the modelling of political, social, and theological scenarios, contextualising his work as *dynamic game theory*. Initially von Neumann and Morgenstern (1944, p. 44) had presented their theory as a static view of the larger domain of interest: "our theory is thoroughly static. A dynamic theory would ... be more complete and ... preferable." But this was not yet possible, von Neumann and Morgenstern believed, because a dynamic theory could not be attempted without a *complete static theory*.

For Brams (1993), strictures of game theory did not exclude the free incorporation of dynamic elements. Brams' *dynamic game theory* combines aspects of the normal-form and extensive-form treatments, allows participants to look-ahead, treats payoffs as ordinal (ordered by preference rather than scalar value), allows imbalance in the power of participants with respect to each other, and allows participants to have different levels of knowledge about the game being played to model the use of deception.

Brams (1994) gave four rules for modelling dynamic games, plus two constraints (Brams referred to these constraints as *rationality rules*): *Rule 1* defines the initial state of a game. The initial state is equivalent with an *outcome*, i.e., one of the cells in the payoff matrix. *Rule 2* says that participant 1 can then choose to change strategy, which would give a new outcome. *Rule 3* says that participant 2 can then choose to change strategy, again giving a different outcome. Unlike traditional game theory the moves do not occur at the same time, but sequentially. *Rule 4* gives the condition for ending the game—this is when no participant wishes to change strategy again, which leaves the current state as the final state. The constraint rules limit the ability to change strategy: *Rule 5* instructs that a participant will not make a move unless it leads to a preferred outcome; while the final rule, *Rule 6* allows either participant to move first, or to wait and move second, which allows either participant to perhaps gauge the other's strategy before moving themselves. The first-mover rule (*Rule 6*) is justified for inclusion, Brams says, as it should be implicit in a generalisation of standard game theory that moves are sequential.

Brams' (1993) aim with dynamic game theory was to construct a model that he felt was more plausible (than standard game theory); in that it would more accurately capture what people think about, and then actually do, in strategic decision-making situations.

2.3.3.3 Preferential Taxonomies

Walliser (1988) constructed a taxonomy of $\mathbf{2 \times 2}$ games using three criteria, yielding 144 games. The first criterion was the existence of, and number of, Nash Equilibria; the second criterion was the presence of Stackelberg equilibria; defined by Walliser as a solution such that one participant will always play their best response to the other's best action. The third criterion was on the existence and relationship of Pareto-optimality to Stackelberg equilibria. By using these criteria, Walliser's taxonomy generalised the dynamics of abstract games. For example, Walliser groups twelve *symmetric* games by the dynamic that the participants can exchange roles and have their preferences unchanged, stating that such games are "invariant to role exchange"; and similarly, twelve *antisymmetric* games "which assume that the players have opposite preferences when they exchange their roles" (1988, p. 164).

Fraser and Kilgour (1986) introduced an enumeration of all 726 distinct $\mathbf{2 \times 2}$ games obtained by considering games where "one or both players may have equal preferences for two or more outcomes" (1986, p. 99)[36]. Their taxonomy is an extrapolation from Rapoport and Guyer's (1966) formulation of the set of 78 strict ordinal $\mathbf{2 \times 2}$ games. Fraser and Kilgour (1986) give a number of reasons for enumerating all 726 games. Firstly, modelling a real-world situation in a strictly ordinal taxonomy enforces a choice

[36] This is a relaxation of the 'strictly' in 'strictly ordinal', so that there can be duplicate preferences in the set of game models.

between one preference or the other, when real-world situations are often resolved with tied preferences[37]). Secondly, that this expanded scope for modelling behaviour increased the sensitivity of analysis of conflict situations. Thirdly, the greater number of similar games allows comparison of dynamics more easily (but, they admit, complicates their differentiation), and fourth, that games outside of the ones identified with semantic labels (i.e., the well-known colloquial tags such as *Stag Hunt*, *Chicken*, etc) were not well-known nor easily accessible, so presenting them all in an extensive taxonomy would enable easier analysis through greater access and visibility.

Underlying these considerations Kilgour and Fraser (1988) had a practical and parsimonious motivation, aiming to sort and categorise game models by their properties rather than by theories of strategy. As such, their taxonomy is intended to be invariant to theories of strategic choice or the specific rules under which a game may be played.

2.3.3.4 A 2 × 2 Topology

The Robinson and Goforth (2005) topology organises the set of strictly ordinal $\mathbf{2 \times 2}$ game models by *adjacency of minimal preference pair difference* between any two of the 144 game models (2005). Robinson and Goforth state that their approach shifts classification of $\mathbf{2 \times 2}$ games from the "space of strategies to the space of payoffs" (2005, p. 9). Each game model in the topology is a $\mathbf{2 \times 2}$ matrix that specifies the preferences that each participant has for the outcomes of a single play of the game. Any one of the four possible outcomes is the joint result of two actions—one from each participant, performed at the same time, and without knowledge of the other's action. Their combined actions (the outcome) map to one of the four cells in the $\mathbf{2 \times 2}$ matrix. When the contents of the cells are given as a pair of preference values in the range [1–4], with 4 being the most preferable and 1 the least, the game model is *ordinal*. A *strictly* ordinal matrix contains each of the values 1, 2, 3, and 4 exactly once only, for each participant, i.e., no preference for an outcome can be indifferent to another, so must be unique. **Figure 2.7** shows **a)** scalar representation of Prisoner's Dilemma with canonical values, and **b)** ordinal representation of the equivalent Robinson and Goforth game model, **g111**, with preferences in the range [1–4].

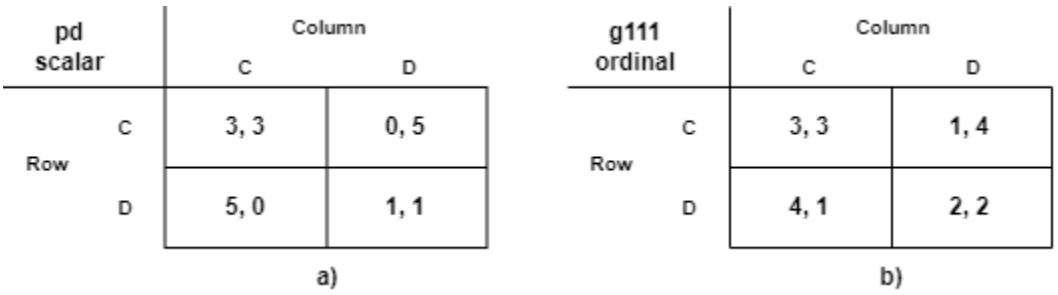


Figure 2.7: Two representations of Prisoner's Dilemma: **a)** scalar, and **b)** ordinal.

To individually identify the 144 game models, Robinson and Goforth (2005) use an indexing system, shown in **Figure 2.8**. The index is made up of the letter $\boldsymbol{g}$ followed by three indices, hence the general form is $\boldsymbol{glrc}$. The first indice identifies the layer $\boldsymbol{l} \in \mathbf{\{1,}$

[37] Literally *six-of-one, half-a-dozen-of-the-other;* Kilgour and Fraser (1988) wanted to capture *indifference* in their models, by allowing games where participants have equal preference between actions.

2, 3, 4}. The remaining two indices represent $\boldsymbol{r}$ (the row index), and $\boldsymbol{c}$ (the column index), respectively, where $\boldsymbol{r}$, $\boldsymbol{c}$ $\in$ **{1, 2, 3, 4, 5, 6}**. The Prisoner's Dilemma game model is identified by the label ***g111***.

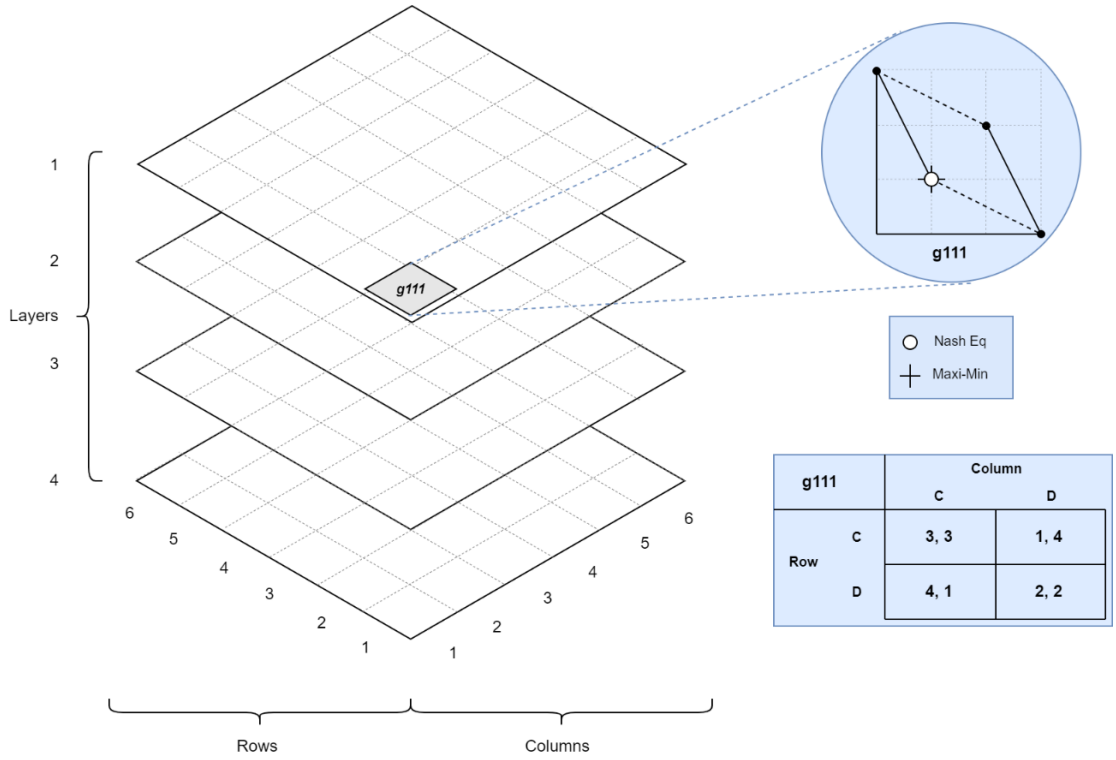


Figure 2.8: $\mathbf{2 \times 2}$ topology indexing schematic. Layer, row, and column indexing is shown. Prisoner's Dilemma's position is highlighted. Layer stack image is adapted from Fig 2.8, *The Topology of the 2x2 Games, A New Periodic Table* (Robinson and Goforth, 2005).

The topology arranges the 144 game models in four layers as shown in **Figure 2.9**. Each layer is composed of thirty-six games. Each game in a layer has the same pair of highest-ranked preference values (Robinson & Goforth, 2005; Perlo-Freeman, 2006). Each pair represents a game outcome and is a tuple formed from the joint action of both participants. Each layer can be drawn as a torus of 37 holes, with Euler number of 0 (Robinson & Goforth, 2005)[38]. Robinson and Goforth describe the mechanism of *preference swaps* between adjacent game models as a property of the minimal necessary condition to change from one game to another. Consisting of only strictly ordinal payoff values precludes modelling game models with tied preferences, however in later work Robinson, Goforth, and Cargill (2007) extend the topological treatment of game models to the set of all *non*-strict ordinal games; i.e., games with tied preference values, adding to the earlier work of Fraser and Kilgour (1986).

A *Robinson-Goforth Space* (RGS) is a graph representation, $\boldsymbol{G_{144}}$, of the dual of the Robinson and Goforth (2005) topology. The *RGS* is examined in more detail in Chapter Three, where the graph $\boldsymbol{G} \subseteq \boldsymbol{G_{144}}$ is defined. The experiments detailed in Chapters Four, Five, and Six use the graph $\boldsymbol{G}$.

[38] The layers are also connected by 'pipes' and 'hotspots' to form a single nested 37-holed torus with Euler number of -72 (Robinson & Goforth, 2005; Perlo-Freeman, 2006).

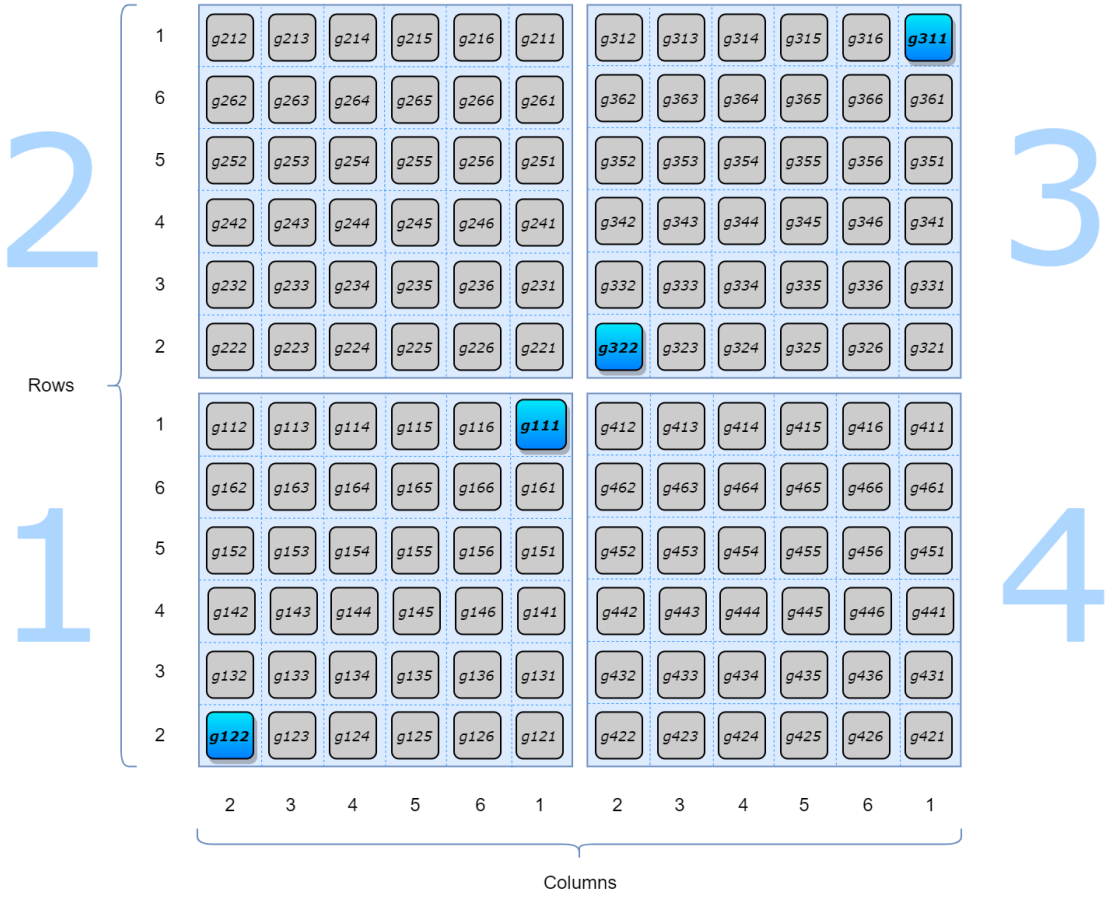


Figure 2.9: Layers and games in the Robinson and Goforth (2005) topology. All game models are shown. The four highlighted games are **g111** (Prisoner's Dilemma), **g122** (Chicken), **g322** (Stag Hunt), and **g311** (Coordination). Layer IDs provide the first digit of each game model name; then row, then column. Figure is adapted from Fig 9.8, *The Topology of the 2x2 Games, A New Periodic Table* (Robinson and Goforth, 2005).

2.3.3.5 Further Topological Treatments

The Perlo-Freeman (2006) topology extends the Robinson and Goforth (2005) topology by selecting a subset of the **2 × 2** ordinal games on the basis of whether, in the game, both players have a dominant preference for a strategy they wish the other participant to play. This forms a set of what Perlo-Freeman terms *cooperate-defect* games. This set maintains existing relationships in the Robinson and Goforth topological space. Perlo-Freeman further separates the cooperate-defect games into two: the set of 36 *interchangeable* games and the set of 21 *non-interchangeable* games. The interchangeable games retain identical preference ordering when participants places are swapped, and the non-interchangeable games do not. Perlo-Freeman shows that the set of interchangeable games can be graphed to a torus. In addition, several other subsets of games selected on type(s) of strategy can also be drawn onto toruses (Perlo-Freeman, 2006).

Crandall et al. (2018a; 2018b) assert that it is not feasible to have an algorithm that is guaranteed to converge to an optimal policy when, in advance, the game model, and the type of the opponent, is unknown—motivating their development of a framework for assessing the relative performance of a set of algorithms where each algorithm is measured against others in regard to establishing and maintaining cooperation in a **2 × 2** game space based on the Robinson and Goforth topology. The objectives of the Crandall et al. (2018a) study were to assess the ability of each participant algorithm to achieve

game independence, *partner independence*, and *responsiveness*. Crandall et al. (2018b) define responsiveness as an algorithm's ability to be not just quick at achieving mutual cooperation, but, in addition, effective at *maintaining* mutual cooperation.

By assessing algorithms on these three metrics and running their study over a large set of game models, Crandall et al. (2018a) judge the algorithms on their *overall* performance—as any one algorithm may excel in a specific game-form but perform poorly in another. In short, their study sought to identify the strongest participants, on average, across all experiment instances. The study was conducted on a set of 720 normal-form game models derived from the Robinson and Goforth (2005) topology.

To expand from 144 game models to 720, Crandall et al. (2018b) apply five functions (identity, power-2, power-3, separated, and square-root, each of which is then normalised) to the ordinal preference values to obtain scalar payoffs, as shown in **Table 2.1**.

Table 2.1: Ordinal to scalar transformations applied to generate 720 game models. The set of 720 game models are an expansion of the 144 Robinson and Goforth (2005) game models. Table is reproduced, slightly adapted, from Supplementary Table 1, *Cooperating with Machines Supplementary Material* (Crandall et al., 2018b).

Function	Ordinal Value				Normalised Values			
	1	**2**	**3**	**4**				
Identity	$\mathbf{1}$	2	3	4	0.00	0.333	0.667	1.00
Power-2	$\mathbf{1}^2$	$\mathbf{2}^2$	$\mathbf{3}^2$	$\mathbf{4}^2$	0.00	0.200	0.533	1.00
Power-3	$\mathbf{1}^3$	$\mathbf{2}^3$	$\mathbf{3}^3$	$\mathbf{4}^3$	0.00	0.111	0.413	1.00
Separated	$\frac{1}{2}(\mathbf{1}-1)$	$\frac{1}{2}(\mathbf{2}-1)$	$2\,(\mathbf{3}-1)$	$2\,(\mathbf{4}-1)$	0.00	0.083	0.667	1.00
Square-Root	$\sqrt{1}$	$\sqrt{2}$	$\sqrt{3}$	$\sqrt{4}$	0.00	0.414	0.732	1.00

Crandall et al. (2018a, 2018b) handle the complications in assessing the mutually-cooperative outcome in a given game model, raised by Rapoport, Guyer and Gordon (1976), and discussed previously in reference to the Prisoner's Dilemma and Chicken game models (§2.3.3.1, see also **Figure 2.6**), by determining which outcome in each game model is equivalent to the (pre-calculated) *Nash Bargaining Solution* (Nash, 1950b) of the game model at hand. A Nash Bargaining Solution is determined for a game model as an optimisation problem, under Nash's four axioms (van Damme, 1986; Harsanyi & Selten, 1972):

- Pareto-efficiency,
- Symmetry,
- Invariance to Equal Payoff Representations, and
- Independence of Irrelevant Alternatives,

of the product of each participant's optimal utility that each participant can extract from the game model. While the ability to obtain a Pareto-optimal solution (Qiao et al., 2006) is known, calculating a Nash Bargaining Solution for a game model is potentially computationally expensive (Nash, 1951; Nash, 1953; Binmore et al., 1986).

In other work with the Robinson and Goforth topology, Bruns (2012) found that performing a swap in the values of payoffs in a game model can lead to the conversion of a game from general-sum to zero-sum, can alter Nash Equilibria, and can also alter the number of Equilibria. Brams (1994) commented on why games may be seen to change,

describing a situation where a participant, upon acquiring new information about the current game, may consider that the game has changed, and so effectively the participant would now be playing to a new game model. Changing from one game to another, via a minimal swap of preference values, exploits the proximity of game models in the Robinson and Goforth model. Bruns (2012) confirmed that Brams' (1994) process to convert from the Prisoner's Dilemma game to the Revelation game can be performed in the Robinson and Goforth model with a sequence of two swaps in the lowest-ranked payoffs of the equivalent games, **g111** and **g241**. This sequence is equivalent to traversing a total of four nodes in a graph representation of the topology, from $\boldsymbol{g111} \rightarrow \boldsymbol{g221} \rightarrow \boldsymbol{g231} \rightarrow \boldsymbol{g241}$. The promise of adopting a graph representation of the Robinson and Goforth topology is that sequences of game model states can be traced through the graph to provide discrete trajectories through what is a discrete preference space.

2.4 In Summary

This chapter has provided the theoretical context for the remainder of the thesis. As an investigation into the cooperative dynamics of agent behaviour, using the tools of game theory and computational learning, this thesis has now reviewed a range of methods, techniques, theory, and practice; to identify and implement a methodology to enable the collection of descriptive game-theoretic statistics of agent behaviour.

The principal tool that is being used in this thesis is the topological arrangement of strictly ordinal $\mathbf{2} \times \mathbf{2}$ matrix game models developed by Robinson and Goforth (2005). The topological arrangement of strictly ordinal $\mathbf{2} \times \mathbf{2}$ game models provides a discrete preference space that carries with it inherent semantic relations from the world of mathematical representation to the world of human affairs—the storied application of game models to the analysis of such disparate subjects as evolutionary theory and nuclear war brinksmanship is well-documented (Poundstone, 1993).

Recent advances in the use of game theory in machine learning see new applications and methods being used effectively both in research and in application to the real world (Fang, 2016). Placing cooperation as the central concept under study in this thesis, within a context of machine learning, motivated the narrative review of the development of concepts of cooperation in, predominantly, the biological sciences; this decision was predicated on the belief that building an understanding of *machinic* cooperation is dependent on knowledge of the modalities with which cooperation is already expressed in the world around us. AI, being computational, provides the ability to obtain discrete measures of cooperation, over time. At any single discrete timestep a snapshot of an agent's internal state may be a potential data source for regulation of AI operation, among other applications; for example, as a data source to aid in the visualisation of human and AI interfaces, regardless of whether the AI is embodied (as a robot or an android of some type), distributed in a global network, or localised as a personal peripheral.

A graph dual, $\boldsymbol{G_{144}}$, of a subset of the features in the Robinson and Goforth model is constructed in Chapter Three. A graph $\boldsymbol{G} \subseteq \boldsymbol{G_{144}}$ is used in the experiments presented in Chapters Four, Five, and Six. Experimental domains for the study of cooperation were reviewed in §2.2 and §2.3. The series of experiments presented in this thesis are informed by the design of the Crandall et al. (2018a) tournament and methodology. A Crandall-

style tournament is presented in Chapter Four. As a whole, the ethos for this thesis is informed by an ethological, observational methodology (Tinbergen, 1963; Rahwan et al, 2019). This methodology informs the research design for the experiments presented in Chapters Four, Five, and Six as well as the extraction and analysis of data regarding an entity and its operational state, by the gathering of descriptive game-theoretic statistics.

Chapter Three

Robinson-Goforth Space

We shall not cease from exploration
And the end of all our explaining
Will be to arrive at where we started
And know the place for the first time

—T.S. Eliot[39]

This chapter stands alone to explicitly construct a graph derivation $\boldsymbol{G}$; $\boldsymbol{G} \subseteq \boldsymbol{G_{144}}$ which is used as a model in the remainder of the thesis. $\boldsymbol{G_{144}}$ itself is obtained from a subset of the features[40] found in the Robinson and Goforth (2005) topology, introduced in §2.3.3.4.

$\boldsymbol{G_{144}}$ is composed of 144 ordinal bi-matrix game models $\boldsymbol{g_i}$; $\boldsymbol{g_i} \in \boldsymbol{G_{144}}$; $\boldsymbol{G_{144}} = (\boldsymbol{V}, \boldsymbol{E})$. There are 144 vertices $\boldsymbol{v} \in \boldsymbol{V}$ in $\boldsymbol{G_{144}}$. Each vertex $\boldsymbol{v_i} \in \boldsymbol{V}$ corresponds to game model $\boldsymbol{g_i} \in \boldsymbol{G_{144}}$. Each node $\boldsymbol{g} \in \boldsymbol{G_{144}}$ is of degree six (6), so has 6 directed edges $\boldsymbol{e} \in \boldsymbol{E}$, plus an identity edge $\boldsymbol{e_I}$, giving 432 undirected edges $\boldsymbol{e} \in \boldsymbol{E}$ between distinct nodes $\boldsymbol{g_i} \neq \boldsymbol{g_i}$ and a total of 576 undirected edges $\boldsymbol{e} \in \boldsymbol{E}$ when including the identity $\boldsymbol{e_I}$; $\boldsymbol{g_i} = \boldsymbol{g_i}$.

3.1 The Reduced RGS Graph, *G*.

$\boldsymbol{G_{144}}$ is synonymous to a Robinson-Goforth Space[41], or RGS. $\boldsymbol{G}$, as a subset of $\boldsymbol{G_{144}}$ is then a *reduced* RGS, or *rRGS*. To specify the difference between $\boldsymbol{G}$ and $\boldsymbol{G_{144}}$ it is first necessary to revisit the Robinson and Goforth (2005) topology, with focus now on the process for forming a network graph dual; in particular it is necessary to describe the procedure for generating edges between nodes in further detail, as it is in the application of the generators, and the resultant generation of the edges between vertices (specifically, of concern is the number of edges between vertices *within a layer* versus the number of edges *between layers*), that the functional difference between $\boldsymbol{G}$ and $\boldsymbol{G_{144}}$ is found.

Robinson and Goforth (2005) specify each game model element in the *RGS* as an abstract element that corresponds to a point in topological space. Over each of these points a set of six, *non-atomically* applied, group operations produce six edges per point. In the *rRGS*, each layer of the topological space has thirty-six game models; where six game models will have six edges, sixteen game models will have five edges, and the remaining

[39] T. S. Eliot, *Little Gidding*, *Four Quartets* (1943).

[40] By way of example, topological analysis such as that by Perlo-Freeman (2006), presented in §2.3.3.5, is not addressed in the model constructed for the research in this thesis.

[41] Introduced in §2.3.3.4, a Robinson-Goforth Space (RGS) is a graph representation, $\boldsymbol{G_{144}}$, of the dual of the Robinson and Goforth (2005) topology.

sixteen game models will have four edges. Edges are generated by the *atomic application* of the six group operations. The graph $\boldsymbol{G}$; $\boldsymbol{G} = (\boldsymbol{V_{144}}, \boldsymbol{E_{480}})$; $\boldsymbol{G} \subseteq \boldsymbol{G_{144}}$ is obtained from the topological space of $\boldsymbol{G_{144}}$ using atomic group operations only, plus the identity edges.

Each group operation represents a minimal deformation of the model. To these six operations is added the identity operation. Deformations to the game model are referred to as transformations, or more loosely, as *swaps*. Each swap is the reordering of one preference pair that any one participant has for an outcome.

In practice, this is achieved by swapping two of the four adjacent preferences (adjacent with respect to their ordinal rank) with each other. The three distinct transformations consist of operations $\mathbf{1} \leftrightarrow \mathbf{2}$, $\mathbf{2} \leftrightarrow \mathbf{3}$, and $\mathbf{3} \leftrightarrow \mathbf{4}$.

Including both participants, these operations are enumerated with $\boldsymbol{R}$ and $\boldsymbol{C}$ (row and column participant) identifiers, with the preference swap as subscript: $\boldsymbol{R_{12}}$, $\boldsymbol{R_{23}}$, $\boldsymbol{R_{34}}$, $\boldsymbol{C_{12}}$, $\boldsymbol{C_{23}}$, and $\boldsymbol{C_{34}}$.

The set of game models obtained by these six generators is the group $\boldsymbol{G_{144}}$ and includes topological features defined by Robinson and Goforth (2005) such as *regions*, *tiles*, *neighbourhoods*, *hotspots*, and *pipes*.

The set of generators $\boldsymbol{R_{12}}$, $\boldsymbol{R_{23}}$, $\boldsymbol{C_{12}}$, $\boldsymbol{C_{23}}$, applied to an arbitrary strictly-ordinal game model, will produce thirty-six game models (including the origin model), and form a single, closed, layer. The implication of an edge between two game models at opposite outer edges on each layer is that the relation wraps from one tile to another, as each row and column are connected in a cyclic manner (Perlo-Freeman, 2006).

The application of one of the generators $\{\boldsymbol{R}, \boldsymbol{C}\}_{\boldsymbol{12}}$ or $\{\boldsymbol{R}, \boldsymbol{C}\}_{\boldsymbol{23}}$ will change a game model from one point in the topology to another, for at most a distance of one point, i.e., to a neighbouring, adjacent, game model, as shown in **Figure 3.1**, which illustrates the application of the $\boldsymbol{R_{12}}$ generator on game model **g111**, resulting in the game model **g121**. **Figure 3.2** shows the neighbourhood that is obtained by applying the six generators to game model **g412**.

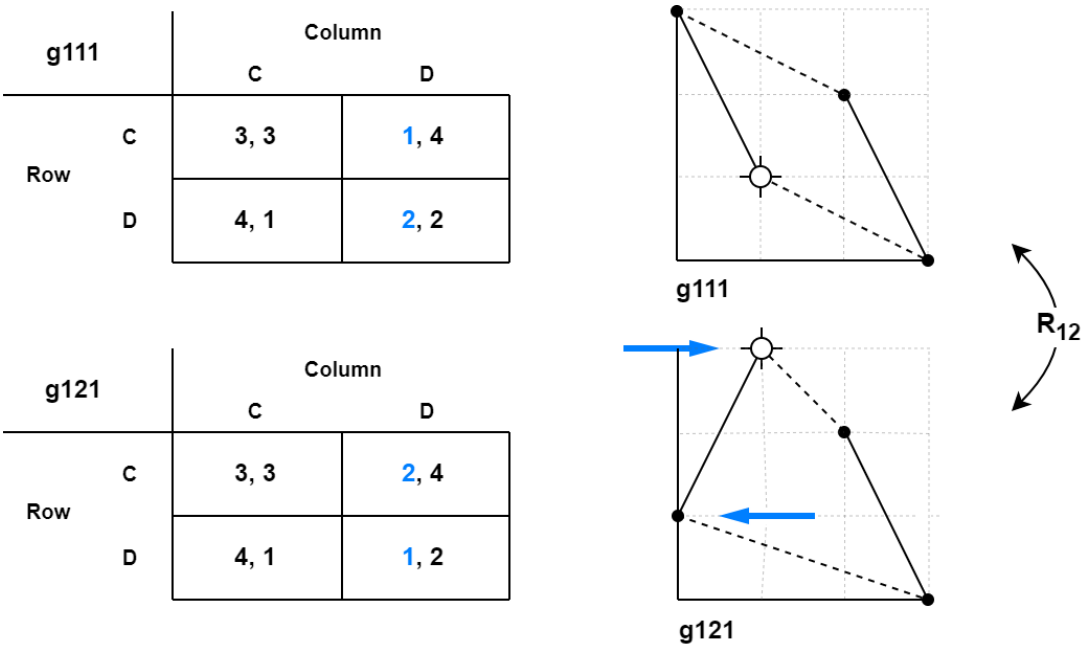


Figure 3.1: Application of $\boldsymbol{R_{12}}$ generator on **g111**. Interchanging row's two lowest values results in **g121**. Figure adapted from Fig 3.1, *The Topology of the 2x2 Games, A New Periodic Table* (Robinson and Goforth, 2005).

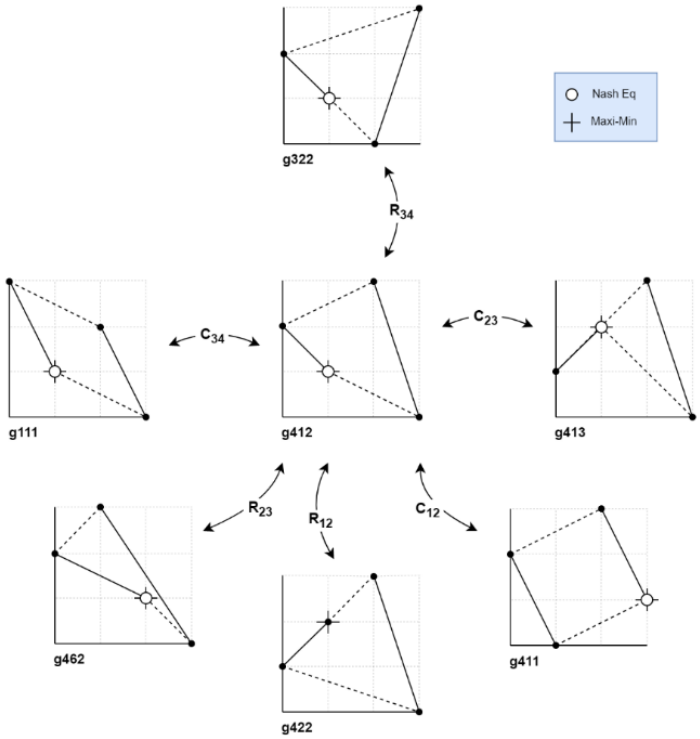


Figure 3.2: Neighbourhood N_{412} of **g412.** Elements of N_{412} are **g412**, **g111**, **g411**, **g422**, **g322**, **g413**, and **g462**. Note that Column's inducement correspondence in **g413** is the partially obscured dotted line that extends between (**1**, **2**) and (**3**, **4**). Figure is adapted from Fig 3.2, *The Topology of the 2x2 Games*, *A New Periodic Table* (Robinson and Goforth, 2005).

The seven games in the neighbourhood are located over three of the four layers of the topology, as shown in **Figure 3.3**.

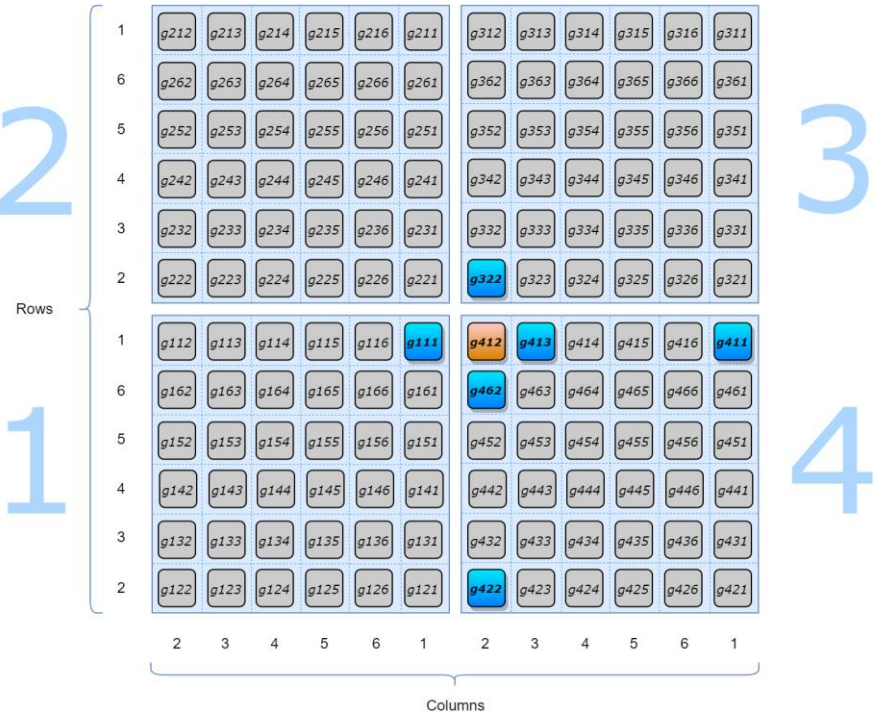


Figure 3.3: Robinson and Goforth layer map of the neighbourhood of game model **g412.** Six game models are obtained from the generators R_{12}, R_{23}, R_{34}, C_{12}, C_{23}, and C_{34}.

Application of the generators $\boldsymbol{R_{34}}$, and $\boldsymbol{C_{34}}$ link game models between layers. However, their atomic application does not result in a closed space. Applying the six generators atomically, without concatenating any generator in sequence with another, gives the undirected graph $\boldsymbol{G}$; $\boldsymbol{G} = (\boldsymbol{V_{144}}, \boldsymbol{E_{336}})$[42], which is a closed space (all edges lead, in either direction, to a vertice within $\boldsymbol{V}$).

It is on this result, i.e., the atomic application of the generators to form a closed set under $\boldsymbol{G_{144}}$, that the subset $\boldsymbol{G}$ is derived.

In conclusion:

- The graph $\boldsymbol{G}$ provides the model for the experiments detailed in Chapters Four, Five, and Six.
- An adjacency list for the graph $\boldsymbol{G}$, as a table, as a plain csv, and in *igraph* (Csardi et al., 2006) format, is given in Appendix B.1 and Appendix B.5.

[42] There are a total 336 undirected edges in the graph $\boldsymbol{G}$, 480 when including the 144 identity edges $\boldsymbol{e_I}$.

Chapter Four

Multi-Model Tournaments

That's what the money is for!

—Don Draper, *Mad Men*[43]

Software frameworks for running $\mathbf{2} \times \mathbf{2}$ tournaments have evolved from Axelrod's (1980a, 1980b, 1984) original tournaments to encompass research and open-source repositories of both general and specialised application. Recent work by Crandall et al. (2018a, 2018b) describes a tournament model for the study of human and machine cooperation, with games of varied types including the Robinson and Goforth (2005) topology.

The open-source repository *Axelrod-Python* (Knight et al., 2015) recreates Axelrod-style tournaments and strategies. Many of the algorithms are written and contributed by the community, in addition to the repository maintainers sourcing original 'Axelrod' code listings. *Gambit* (McKelvey, McLennan & Turocy, 2016) is a collection of tools for computational game theory which also has a web-GUI tool for model exploration. *NashPy* (Knight & Campbell, 2017) is a multi-layered library for $\mathbf{2} \times \mathbf{2}$ equilibria-computation tools for both normal- and extensive-form; in addition, *NashPy* provides tools for evolutionary paradigms.

Beyond the value that all of these hold, some characteristics of game-theory repositories, that likely hold for any domain's software output, are:

- every model has its own assumptions, some explicit, some not;
- the coverage, accuracy, and availability of documentation ranges from excellent to unavailable;
- models may not be extant or are inaccessible.

The first characteristic can have critical ramifications. It is not always clear what a third-party codebase 'takes-as-given', and just 'reading the code' is not always illuminating. The second characteristic—inaccurate and/or missing documentation—at the extreme can render software unusable or impractical. The final characteristic, for empirical game theory implementation, represents a loss of heritage, which is in-part responsible for motivating efforts to recover, and recreate, Axelrod-era (and later) codebases. Of the two Axelrod tournaments, only the second survives in a complete form,

[43] From the *Mad Men* episode 'The Suitcase' (Weiner & Getzinger, 2010); Don and Peggy are discussing compensation, reward, and expectations—their perspectives differ, it seems.

as a FORTRAN[44] source code listing (Knight et al., 2016; Campbell & Knight, 2020). However, much other code, pre-widespread internet access, has never been widely available. Some code is closed-source, and some is only available on request.

There is then a lack of open computational research environments for performing computational game theory research. Apart from motivating the creation of such a computational model, this chapter provides an opportunity to recontextualise the traditional game theory tournament (see §2.3.1.1 and §2.3.1.2) using the 144 game models of the *rRGS*—rather than using either the Crandall form of 720 game models (Crandall et al., 2018a), or the form of a singular model, as found in previous tournament studies (Axelrod, 1980a, 1980b, 1984; Knight et al, 2015)—presents an experimental space that is clearly derived from the Crandall study and is, within that remit, a novel implementation of a multi-model tournament, with its focus on *perfect-* and *incomplete-information*.

Here, the multi-model tournament consists of two rounds; in each round all participants play each other once only (a strict round-robin, see §4.2.1). The first round consists of a round-robin per algorithm-group and the final round-robin consists of qualifiers from the previous round. The principal metric of interest in this tournament is the correlation of the total reward (TR) that an agent gains across the complete set of matches it participates in, with the total number of times that the agent shares in a 'mutually cooperative' outcome (MCR). Assessing this correlation follows analysis performed by Crandall et al. (2018a; 2018b), in order to confirm their general finding (across a variety of games, game models, and studies) that "proportion of rounds that players played mutually cooperative solutions … was strongly correlated with the payoffs a player received" (2018a, p. 9).

This correlation is of interest as it is a fundamental indicator of whether an increased probability of an agent cooperating leads to improved reward, across the entire set of game models, i.e., a variety of environmental situations; despite such change in behaviour as potentially requiring compromise in the short-term, for example, by exposure to risk of the ***S***[45] outcome in the Prisoner's Dilemma game model. The trade-off between short-term compromise and long-term gain is of interest as a tenet of the *folk theorem* (Crandall & Goodrich, 2005; Banerjee & Sen, 2007). Replication of this general result in this study is desired as a means for asserting the validity of the overall computational research environment being developed in this thesis. Furthermore, numeric verification by analysis of deterministic strategies can confirm the validity of the computational research environment. The results of this verification step are presented in §4.2.1.

This chapter is organised as follows: firstly, in §4.1, the Multi-Model Tournament experiment type is described, enumerating the algorithms used, game models played over, and evaluation metrics. The derivation of 'mutual cooperation', per individual game model, is given in §4.1.3. Results are given in §4.2, followed by a discussion in §4.3. Supplementary material is located in Appendix B.2.

[44] FORTRAN in and of itself is not the problem, more the dwindling number of the FORTRAN-literate.
[45] ***S***, or 'sucker' outcome, is the result of cooperating when the other participant defects, see §2.3.1.

4.1 Multi-Model Tournament Experiment Type

The *Multi-Model tournament* experiment type is an aggregate game mode, being composed of a single strict-round-robin tournament of competing algorithms, per game model. A single strict-round-robin tournament is run over one game model (usually, but not always, the Prisoner's Dilemma, see §2.3.1.1). In a strict-round-robin tournament each algorithm, in a set of algorithms, plays every other algorithm in the set once only, giving $\boldsymbol{n(n-1)/2}$ matches[46]. The decision to use the strict form of a round-robin is informed by the discussion and analysis of dyadic interaction in Rapoport et al.'s (2015) multi-stage, round-robin, elimination tournaments.

A multi-model tournament runs over *every* game model in the RGS, giving 144 distinct game environments (Crandall et al., 2018a). Each match consists of a single episode of 1000 timesteps. This episode length is the median of the three values (100, 1k, and 50k) used by Crandall et al. (2018b). The set of game models consists of all vertices $\boldsymbol{V}$ from the *RGS* graph, $\boldsymbol{G}$, introduced in Chapter Three. The interaction of agents with the environment is constrained:

- to discrete temporal-interval evaluations,
- to a binary action-space, and
- to a single scalar value input (reward) per temporal-interval.

The scalar value is the rank of the ordinal preference in the range [1,4]. Each match in this experiment series is modelled as an *Imperfect Information Markov Game* (§2.2.2).

4.1.2 Algorithms

Each of the algorithms used in this experiment series is one of three types: *game-theoretic*, *binary bandit*, or *foundational Reinforcement Learning* (fRL). The selection of these groups was discussed in §2.2.3.

4.1.2.1 Game-Theoretic

Game-theoretic algorithms have had, and continue to have, widespread use: modelling processes in the biological sciences including evolution and predator-prey interactions (McNamara, 2022; Alpern et al., 2019; Brown et al., 1999); modelling the dynamics of individual and collective behaviours in psychology (Azar, 2019; Geanakoplos, 1989); in the modelling of economic theory (Henrich et al., 2001; Rabin, 1993; Kreps et al., 1982; Schelling, 1981), and in deep learning (Hazra & Anjaria, 2022).

There were two main criteria for the selection of the six game-theoretic, normal-form[47] algorithms shown in **Table 4.1**. The first was to not favour selection on the basis of an algorithm's target domain, but to choose algorithms based in a broader concept of domain generalisation (see §2.2.3.2). Secondly, to principally select static, 'pure-strategy'

[46] In a non-strict-round-robin each participant plays all other participants, *and also itself*. A participant playing itself is treated separately in this thesis as an instance of the symmetric selfplay experiment type, see Appendix A.

[47] The term 'normal-form' is synonymous with 'strategic-form'. All extensive-form algorithms are excluded from the selection process, favouring instead those algorithms that operate on normal-form domain representations (i.e., $\mathbf{2 \times 2}$ matrices), see §2.3.3.4.

algorithms (i.e., deterministic automata) that, while showcasing foundational game-theoretic concepts such as *Tit-for-Tat* and *Naïve Bully*, can be used for validation of the experimental framework as they have entirely deterministic behaviour. The algorithm *Naïve Bully* defects on every first move, then plays the opposite of what has been played by their opponent in every timestep thereafter. In counterpoint, the *Random* algorithm has no fixed responses, makes no inference, and performs no analysis: *Random*'s (mostly[48]) unbiased stochasticity provides a reference point to compare the performance of other algorithms[49].

The algorithm *Fictitious Play* (see §2.2.4) plays its 'best-response'; so, it is stochastic if its opponent is stochastic—a deterministic opponent would generate a deterministic response from *Fictitious Play* (Fudenberg & Levine, 1998). *Fictitious Play* requires relaxation of the constraint over *state information visibility* (see §1.3.2). The constraint over the agent/environment boundary also requires relaxation as *Fictitious Play* must be able to observe other agent's actions (see §1.3.4).

Table 4.1: Properties of game-theoretic algorithms. **Visibility** indicates relaxation of constraints §1.3.2 and §1.3.4 (**MDP**: Markov Decision Process. **POMDP**: *Partially-Observable Markov Decision Process*). **Stochastic** indicates if the algorithm has non-deterministic action-selection and/or state-transitions, or is a 'pure' strategy, i.e., a fixed strategy, equivalent to a deterministic automaton. **Tag** is used in tables and figures to identify each algorithm.

Algorithm	Tag	Visibility	Stochastic
Always Cooperate	*allc*	POMDP	N
Always Defect	*alld*	POMDP	N
Fictitious Play	*fictitious_play*	MDP	Y
Naïve Bully	*bully_naïve*	MDP	N
Random	*random*	POMDP	Y
Tit-for-tat	*tft*	MDP	N

4.1.2.2 Binary Bandits

The *Binary Bandit* algorithms (see §2.2.4) are reduced forms of the *Bandit* algorithms given by Sutton and Barto (1998; 2018), who describe the full *Bandit* algorithms as precursors to complete reinforcement learning algorithms. The methods and techniques that these algorithms are constructed from are also found in *fRL* algorithms. For example, methods for action-selection predominantly use one of two approaches: *ε-greedy*, or *softargmax*. As such, bandits are ablative forms of their RL counterparts.

Sutton and Barto (1998) highlight issues that Bandit algorithms have on certain types of problems. A $\mathbf{2 \times 2}$ matrix game with stochastic probability of either *failure* or *success* presents all algorithms with an action space that has only two options. Applied to Bandit algorithms, this produces the *Binary Bandit* class of algorithms. Where an algorithm has only two options the calculation of *success* probability can be either straightforward or difficult. This is because, in the simple case, one action has a greater than half probability of succeeding, and the other action a less than half probability of failure. Inferring that the action that produced the success is then more likely to produce the same reward in

[48] Software library methods for generating pseudorandom values can have bias themselves (Lemire, 2019; Cook, 2018). Here, in any given experiment instance, *all* agents will be subject to the *same* bias due to being run on a common platform.

[49] As an aside, *Random* is a winning strategy in the game Matching Pennies (Bhattacharya, 2021).

future is then favoured. However, when the probability of success for either action is low (or conversely, both high), then in the first case (low probability of success) the bandit will infer that the action that produced the failure results in the alternative option to have a high probability of success even though, as stated, it too has a low chance of producing an improved reward. This pattern also holds for problems with high probability of success. In this case, the alternative action is inferred to have a low probability of producing improved reward even though it too actually has a high probability of producing the reward. This dichotomy is illustrated in **Figure 4.1**. As RL algorithms are functional descendants of Bandit methods (Sutton & Barto, 2018) the dichotomy that Binary Bandits encounter may expose a weakness in *fRL* algorithms over $\mathbf{2} \times \mathbf{2}$ domains as well.

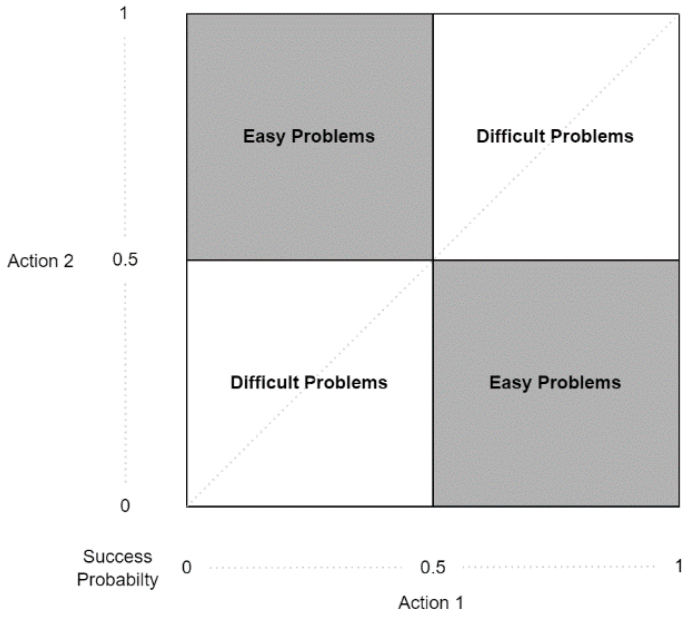


Figure 4.1: Binary Bandit unit square. The divisions of the unit square illustrates the issues in decision-making for Binary Bandits. Adapted from Fig. 2.2, *Reinforcement Learning: An Introduction* (Sutton & Barto, 1998).

The Binary Bandits selected for this experiment series are listed in **Table 4.2**. These algorithms have been selected on several criteria. Firstly, that they conform to the *on-line* constraint (§1.3.1) which excludes post-hoc-analysis and *Monte Carlo* algorithms. Of the algorithms selected, three force the relaxation of the constraint that the reward signal is the only information crossing the agent/environment boundary (see §1.3.4). These three algorithms are:

- *Supervised Learning*, *Direct*,
- *Supervised Learning*, *Learning Automata*, *Linear-Reward-Inaction*, and
- *Supervised Learning*, *Learning Automata*, *Linear-Reward-Penalty*.

These three algorithms all require knowledge of the other agent's previous action at the time of the current timestep, however, they do not maintain explicit history of this knowledge beyond the previous timestep, in other words, memory depth over their opponent's actions is equal to one.

4.1.2.3 Foundational Reinforcement Learning (*fRL*)

The Reinforcement Learning (RL) algorithms implemented in this thesis are drawn from Sutton and Barto (1998; 2018), Szepesvári (2010), and Watkins (1989); and were introduced in §2.2.4. as 'foundational' RL (*fRL*) algorithms. The fourteen algorithms selected conform to the constraints outlined in §1.3 and are listed in **Table 4.3**. These algorithms are separable by their use of hyperparameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ (learning rates), $\boldsymbol{\gamma}$ (discount return rate), $\boldsymbol{\lambda}$ (trace discount rates), and by their action-selection method, either *softargmax*, or *ε-greedy*. The algorithm *Watkins Q, Linear Function Approximation* is a linear function approximator while the remaining algorithms are so-called 'tabular' models—using traces, Q-tables, and other, similar data structures, for discrete internal state representation. To conform to the Markov POMDP model (§2.2.1), all of the algorithms have explicit history over only the most recent[50] timestep, as well as the current timestep. The algorithm *Watkins Q, Linear Function Approximation* is implemented using a feature vector of the outcomes from the last timestep, plus an associated weight vector.

4.1.3 Evaluation Metrics

There are two metrics of interest in the multi-model tournament. First, an agent's total reward for the entire tournament. Second, the count of an agent's 'acts of mutual cooperation' over the entire tournament. A single 'mutually-cooperative' outcome counts as one occurrence of a mutually-cooperative outcome for each agent. The same pair of actions will not necessarily result in cooperation in other games (as they would in Prisoner's Dilemma) because the strategic dynamics (Nash Equilibria, Nash Bargaining Solution, Maxi-Min) solutions change per game model, so the same action pair has a different teleological outcome with respect to both cooperation and reward than in the Prisoner's Dilemma. The definition for mutual cooperation used in this experiment series follows the model outlined by Crandall et al. (2018a; 2018b), as being the Nash Bargaining Solution (NBS) outcome(s) in a given game model.

In the case of a game model with two NBS locations, analysis to apply any of a range of methods to calculate the exact ratio of mutual cooperation were not pursued, in favour of the raw count of acts of mutual cooperation (being the joint outcome that maps to the NBS locations in each game model). The NBS locations for the first sixteen game models in Layer One of the *rRGS* graph, $\boldsymbol{G}$, are shown in **Table 4.4**.

An expanded version of **Table 4.4**, for *all games in all layers*, which includes the payoff vectors and cell locations in canonical form, as well as in the alternative cartesian form, is located in Appendix B.2.2.

[50] An algorithm's explicit memory depth (the explicit stored history of outcomes from previous timesteps) is configured by a *memory_depth* option which is set to one (1) in all algorithms. Those algorithms that incorporate various data structures for state representation (e.g., Q-tables, or *eligibility* and *replacing* traces—which can be thought of as weights over states on longer timescales than just the immediate past), also apply the constraint on memory depth to those data structures. See Appendix A.3.

Table 4.2: Binary Bandit Algorithms. All seventeen (17) algorithms are stochastic. **Visibility** indicates constraint over SIV (§1.3.2), and environment boundary (§1.3.4). **Tag** is used in figures.

Algorithm	Tag	Visibility
Incremental, Action Preferences, Softmax	*bandit_inc_softmax_ap_2ed*	POMDP
Non-Incremental, Action Preferences, Softmax	*bandit_noninc_softmax_ap_2ed*	POMDP
Pursuit, Sample-Average	*bandit_pursuit_sav*	POMDP
Reinforcement Comparison	*bandit_reinfcomp*	POMDP
Sample-Average Incremental	*bandit_sav_inc*	POMDP
Sample-Average, Incremental, Optimistic, Greedy	*bandit_sav_inc_optimistic_greedy*	POMDP
Sample-Average, Incremental, Softmax	*bandit_sav_inc_softmax*	POMDP
Sample-Average, Non-Incremental	*bandit_sav_noninc*	POMDP
Sample-Average, Non-Incremental, Softmax	*bandit_sav_noninc_softmax*	POMDP
Supervised Learning, Direct	*bandit_sl_direct*	MDP
Supervised Learning, Learning Automata, Linear-Reward-Inaction	*bandit_sl_la_lri*	MDP
Supervised Learning, Learning Automata, Linear-Reward-Penalty	*bandit_sl_la_lrp*	MDP
Weighted Average	*bandit_wa*	POMDP
Weighted-Average, Optimistic, Greedy	*bandit_wa_optimistic_greedy*	POMDP
Weighted Average, Softmax	*bandit_wa_softmax*	POMDP
Weighted Average, Softmax, Action Preferences	*bandit_wa_softmax_ap_2ed*	POMDP
Weighted Average, Upper Confidence Bounds	*bandit_wa_ucb*	POMDP

Table 4.3: Foundational RL Algorithms. All fourteen (14) algorithms adhere to all constraints (§1.3) and are stochastic. **Tag** is used in figures.

Algorithm	Tag	Hyperparameters			
		Learning	Discount	Trace Discount	Action Selection
Actor/Critic	*actor_critic_1ed*	α, β	γ		$softargmax$
Actor/Critic, Eligibility Traces	*actor_critic_1ed_eligibility_traces*	α, β	γ	λ	$softargmax$
Actor-Critic, Replacing Traces	*actor_critic_1ed_replacetrace*	α, β	γ	λ	$softargmax$
Double Q-Learning	*double_qlearning*	α	γ		$\varepsilon - greedy$
Expected SARSA	*expected_sarsa*	α	γ		$\varepsilon - greedy$
Q-Learning	*qlearning*	α	γ		$\varepsilon - greedy$
R Learning	*rlearning*	α, β			$\varepsilon - greedy$
SARSA	*sarsa*	α	γ		$\varepsilon - greedy$
SARSA Lambda	*sarsa_lambda*	α	γ	λ	$\varepsilon - greedy$
SARSA Lambda, Replacing Traces	*sarsa_lambdareplacetrace*	α	γ	λ	$\varepsilon - greedy$
Watkins (naive) Q, Lambda	*watkins_naïve_q_lambda*	α	γ	λ	$\varepsilon - greedy$
Watkins (naive) Q, Lambda, Replacing Traces	*watkins_naïve_q_lambda_replacetrace*	α	γ	λ	$\varepsilon - greedy$
Watkins Q, Lambda	*watkins_q_lambda*	α	γ	λ	$\varepsilon - greedy$
Watkins Q, Linear Function Approximation	*watkins_q_lfa*	α	γ		$\varepsilon - greedy$

Table 4.4: RGS Layer One Extract; Mutual Cooperation Locations for Sixteen Game Models. The first sixteen game models of RGS Layer One show cell locations for Nash Equilibria (**NE**), Nash Bargaining Solutions (**NBS**), Maximin (**Maximin**), and Mutual Cooperation (**MC1, MC2**). Cell locations are indicated using canonical (**RGS**) cell indexing. Mapping of cell locations to canonical semantic cell outcomes is as follows: CC ↔ (0,0), CD ↔ (0,1), DC ↔ (1,0), DD ↔ (1,1). Outcome payoff values indicated by **V**. Note lexical ordering of game models.

Sources: NE1: Nash Equilibrium One (Robinson & Goforth, 2005; Bruns 2015; Crandall et al, 2018b). **NE2:** Nash Equilibrium Two (Robinson & Goforth, 2005; Crandall et al, 2018b). **NBS1:** Nash Bargaining Solution One (Crandall et al., 2018b). **NBS2:** Nash Bargaining Solution Two (Crandall et al., 2018b). **Maximin:** Maxi-min solution (Robinson & Goforth, 2005). **MC:** Synthesis: Translation from cartesian payoff vector ordering to canonical payoff vector ordering; Ordering of **MC #1** and **MC #2** as per Crandall et al. (2018b).

RGS Layer One Extract, Mutual Cooperation Locations							
Game Model	**NE1**	**NE2**	**NBS1**	**NBS2**	**Maximin**	**MC1**	**MC2**
g111	1,1	-	0,0	-	1,1	0,0	-
g112	1,0	-	1,0	-	1,0	1,0	-
g113	1,0	-	1,0	0,1	1,0	1,0	0,1
g114	1,0	-	1,0	0,1	1,1	1,0	0,1
g115	1,1	-	1,0	0,1	1,1	1,0	0,1
g116	1,1	-	1,1	-	1,1	1,1	-
g121	0,1	-	0,1	-	0,1	0,1	-
g122	1,0	0,1	0,0	-	0,0	0,0	-
g123	1,0	0,1	1,0	0,1	0,0	1,0	0,1
g124	1,0	0,1	1,0	0,1	0,1	1,0	0,1
g125	0,1	-	1,0	0,1	0,1	1,0	0,1
g126	0,1	-	1,0	0,1	0,1	1,0	0,1
g131	0,1	-	1,1	0,1	0,1	1,1	0,1
g132	1,0	0,1	1,0	0,1	0,0	1,0	0,1
g133	1,0	0,1	1,0	0,1	0,0	1,0	0,1
g134	1,0	0,1	1,0	0,1	0,1	1,0	0,1
g135	0,1	-	0,1	-	0,1	0,1	-
g136	0,1	-	0,1	-	0,1	0,1	-
g141	0,1	-	1,0	0,1	1,1	1,0	0,1
g142	1,0	0,1	1,0	0,1	1,0	1,0	0,1
g143	1,0	0,1	1,0	0,1	1,0	1,0	0,1
g144	1,0	0,1	1,0	0,1	1,1	1,0	0,1
g145	0,1	-	0,1	-	1,1	0,1	-
g146	0,1	-	0,1	-	1,1	0,1	-

4.2 Results

In this section, the proportion of the *potential* mutual cooperation events each agent engages in is calculated along with the proportion of *potential* reward that each agent receives. If an algorithm participates in an act of mutual cooperation in every timestep then their rate of mutual cooperation (MCR) is 1. Similarly, if an algorithm participates in mutual cooperation in exactly half, or 500, of 1000 timesteps, their *MCR* is .5. The algorithm's total reward (TR) in any single game model, over 1000 timesteps, is in the range [1000–4000], as each timestep reward is the scalar representation of the ordinal preference, with range [1–4]. Results are presented for each algorithm group's round-robin tournament, plus the final round consisting of the 'winners' of each previous round-

robin. For each algorithm group, the results show two metrics: *TR*, an agent's received proportion of available reward; and *MCR*, an agent's proportion of engagement in available opportunities for mutual cooperation. Both metrics are first tabulated for the single game model **g111**, and then for all game models in ***G***. The two metrics are evaluated using Pearson's coefficient of correlation[51] to ascertain linear association, as a proxy for an agent's ability to learn cooperative strategies as a means to gaining optimal returns (see the discussion of the *folk theorem* in the introduction to this chapter, and also §4.3).

4.2.1 Match Pairing Validation

Table 4.5 lists every match pairing in the game-theoretic algorithm set over the single game model **g111** and confirms that the reward for each agent, and the number of mutual cooperation events, conforms to expectations; for example, both *Always Cooperate* and *Tit-for-Tat* will always cooperate with each other at every timestep, giving 1000 mutual cooperation events for the match; likewise, the pairing of *Always Defect* and *Naïve Bully* records zero mutual cooperation. See Appendix B.2.3 for more detail.

Table 4.5: Single-Model Tournament Match Summary: Game-theoretic Match Results. All matches are over single game model **g111**. **MC** (*Mutual Cooperation*) is the total number of outcomes that map to one of the game model's mutual cooperation locations, listed in **Table 4.4**.

Single-Model Tournament Summary Match Results -- Game Theoretic, g111, 1k				
Agent Zero	**Agent One**	**Reward Agent Zero**	**Reward Agent One**	**MC**
allc	alld	1000	4000	0
allc	bully_naïve	1000	4000	0
allc	fictitious_play	3000	3000	1000
allc	tft	3000	3000	1000
allc	random	2012	3494	506
alld	bully_naïve	3998	1001	0
alld	fictitious_play	2002	1999	0
alld	tft	2002	1999	0
alld	random	2960	1520	0
bully_naïve	fictitious_play	1001	3998	0
bully_naïve	tft	2500	2500	250
bully_naïve	random	2572	2428	256
fictitious_play	tft	3000	3000	1000
fictitious_play	random	1986	3501	489
tft	random	2521	2524	289

4.2.2 Two-Stage Round-Robin Tournament

The first round of the two-stage round-robin tournament consists of three sets of matches, one set for each algorithm group. The second round selects three of the top four

[51] Pearson's test for correlation on the strength and direction of a linear association between two variables (https://en.wikipedia.org/wiki/Pearson_correlation_coefficient) is used here in light of three factors: firstly, to follow the general model of the work of Crandall et al. (2018a, 2018b) which itself is informed by the work of Axelrod (1984); secondly, to corroborate with Axelrod's findings concerning the "norm of reciprocity" (1984, p. 5); and thirdly, given Press and Dyson's findings that zero-determinant strategies reveal linear relations in the scores the players receive, which, in turn, "depend linearly on their corresponding payoff matrices" (2012, p. 10410). Furthermore, Press and Dyson assert that zero-determinant strategies exist in all iterated 2x2 games (2012).

algorithms from each group, and then adds several algorithms from across the groups. These additional algorithms are added due to the utility they offer in comparison to other algorithms; for example, *Always Defect* is promoted through to the second round as its deterministic nature, of always defecting, provides a lower *MCR* bound.

4.2.2.1 Game-Theoretic Round-Robin

There are six game-theoretic algorithms in the tournament. The number of matches in a round-robin tournament over a single game model is 15. Each match is independent. There are a total of 2160 matches over 144 game models with the algorithms playing over 15 matches of 1,000 timesteps each, per game model.

Table 4.6 summarises results for the game-theoretic algorithms with respect to *TR* and *MCR*, over the single game model **g111**. A plot of this data is shown in **Figure 4.2**. The Pearson correlation co-efficient of the linear relationship between *TR* and *MCR* shows a negative correlation of $r(4) = -.28,\ p = -.595$.

Table 4.7 summarises the results for the game-theoretic algorithms with respect to *TR* and *MCR* for all game models in $\boldsymbol{G}$. A plot of this data is shown in **Figure 4.3**. The Pearson correlation co-efficient of the linear relationship between *TR* and *MCR* shows a negative correlation of $r(4) = -.13,\ p = .802$.

Table 4.6: Single-Model Tournament Summary: Game-theoretic Algorithm Performance over game model **g111**. **TR**: Total Reward, or the proportion of available reward that an algorithm receives. **MCR**: Mutual Cooperation Rate, being the share of total available opportunities for mutual cooperation in which an algorithm shares. Bolded values indicate the highest value in each column.

Single-Model Tournament Summary Algorithm Performance -- Game Theoretic, g111, 1k				
Algorithm	**Total Reward**	**Total MC**	**TR**	**MCR**
allc	10012	2506	.5006	.5012
alld	**14962**	0	**.7481**	.0
bully_naïve	11074	506	.5537	.1012
fictitious_play	13983	2489	.69915	.4978
tft	13020	**2539**	.651	**.5078**
random	13467	1540	.67335	.308

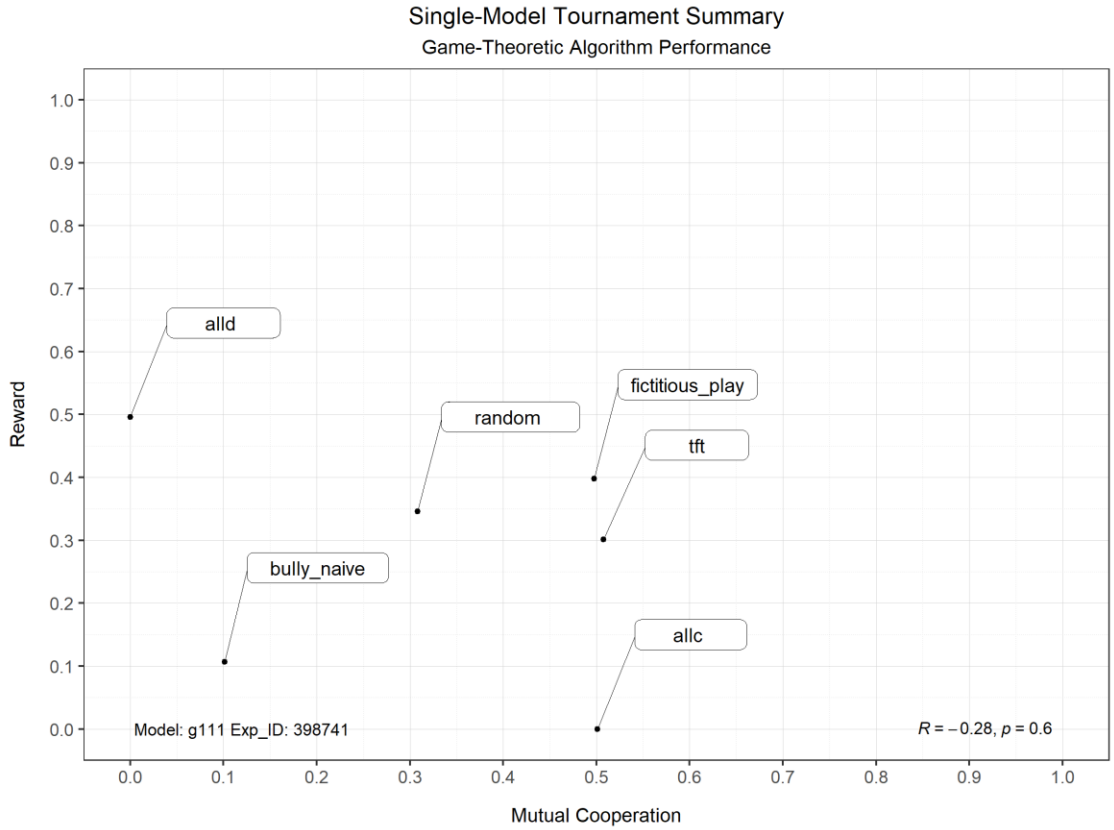


Figure 4.2: Single-Model Tournament Summary, Game-theoretic Algorithm Performance.

Table 4.7: Multi-Model Tournament Summary: Game-theoretic Algorithm Performance. **TR**: Total Reward, or the proportion of available reward that an algorithm receives. **MCR**: Mutual Cooperation Rate, being the share of total available opportunities for mutual cooperation in which an algorithm shares. Bolded values indicate the highest value in each column.

Multi-Model Tournament Summary Algorithm Performance -- Game Theoretic, 1k				
Algorithm	**Total Reward**	**Total MC**	**TR**	**MCR**
allc	1732501	**310567**	.6016	**.4313**
alld	1800222	162974	.6251	.2264
bully_naïve	1770973	224404	.6149	.3117
fictitious_play	1802096	250452	.6257	.3478
tft	**1846148**	291841	**.641**	.4053
random	1823538	253146	.6332	.3516

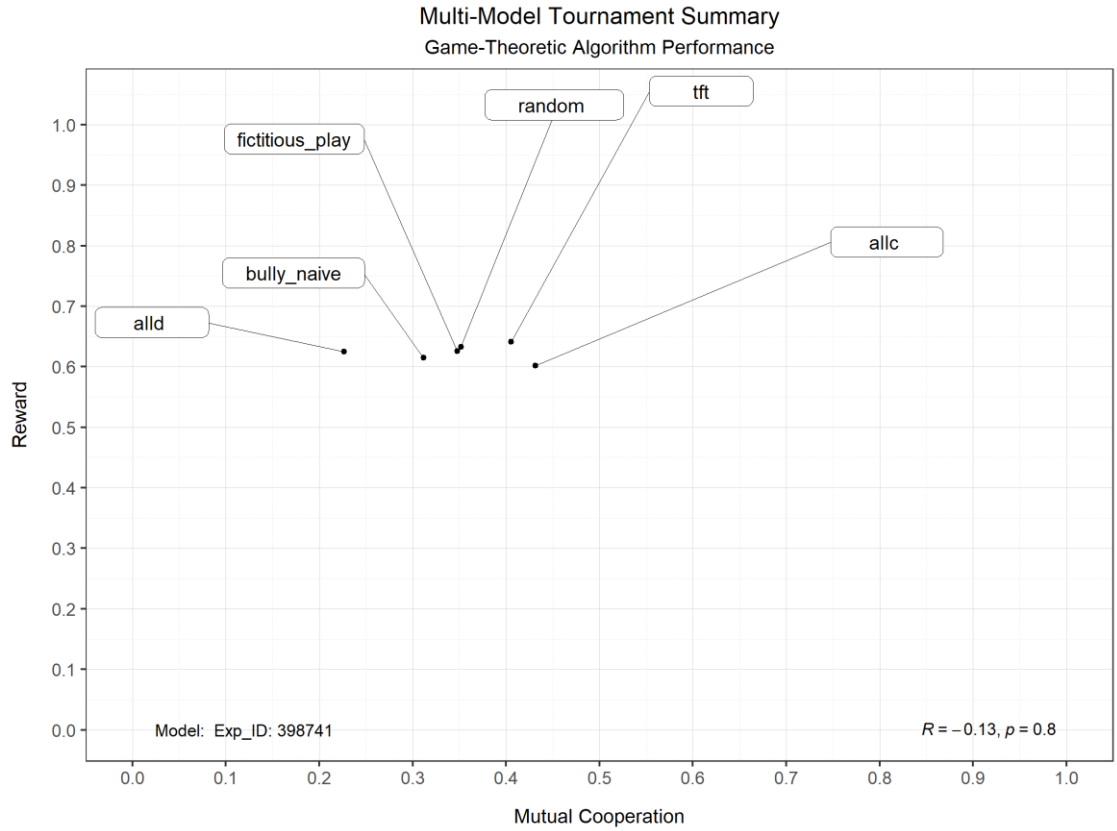


Figure 4.3: Multi-Model Tournament Summary, Game-theoretic Algorithm Performance.

4.2.2.2 Binary Bandit Round-Robin

There are seventeen (17) Binary Bandit algorithms in the tournament. The number of matches in a round-robin tournament over a single game model is 136. Each match is independent. There are a total of 19,584 matches over 144 game models with the algorithms playing over 136 matches of 1,000 timesteps each, per game model.

Table 4.8 summarises the results for the Binary Bandit algorithms with respect to *TR* and *MCR* for all game models in **G**. A plot of this data is shown in **Figure 4.4**. The Pearson correlation co-efficient of the linear relationship between *TR* and *MCR* shows a positive correlation of $r(15) = .99,\ p = 7.7 \times 10^{-14}$.

Table 4.9 summarises results for the Binary Bandit algorithms with respect to *TR* and *MCR* over the single game model **g111**. A plot of this data is shown in **Figure 4.5**. The Pearson correlation co-efficient of the linear relationship between *TR* and *MCR* shows a negative correlation of $r(15) = -.87,\ p = 7.33 \times 10^{-6}$.

Table 4.8: Single-Model Tournament Summary: Binary Bandit Algorithm Performance over game model **g111**.

TR: Total Reward, or the proportion of available reward that an algorithm receives. **MCR**: Mutual Cooperation Rate, being the share of total available opportunities for mutual cooperation in which an algorithm shares. Bolded values indicate the highest value in each column.

Single-Model Tournament Summary Algorithm Performance -- Binary Bandits, g111				
Algorithm	**Total Reward**	**Total MC**	**TR**	**MCR**
bandit_inc_softmax_ap_2ed	33274	483	.5199	.0302
bandit_noninc_softmax_ap_2ed	31270	483	.4886	.0302
bandit_pursuit_sav	36767	70	.5745	.0044
bandit_reinfcomp	37492	137	.5858	.0086
bandit_sav_inc	35539	119	.5553	.0074
bandit_sav_inc_optimistic_greedy	36444	20	.5694	.0013
bandit_sav_inc_softmax	28093	864	.439	.054
bandit_sav_noninc	35175	249	.5496	.0156
bandit_sav_noninc_softmax	28201	901	.4406	.0563
bandit_sl_direct	36922	5	.5769	.0003
bandit_sl_la_lri	**38098**	35	**.5953**	.0022
bandit_sl_la_lrp	36403	31	.5688	.0019
bandit_wa	33951	**907**	.5305	**.0567**
bandit_wa_optimistic_greedy	37666	129	.5885	.0081
bandit_wa_softmax	28241	889	.4413	.0556
bandit_wa_softmax_ap_2ed	34763	670	.5432	.0419
bandit_wa_ucb	36174	190	.5652	.0119

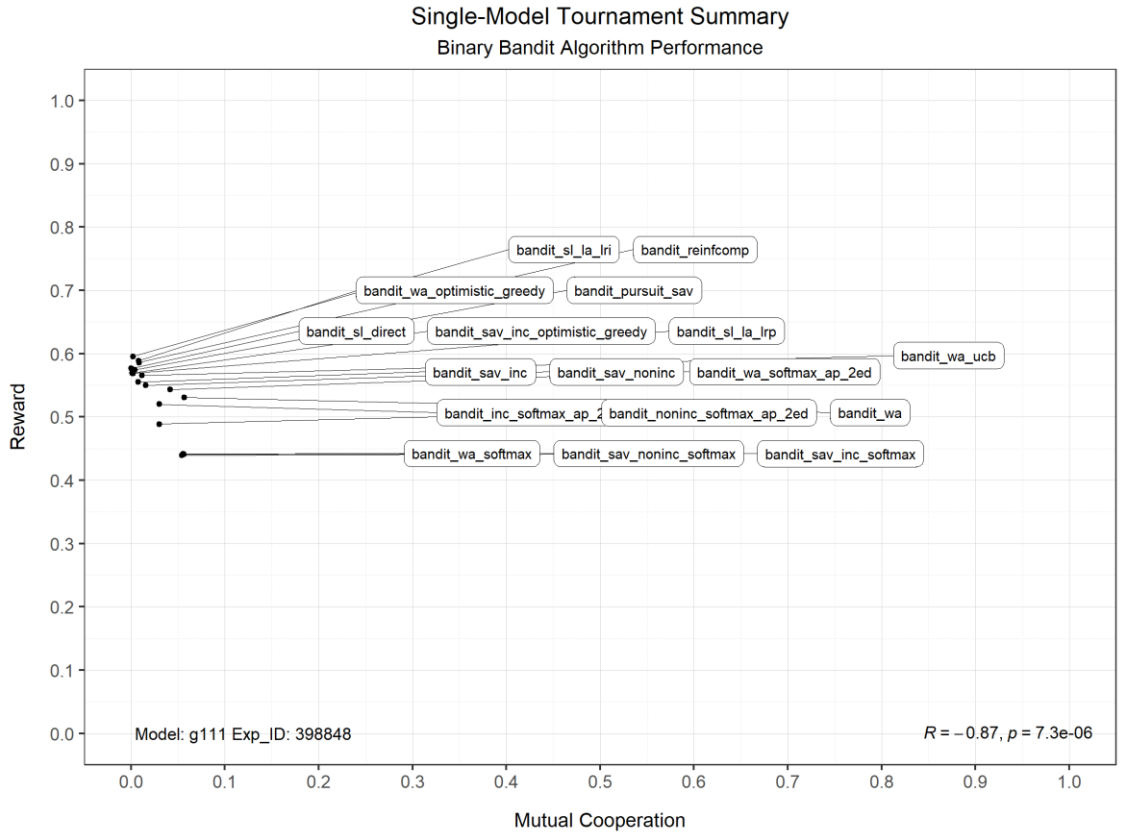


Figure 4.4: Single-Model Tournament Summary, Binary Bandit Algorithm Performance.

Table 4.9: Multi-Model Tournament Summary: Binary Bandit Algorithm Performance.

TR: Total Reward, or the proportion of available reward that an algorithm receives. **MCR**: Mutual Cooperation Rate, being the share of total available opportunities for mutual cooperation in which an algorithm shares. Bolded values indicate the highest value in each column.

Multi-Model Tournament Summary
Algorithm Performance -- Binary Bandits

Algorithm	Total Reward	Total MC	TR	MCR
bandit_inc_softmax_ap_2ed	6927532	1337861	.7517	.5807
bandit_noninc_softmax_ap_2ed	5891208	989714	.6392	.4296
bandit_pursuit_sav	7224221	1489458	.7839	.6465
bandit_reinfcomp	7330075	1472716	.7954	.6392
bandit_sav_inc	7214689	1459792	.7828	.6336
bandit_sav_inc_optimistic_greedy	7098920	1488346	.7703	.646
bandit_sav_inc_softmax	5912474	1037853	.6415	.4505
bandit_sav_noninc	7148517	1467276	.7757	.6368
bandit_sav_noninc_softmax	5884778	1038146	.6385	.4506
bandit_sl_direct	7327025	1487791	.795	.6457
bandit_sl_la_lri	7184732	1448287	.7796	.6286
bandit_sl_la_lrp	7189809	1463851	.7801	.6354
bandit_wa	7091791	1464291	.7695	.6355
bandit_wa_optimistic_greedy	**7360559**	**1519764**	**.7987**	**.6596**
bandit_wa_softmax	5901190	1036701	.6403	.45
bandit_wa_softmax_ap_2ed	6943699	1350297	.7534	.5861
bandit_wa_ucb	7096130	1480994	.77	.6428

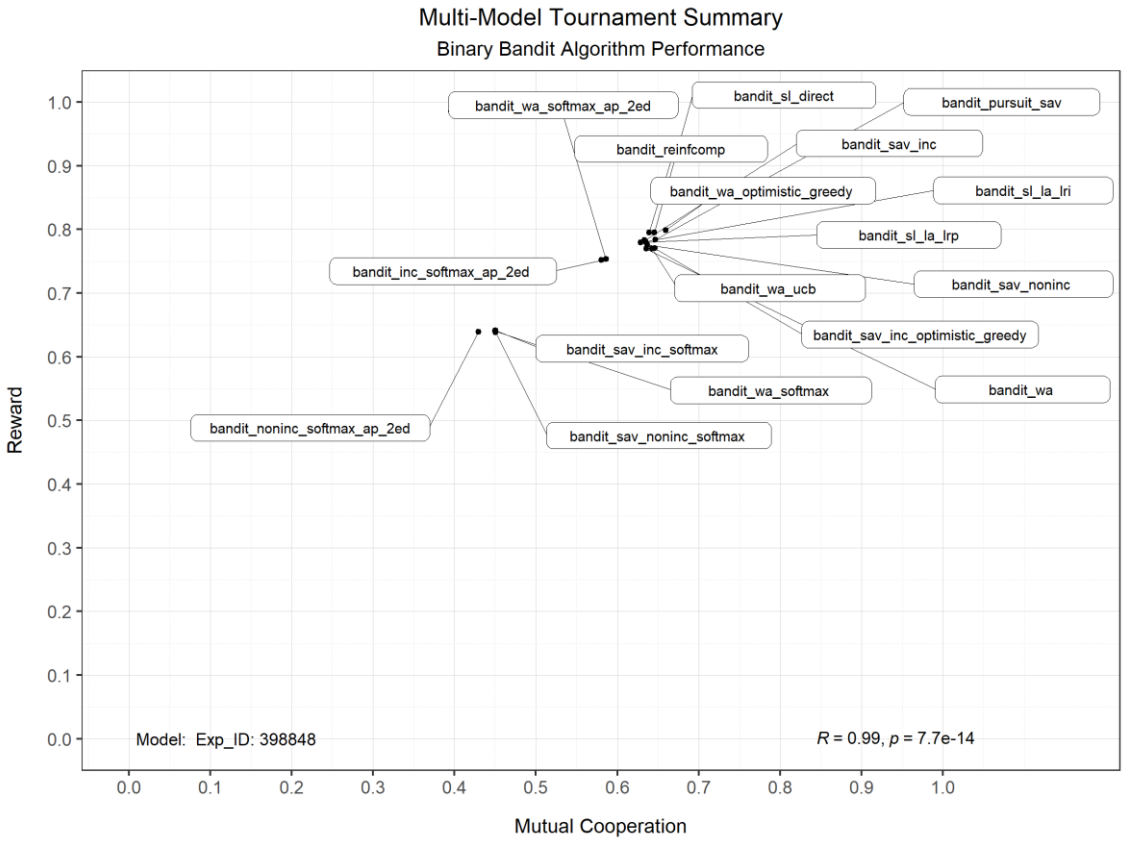


Figure 4.5: Multi-Model Tournament Summary, Binary Bandit Algorithm Performance.

4.2.2.3 Foundational RL Round-Robin

There are fourteen *fRL* algorithms in the tournament. The number of matches in the round-robin tournament, over a single game model, is 91. Each match is independent. There are a total of 13,104 matches over 144 game models with the algorithms playing over 91 matches of 1,000 timesteps each, per game model.

Table 4.10 summarises results for the *fRL* algorithm set with respect to *TR* and *MCR*, over the single game model **g111**. A plot of this data is shown in **Figure 4.6**. The Pearson correlation co-efficient of the linear relationship between *TR* and *MCR* shows a negative correlation of $r(12) = -.59,\ p = .026$.

Table 4.11 summarises the results for the *fRL* algorithms with respect to *TR* and *MCR*, for all game models in $\boldsymbol{G}$. A plot of this data is shown in **Figure 4.7**. The Pearson correlation co-efficient of the linear relationship between *TR* and *MCR* shows a positive correlation of $r(12) = .5,\ p = .067$.

Table 4.10: Single-Model Tournament Summary: Foundational RL Algorithm Performance over game model **g111**.

TR: Total Reward, or the proportion of available reward that an algorithm receives. **MCR**: Mutual Cooperation Rate, being the share of total available opportunities for mutual cooperation in which an algorithm shares. Bolded values indicate the highest value in each column.

Single-Model Tournament Summary Algorithm Performance – Foundational RL, g111, 1k				
Algorithm	**Total Reward**	**Total MC**	**TR**	**MCR**
actor_critic_1ed	**40126**	6164	**.7717**	.4742
actor_critic_1ed_eligibility_traces	35821	3695	.6889	.2842
actor_critic_1ed_replacetrace	38363	3671	.7378	.2824
double_qlearning	36012	**9560**	.6925	**.7354**
expected_sarsa	31190	7793	.5998	.5995
qlearning	34262	9509	.6589	.7315
rlearning	39535	5361	.7603	.4124
sarsa	31745	8180	.6105	.6292
sarsa_lambda	35472	9023	.6822	.694
sarsa_lambda_replacetrace	33850	9152	.651	.704
watkins_naive_q_lambda	36021	8236	.6927	.6335
watkins_naive_q_lambda_replacetrace	34636	9199	.6661	.7076
watkins_q_lambda	34569	9394	.6647	.7226
watkins_q_lfa	38191	4379	.7344	.3368

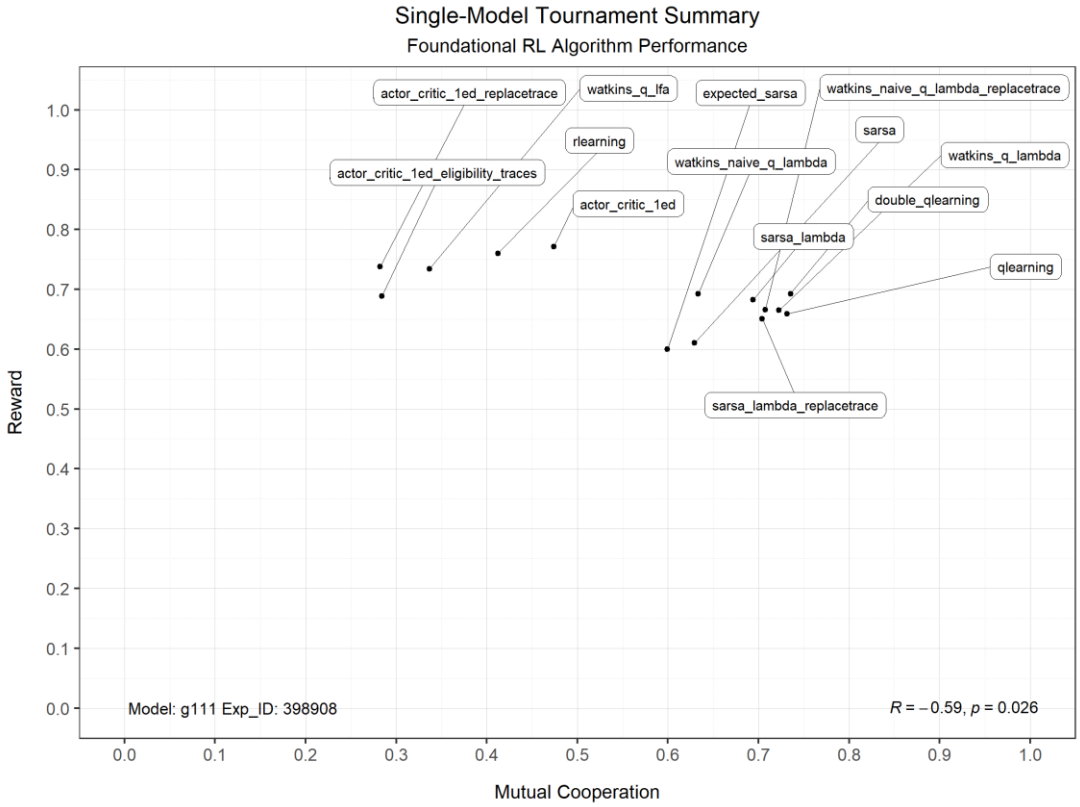


Figure 4.6: Single-Model Tournament Summary, Foundational RL Algorithm Performance.

Table 4.11: Multi-Model Tournament Summary: Foundational RL Algorithm Performance.

TR: Total Reward, or the proportion of available reward that an algorithm receives. **MCR**: Mutual Cooperation Rate, being the share of total available opportunities for mutual cooperation in which an algorithm shares. Bolded values indicate the highest value in each column.

Multi-Model Tournament Summary Algorithm Performance – Foundational RL, 1k				
Algorithm	**Total Reward**	**Total MC**	**TR**	**MCR**
actor_critic_1ed	5351990	900724	.7147	.4812
actor_critic_1ed_eligibility_traces	5085750	791408	.6792	.4228
actor_critic_1ed_replacetrace	5211990	813748	.696	.4347
double_qlearning	5399190	1046442	.721	.559
expected_sarsa	5360553	1053961	.7159	.563
qlearning	5305425	1062291	.7085	.5675
rlearning	**6350832**	**1163147**	**.8481**	**.6213**
sarsa	5292572	1058638	.7068	.5655
sarsa_lambda	5258845	1031845	.7023	.5512
sarsa_lambda_replacetrace	5169090	1035076	.6903	.5529
watkins_naive_q_lambda	5309447	1046734	.7091	.5592
watkins_naive_q_lambda_replacetrace	5165850	1037576	.6899	.5543
watkins_q_lambda	5194671	1038007	.6937	.5545
watkins_q_lfa	5316508	832559	.71	.4447

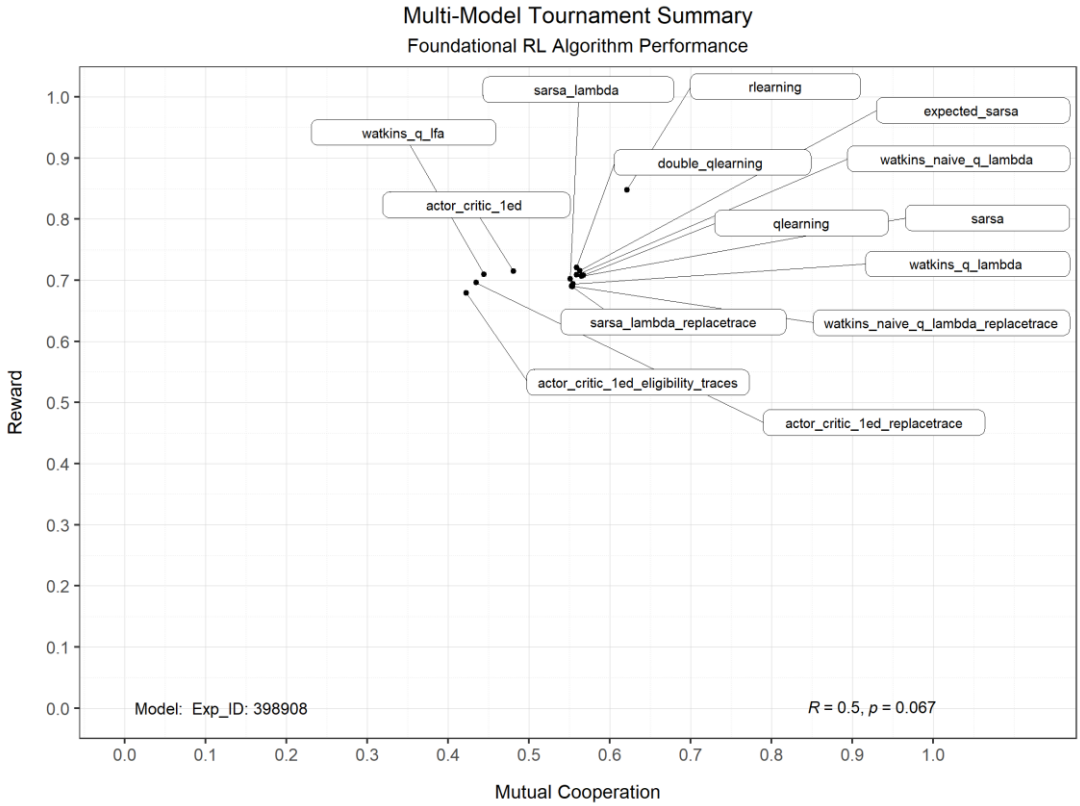


Figure 4.7: Multi-Model Tournament Summary, Foundational RL Algorithm Performance.

4.2.2.4 Final Round-Robin

To complete the tournament experiments, a selection of algorithms from each group participate in a final round of tournament competition. The algorithms are selected for this round on two criteria: firstly, the top two or three performers from each algorithm group go through; and secondly, on an algorithm's characteristics making it a useful comparator, for example, the *Random* algorithm gives a useful yardstick for assessing the other algorithms. In addition, the *Tit-for-Tat* algorithm and *Fictitious Play* are also included in this final round.

The number of matches over a single game model is 66. There are a total of 9504 matches over 144 game models with twelve algorithms playing over 66 matches of 1,000 timesteps each, per game model.

Table 4.12 summarises results for the $Final\ Round$ set with respect to *TR* and *MCR*, over the single game model **g111**. A plot of this data is shown in **Figure 4.8**. The Pearson correlation co-efficient of the linear relationship between *TR* and *MCR* shows a positive correlation of $r(10) = -.6419,\ p = .02444$.

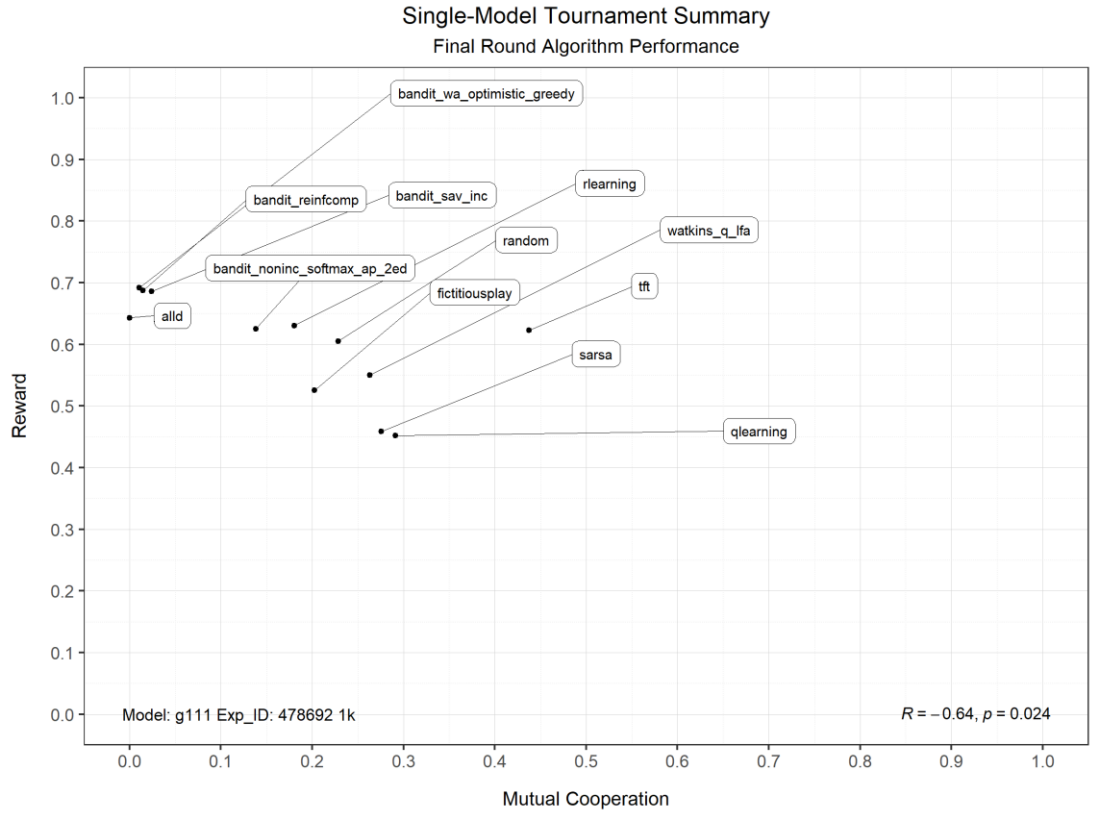


Figure 4.8: Single-Model Tournament Summary, Final Round Algorithm Performance.

Table 4.13 summarises the results for the *Final Round* set with respect to *TR* and *MCR*, for all game models in $\boldsymbol{G}$. A plot of this data is shown in **Figure 4.9**. The Pearson correlation co-efficient of the linear relationship between *TR* and *MCR* shows a positive correlation of $r(10) = .732487,\ p = 6.743 \times 10^{-3}$.

Table 4.12: Single-Model Tournament Summary: Final Round Algorithm Performance over game model **g111**.

TR: Total Reward, or the proportion of available reward that an algorithm receives. **MCR:** Mutual Cooperation Rate, being the share of total available opportunities for mutual cooperation in which an algorithm shares. Bolded values indicate the highest value in each column.

Single-Model Tournament Summary Algorithm Performance –Final Round, g111, 1k				
Algorithm	**Total Reward**	**Total MC**	**TR**	**MCR**
alld	28294	0	.6431	0
bandit_noninc_softmax_ap_2ed	27509	1523	.6252	.1385
bandit_reinfcomp	30253	164	.6876	.0149
bandit_sav_inc	30189	265	.6861	.0241
bandit_wa_optimistic_greedy	**30443**	119	**.6919**	.0108
fictitiousplay	23129	2228	.5257	.2025
qlearning	19886	3202	.452	.2911
random	26617	2515	.6049	.2286
rlearning	27740	1988	.6305	.1807
sarsa	20176	3034	.4585	.2758
tft	27403	**4814**	.6228	**.4376**
watkins_q_lfa	24204	2892	.5501	.2629

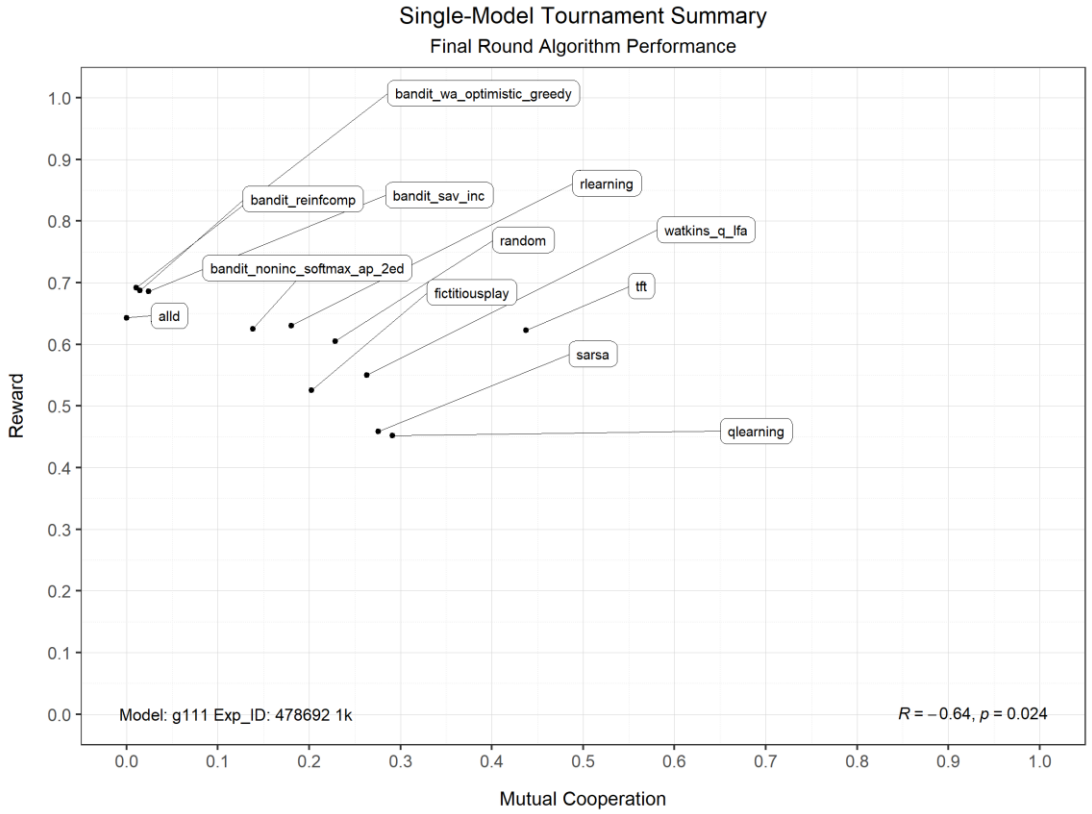


Figure 4.8: Single-Model Tournament Summary, Final Round Algorithm Performance.

Table 4.13: Multi-Model Tournament Summary: Final Round Algorithm Performance.

TR: Total Reward, or the proportion of available reward that an algorithm receives. **MCR**: Mutual Cooperation Rate, being the share of total available opportunities for mutual cooperation in which an algorithm shares. Bolded values indicate the highest value in each column.

Multi-Model Tournament Summary
Algorithm Performance – Final Round, 1k

Algorithm	Total Reward	Total MC	TR	MCR
alld	3931194	436185	.6205	.2754
bandit_noninc_softmax_ap_2ed	4080019	604209	.6439	.3814
bandit_reinfcomp	**5116161**	912977	**.8075**	.5764
bandit_sav_inc	5001182	902584	.7893	.5698
bandit_wa_optimistic_greedy	5078001	917000	.8015	.5789
fictitiousplay	4241089	715662	.6694	.4518
qlearning	4049706	858120	**.**6392	**.**5417
random	4020769	622367	.6346	.3929
rlearning	4923296	**926831**	.777	**.5851**
sarsa	4062560	859571	.6412	.5427
tft	4150896	771108	.6551	.4868
watkins_q_lfa	4125885	689684	.6512	.4354

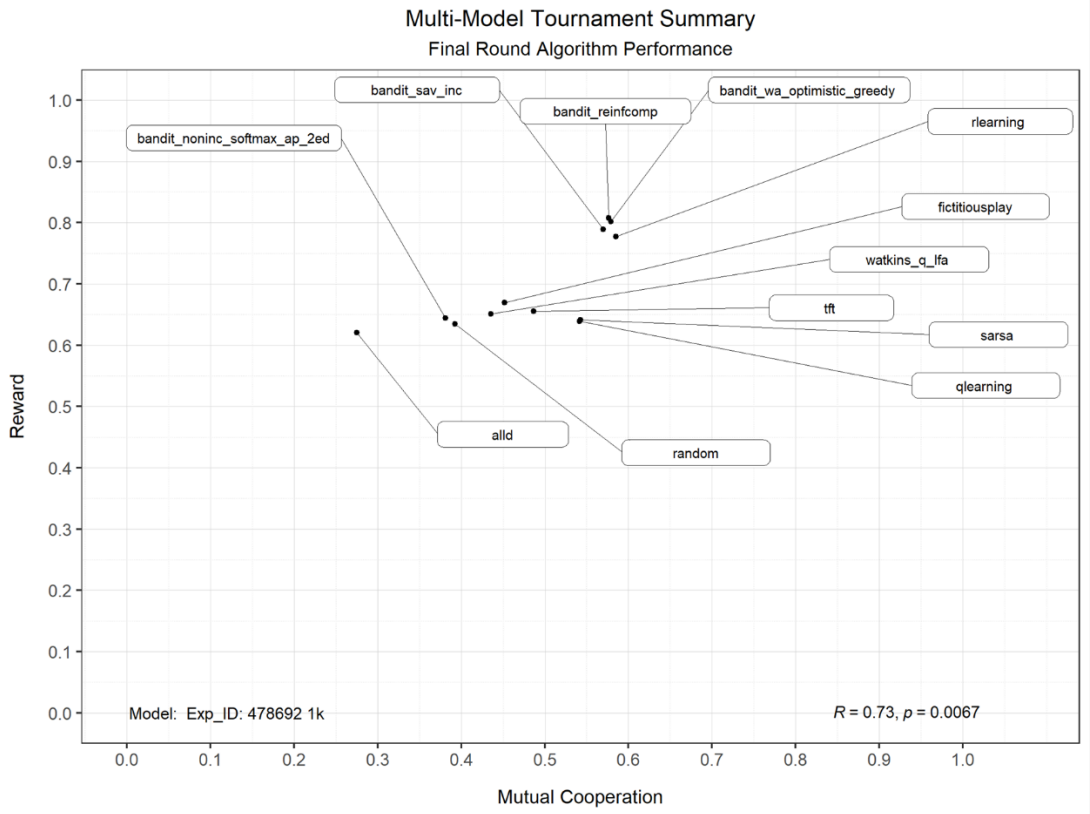


Figure 4.9: Multi-Model Tournament Summary, Final Round Algorithm Performance.

4.3 Discussion

The multi-model experiment type allows the assessment of algorithm performance in individual game models as well as across the set of game models in $\boldsymbol{G}$. The primary interest in this experiment series is the correlation, if any, between the metrics of reward, *TR*, and cooperation, *MCR*, as discussed in the introduction any correlation between these two metrics can be viewed as a proxy for the *folk theorem* tenet of 'compromise'. Summary results of correlation of each algorithm's gained share of potential mutual cooperation with their received share of available total reward, for all experiment instances, is shown in **Table 4.14**.

Table 4.14: Correlation of **TR** (reward) and **MCR** (mutual cooperation rate): All Algorithm Sets, All Models.

Correlation Co-efficient Summary Model by Algorithm Set, Length					
Algorithm Set	**Length (timesteps)**	**degrees of freedom**	**Model**	**r, Pearson**	**p-value**
Game-Theoretic	1k	4	Single Multi	−.28 −.13	.595 .802
Binary Bandit	1k	15	Single Multi	−.87 .99	7.33×10^{-6} 7.7×10^{-14}
Foundational RL	1k	12	Single Multi	−.59 .5	.026 .067
Foundational RL	10k	12	Single Multi	.56 .93	.036 1.72×10^{-6}

Beginning with the game-theoretic round-robin, the data from each match pairing is tabulated per algorithm, e.g., as shown in **Table 4.6**. As expected, across all match pairings, *Always Defect* never engages in a single act of mutual cooperation. The algorithm *Naïve Bully*, which first plays an act of defection, then forever after plays the opposite of what its opponent has played, does not fare much better in gaining a share of available mutual cooperation, at a low .1012 (~10%). In keeping with Axelrod's (1980a, 1980b) findings, the algorithm *Tit-for-Tat* has the highest share of mutual cooperation of all the algorithms over this game model (**g111**).

In contrast, when assessed over all game models in $\boldsymbol{G}$, *Tit-for-Tat* no longer scores the highest share of mutual cooperation, but does come out with the highest reward (see **Table 4.7**).

The performance of *Always Defect*, across all games (with all of their varying strategic dynamics), is somewhat better with respect to mutual cooperation than for the single game model **g111**. This is due to *Always Defect's* action, of always defecting, mapping to an NBS location in some games. Similarly, for *Tit-for-Tat* the variety of games lowers the share of mutual cooperation that the algorithm experiences, however *Tit-for-Tat's* share of *TR* is now the highest of all six game-theoretic algorithms.

A general comment that can be made is that the variety of strategic dynamics (NBS locations, Nash equilibria, Maximin) allows all six algorithms to perform better across the *rRGS* than they do on **g111**, as can be seen in **Figure 4.2** and **Figure 4.3**. Further, evaluating any supposed link between *MCR* and *TR*, with these algorithms, is not illuminating. Neither the single-model ($r(4) = -.28,\ p = -.595$), nor the multi-model ($r(4) = -.13,\ p = .802$) exhibit any statistically significant correlation between the two metrics *TR* and *MCR*.

As the algorithms (four of which are deterministic, the remaining two are stochastic) are mostly playing fixed actions, i.e., a pure strategy, the correlation test is more influenced by the deterministic nature of the algorithms than by an agent learning to exploit the hypothesised relationship between *TR* and *MCR*, or by the strategic dynamic of the game model itself.

The next two experiment instances examine the performance of the Binary Bandit algorithm set. Firstly, in the single-model case, **Figure 4.4** shows all of the bandits clustering at a mid-range for *TR*, and at very low levels for *MCR*. Correlation of the two metrics, *TR* and *MCR*, for this instance, is statistically significant, and is negative: $r(15) = -.87,\ p = 7.33 \times 10^{-6}$. In the multi-model case, shown in **Figure 4.5**, all of the algorithms have improved, with a tight clustering towards the high end of their achievement. Correlation of the two metrics, *TR*, and *MCR*, for the multi-model instance, is statistically significant, and is positive: $r(15) = .99,\ p = 7.7 \times 10^{-14}$. In the first instance this suggests that the game model **g111** is inversely correlating cooperation with reward; in other words, Prisoner's Dilemma does not make it easy to unilaterally cooperate.

The strong positive correlation for the entire model $\boldsymbol{G}$ suggests the other 143 game models afford more opportunity for engaging in acts of cooperation than does the canonical form **g111**. An important observation at this point is that bias across the *RGS* is not uniform, for example, layer three is biased towards high payoff, mutually cooperative, outcomes (Bruns, 2010).

Also of interest in the Binary Bandit multi-model tournament is the performance of the 'softmax' variants compared to the rest of the bandits, as the softmax variants perform worse across the board than their brethren, as can be seen in **Figure 4.5**, where the six softmax variants are in two groups, both of which are lower on both axes than the other eleven algorithms.

The correlations of the next two experiment instances, the single-model *fRL* algorithm set, $r(12) = -.59,\ p = .026$, and the multi-model *fRL* algorithm set, $r(12) = .5,\ p = .067$, are both less strongly correlated than were the Binary Bandits; but, as for the Binary Bandits, the *fRL* algorithms show a similar pattern in the game model **g111**, being moderately inversely correlated with cooperation; while also being moderately positively correlated for the overall model $\boldsymbol{G}$.

With the exception of *R Learning*, the algorithms in the multi-model instance, **Figure 4.7**, are more tightly clustered than the algorithms in the single-model instance, **Figure 4.6**. The robust performance of *R Learning* over the complete model $\boldsymbol{G}$ is in contrast to the generally weaker performance of the *Actor/Critic* variants.

These findings confirm several matters in the context of this research. Firstly, that Prisoner's Dilemma is a functionally challenging environment in which to cooperate. Secondly, that deterministic algorithms are at the mercy of the environment, as they cannot adapt. Thirdly, while showing overall improved ability to adapt to a range of environments, Binary Bandits appear to be susceptible, as Sutton and Barto (1998) conjectured, to a 'difficulty', such that it can be argued this difficulty is manifested by the unique strategic dynamics of Prisoner's Dilemma (see §4.1.2.2). Fourthly, the *fRL* algorithms demonstrate better ability to adapt, i.e., *learn*, than the bandits do in the game model **g111**, but perform slightly less well than the bandits in the multi-model. This lower performance over the whole model may be due to *fRL* algorithms converging on minima more readily, or it could be due to initial hyperparameter settings[52] affecting the algorithms and their subsequent trajectory through reward space. Regardless, it would be instructive to perform generalised linear model analysis on the separable components of the *fRL* algorithms; in addition, further ablation studies would likely be informative. Lastly, the variation in correlation discussed above reveals interesting properties both of the algorithms in the study, and the game models themselves, particularly that where an algorithm can modify its strategy to engage in increased mutual cooperation that algorithm will have a proportionate increase in reward (*fRL*, multi-model, 10k: $r(12) = .93,\ p = 1.72 \times 10^{-6}$).

The derivation of this proxy for the folk theorem tenet ordinarily referred to as 'compromise' is not intended in this study to be exhaustive or highly detailed, however this result does confirm that the computational research framework constructed for the experiments in this thesis affords expected learning agent behaviour to be observed.

In addition, scoping, writing, testing, running, and analysing the results from a multi-model experiment type has produced much data about algorithms and their learning processes, however fine-grained analysis of these details is out-of-scope for this thesis. Equally of interest is the use of the model $\boldsymbol{G}$ as a domain of environments. That the character of the game model **g111** (Prisoner's Dilemma) is difficult is unsurprising,

[52] Initial hyperparameter values, for those algorithms that have hyperparameters, were selected through prior parameter studies and are listed in Appendix B.2.2**.**

however, it is of practical utility to be able to contrast the behaviour of an algorithm in one standardised game model with the algorithm's behaviour in another, standardised, game model.

Chapter Five

Representational Equivalence

"To see a world in a grain of sand
And a heaven in a wild flower
Hold infinity in the palm of your hand
and eternity in an hour."

—William Blake[53]

This chapter aims to establish whether algorithms exhibit consistency in their behaviour across isomorphic representations of the Prisoner's Dilemma game model (see §2.3.1), in a Markov Game environment $\boldsymbol{M}$ (see §2.2.2), under the constraints, and with the assumptions, specified in §1.3. Various ad-hoc observations gathered in the scoping and implementation of the multi-model tournament experiment series (see previous chapter) suggest that agents, or rather, algorithms, *do not* exhibit equivalent, or consistent, behaviour under this condition of change in the representation of the game model. The environment, $\boldsymbol{M}$, that encompasses the game model, is defined with an observation function $\boldsymbol{O}$. An agent's interface with the game environment is by its own observation function $\boldsymbol{O_i}$, through which it is able to receive inputs (rewards) from the environment. This experiment series considers the agent's handling of said inputs as reified internal state of the observation function $\boldsymbol{O_i}$ and examines the effects on game outcome distributions as a function of the treatment applied to the representation: all else is held constant.

The literature on this topic is varied, ranging from a lack of acknowledgment of any consequence of the effect, to an acceptance of the effect such that it motivates methodologies (specifically, the quintupling of the elements in the game model space) to handle the variance in behaviour (Crandall et al., 2018b). For example, there is the view that linear transformations of game models are functionally equivalent (Robinson & Goforth, 2005). Utility theory (von Neumann & Morgenstern, 1944) is defined over positive linear transformations of additivity and multiplicativity. Both normative game theory and utility theory posit that the dynamics of a game model remain unchanged when the model's payoff values undergo a positive linear transformation (see §2.3.1.1, and §2.3.2). The expectation under these views is that a variant of Prisoner's Dilemma with canonical values will provide the participants of the game with functionally equivalent decision-making situations as would a variant of the model where canonical values have been, for example, normalised (normalisation being just one of an infinite number of

[53] From *Auguries of Innocence* by William Blake (1803; 1983)

possible positive transformation functions). 'Functionally equivalent' here means that a rational player will retain their normative policy between isomorphic variants of the game model. However, in repeated studies with humans, Rapoport and Chammah (1965) found that the observed behaviour of participants did not conform to the rational expectations of game theory. Kreps et al. (1982) came to similar conclusions regarding players and their 'irrationality', likewise, Sally (1995, p. 58), in a 35-year meta-review of the field, is led to assert that "a model of pure self-interest is usually inconsistent with the results of experimental decision-making". Sally relates how the dichotomy between expectation and observation was experienced in the very first explicit trial of the Prisoner's Dilemma, by its creators Flood and Dresher. Sally concludes that the review found that expectations of rationality were not being confirmed by experimental results. Hamburger (1973) asserts that transformations of a game model *can* affect the strategic properties, or dynamics, of a game; further, that some transformations do not retain isomorphism from the original model, and as such become a new game with an altered set of dynamics.

Looking to the literature of other computational paradigms, within a game-theory aegis, adds several detailed explorations of this issue of representation and stability of behaviour. Primarily, the work of Ashlock and Kim (2008) investigated the role of representation in an evolutionary computational context and found, in one series of experiments, that "all three representations sample the strategy space in a radically different manner" (2008, p. 647). In another series of experiments, Ashlock and colleagues concluded that "changing the payoff matrix, within the bounds permitted by the defining inequalities of prisoner's dilemma, yields different results" (Ashlock et al., 2010, p. 225). Also in an evolutionary context—specifically an empirical investigation into a topological representation of Prisoner's Dilemma—Robinson and Goforth (2005, p. 156) assert that while the topological properties of ordinal games define the relationships between the games, the topology "is insufficient for describing and predicting patterns of behaviour" which suggests that an agent's behaviour is not necessarily a function of the game model.

Returning to the online computational paradigm, Crandall et al. (2018b, pp. 8–10) acknowledge the issue in supplementary material: for various learning algorithms, "actual payoff values assigned to the ordinal preferences … can, and often do, impact the behaviors of some algorithms in repeated games". Marwala (2021) writes that as machines are bound by similar constraints as human subjects (e.g., *imperfect-* and *incomplete-information*), machines will, in turn, be limited. Simon's (1955) *bounded rationality*, which considers decision-making situations as generally being limited by the available information, such that 'satisficing', or finding acceptable solutions, is the norm; rather than optimisation ('maximisation'). Conlisk (1996) surveys economics literature to ask if *bounded rationality* is a practical alternative model to the rational *unbounded rationality* view, arriving at a determination that both have important roles and offer substantial insights to the problems to which they are applied, but as a heuristic, rather than an idealised model.

Whether an algorithm has *complete-*, *incomplete-*, *perfect-*, or *imperfect-information*, it may be crucial for the algorithm to have awareness of its operational limitations, so as to make informed inference, and build models of the world within a defined scope. Knowledge of how these constraints affect the machine's decision-making capability is therefore of critical importance to the context of this thesis; being *machinic*; i.e., of and

about algorithmic decision-making in emergent software systems. It is potentially problematic to accept the view of *rational consistency* being applied to learning agents without an empirical examination of the issue. Rather than defaulting to an acceptance of the view, *a priori*, that machines are 'neutral', i.e., unbounded, such that they will automatically conform to the expectations of rationality when under study; it is preferable to look more closely at the issue by, first, constructing a null hypothesis: that computational learning agents will exhibit *equivalent policy* under conditions of positive linear transformation, and then second, put the hypothesis to the test. The term *equivalent* means that where an agent may exhibit a certain distribution over the four game outcomes of a game model under one specific representation of the environment, we would expect the agent to exhibit a similar distribution *when the only factor that has changed* is the representation of the environment where the representation has to have been transformed in a way that, theoretically, preserves the dynamic structure of the game, i.e., the representation is isomorphic.

The null hypothesis can be re-stated as an expectation over an experiment series: that for any isomorphic transformation of a given game model, the outcome distribution for any one algorithm should be approximately equivalent for that same algorithm between the original and the transformed game model. If the null hypothesis does not hold then the assertion can be made that significant variance in the behaviour of algorithms is indeed a real effect: that is, agent behaviour is not equivalent under isomorphic representations of environmental reward filtered through the agent's observation function, $\boldsymbol{O_i}$. Robinson and Goforth also asserted that "any ordinally equivalent game is also a Prisoner's Dilemma" (2005, p. 6). If this is the case, then there should be no observed, significant, variance in algorithm performance. On the other hand, if this is not the case—if the agents themselves exhibit statistically variant distributions in their behaviour—then we may have to consider that agents are *not* purely 'rational', in the normative game theory sense. In one study, the reward values an agent receives may map directly to the Prisoner's Dilemma game model, but if agent behaviour is statistically, i.e., significantly, variant for an isomorphic representation of the Prisoner's Dilemma, then any prediction of agent behaviour will have to be made with less confidence.

The motivation for studying the behaviour of machines in an ethological manner was introduced in §1.1.1. An initial key step is to assume a *tabula rasa*, and then observe and measure how an agent actually behaves in an environment—rather than beginning outright with a pre-existing normative expectation. A further consideration in assessing whether this is a valid concern to pursue is to ask how likely is it that an environment will change such that this issue becomes relevant. By definition, a non-stationary environment (see §2.2) implies that an agent's relationship to its environment can change as the environment is mutable. Being the contact point to the environment, the representation internal to the agent's observation function $\boldsymbol{O_i}$ may in turn fluctuate. Additionally, an agent's internal state representation may transform as part of normal algorithmic operation; or it may be that an agent's method of utility extraction from reward signals is altered by design, or computationally. Alternatively, an agent or algorithm may be entirely transplanted from one use case or application to another.

A generalised perspective on these processes captures those occasions in everyday life that entail an implicit cast between representations, for example, whenever we (in the real-world) fix a price-point to a preference, or conversely, when compromising on

features of a purchase with a known price. In other words, whenever we translate from scalar (or cardinal) values to ordinal preferences, and vice versa.

To recap this introduction, this chapter sets out an experiment series that has the *sole goal* of confirming whether behavioural non-equivalence in learning agents, under isomorphic representations, *is a real effect*, under the constraints and assumptions set out in §1.3.

The next section, §5.1, defines formal equivalence with respect to transformations of the game model representation; this is followed, in §5.2, by an account of the methodology adopted for this experiment series; before presenting the results of the experiments in §5.3. Discussion of immediately relevant issues that arise from the findings is presented in §5.4.

5.1 Asserting Formal Equivalence

This experiment series examines four representations of the Prisoner's Dilemma in normal-form, shown in **Figure 5.1a) - d)**. A depiction of the canonical scalar-valued Prisoner's Dilemma is shown in **a)**; while **b)** shows the equivalent *RGS* game model **g111** represented by scalar values that denote the ordinal preference rank of the outcome for each player (see §2.3.3.4); **c)** depicts the canonical scalar-valued Prisoner's Dilemma transformed to normalised values; lastly **d)** shows the ordinal representation where the canonical scalar values have been transformed to normalised values.

a)

pd scalar		Column C	Column D
Row	C	3, 3	0, 5
	D	5, 0	1, 1

c)

pd scalar_norm		Column C	Column D
Row	C	0.6, 0.6	0, 1
	D	1, 0	0.2, 0.2

b)

g111 ordinal		Column C	Column D
Row	C	3, 3	1, 4
	D	4, 1	2, 2

d)

g111 ordinal_norm		Column C	Column D
Row	C	0.667, 0.667	0, 1
	D	1, 0	0.333, 0.333

Figure 5.1: Four Game Model Representations. In turn: **a)** canonical Prisoner's Dilemma with scalar values; **b)** scalar values are replaced by the payoff's ordinal preference rank; **c)** scalar values are normalised; **d)** ordinal preference values are normalised. The Robinson and Goforth (2005) topology refers to the ordinal Prisoner's Dilemma as game model **g111**, see §2.3.3.4 and Chapter Three**.**

To assess formal equivalence, the four social dilemma inequalities (discussed in §2.3.2) provide a means to assert the preservation of the structural dynamic of a game model under various transformations, while retaining the strategic dynamics attributed to the

original form as an instance of a specific dilemma. Each of the four inequalities operate over the four game outcomes shown in **Figure 5.2**.

Formal equivalence of **pd:scalar** with the three alternative representations under examination in these experiments is established by observing that the social dilemma inequalities that define Prisoner's Dilemma do hold, as detailed in **Table 5.1**, confirming that these four representations of Prisoner's Dilemma are structurally equivalent. **Figure 5.3** illustrates the relationship between each representation.

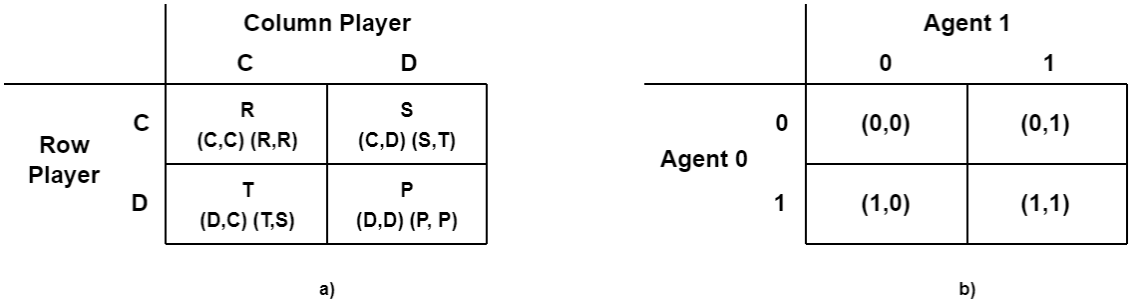


Figure 5.2: Semantic labels attributed to outcomes in the Prisoner's Dilemma. In **a)** the labels **R**, **S**, **T**, **P** indicate the game outcome in the top row of each cell, obtained from the joint play of actions **C** (cooperate) or **D** (defect) as a shared outcome **CC** (**R**, **R**), **CD** (**R**, **T**), **DC** (**T**, **S**), **DD** (**P**, **P**). In **b)** the normal form matrix, sans semantic labels, generalised to indicate game outcome by cell index. Joint actions **(0,0)**, **(0,1)**, **(1,0)**, **(1,1)** correspond to semantic outcomes **CC**, **CD**, **DC**, **DD**, and likewise to shared outcomes **R**, **S**, **T**, **P**. The generalised normal-form informs the evaluation of studies in Chapter Four, Five, and Six. *Reproduction extracted from* **Figure 2.2**, §2.3.1**,** *this document.*

Table 5.1: Social dilemma inequalities over four representations of Prisoner's Dilemma.

	Inequality	Scalar	Scalar Normalised	Ordinal	Ordinal Normalised	Equivalent
(1)	$R > P$	$3 > 1$	$0.6 > 0.2$	$3 > 2$	$0.667 > 0.333$	$True$
(2)	$R > S$	$3 > 0$	$0.6 > 0$	$3 > 1$	$0.667 > 0$	$True$
(3)	$2R > T + S$	$6 > 5 + 0$	$1.2 > 1 + 0$	$6 > 4 + 1$	$1.334 > 1 + 0$	$True$
(4)	$T > R$ ***or***, $P > S$	$5 > 3$ $1 > 0$	$1 > 0.6$ $0.2 > 0$	$4 > 3$ $2 > 1$	$1 > 0.667$ $0.333 > 0$	$True$

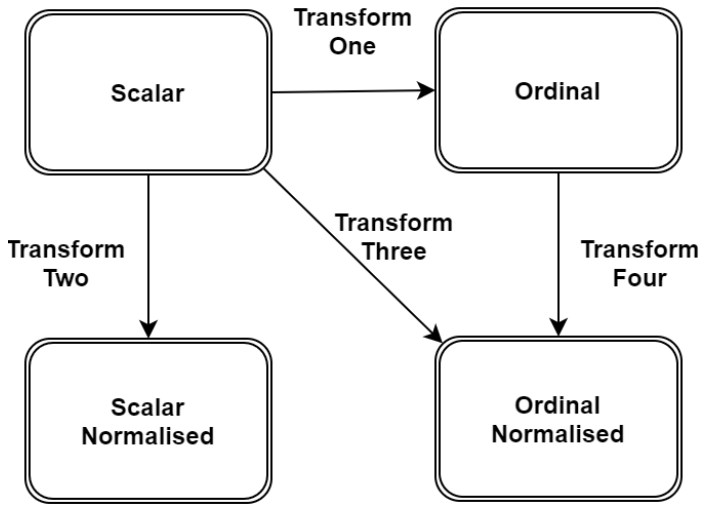


Figure 5.3: Four isomorphic operations over the Prisoner's Dilemma representation.

5.2 Methodology

In the highly constrained setting of a Markov Game environment (see §2.2.2), a constancy in experimental procedure can be enforced. Between two instances of an experiment, all aspects of the implementation can be held constant: the agents and algorithms utilised; the configuration of hyperparameters for each algorithm; the platform upon which the experiments are performed—all remain the same, with the only exception to exogenous conditions not remaining unchanged is a single treatment to the experimental design; specifically, the isomorphic transformation of the representation of the game model's agent inputs (reward, or payoffs) in the agent's observation function $\boldsymbol{O_i}$. Endogenously, learning algorithms incorporate stochasticity into their process, and this must be considered as a potential source of variance in behaviour as well. This consideration is addressed in Appendix A.2.

The Wilcoxon Signed Rank test (Wilcoxon, 1945) is commonly used to analyse variance between two populations that have been differentiated by a single treatment. Here, the populations are the sets of behavioural profiles obtained by pairing learning agents in a *symmetric selfplay* experiment type. The treatment is the transformed game model representation. Applying the Wilcoxon Signed Rank test allows the definition of consistency to be the *non-observation of significant variance* between two populations that have been contrasted by a single treatment, variance being measured over the distribution of the game model outcomes.

The null hypothesis for this experiment series is that *an algorithm's behavioural profile will not exhibit statistically significant variance between structurally equivalent representations of the game model under examination*. If the behavioural profile of an algorithm shows significant variance in comparison to another behavioural profile, where the only difference is a structure-preserving operation over the game model representation, the evidence for rejecting the null hypothesis would be strengthened. Further, this measure of variance may provide insight to an algorithm's cooperative stability across varying environments. In addition to analysis of the aggregate distribution of a behavioural profile further granularity can be achieved by disaggregating each distribution to its four component game outcomes. The statistical test workflow is the same for individual outcome distributions as for aggregate distributions. Each of the experiment instances in this series are implemented as *symmetric selfplay* repeated stage-games along two dimensions—defined by two hyperparameters—where each hyperparameter ranges from (0,1], in 0.1 increments; giving 100 observations for each algorithm. Each observation is the terminal episode-mean distribution of the four cooperative game outcomes CC, CD, DC, and DD[54] over 500 episodes of 1000 timesteps each. As in Chapter Four, the episode length is selected as the median of the episode lengths (100, 1k, and 10k timesteps) used by Crandall et al. (2018b). The 100 instances given by the choice of parameter values gives a grid-search, or parameter sweep, over the entire range of each hyperparameter. The response variable is the distribution of the four game outcomes, measured as the count of each outcome at the termination of each episode, averaged over all episodes. Each experiment instance conforms to the constraints outlined in §1.3.

[54] CC, CD, DC, and DD map to (0,0), (0,1),(1,0), and (1,1), see §2.3.1.

Four representations of the Prisoner's Dilemma game model are shown in **Figure 5.1**: *scalar*, *normalised scalar*, *ordinal*, and *normalised ordinal*. By considering the scalar form as the canonical representation the aim is to compare the behaviour of algorithms between the canonical form, and each of the other three forms. Despite the infinite space of possible transformation functions, only four common mappings are examined.

The four mappings, or transformations, are shown in **Figure 5.3** where: *Transform One* is the operation that converts between the canonical Prisoner's Dilemma game model and the corresponding ordinal game model in the *RGS* (**g111**); *Transform Two* converts between the canonical form and the normalised form, as used, for example, by Crandall and Goodrich (2011); *Transform Three* is the operation that converts between the canonical form and the normalised ordinal form, as used by Crandall et al. (2018a); while *Transform Four* represents the operation of converting from the scalar representation to normalised ordinal.

The collection of experiment instances that is formed by running each algorithm in a single game model representation gives a set of *behavioural profiles* for that representation. The grouping of two such sets is referred to as an experiment group:

- Experiment Group One examines the behavioural profiles derived from the canonical Prisoner's Dilemma scalar payoff values, [0,1,3,5], denoted here as **pd:scalar**; and the ordinally-valued game model representation denoted by **g111:ordinal**, having values [1,2,3,4].
- Experiment Group Two examines **pd:scalar** against the normalised canonical Prisoner's Dilemma **pd:scalar_norm** having values [0, 0.2, 0.6, 1].
- Experiment Group Three examines **pd:scalar** with the normalised ordinal game model **g111:ordinal_norm** having values [0.000, 0.333, 0.667, 1.000].
- Experiment Group Four examines the behaviour of each algorithm between the two representations **g111:ordinal** and **g111:ordinal_norm**, each with values [1,2,3,4], and [0.000, 0.333, 0.667, 1.000], respectively.

The fourteen learning algorithms in this experiment series are listed in **Table 5.2**.

Table 5.2: Algorithms implemented for the equivalence study, and their parameters. Parameter superscript [1,2] denotes paired parameters, computed in range (0,1] with increment of 0.1 giving 100 observation sets for each algorithm, per game model representation.

Algorithm	Parameters			
	Learning	Discount	Trace Discount	Action Selection
Actor/Critic	$\alpha^1, \beta = 0.9$	γ^2		$softargmax$
Actor/Critic, Eligibility Traces	$\alpha^1, \beta = 0.9$	γ^2	$\lambda = 0.9$	$softargmax$
Actor-Critic, Replacing Traces	$\alpha^1, \beta = 0.9$	γ^2	$\lambda = 0.9$	$softargmax$
Q-Learning	α^1	γ^2		$\varepsilon = 0.1$
Double Q-Learning	α^1	γ^2		$\varepsilon = 0.1$
Expected SARSA	α^1	γ^2		$\varepsilon = 0.1$
R Learning	α^1, β^2			$\varepsilon = 0.1$
SARSA	α^1	γ^2		$\varepsilon = 0.1$
SARSA Lambda	α^1	γ^2	$\lambda = 0.9$	$\varepsilon = 0.1$
SARSA Lambda, Replacing Traces	α^1	γ^2	$\lambda = 0.9$	$\varepsilon = 0.1$
Watkins (naive) Q, Lambda	α^1	γ^2	$\lambda = 0.9$	$\varepsilon = 0.1$
Watkins (naive) Q, Lambda, Replacing Traces	α^1	γ^2	$\lambda = 0.9$	$\varepsilon = 0.1$
Watkins Q, Lambda	α^1	γ^2	$\lambda = 0.9$	$\varepsilon = 0.1$
Watkins Q, Linear FA	α^1	γ^2		$\varepsilon = 0.1$

The algorithms in this experiment series are identical to the set of *fRL* algorithms introduced in §2.2.4 and §4.1.2.3. For each algorithm, the parameters being varied are the principal learning rate parameter $\boldsymbol{\alpha}$, and, in all cases except *R-Learning*, the discount factor parameter $\boldsymbol{\gamma}$. Given that *R-Learning* does not use $\boldsymbol{\gamma}$, evaluation is performed over both of its learning rate parameters—$\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$—instead. Fixed values for other parameters in each algorithm are as commonly found in the literature and these are given in **Table 5.2**.

5.2.1 Evaluation Metrics

The evaluation metric of interest is the variance of the distribution of cooperative game outcomes *between* two representations of a game model, as observed in a *symmetric selfplay* paired-parameter study of the fourteen *fRL* algorithms listed in **Table 5.2**.

Each symmetric selfplay match of 1000 timesteps generates a separate vector for each game outcome, one datapoint for each timestep. A single game outcome's distribution can be combined with the other three game outcome distributions to give an aggregate distribution, in addition to the four individual outcome distributions. The aggregated distribution of outcomes attained from a single game model representation forms the behavioural profile of the algorithm under study, for that game model representation.

Also of interest is the metric of an algorithm's peak rate of engaging in the mutual cooperation outcome, or *MCR*. Notwithstanding the inclusion of this metric, as a parameter study the goal is not to necessarily maximise performance (i.e., performance as measured by *total reward over a series of timesteps*, or *TR*). Rather, the goal is to observe any *variance* in an algorithm's behaviour, as exhibited between game model representations. As was mentioned in the discussion of the ethological methodological approach, in §1.1.1, performance maximising algorithms can hinder the process of observing the breadth of machine behaviour (Rahwan et al., 2019). The algorithms in this study are not limited in their search for optimality in any way. Nevertheless, to underline the point, the primary focus of this study is not on 'reward', but on variance.

5.3 Results

The results of the analysis of all fourteen algorithms across each of the game model representations are presented as a set of four experiment groups, each group mapping to a single transformation of the model representation, as described in §5.2.

5.3.1 Normality

The aggregate distribution of each behavioural profile returns a non-normal distribution as found by the Shapiro-Wilk Normality test (Shapiro & Wilk, 1953). Complete normality test data (every algorithm's aggregate and individual outcomes) is located in Appendix B.3.2. The Wilcoxon Signed Rank test is used to establish variance as analysis of the majority of the distributions (278 of 280) via a Shapiro-Wilk Normality test indicate a non-normal distribution—for both aggregate and individual outcome views—as can be

seen in the visual qqplot of the *Watkins Q Linear Function Approximation* algorithm in **Figure 5.4** which gives a Shapiro-Wilk score $W = 0.95401,\ p = 7.698 \times 10^{-10}$.

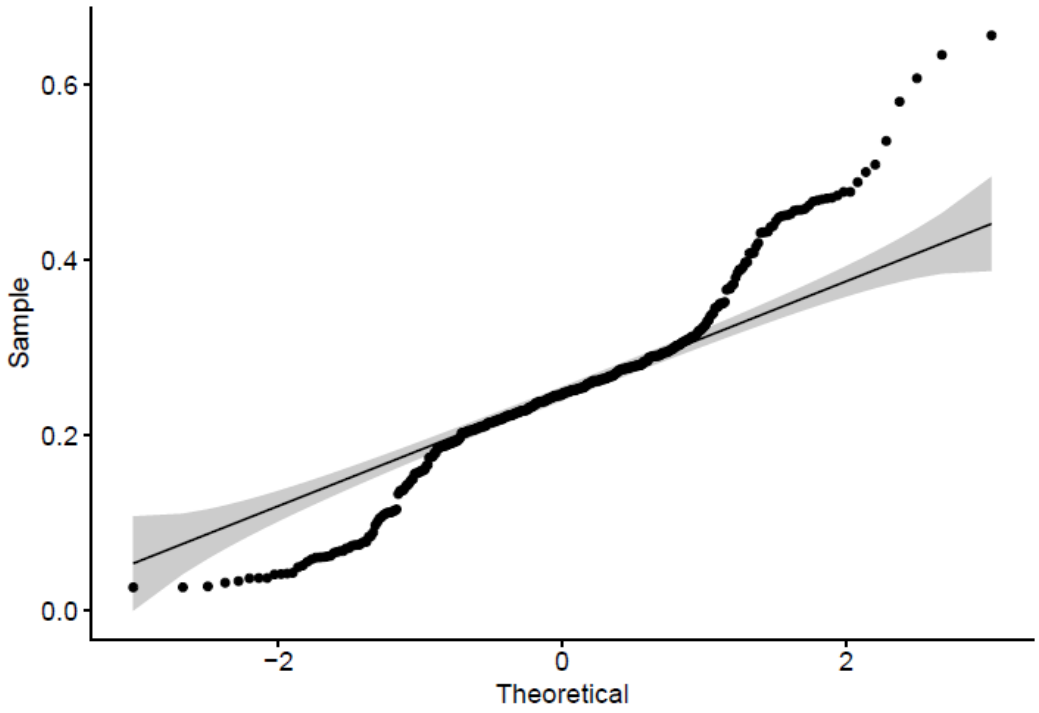


Figure 5.4: QQPlot of aggregated outcomes for *Watkins Q Linear Function Approximation*. Outcomes for **pd:scalar** game model. Shapiro-Wilk normality test $W = 0.95401,\ p = 7.698 \times 10^{-10}$. See Appendix B.3.2.

5.3.2 Peak Cooperative Outcome

In each experiment group, the episode-mean frequency of the peak cooperative outcome (MCR) is included in the results to demonstrate that where some algorithms appear to achieve near-parity in optimal behaviour between some representations, other algorithms exhibit substantial disparity.

5.3.3 Behavioural Profile Visualisation

To visualise the reward surface of an algorithm's behavioural profile the aggregate distribution of the four game outcomes can be separated into individual outcomes and plotted as a faceted 3D surface map, as shown in **Figure 5.5**. The reward surface shows the distribution of each game outcome (on the vertical axis, i.e., the z-axis) obtained by the algorithm *Q-Learning* in *symmetric self-play* in the canonical Prisoner's Dilemma, with representation **pd:scalar**.

The sum of any parameter-pair intersection over all four outcomes is unity. Each facet in **Figure 5.5** illustrates the results for one outcome. Therefore, in each outcome (facet) of the graph the intersection of two parameter values (e.g., $\boldsymbol{\alpha = 0.1, \gamma = 0.9}$) represents an observation of 500 episodes of 1000 timesteps each.

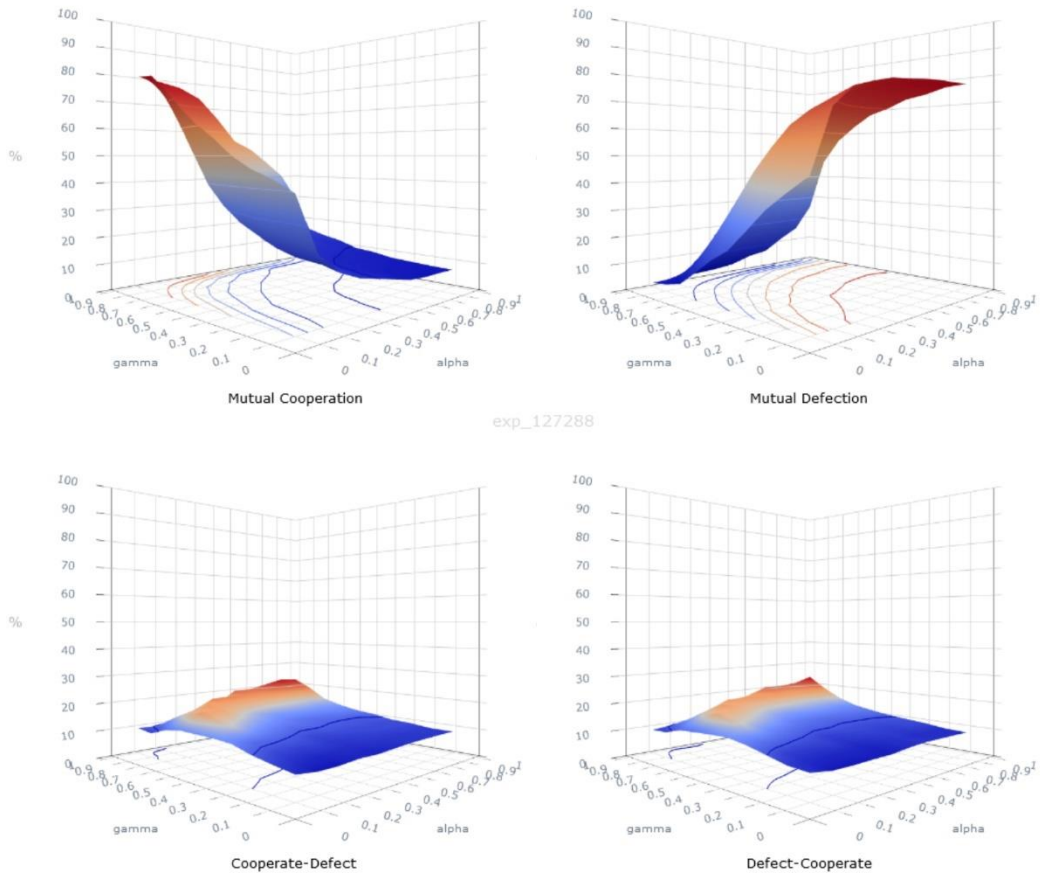


Figure 5.5: Behavioural Profile *Q-Learning*, **pd:scalar**. Outcome Distributions of algorithm *Q-Learning* in a selfplay parameter study in the **pd:scalar** game model representation where each parameter-pair observation is the episode-mean of 500 episodes of 1000 timesteps. Each facet in the figure shows the episode-mean distribution for 100 parameter-pair observations for the labelled outcomes—**Mutual Cooperation** (CC), **Mutual Defection** (DD), **Cooperate-Defect** (CD), and **Defect-Cooperate** (DC). The $\boldsymbol{x}$ and $\boldsymbol{y}$ axes are the respective parameters as depicted in the axis title. The z-axis at any given $\boldsymbol{x}$, $\boldsymbol{y}$ intersection is the percentage of the outcome (labelled per facet) obtained by the agent for the specified hyperparameter pairing. The sum of any $\boldsymbol{x}$, $\boldsymbol{y}$ point across all four outcomes is unity. Exp_ID: exp_127288.

5.3.4 Experiment Group One

Experiment Group One assesses the mapping between **pd:scalar** and **g111:ordinal**. Results of the Wilcoxon tests are shown in **Table 5.3**. Of the fourteen algorithms, eleven have a p-value $< .05$, which indicates that these eleven algorithms do not exhibit equivalence of behaviour between the two representations in this experiment group.

Only three algorithms—*Actor/Critic*, *Actor/Critic with Eligibility Traces*, and *Watkins Q Linear Function Approximation*—do not exhibit variance that can be regarded as significant.

The assessment for statistically significant variance is not immediately apparent when observing boxplots of the algorithms. For example, the boxplot (**Figure 5.6**) of the algorithm *Actor/Critic with Replacing Traces* appears to indicate equivalence through a similar range and median, however it cannot statistically be regarded as equivalent on the strength of the Wilcoxon test ($V = 45337,\ p = .01835$).

The paired lines in **Figure 5.6** map between observation points in the aggregate dataset and show that the mean distribution of any one point in the scalar data does not, as a rule, map to the same point in the ordinal behavioural profile.

Table 5.3: Exp Group One: Scalar (S) ~ Ordinal (O) Aggregated Distribution. Bold indicates statistical significance.

Algorithm	Peak % MCR		Wilcoxon			
	S	O	V	p-value	CI L	CI U
Actor/Critic	32.1	30.2	35742	.071	-0.0176	0.0006
Actor/Critic with Eligibility Traces	46.3	44.9	40978	.705	-0.0036	0.0043
Actor/Critic with Replacing Traces	51.5	42.6	45337	**.018**	0.0019	0.0200
Q-Learning	80.1	81.5	46086	**.010**	0.0010	0.0061
Double Q-Learning	68.9	32.7	34431	**.014**	-0.0059	-0.0006
Expected SARSA	75.7	77.3	27735	**9.08×10^{-8}**	-0.0185	-0.0070
R Learning	25.6	28.3	45520	**.019**	0.0024	0.0177
SARSA	79.9	80.3	46848	**.004**	0.0012	0.0053
SARSA Lambda	68.8	86.7	49435	**5.47×10^{-5}**	0.0171	0.0411
SARSA Lambda, with Replacing Traces	89.0	89	45731	**.015**	0.0005	0.0057
Watkins (naïve) Q, Lambda	78.4	88.1	50043	**1.73×10^{-5}**	0.0270	0.0505
Watkins (naïve) Q, Lambda, Replacing Traces	76.4	89.1	49320	**6.76×10^{-5}**	0.0193	0.0487
Watkins Q, Lambda	87.2	87.8	49355	**6.35×10^{-5}**	0.0029	0.0089
Watkins Q, Linear Function Approximation	47.1	46.2	38262	.478	-0.0013	0.0006

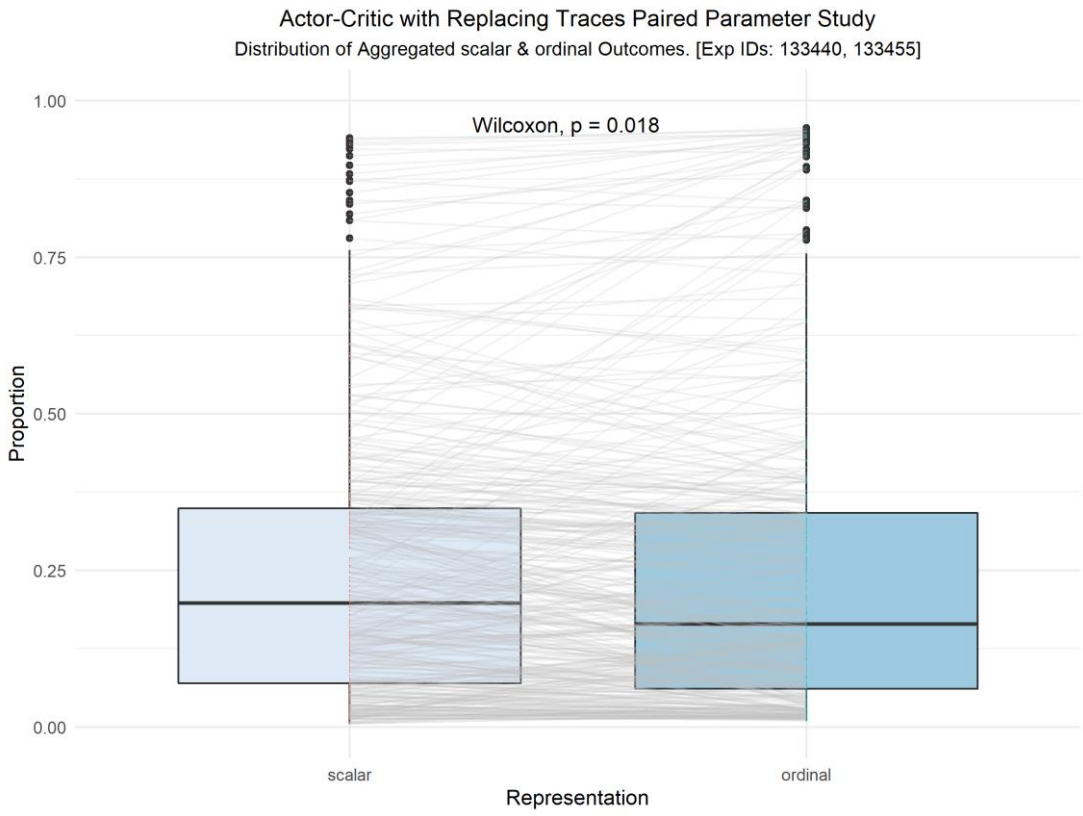


Figure 5.6: Distribution of scalar and ordinal outcomes for *Actor/Critic with Replacing Traces*.

Results of the Wilcoxon Signed Rank test for the aggregate outcomes for all experiments are supplied in Appendix B.3.3. The distribution is more apparent in the raincloud[55] plot shown in **Figure 5.7**, where it is observed that while the overall range and IQR (Inter Quartile Range) metrics of the CC, CD, and DC outcomes would suggest

[55] Raincloud plots (Allen et al., 2021) combine boxplots, violin plots, and points into a single figure.

equivalence, the DD outcome clearly does not, and as can be seen in **Figure 5.8**, each outcome exhibits statistically significant variance. Results of the Wilcoxon Signed Rank test for the DD outcome (DD: $V = 928, p = 4.04 \times 10^{-8}$), and all other results of equivalence testing between both aggregate and individual outcomes, for all experiments, are located in Appendix B.3.3.

An observation that does not map one-to-one to its corresponding observation in the paired-profile suggests some variance, but this does not necessarily indicate that the variance observed from a visual comparison is statistically significant, as can be seen in the boxplot of the algorithm *Actor/Critic's* aggregate outcomes in **Figure 5.9**, where the Wilcoxon test of the aggregated outcomes of the algorithm, between scalar and ordinal representations, fails to provide support for rejecting the null hypothesis ($V = 38811, p = 0.5776$). A raincloud plot of the data illustrates the shape of the distribution of game outcomes in **Figure 5.10**.

Drilling down into the Wilcoxon comparisons of the individual outcomes for this algorithm reveals strong indications that the variance at this lower scale of observation is significant (CC: $V = 4290, p = 1.30 \times 10^{-9}$; CD: $V = 763, p = 1.39 \times 10^{-9}$; DC: $V = 752.5, p = 1.11 \times 10^{-9}$; DD: $V = 4175, p = 1.42 \times 10^{-9}$), as can be seen in **Figure 5.11**.

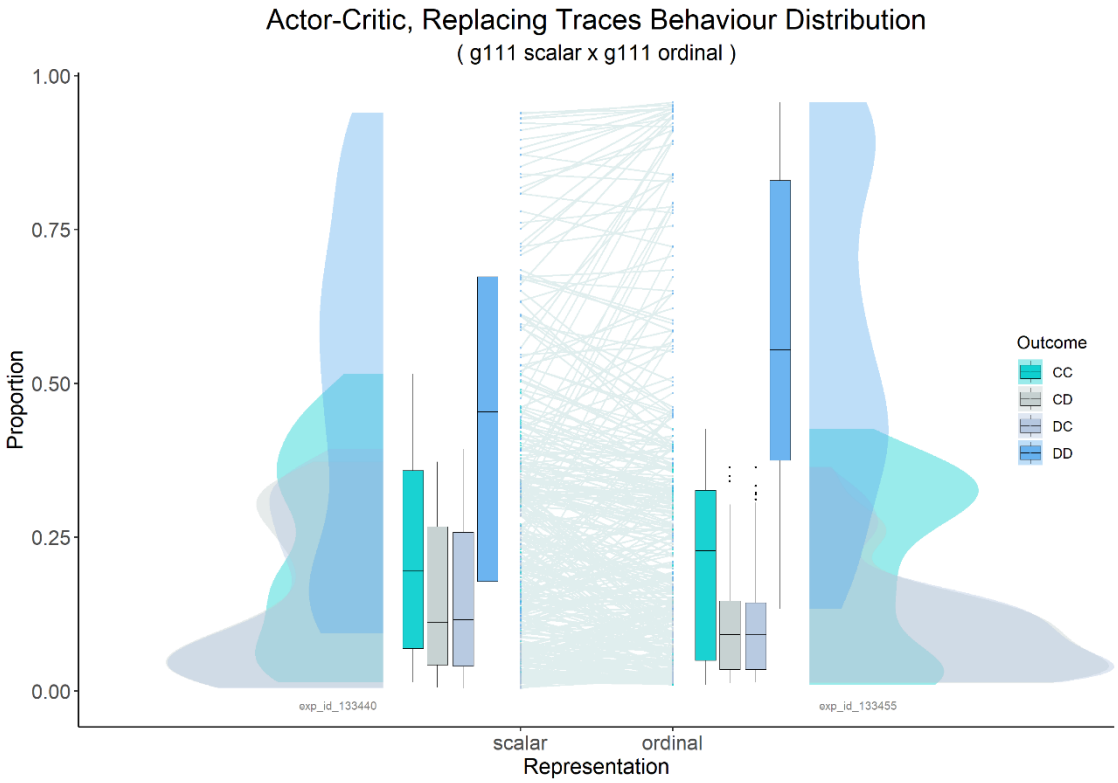


Figure 5.7: Distribution of scalar and ordinal game outcomes for *Actor/Critic with Replacing Traces*.

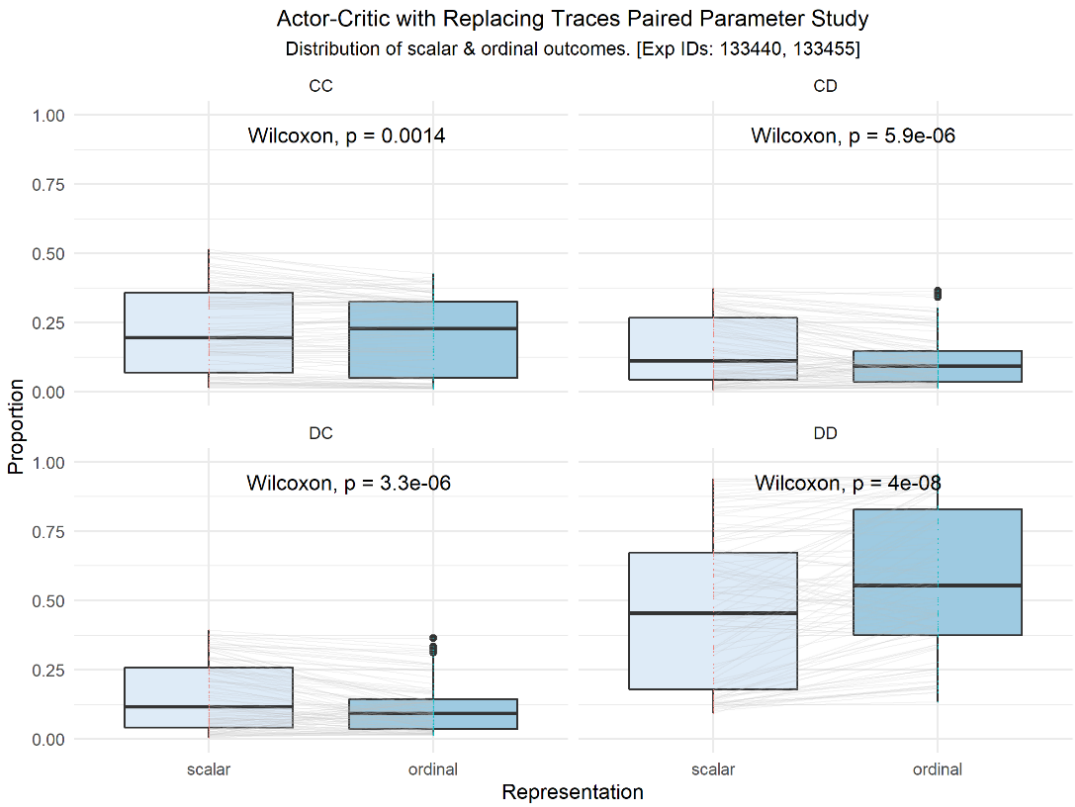


Figure 5.8: Distribution of scalar and ordinal outcomes for *Actor/Critic with Replacing Traces*.

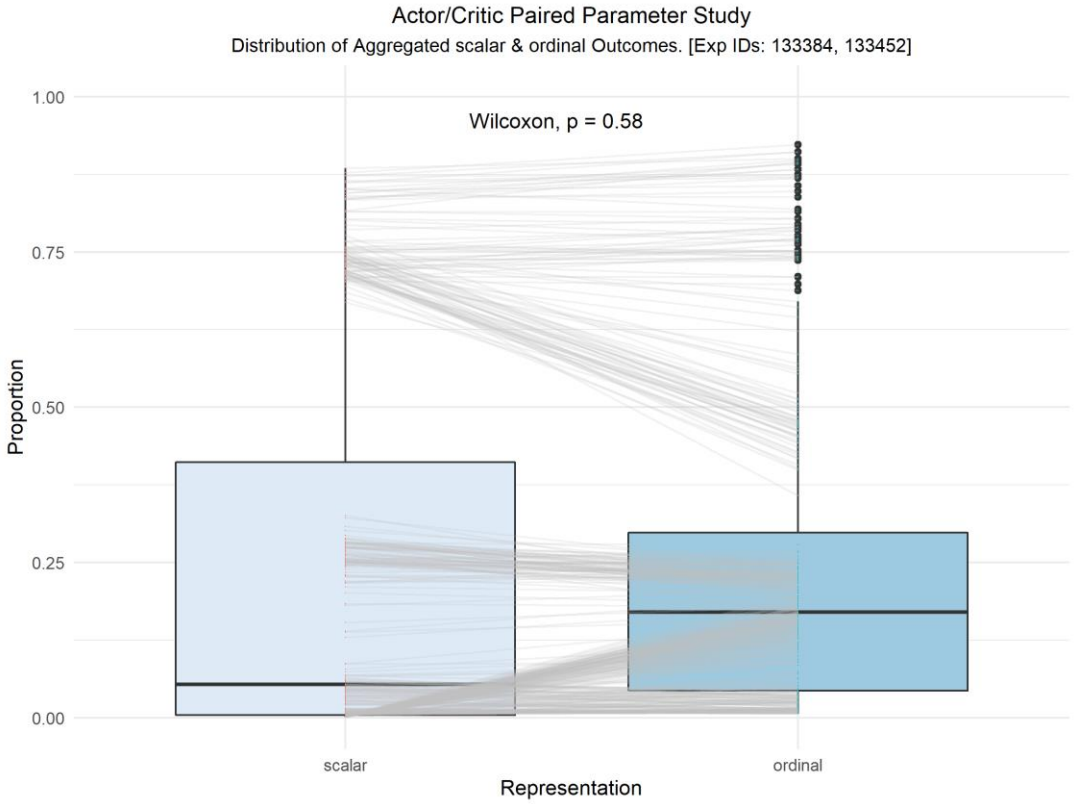


Figure 5.9: Aggregated distribution of scalar and ordinal outcomes for *Actor/Critic*.

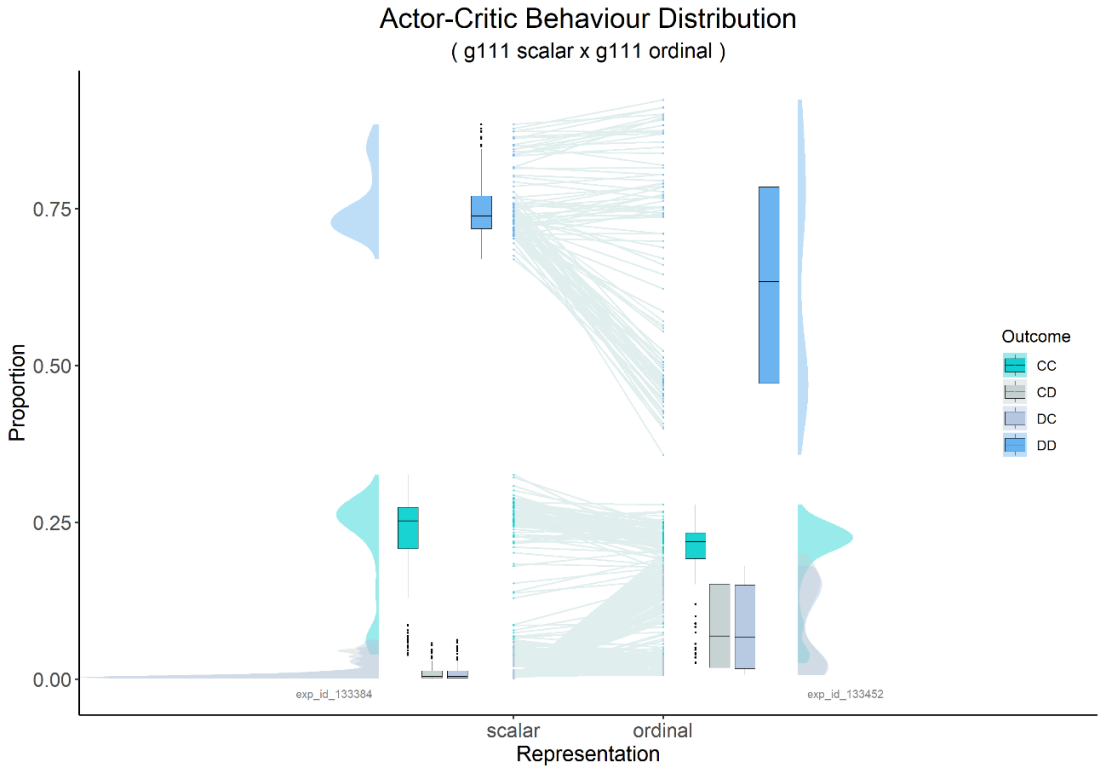


Figure 5.10: Distribution of scalar and ordinal game outcomes for *Actor/Critic*.

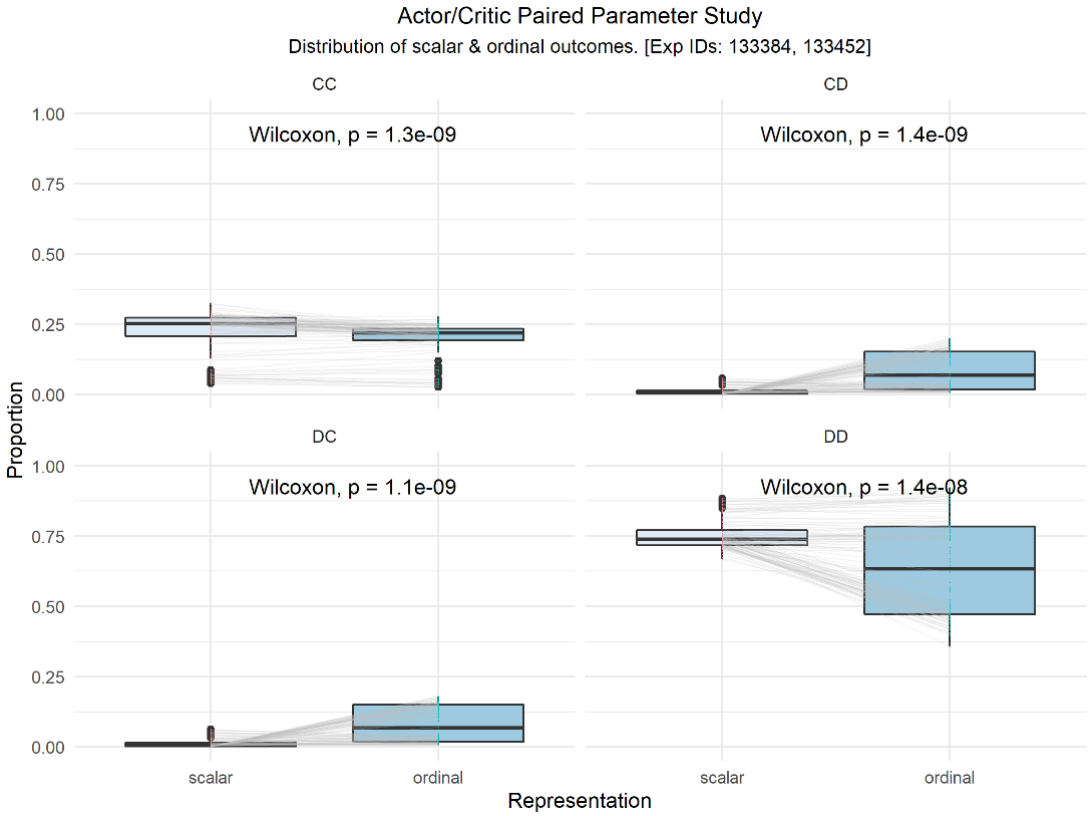


Figure 5.11: Distribution of scalar and ordinal outcomes for *Actor/Critic*.

5.3.5 Experiment Group Two

Results of the Wilcoxon Signed Rank tests for Experiment Group Two are shown in **Table 5.4**. It is apparent that, of the algorithms in this experiment group, four (4) report a p-value $< .05$. This indicates that these four algorithms do not exhibit equivalence between scalar and normalised scalar representations. In contrast to the aggregated outcomes of *Experiment Group One*, those algorithms in *Experiment Group Two*, whose variance is not significant, display paired observations with much less crossing of lines. This is evident in the boxplot of aggregated outcomes for the algorithm *Q-Learning* in **Figure 5.12**, in the raincloud plot that illustrates the shape of the distribution of game outcomes in **Figure 5.13**, and again, in the boxplot for individual outcomes in **Figure 5.14**.

Table 5.4: Exp Group Two: Scalar (S) ~ Normalised Scalar (NS) Aggregated Distribution. Bold indicates significance.

Algorithm	Peak % MCR		Wilcoxon			
	S	NS	V	p-value	CI L	CI U
Actor/Critic	32.1	14.0	41239	.623	-0.0049	0.0074
Actor/Critic with Eligibility Traces	46.3	33.7	46744	.004	0.0026	0.0120
Actor/Critic with Replacing Traces	51.5	44.9	39599	.829	-0.0110	0.0091
Q-Learning	80.1	80.2	40091	.997	-0.0004	0.0004
Double Q-Learning	68.9	68.5	39339	.742	-0.0005	0.0004
Expected SARSA	75.7	75.7	37225	.214	-0.0006	0.0001
R Learning	25.6	23.7	39866	.920	-0.0004	0.0004
SARSA	79.9	80.2	40180	.904	-0.0003	0.0004
SARSA Lambda	68.8	86.6	49945	**2.09×10^{-5}**	0.0123	0.0308
SARSA Lambda, with Replacing Traces	89.0	89.1	39177	.690	-0.0005	0.0003
Watkins (naïve) Q, Lambda	78.4	87.2	49853	**2.50×10^{-5}**	0.0273	0.0488
Watkins (naïve) Q, Lambda, Replacing Traces	76.4	89.2	48331	**2.55×10^{-5}**	0.0120	0.0382
Watkins Q, Lambda	87.2	87.2	39692	.928	-0.0005	0.0004
Watkins Q, Linear Function Approximation	47.1	24.9	39939	.945	-0.0004	0.0004

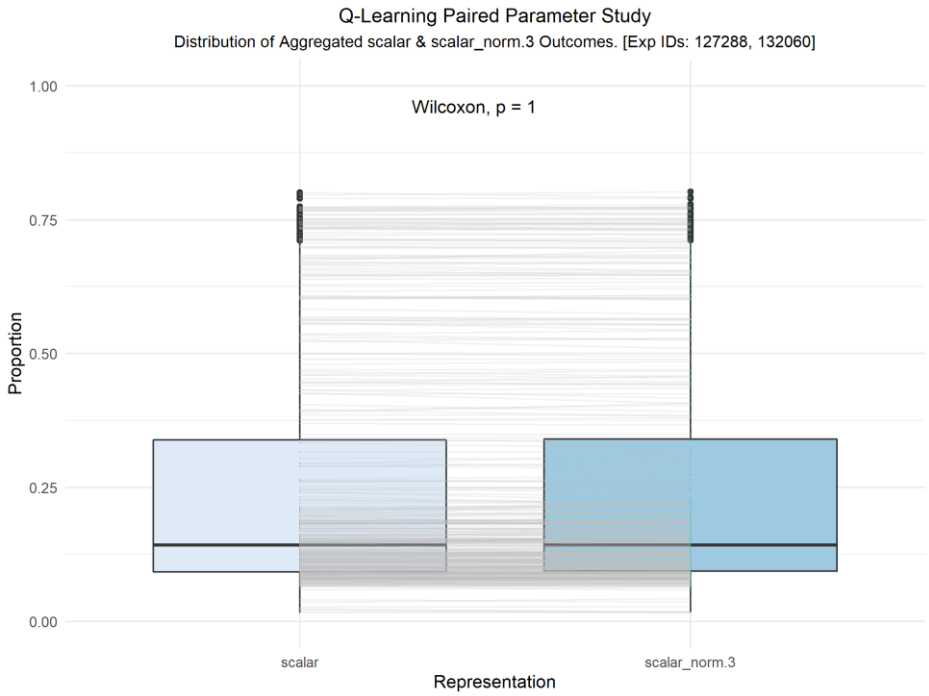


Figure 5.12: Aggregated distribution of scalar and normalised-scalar outcomes for *Q-Learning*.

Figure 5.13: Distribution of scalar and scalar_norm game outcomes for *Q-Learning*.

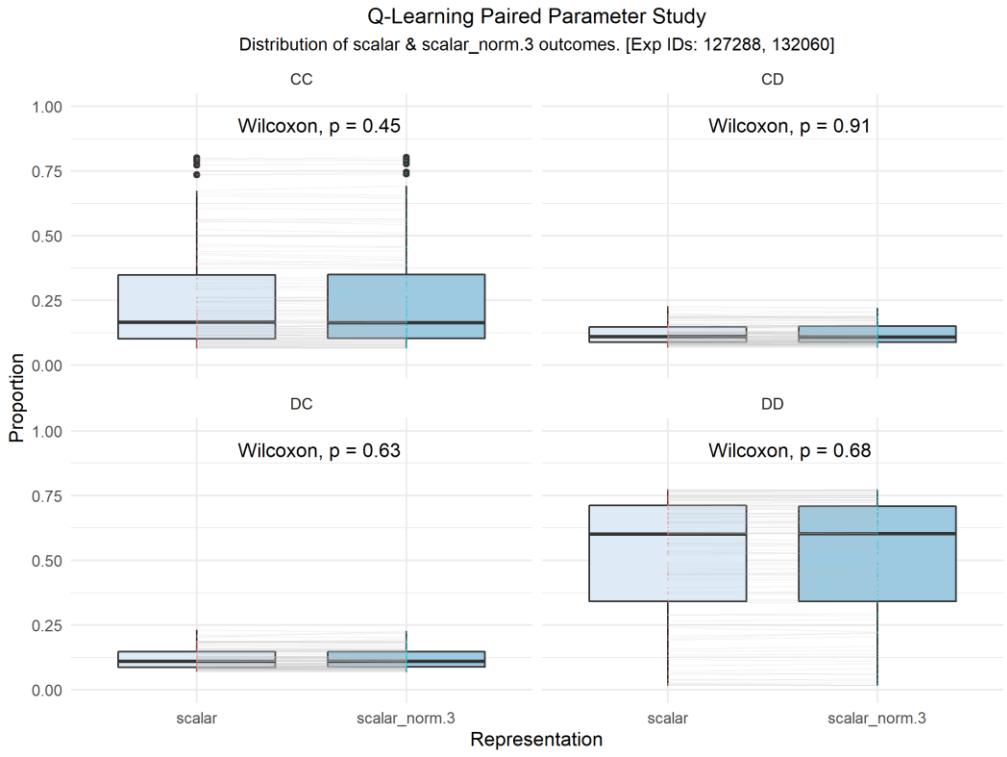


Figure 5.14: Distribution of scalar and ordinal game outcomes for algorithm *Q-Learning*.

5.3.6 Experiment Group Three

Results of the Wilcoxon Signed Rank tests for Experiment Group Three are shown in **Table 5.5**. Twelve of the algorithms have p-value < .05 indicating that these algorithms do not exhibit equivalence of behaviour between the two representations in this experiment group. Of special interest in *Experiment Group Three* is the difference in behaviour observed in the algorithm *Watkins Q Lambda*—the variation between representations, of both the CC and DD outcome, is considerable (**Figure 5.15**).

Table 5.5: Exp Group Three: Scalar (S) ~ Normalised-Ordinal (NO) Aggregated Distribution. Bold indicates statistical significance.

Algorithm	Peak % MCR		Wilcoxon			
	S	NO	V	p-value	CI L	CI U
Actor/Critic	32.1	19.8	47086	**.003**	0.0048	0.0195
Actor/Critic with Eligibility Traces	46.3	37.4	48229	**3.03×10^{-4}**	0.0048	0.0163
Actor/Critic with Replacing Traces	51.5	39.0	44761	**.044**	0.0003	0.0253
Q-Learning	80.1	80.4	47276	**.001**	0.0023	0.0075
Double Q-Learning	68.9	68.9	43233	.176	-0.0006	0.0034
Expected SARSA	75.7	76.9	49542	**2.88×10^{-5}**	0.0021	0.0049
R Learning	25.6	24.1	45745	**.015**	0.0089	0.0242
SARSA	79.9	80.9	47300	**.002**	0.0019	0.0068
SARSA Lambda	68.8	87.0	49205	**8.33×10^{-5}**	0.0122	0.0311
SARSA Lambda, with Replacing Traces	89.0	89.2	48050	**5.90×10^{-4}**	0.001	0.0033
Watkins (naïve) Q, Lambda	78.4	87.8	49985	**1.94×10^{-5}**	0.0269	0.0496
Watkins (naïve) Q, Lambda, Replacing Traces	76.4	89.1	49091	**1.02×10^{-4}**	0.0154	0.0404
Watkins Q, Lambda	87.2	86.9	50420	**5.03×10^{-6}**	0.0020	0.0047
Watkins Q, Linear Function Approximation	47.1	46.8	38471	.482	-0.0007	0.0004

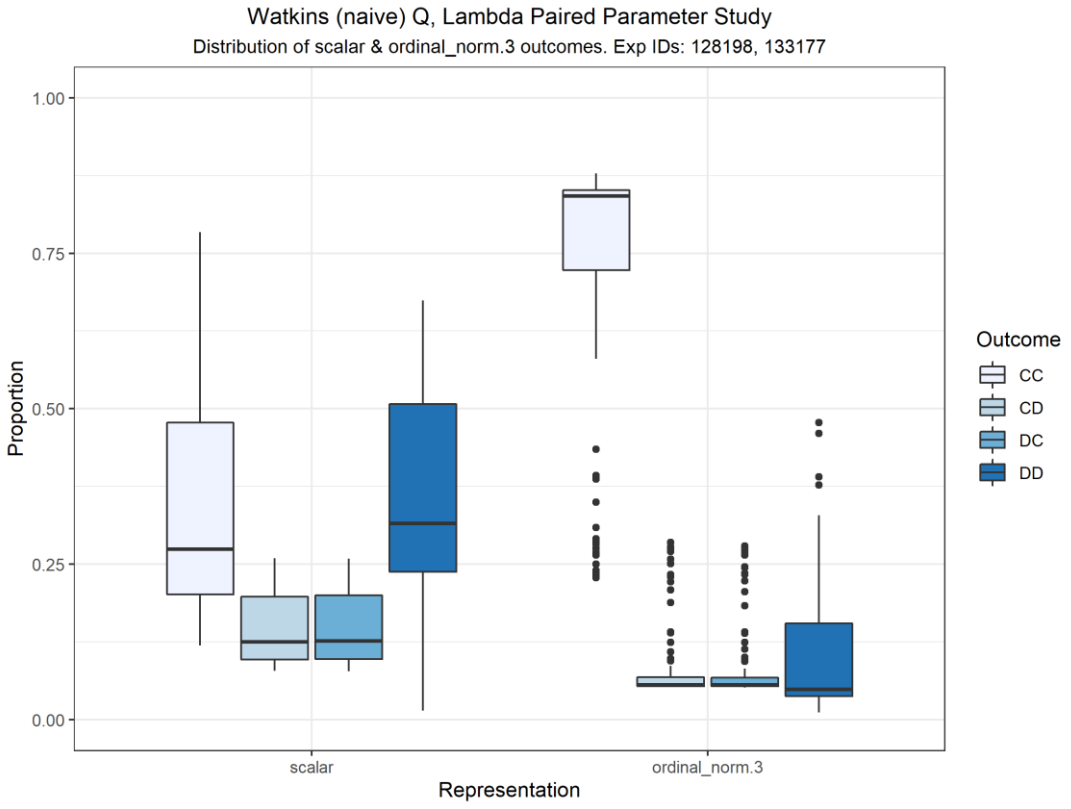


Figure 5.15: Grouped boxplot of outcomes for algorithm *Watkins (naive) Q, Lambda*.

The behaviour can be seen to have changed from exhibiting an approximately equal range in the **pd:scalar** distribution of CC and DD outcomes to one that has a higher proportion of CC (mutual cooperation) and a lower proportion of DD (mutual defection) in the **g111:ordinal_norm** representation. The raincloud plot (**Figure 5.16**) illustrates that the distribution of game outcomes between the two representations varies substantially.

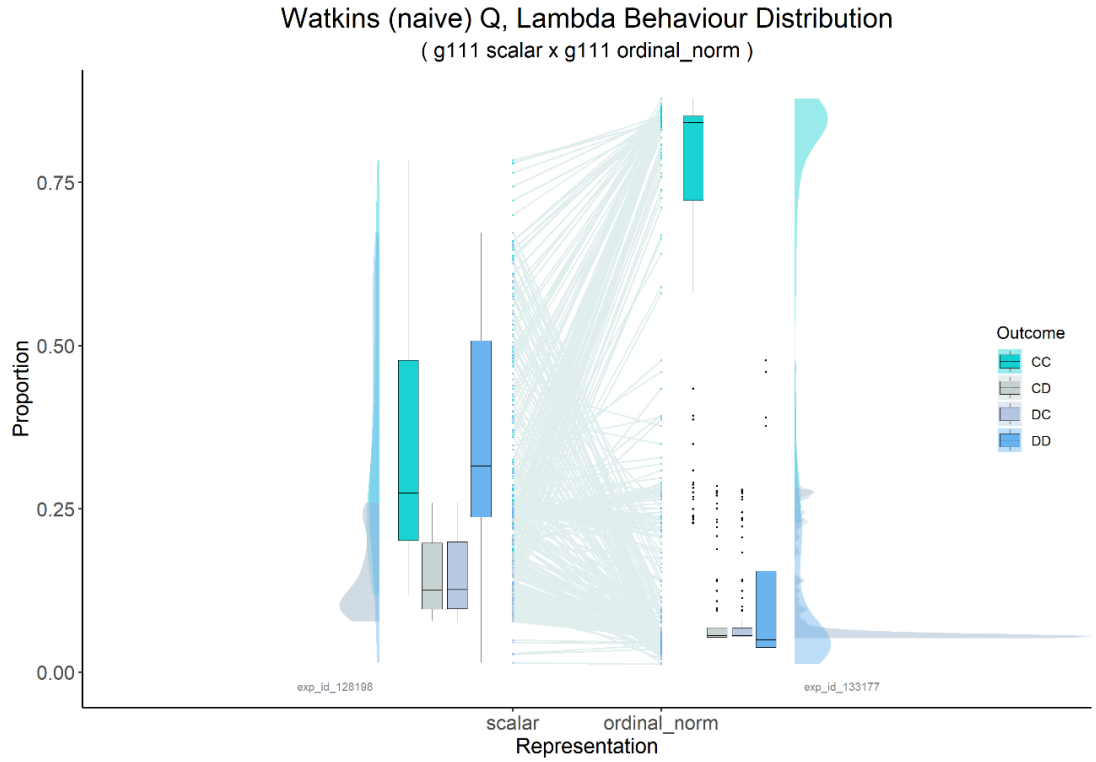


Figure 5.16: Distribution of scalar and ordinal game outcomes for *Watkins (naïve) Q, Lambda*.

5.3.7 Experiment Group Four

Results of the Wilcoxon Signed Rank tests for Experiment Group Four are shown in **Table 5.6**. Of the fourteen algorithms, seven have p-values $< .05$, indicating that these seven algorithms do not exhibit equivalence of behaviour between the two representations in this experiment group. The set of algorithms that return significant p-values differs from those that exhibit this property in *Experiment Groups One*, *Two*, and *Three*.

The algorithm *Expected SARSA* shows substantial variance between the two representations, as can be seen in **Figure 5.17**. In contrast, *Watkins Q, Linear Function Approximation*, does not exhibit statistically significant variance, but it does exhibit behaviour characterised by virtually equivalent distributions for all outcomes, over both representations, as can be seen in **Figure 5.18**.

For example, the algorithm *Watkins Q Linear Function Approximation* does not exhibit the property of equivalence, in the aggregate view in *Experiment Group Four*, which is in contrast to the results for this algorithm in the first three experiment groups. However, it is the weakest result for this algorithm in respect of extracting any meaning from the result of non-significance ($V = 44057,\ p = .07136$).

Table 5.6: Exp Group Four: Ordinal (O) ~ Normalised Ordinal (NO) Aggregated Distribution. Bold indicates significance.

Algorithm	Peak % CC		Wilcoxon			
	O	NO	V	p-value	CI L	CI U
Actor/Critic	30.2	19.7	48983	**1.24×10^{-4}**	0.0285	0.0712
Actor/Critic with Eligibility Traces	44.9	37.4	45950	**.012**	0.0015	0.0147
Actor/Critic with Replacing Traces	42.6	39.0	43710	.119	-0.0016	0.0141
Q-Learning	81.5	80.4	48454	**3.06×10^{-4}**	0.0022	0.0066
Double Q-Learning	32.7	68.9	51223	**1.53×10^{-6}**	0.0067	0.0140
Expected SARSA	77.3	76.9	52508	**8.21×10^{-8}**	0.0113	0.0248
R Learning	28.3	24.1	49944	**1.32×10^{-5}**	0.0011	0.0026
SARSA	80.3	80.9	49862	**2.45×10^{-5}**	0.0039	0.0090
SARSA Lambda	86.7	87.0	36226	.094	-0.0036	0.0002
SARSA Lambda, with Replacing Traces	89	89.2	39935	.943	-0.0019	0.0017
Watkins (naïve) Q, Lambda	88.1	87.8	42598	.280	-0.0005	0.0017
Watkins (naïve) Q, Lambda, Replacing Traces	89.1	89.1	38132	.395	-0.0020	0.0008
Watkins Q, Lambda	87.8	86.9	35790	.075	-0.0037	0.0001
Watkins Q, Linear Function Approximation	46.2	46.8	44057	.071	-0.0001	0.0014

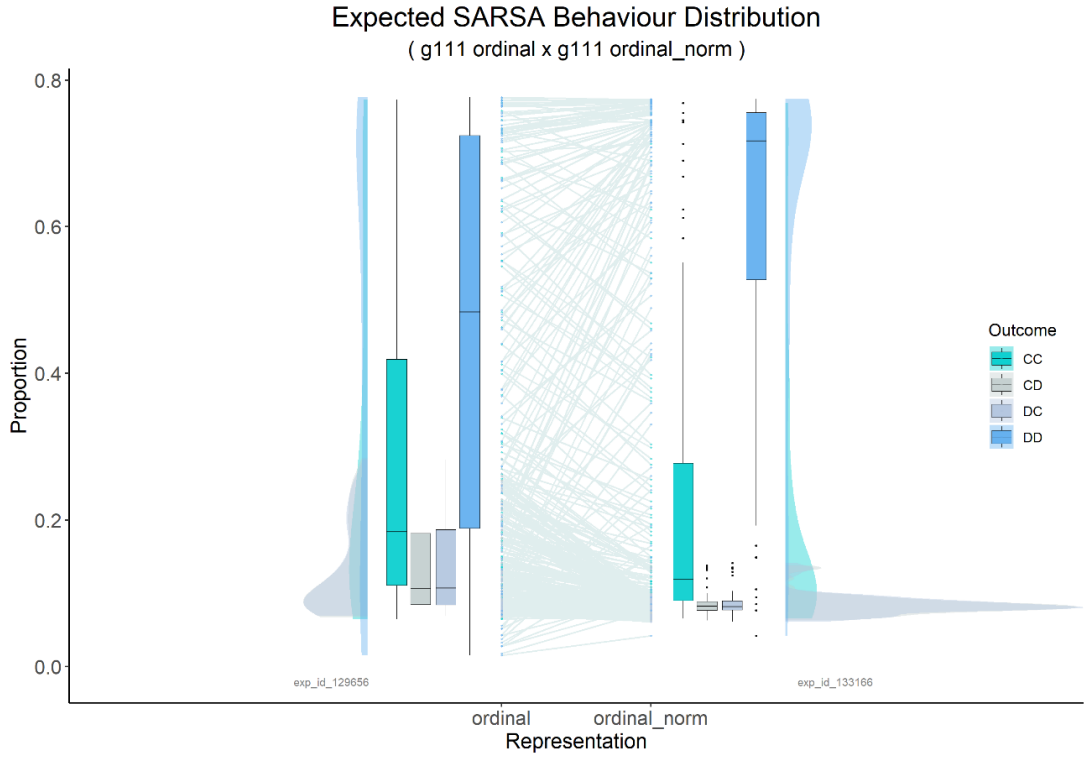


Figure 5.17: Distribution of ordinal and normalised-ordinal outcomes for *Expected SARSA*.

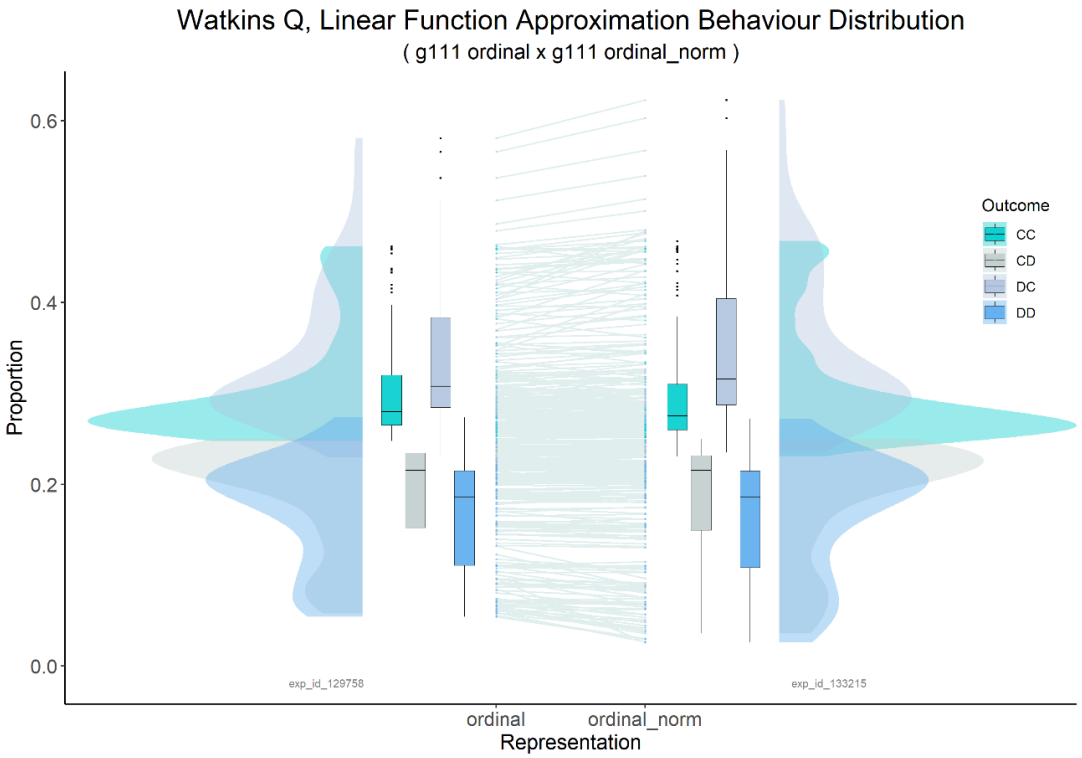


Figure 5.18: Distribution of ordinal and normalised-ordinal outcomes for *Watkins Q. Linear Function Approximation*.

5.4 Discussion

The results of this experiment series indicate that the behaviour of the algorithms studied can vary substantially as a product of the input representation of an otherwise equivalent game model. Given the common use of semantic interpretations of repeated game outcomes, the validity of the expectation that conclusions drawn over varying representations—even those that conform to the social dilemma inequalities—should be invariant with respect to behaviour is, then, unclear.

5.4.1 Principal Finding

Overall, these results suggest the broad finding to be wary of unconditionally generalising learning algorithms, in this case, between representations. However, within a representation it is potentially the converse—in that we can compare numerous algorithms within one single representation. But, if the input signal changes scale, function, or otherwise transforms, supposedly invariantly, then we are faced with the question of whether the algorithm is now displaying bias induced by the change in representation. On the other hand, where algorithms are often ‘tuned’ in respect of their hyperparameters this issue may be somewhat redundant once that tuning, and optimisation, has been performed satisfactorily—but exhaustively exploring tuning is not always possible in practice, and non-stationary environments may de-calibrate those very same tunings. The cause of this phenomenon is unclear from this study, which primarily set out only to confirm the effect was real, and as such establishing a cause for the apparent effect was not the aim. The principal finding, that there is an effect, is supported

by the data. Of more immediate concern is to understand more about the effect, for example, to ask how much the variance in behaviour actually affects an algorithm—does the range of an outcome's distribution shift only slightly, or is the effect more pronounced?

5.4.2 Distribution Shift

It may well be that the statistical assumption of a paired treatment—such that every parameter-pair in one behavioural profile is strictly compared to its exact same pairing in a second behavioural profile—enforces too strict an interpretation on assessing equivalence of behaviour. Relaxing this interpretation may reveal that the range, median and mean of each aggregate outcome do conform to some acceptable measure of equivalence, such as the results for the algorithm *Watkins Q Linear Function Approximation* which do indicate an appreciable lack of variance in each experiment at aggregate level, but, however, do devolve to a mix of significant and non-significant levels of variance at the individual outcome scale. Even if it is the case that the test applied in this study is too strict, there remains the fact that not all of the algorithms tested are seen to retain a behavioural profile that cannot be considered as substantially different between one, or more, of the representations—in that their range, median and mean values differ greatly. This shift in the distribution of the behavioural profiles, at the individual distribution level, is clear in both the mutual-cooperation outcome and the mutual-defection outcome of the algorithms *Watkins (naïve) Q Lambda* (**Figure 5.15**) and *SARSA Lambda* (**Figure 5.19**).

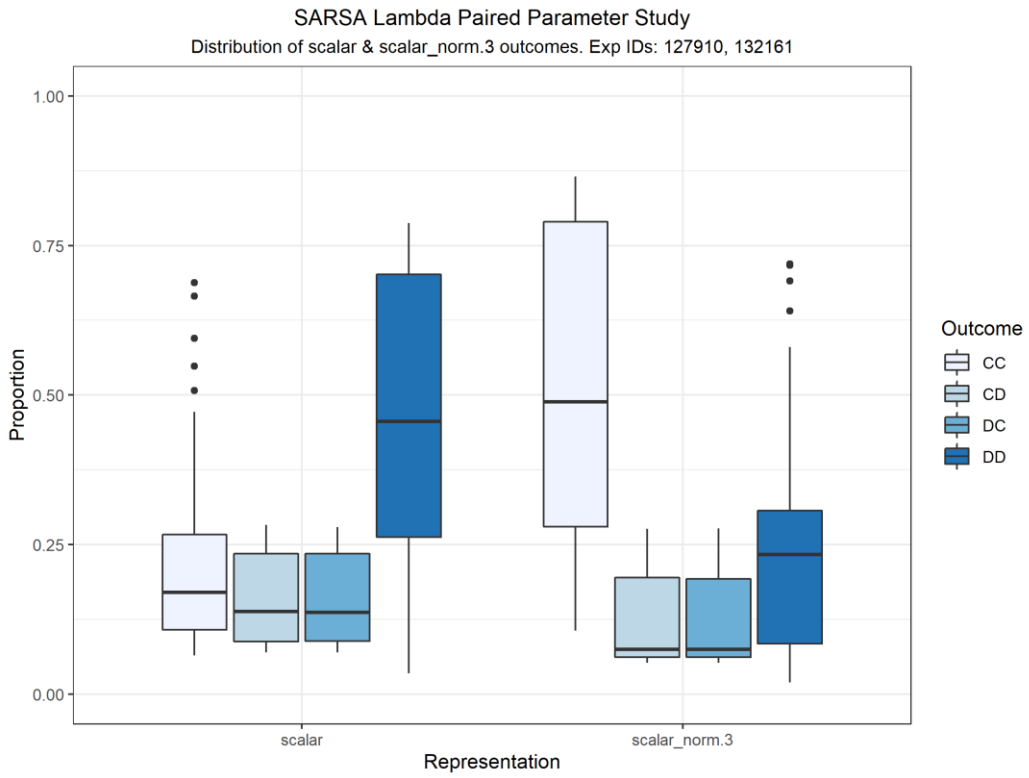


Figure 5.19: Grouped boxplot of scalar and normalised-scalar outcomes for *SARSA Lambda*.

5.4.3 Expectations of Stability of Behaviour

There is potentially an expectation of seeing increased stability when the structure-preserving operation is a transformation from **pd:scalar** to **pd:scalar_norm**; and likewise, in the transformation of **g111:ordinal** to **g111:ordinal_norm**. However, the algorithms that exhibit a measure of stability between these representations are not an exact match, as shown in **Table 5.7**.

Table 5.7: Variance Results, all Algorithms. Results are separated into two columns: those that exhibit significant variance (left), and those that do not (right). Of all algorithms assessed, only *Watkins Q Linear Function Approximation* (bold in table) exhibits non-significant variance in *all* experiments.

Experiment Group	p-value < .05	p-value > .05
One	Actor/Critic with Eligibility Traces Q-Learning Double Q-Learning Expected SARSA R Learning SARSA SARSA Lambda SARSA Lambda, with Replacing Traces Watkins (naive) Q, Lambda Watkins (naive) Q, Lambda with Replacing Traces Watkins Q, Lambda	Actor/Critic Actor/Critic with Replacing Traces **Watkins Q, Linear Function Approximation**
Two	Actor/Critic Actor/Critic with Eligibility Traces SARSA Lambda Watkins (naive) Q, Lambda Watkins (naive) Q, Lambda with Replacing Traces	Actor/Critic with Replacing Traces Q-Learning Double Q-Learning Expected SARSA R Learning SARSA SARSA Lambda, with Replacing Traces Watkins Q, Lambda **Watkins Q, Linear Function Approximation**
Three	Actor/Critic with Eligibility Traces Actor/Critic with Replacing Traces Q-Learning Expected SARSA R Learning SARSA SARSA Lambda SARSA Lambda, with Replacing Traces Watkins (naive) Q, Lambda Watkins (naive) Q, Lambda with Replacing Traces Watkins Q, Lambda	Actor/Critic Double Q-Learning **Watkins Q, Linear Function Approximation**
Four	Actor/Critic Actor/Critic with Eligibility Traces Q-Learning Double Q-Learning Expected SARSA R Learning SARSA	Actor/Critic with Replacing Traces SARSA Lambda SARSA Lambda, with Replacing Traces Watkins (naive) Q, Lambda Watkins (naive) Q, Lambda with Replacing Traces Watkins Q, Lambda **Watkins Q, Linear Function Approximation**

This suggests that the components that contribute to instability (or conversely, stability) in each algorithm vary. It may be that instability is endemic to each algorithm of this class (*fRL*) and as such the instability is the outcome of a process or mechanism orthogonal to the representation itself, such that a change in representation is but one possible perturbation method that could be applied to bring out the effect as observed in this experiment series. The apparent ability of *Watkins Q Linear Function Approximation* to exhibit less variance under changing representations than the other algorithms may indicate that discrete data structures are more susceptible to this issue than those that approximate a continuous space.

It may be thought that the effect that has been observed can be attributed to being an artifact of the process of normalisation, i.e., transforming the scalar values by normalising in the range (0,1). However, this apparent explanation can be rejected on the basis of the results from Experiment Group One. In this group, unlike the other three, there is no normalisation occurring—the mapping in Experiment Group One is from the canonical Prisoner's Dilemma on the one hand, to the equivalent ordinal rank of the canonical payoffs (i.e., the ranked preference for each payoff), on the other. That the effect is observed under this condition, as well being observed in the other three experiment groups serves to reject the 'effect of normalisation' as having sole explanatory power.

It may be that the manifold of the values accessible to the internal state of each algorithm varies under different representations. Within a representation a logical assertion is that algorithms can be compared—with the understanding that the given representation may define the depth and location of local minima as a possibly unique space to each algorithm, per representation.

With regard to expectations of behaviour, we are not now assured that the behaviours available to an algorithm will conform to pre-existing, intuitive, expectations derived from a semantic interpretation of the dilemma.

5.4.4 Sources of Stochasticity

In regard to potential sources of stochasticity in the software framework that may be imparting bias onto the algorithms and so affecting their behaviour, the principal source of stochasticity in the model is found in the action-selection method for each algorithm, either $\boldsymbol{softargmax}$[56], or $\boldsymbol{\varepsilon - greedy}$, as listed in **Table 5.2**. As, for each algorithm, the respective action-selection method does not vary, it is not expected that the observed variance in behaviour is as a result of the action-selection method. Further detail on sources of variance in the experimental framework are discussed in Appendix A.

5.4.5 Validity of the Inequalities

A question not yet addressed in this chapter concerns the social dilemma inequalities themselves. The expectation that the inequalities (see §2.3.2), derived from the Prisoner's Dilemma (§2.3.1), will influence the behavioural profile that a participant will exhibit is a common game theory formalism in research to which this thesis is adjacent and

[56] Here $\boldsymbol{softargmax}$ is implemented with shift-invariant exponential normalisation, and there is no temperature term as per the Sutton & Barto (2018, p. 37) implementation. See Appendix A.

contextualised by (Axelrod & Hamilton, 1981; Nowak & Sigmund, 1993; Macy & Flache, 2002; Banerjee & Sen, 2007; Masuda & Ohtsuki, 2009; Han et al., 2011a; Han et al., 2011b; Han et al., 2012; Press & Dyson, 2012; Leibo et al., 2017; Wang et al., 2018). As stated in the introduction to this chapter, the inequalities have also been used as a basis for evaluating equivalence of behaviour in previous studies. While not in any way suggesting that the inequalities are necessarily flawed, the question that can be asked is whether the inequalities are overly generally specified for the computational domain, as the space of values the inequalities admits does not have any imperative over an action gained via a human interpretation that could potentially either close out, or make more attractive so increase the likelihood of, further spaces of possible values in an agent's reachable space. Also, bearing in mind that the inequalities are cast in semantic terms (*reward*, *punishment*, *fear*, *greed*, etc), it may be the case that these semantics, when explored computationally, on probability terms, just do not map to the same outcomes as understood by human participants.

A frequentist expectation may propose that a computational agent would reduce to the rational policy (under all isomorphic representations) but that is not what has been found in this experiment series, as evidenced by the lack of consistency across the observed behavioural distributions. Further investigation into the relationship between an agent's internal state and the agent's behavioural distribution would be of value in this research domain. While of particular interest to the research objective presented in this thesis the hypothesis of the inequalities being too broadly construed for fine-grained computational agent behaviour profiling is not explored further here. The impact of the principal finding of this experiment series is, however, of particular importance to the thesis, and this is discussed next.

5.4.6 Impact on Thesis

The principal finding of this experiment series, presented in §5.4.1, raises an important question for the objective of this thesis. The principal aim of the thesis is to develop a method for identifying a *current* strategic dynamic, in the interaction of an agent with its environment, through *mapping* into the *rRGS* graph $\boldsymbol{G}$, by the agent's observation function $\boldsymbol{O_i}$, the identity function, with respect to each node in $\boldsymbol{G}$, can be obtained.

To this point, it has been assumed that the only available input to an agent to determine this mapping is the reward gained from the environment in response to an action. However, there may be significant variation in the performance of algorithms under conditions of change modelled in this chapter which raises doubt about whether a mapping derived solely from the reward signal would identify the same node in $\boldsymbol{G}$ as may be exhibited by the agent, post transformation of the inputs via the agent's observation function $\boldsymbol{O_i}$, given that the former would also require a transformation of some kind in order to map to the ordinal values of the *rRGS*.

As a result of this uncertainty, it must be considered that attempting to use only scalar reward values to map directly into the *rRGS* does not appear promising due to the amount of variation that has been observed, the lack of a ready explanation, and, crucially, without having an inverse function to neutralise the effect. Therefore, a different conception of what it is that is being measured when speaking of 'mapping into the *rRGS*' may be worth considering. The prime candidate for gathering further data is the agent itself, specifically,

in the behaviour that an agent exhibits. The next chapter explores the coupling of observations over agent preference to actions as well as the reward signal, to form a mapping to a node in the *rRGS* graph $\boldsymbol{G}$.

5.5 Summary

The results of this experiment series suggest that if isomorphic representations induce a discrete and non-identical reachable space in respect to the values that an internal agent state may take, then, in turn, the possible behaviours that an agent can embody will also vary; and so, agents experiencing such complexity may display less predictable behaviour. This property of behavioural variance is a vector for *representation-induced algorithmic bias*. As a contributor to overall bias in a system (see §1.1) it is likely that representation-induced bias is but one iron in the fire, so to speak, another component in the spectrum of contributory effects that together comprise bias in computational systems.

In conclusion, the null hypothesis of this experiment series—that an algorithm's behavioural profile will not vary between equivalent representations of the game model—*is not supported* for the majority of the learning algorithms examined in this experiment series, as shown in **Table 5.7**, where only *Watkins Q Linear Function Approximation* exhibits anything approaching non-significant variance in all experiment groups. It is of note that the result for *Watkins Q Linear Function Approximation* is somewhat weak in *Experiment Group Four* ($V = 44057,\ p = .07136$). Regardless, the ability of *Watkins Q Linear Function Approximation* to maintain equivalence across these four representations suggests that interrogating contemporary policy-gradient reinforcement learning algorithms would be of value in future work.

Chapter Six

Game Model Recognition

I wanted you. And I was looking for you
But I couldn't find you
I wanted you. And I was looking for you all day
But I couldn't find you. I couldn't find you

—Laurie Anderson, *Walking and Falling*[57]

This chapter describes an experiment series that has a singular initial objective. That objective is to develop a method using preference mapping to identify the strategic dynamic inherent to a game model $\boldsymbol{g} \in \boldsymbol{G}$, if manifest in a game $\boldsymbol{g}' \notin \mathbf{G}$, in the context of a Markov game $\boldsymbol{M}$ (see §2.2.2). In this experiment series, the game models $\boldsymbol{g}'_{\boldsymbol{i}}$ are represented by canonical game theory games. The purpose of this method is to be a mechanism for identifying the occurrence of a state of *cooperative intent*, as discussed in §1.2.1, under the constraints outlined in §1.3[58].

A predicate for this experiment series is the expectation that, in a dynamic cooperation situation $\boldsymbol{g}'$, for example, an iterated Prisoner's Dilemma with canonical values, participants in $\boldsymbol{g}'$ will exhibit a pattern in their action sequences that corresponds to the pattern of play observed when participating in the game model $\boldsymbol{g}$ (i.e., **g111**, the ordinal form of the canonical Prisoner's Dilemma); then the participants in the game—i.e., the *Agents*, but also potentially an *Observer* entity, exogenous to the participants—will be able to match the pattern of strategic dynamics in the game $\boldsymbol{g}'$ to a game $\boldsymbol{g} \in \boldsymbol{G}$, and so perform an act of *game recognition*.

The suitability of a game $\boldsymbol{g}'$ to be decomposed in this way is a function of the method used to decompose, isolate, or otherwise analyse the cooperative dynamics of the situation (in the current example, Prisoner's Dilemma in canonical form). In other words, the suitability of the method rests on the abstraction used to interface to the game situation—in this case, a Markov game—however, with appropriate abstraction the generalised application of the preference mapping method is conceptually feasible. Preference mapping generalisation is returned to in §6.4.

The simplest applicable case for performing mapping of any kind is a one-to-one map between the members $\boldsymbol{g} \in \boldsymbol{G}$ and the members $\boldsymbol{g}'' \in \boldsymbol{G}$, where the only difference between a game model $\boldsymbol{g}$, and a game model $\boldsymbol{g}''$, is the *conceptual type* of the payoff value—the two sets of games have the *same values* in each game in $\boldsymbol{G}$. For $\boldsymbol{g} \in \boldsymbol{G}$, the type is a value

[57] Laurie Anderson, *Walking and Falling*, Track 4, *Big Science*, 1982; also see (well, listen to) *Walking and Falling*, on the live recording *United States Live (Part 2)*, 1984.

[58] This experiment series does not address the proposed mechanism for reflection, introduced in §1.2.1.1.

signifying its ordinal rank, for $\boldsymbol{g}'' \in \boldsymbol{G}$ the type is an integer signifying a reward. For both, the payoff value is the reward $\boldsymbol{r} \in \mathbb{R}^4 = \{1, 2, 3, 4\}$. This conceptual separation between the game types assists with the step-wise development of the mapping method. Firstly, it allows *direct mapping*, which simply compares the incoming rewards to each $\boldsymbol{g} \in \boldsymbol{G}$, and flags if it finds a match. The experiment to evaluate this first method is described in §6.3.1. Secondly, rather than mapping scalar rewards directly, the *preference mapping* method progressively tracks an entity's preferences over its own actions, and then uses those preferences as a key to map into a game $\boldsymbol{g} \in \boldsymbol{G}$. An experiment to evaluate this second method is described in §6.3.2. A third and final set of experiments (described in §6.3.3) evaluates the second method (preference mapping) on several games $\boldsymbol{g}' \notin \mathbf{G}$, with the objective being to identify the *rRGS* analog to the game $\boldsymbol{g}'$, if it exists. The preference mapping method processes the preference an agent exhibits to actions, over time, to identify the node whose game model matrix is being used to constitute the environment $\boldsymbol{M}$. Successful recognition of a game model is registered as an activation of a node in the RGS graph, an instance of which is held by each agent. An external observer also registers the activation, in its own copy of the RGS graph. The preference mapping algorithm gives agents a view of the world built from observing their own behaviour in response to received reward, specifically, the preference they show for the actions that form the outcomes that produce the reward, over time. Visibility of the outcome is a result of knowing the other participant's actions from previous timesteps, as well as the agent's own actions over time. This is made possible by relaxing the *state information visibility* (*SIV*) constraint over an agent's recognition component, from a state of *imperfect information* to a state of *perfect information*. The observer component compiles both agents' recognition component, and also gains the ability to see the payoff values of both agents, giving the observer a *SIV* of *perfect* and *complete information* (see §1.3.2 and §2.2.2). The agents' learning algorithm components continue to have *imperfect* and *incomplete SIV* and also maintains Markov constraints (see §2.2.2). Each agent's recognition component, however, retains an explicit history of its own rewards which is also a Markov relaxation, but again, for the recognition component only.

The next section, §6.1, outlines the methodology for this experiment series. The following section, §6.2, describes two methods for the identification of $\boldsymbol{g} \in \boldsymbol{M}$; where $\boldsymbol{g}$ (an ordinal game model from the *rRGS* graph $\boldsymbol{G}$) is being used to constitute a Markov Game $\boldsymbol{M}$. The game model $\boldsymbol{g}$ has not undergone a positive linear transformation of any kind. The first of the two methods, *direct mapping*, is described in §6.2.1. It maps directly from the reward values (obtained from the game model matrix $\boldsymbol{g}$) to the model $\boldsymbol{g}$ (in the *rRGS*) and is the simplest experimental case as the reward values are identical to the values in the game model $\boldsymbol{g}$. The second method, *preference mapping*, is described in §6.2.2. This method operates on the same set of equivalent values as in the first method but disassociates from mapping on the reward values. Instead, this method places a layer of indirection to map the received reward values onto the agent's own, observed, *preference* for the actions that led to those rewards being received. To reiterate, each mapping method is isolated[59] from the other.

[59] Both methods accept the same input and emit values of equivalent type; however, each method's pipeline is completely isolated from the other, such that the agent, although executing the functions, does not inform either method of any part of its own action-choice processing.

The results of evaluating both methods are given in §6.3. The *preference mapping* method is applied to a set of games $g' \notin \mathbf{G}$, in §6.3.3. These games are canonical versions of Prisoner's Dilemma (**g111**), Stag Hunt (**g322**), Chicken (**g122**), and the Coordination (**g311**) game (see §2.3.1, §2.3.2, and §2.3.3.4). A discussion of the results of the experiments in this chapter then follows, in §6.4.

6.1 Methodology & Scope

In the Prisoner's Dilemma, the action that produces the highest reward at any given timestep is not necessarily the most preferred action over the longer term; in strategic terms this is characterised by neither player benefiting from unilaterally breaking an equilibrium (Shubik, 1970), as such, the deficient outcome *defect-defect* is a *non-cooperative* equilibrium.

The conception that this unique game model would induce a pattern in a participant's behaviour was explored by Rapoport and Chammah (1965), who presupposed that the pattern of behaviour that an agent displays would be a signatory identifier that reflects the environment's strategic dynamic. Rapoport and Chammah speculated that the particular dynamics of a given situation would either induce, or require, a pattern of play from the participants—a pattern, they posited, unique to that set of strategic dynamics. Their analysis over a number of psychological studies explored a variety of techniques to find supporting evidence for the conjecture, however, the question, by their own account, remains open.

The question of whether it is the behaviour causing the dynamic, or the dynamic causing the behaviour, is not at stake in this experiment series, as the dynamic is provided by the game model matrix. Causality, however, remains an important question that informs the next consideration: what is being assessed is *not necessarily*[60] the dynamic of the environment, but an assessment of a dynamic *in the agent's behaviour*, posited to have been induced in the agent by the agent's *experience* of the environment, via its observation function $\boldsymbol{O_i}$. Another way of picturing this is to think of the recognition component as acting on a view into the agent's *perception* of the environment, the information available from that perception is manifested in the behaviour of the agent in a process that is identifiable as an instance of *bounded rationality* governing and defining the information set from which an entity makes its decisions (Simon, 1955).

The distinction sometimes placed upon agents of being an individual entity, or as being part of the environment, is one of perspective. From any single agent's point-of-view everything else *is* 'the world', a definition that conforms to the game theory concept of a *game-against-nature* (Colman, 1995). As a *game-against-nature*, each experiment instance in this series has at most two agents. Neither agent views the other as a like entity; as far as each agent is concerned, the other agent is indistinguishable from the environment. However, the external observer in these experiments has *perfect* and *complete* information visibility into both agent's actions and rewards, as well as *perfect* and *complete* state information visibility (*SIV*) over the environment. The observer is exogenous to the agents but endogenous to the environment. To establish that the

[60] The qualifier 'not necessarily' indicates that the dynamics of the environment, and the dynamics of the behaviour of the individual agents, can, but do not have to, concur.

recognition algorithm accurately identifies a game model, each experiment instance captures the recognition success rate (*gamelock*, or $\boldsymbol{gL}$), and the number of timesteps to the point of game model recognition (*time-to-lock*, or $\boldsymbol{TTL}$). To summarise, each agent knows:

- their own action in the last timestep,
- the other participant's action in the last timestep, and therefore,
- they know the outcome of the last timestep.
- Each agent also knows their own reward from the last timestep,
- but they do not know the other participant's reward(s).

Further, the external observer relaxes constraints such that it:

- has visibility into the actions of all agents, and
- has visibility into the payoffs received by all agents.

For both agents, the memory depth directive is as for the internal state structures in the foundational RL (*fRL*) algorithms (see §4.1.2.3)—an agent may not store an explicit history of the inputs in any structure but can store computational results. The Observer component is not restricted at all in this way, so may store a complete history of inputs. The Observer is not a *type* of agent, and has no 'learning algorithm' component, but is a separate entity within the environment.

To reiterate, this relaxation to the agent's recognition component does not extend to the agent's strategy component (i.e., the agent's learning algorithm), which continues to interface to the environment under the same constraints as in previous chapters.

In total, for every game model $\boldsymbol{g}$, an experiment instance conducts an *asymmetric-selfplay* experiment type of 100 episodes of 1000 timesteps each, resulting in 14,400 episodes over the whole graph $\boldsymbol{G}$, per method.

Finally, an observation from working with the experiment framework that informs an agent's initial state: to achieve the objective of mapping by preference requires a certain amount of stochasticity in an agent's actions, as a deterministic algorithm cannot gather enough varied data to identify a game model. For this experiment series, both agents (*agent zero*, or $\boldsymbol{A_0}$; and *agent one*, or $\boldsymbol{A_1}$) use the *Q-Learning* reinforcement learning algorithm, with asymmetric hyperparameters ($\boldsymbol{A_0}$ $\{\boldsymbol{\alpha} = 0.1,\ \boldsymbol{\gamma} = 0.9\}, \boldsymbol{A_1}$ $\{\boldsymbol{\alpha} = 0.9,\ \boldsymbol{\gamma} = 0.1\}$). The conjecture guiding the choice of asymmetric hyperparameters is that asymmetry in learning and decay parameters will lower the likelihood of pre-emptively locking-in (see §2.2.3.3) to an equilibrium *prior* to performing the *range* of exploration required for the preference mapping method to successfully perform game model recognition. *Q-Learning* was chosen as the vehicle for assessing the recognition component on a heuristic assessment of its generally acceptable rates for mutual cooperation achieved in Chapter Four, and for its performance in the representational equivalence experiment series presented in Chapter Five.

While *Q-Learning* demonstrated significant variance in three of the four representations examined in this later experiment series, the structure of the point mappings (see **Figure 5.13**) suggests that the algorithm is not 'wildly' variant. As this current experiment series benefits from some variance in agent behaviour, but also needs some identifiable patterns to emerge, *Q-Learning* presented as a reasonable candidate for this first *recognition-oriented* experiment series. Future work would benefit from

assessing a variety of algorithms to quantify their effect, if any, on the recognition component.

6.2 Game Model Recognition

The first test of the game model recognition algorithm is to conduct a comparison between the two methods *direct reward* and *preference mapping*. For each method, an experiment instance is run for each game model $\boldsymbol{g} \in \boldsymbol{G}$. This gives 144 experiment instances for each of the two methods.

The first method, *direct mapping*, relies on a bijective relation between the scalar reward $\boldsymbol{r}$; $\boldsymbol{r} \in \mathbb{R}^4 = \{1, 2, 3, 4\}$; $\boldsymbol{r} \in \boldsymbol{g} \in \boldsymbol{G}$, and the ordinal preference $\boldsymbol{p}$; $\boldsymbol{p} \in \mathbb{R}^4 = \{1, 2, 3, 4\}$; $\boldsymbol{p} \in \boldsymbol{g} \in \boldsymbol{G}$. For the game model **g111**, the reward that each participant receives from a mutual cooperation outcome is three (3), which maps, in **Figure 6.1a)**, to the intersection of the $\boldsymbol{C}$ row and $\boldsymbol{C}$ column. The cell location index $(\boldsymbol{C}, \boldsymbol{C})$ is equivalent to the cell index (0,0) in **b)**. As an ordinal map, the preference rank of three is functionally equivalent to the scalar reward 3. The direct mapping method is detailed further in §6.2.1.

The second recognition method, *preference mapping*, decouples the direct mapping between the scalar reward and the ordinal preference rank. Mapping is attempted using a value emitted from a function that is effectively concurrent to, and completely isolated from, an agent's strategy algorithm. Conceptually, this function exists as a component within the agent's observation function $\boldsymbol{O_i}$ (see §2.2.2) and is executed at every timestep to enable a progressive re-normalisation of the entire reward history. The preference mapping method is detailed further in §6.2.2.

		Column C	Column D
Row	C	3, 3	1, 4
	D	4, 1	2, 2

a)

		A_1 0	A_1 1
A_0	0	(0,0)	(0,1)
	1	(1,0)	(1,1)

b)

Figure 6.1: Mapping under canonical payoff ordering. In **a)**, the game model **g111** is shown, and in **b)** the cell index map is pictured, where $\boldsymbol{A_0}$ is **Agent Zero**, and $\boldsymbol{A_1}$ is **Agent One**.

6.2.1 Direct Mapping by Reward

Using this method, agents attempt to map the reward value directly onto their payoff in an outcome pair of a given cell index location, such as $(0, 1)$. An outcome is the result of a joint action by two agents, so the tuple returned from the game model by that outcome contains two values, one for each agent. Each agent is only able to access their own reward value from the outcome tuple.

Having received a reward value, the agent's recognition component looks into the *rRGS* and collects a list of all game models that have that same value in the location mapped to by the outcome's cell index. An empty set is returned if none of the game models has that exact value in the identified cell index, for that agent. However, a

successful *value-mapping* produces a set of game model candidates, where each candidate has the mapped value in the same cell index for that agent. The candidate sets accumulate as an aggregation of successful value-mappings at every timestep, which results in four sets of candidate game models; one set for each game outcome.

If the intersection of these candidate sets resolves to a single game model, then that game model is a key to index to a distinct game model in $\boldsymbol{G}$. A bijective mapping between the obtained key and an indexed game model is the end-point of a successful recognition process. By way of example, a successful recognition of the game model **g234** would occur when the obtained key is **g234** and the indexed game model is also **g234**, giving a successful game model *lookup*. The agent's recognition component's *lookup()* function is listed in **Algorithm 6.1**. To summarise this listing, the *lookup()* function is called every timestep. Each call produces either a set of candidate game models or an empty set. A candidate game model is a game model that has the identical value for the outcome as that which the agent supplied to the lookup function as $\boldsymbol{v}$. In this experiment instance, the agent directly passes the value from the observation function $\boldsymbol{O_i}$, without filtering or processing in $\boldsymbol{O_i}$, to the *lookup()* function.

Algorithm 6.1 Agent Lookup.

Description: Accepts an outcome and a reward, maps the outcome to every game model in the RGS and extracts the game model's values for that outcome, then compares the reward to the payoff value the agent would have received under that game model. If a match, then add the game model ID to a list and return the list.

Input: *k* is most recent outcome; *v* is most recent reward.
Output: *candidate_matches* is a list of candidate model IDs.

Dependency: *RGS*()::*reference_rgs* is a data structure containing all game models in the RGS, where *r* in *reference_rgs* is a key::value pair of game model ID and 2x2 game matrix.

```
1:  procedure Agent()::lookup(k,v)
2:
3:    for r in reference_rgs do
4:          model_id ← r[0]
5:          outcome ← r[k]
6:          mapped_reward ← outcome[agent]
7:
8:    if mapped_reward == v then
9:       candidate_matches ← model id
10:
11:    return candidate_matches
```

Algorithm 6.1: Agent lookup function. Each agent has a recognition component that performs the mapping process. The *lookup()* function performs indexing into the *rRGS*, attempting to match the reward $\boldsymbol{v}$ at the cell index location given by the outcome $\boldsymbol{k}$ in each game model $\boldsymbol{g}$.

Following the aggregation of candidate sets, for each outcome that has been visited the agent (or Observer) executes a set intersection over all four sets of candidates. For the Observer component, if the intersection is reduced to a single candidate element, then this

signifies the occurrence of a gamelock ***gl***. For agents, with their reduced *state information visibility*, a successful intersection of the candidate sets results in a set of twelve candidate game models and is called an *interlock*. The method to perform the set intersection is given in **Algorithm 6.2**.

Algorithm 6.2 Agent Identify Game Model by Reward.

Description: This function calls lookup() at every timestep and accumulates the candidate game models in four sets, one set of candidates for each game outcome. A set intersection over these four sets gives either an empty set, or a set with twelve game model IDs which comprises an interlock.

Input: *pr* is most recent reward received by the agent.
ps is most recent step's outcome participated in by the agent. At the first timestep in an episode, *ps* will be empty.

Output: *new_interlock* is an array of game Model IDs. Dependent on context it can be returned to the calling function or be used in a class hierarchy. An interlock is only ever of length 0 or length 12.

Dependency: *RGS*()::*outcome count* is a key::value data structure that records a count for each of four outcomes. An outcome is the *key*, and the count is *value*. *RGS*()::*gm* is a key::value data structure where the key is an outcome pair, and the value is a set of game model IDs.

```
1: procedure Agent()::identify_game_model_reward(pr, ps)
2:
3:    if ps then
4:       outcome count[ps] ← pr
5:
6:    for k, v in outcome_count do
7:       if v then
8:          candidates ← lookup(k, v)
9:          gm[k] ∩ candidates
10:
11:   new_interlock ← gm[0,0] ∩ gm[0,1] ∩ gm[1,0] ∩ gm[1,1]
```

Algorithm 6.2: Agent Identify Game Model by Reward

6.2.2 Preference Mapping by Reward and Behaviour

In the second experiment instance the agent maps into ***G*** by constructing a key from its own per-timestep progressive action-response, i.e., the agent's current action preference, updated with the most recent outcome, over the history of interaction.

The algorithm is progressive as the history of rewards over the current episode is re-normalised at every timestep[61]. This function, *Progressive Min-Max Normalisation*, is shown in **Algorithm 6.3**. The value that is returned from this function is passed to a

[61] The re-normalisation process differs slightly dependent on the source data stream.

preference-oriented *lookup()* function analogous to the value-oriented *lookup()* procedure described in **Algorithm 6.1**. The preference mapping method reveals the preference that an agent shows for their own actions, over all four outcomes.

The aggregate of preferences may produce candidate game models, which, after an intersection operation, may produce an *rRGS* key. As in the *direct mapping* method, this key can then be used to map into $\boldsymbol{G}$ to return either a single node (gamelock, or $\boldsymbol{gl}$), or a set of nodes (interlock, or $\boldsymbol{il}$). An interlock consists of those game models that have partial matches to the key, i.e., when the key does not map one-to-one on all four preferences, because of missing or different values. In this case the key may only have two or three of the four ordinal values in the same outcome position.

Algorithm 6.3 Progressive Min-Max Normalisation

Description: Progressively min-max normalise a time-ordered array of scalar reward values.
1. If necessary, shift all values into positive range.
2. Do min-max norm: scale to (0,1) by x_i / *max*(*reward history*).
3. Pass most recent, re-normalised, value to binning function.
4. Return output of binning function.

Input: *reward_history* is an array of received reward.
Output: *bin* is a scalar in range (1,4).

Dependency *RGS*()::*bin_normalised_reward*() Function that accepts as a parameter a value in the range (0,1) and returns bin number.

```
1: procedure mm_normalise_bin_reward(reward_history[])
2:
3:     len = length(reward_history)
4:
5:     for r in reward_history do
6:         if r < 1 then
7:             has_negative_or_zero ← true
8:
9:     if has_negative_or_zero then
10:        min_r = min(reward history)
11:        d = 0 – min_r + 1
12:        for index in reward_history do
13:            reward_history[index] += abs(d)
14:
15:     ∀ r in reward_history do
16:         mm_norm[] ← max(reward_history)
17:     bin ← bin_normalised_reward(mm_norm[len-1])
18:
19:     return bin
```

Algorithm 6.3: Function for progressive normalisation. The specific normalisation method varies dependent on the range of the reward values in the source data stream. Progressive re-normalisation operates over every received reward in the history, the most recent reward is then binned (0,.25,.5,.75). The bin ID is returned to the calling entity and this value is used as the ordinal preference of the matching action in the outcome under consideration.

6.3 Comparing the Methods

A functional limitation that applies to both methods (*direct*, and *preference*), due to the constraints of *perfect information* (agents can see others' actions) and *incomplete information* (agents cannot see other agent's payoffs) is that agents are restricted to an *interlock*, which identifies twelve game models as possible candidate matches. This is because the agent only has visibility as shown in **Figure 6.2b)**. Using the Axelrod (Axelrod & Hamilton, 1981; Hofstadter, 1983; Axelrod, 1984) form of the Prisoner's Dilemma game model, in **b)** the agent $\boldsymbol{A_0}$ can see their own payoffs and also agent $\boldsymbol{A_1}$'s action. The observer has full visibility, which reveals payoff values as well as knowledge of the outcome, as depicted in **a)**.

Axelrod	Column / A_1	
	C	D
C	3, 3	0, 5
Row / A_0		
D	5, 0	1, 1

a)

Axelrod SIV view for A_0	A_1	
	C	D
C	3, C	0, D
A_0		
D	5, C	1, D

b)

Figure 6.2: State Information Visibility (**SIV**) view for Row participant. In **a)** Axelrod's canonical Prisoner's Dilemma under *perfect* and *complete* SIV. In **b)** the same game model showing the Row ($\boldsymbol{A_0}$) participant's view under *perfect* and *incomplete* SIV.

In the results that follow a table is presented for each mapping method, summarising the success rate (*gamelock*, or $\boldsymbol{gL}$), and the time-to-recognition (*time-to-lock*, or $\boldsymbol{TTL}$) over the graph $\boldsymbol{G}$. Each table provides a breakdown per *rRGS* layer (see **§2.3.3.4**, and Chapter Three); and also provides a breakdown for a subset of games in $\boldsymbol{G}$, chosen for their representative spread, i.e., the chosen games are not lonesome outliers. Results for all games are located in **Appendix B.4.2**.

6.3.1 Direct Mapping

The direct reward mapping method obtains a gamelock ($\boldsymbol{gL}$) in 98.7% of episodes. The mean time-to-lock ($\boldsymbol{TTL}$) over all episodes was 103.62 timesteps ($\boldsymbol{\sigma} = 149.15$). No game model went unrecognised. There were no false positives (i.e., assessing a gamelock in $\boldsymbol{g}$ as **g112**, for example, when $\boldsymbol{M}$ was constituted by **g243**). A summary of the results for the observer is shown in **Table 6.1**, and a summary for each agent is listed in **Table 6.2**. Full results, for the whole *rRGS*, are located in Appendix B.4.2.

6.3.2 Preference Mapping

The preference mapping method obtains a gamelock ($\boldsymbol{gL}$) in 98.2% of episodes. The mean time-to-lock ($\boldsymbol{TTL}$) over all episodes was 140.32 timesteps ($\boldsymbol{\sigma} = 170.25$). No game model went unrecognised. There were no false positives (i.e., assessing a gamelock in $\boldsymbol{g}$ as **g114**, for example, when $\boldsymbol{M}$ was constituted by **g211**).

A summary of the results for the observer is shown in **Table 6.3**, and a summary for each agent is listed in **Table 6.4**. Full results, for the entire RGS, are located in Appendix B.4.2.

Table 6.1: Gamelock by Reward, Summary. Strategy algorithm is *Q-Learning*, with asymmetric parameters (see §6.1). Each episode is 1000 timesteps. ***gL*** is gamelock count, i.e., the act of recognition. ***TTL*** is time-to-lock, i.e., how many timesteps until recognition occurs. The subsets of games listed in this table are a representative sample. Results for all games are located in Appendix B.4.2.

† Standard Deviation (σ) for population (all episodes, all game models) this row only, all others sample.
†† Standard Deviation (σ) over all episodes for the thirty-six game models in each layer.
††† Standard Deviation (σ) over all episodes for each individual game in each group.

Gamelock by Reward, Observer Summary Q-Learning, RGS, 1k							
Model	**Episodes**	***gL* Count**	***gL* Rate**	**Mean TTL**	**TTL σ**	**Min TTL**	**Max TTL**
All †	14400	14219	.987	103.62	149.15	4	999
Layer 1 ††	3600	3591	.998	70.75	91.42	4	989
Layer 2	3600	3545	.985	133.22	181.7	4	999
Layer 3	3600	3494	.971	139.05	182.2	4	998
Layer 4	3600	3589	.997	71.47	101	4	998
g111 †††	100	100	1	48.93	33.44	4	161
g112	100	100	1	58.68	47.48	6	317
g113	100	100	1	53.29	39.2	7	218
g114	100	100	1	60.71	43.8	4	246
g115	100	100	1	54.01	40.17	8	222
g116	100	100	1	53.21	30.53	4	136
g261	100	93	.93	210.83	251.1	13	942
g262	100	93	.93	181.13	211.9	5	807
g263	100	98	.98	201.27	261.3	9	922
g264	100	98	.98	187.88	222	5	864
g265	100	100	1	201.68	219.3	11	883
g266	100	96	.96	226.99	259	9	951
g361	100	93	.93	198.7	207	4	832
g362	100	90	.9	184.61	206.6	8	949
g363	100	93	.93	185.72	236.7	4	966
g364	100	92	.92	161.24	204.1	4	870
g365	100	99	.99	196.78	238.6	6	977
g366	100	99	.99	215.06	246.7	4	921
g421	100	100	1	56.31	34.31	4	200
g422	100	100	1	57.59	35.73	7	188
g423	100	100	1	57.14	39.2	6	212
g424	100	100	1	48.63	35.98	8	219
g425	100	100	1	51.52	34.17	5	156
g426	100	100	1	55.62	35.77	6	185

Table 6.2: Interlock by Reward, Agent Summary. Note that **TTL** (time-to-lock) is the point (timestep) at which the recognition process completes. With the direct reward method this occurs at the same time for both agents. The sequence of candidate matching leading up to this recognition point can vary or be identical. Compare this to Interlock by Preference, Agent Summary (**Table 6.4**) in which agents resolve recognition together and separately.

Strategy algorithm is *Q-Learning*, with asymmetric parameters (see §6.1). Each episode is 1000 timesteps. $\boldsymbol{gL}$ is gamelock count, i.e., the act of recognition. $\boldsymbol{TTL}$ is time-to-lock, i.e., how many timesteps until recognition occurs. The subsets of games listed in this table are a representative sample. Results for all games are located in Appendix B.4.2.

† Standard Deviation ($\boldsymbol{\sigma}$) for population (all episodes, all game models) this row only, all others sample.
†† Standard Deviation ($\boldsymbol{\sigma}$) over all episodes for the thirty-six game models in each layer (L).
††† Standard Deviation ($\boldsymbol{\sigma}$) over all episodes for each individual game in each group.

Interlock by Reward, Agent Summary

Q-Learning, RGS, 1k

		iL Count		iL Rate		Mean TTL		TTL σ		Min TTL		Max TTL	
Model	**Episodes**	A_0	A_1	A_0	A_1	A_0	A_1	A_0	A_1	A_0	A_1	A_0	A_1
All †	14400	14221	14221	.988	.988	103.76	102.87	149.52	149.77	4	4	999	999
L1 ††	3600	3591	3591	.998	.998	70.75	70.02	91.42	91.26	4	4	989	989
L2	3600	3545	3545	.985	.985	133.22	132.35	181.74	182.36	4	4	999	999
L3	3600	3496	3496	.971	.971	139.6	138.21	183.26	183.3	4	4	1000	1000
L4	3600	3589	3589	.997	.997	71.47	70.9	100.96	101.29	4	4	998	998
g111 †††	100	100	100	1	1	48.93	48.23	33.44	33.55	4	4	161	161
g112	100	100	100	1	1	58.68	58.28	47.48	47.68	6	6	317	317
g113	100	100	100	1	1	53.29	52.18	39.2	38.96	7	7	218	218
g114	100	100	100	1	1	60.71	60.67	43.8	43.65	4	4	246	246
g115	100	100	100	1	1	54.01	53.5	40.17	40.37	8	8	222	222
g116	100	100	100	1	1	53.21	52.09	30.53	30.1	4	4	136	136
g261	100	93	93	.93	.93	210.83	210.59	251.1	251.62	13	13	942	942
g262	100	93	93	.93	.93	181.13	181.06	211.9	212.29	5	5	807	807
g263	100	98	98	.98	.98	201.27	200.86	261.3	262.15	9	9	922	922
g264	100	98	98	.98	.98	187.88	186.78	222	223.03	5	5	864	864
g265	100	100	100	1	1	201.68	201.25	219.3	219.82	11	11	883	883
g266	100	96	96	.96	.96	226.99	226.06	259	259.92	9	9	951	951
g361	100	93	93	.93	.93	198.7	191.01	207	199.84	4	4	832	832
g362	100	90	90	.9	.9	184.61	184.1	206.6	207.24	8	8	949	949
g363	100	94	94	.94	.94	195.72	194.13	249.6	250.84	4	4	1000	1000
g364	100	92	92	.92	.92	161.24	158.5	204.1	204.83	4	4	870	870
g365	100	99	99	.99	.99	196.78	193.1	238.6	239.22	6	6	977	977
g366	100	99	99	.99	.99	215.06	214.41	246.7	247.45	4	4	921	921
g421	100	100	100	1	1	56.31	56.12	34.31	34.27	4	4	200	200
g422	100	100	100	1	1	57.59	57.25	35.73	35.84	7	7	188	188
g423	100	100	100	1	1	57.14	56.19	39.2	39.21	6	6	212	212
g424	100	100	100	1	1	48.63	48.52	35.98	35.96	8	8	219	219
g425	100	100	100	1	1	51.52	51.12	34.17	34.32	5	5	156	156
g426	100	100	100	1	1	55.62	55.3	35.77	35.87	6	6	185	185

Table 6.3: Gamelock by Preference, Summary. Strategy algorithm is *Q-Learning*, with asymmetric parameters (see §6.1). Each episode is 1000 timesteps. ***gL*** is gamelock count, i.e., sum of the individual acts of recognition. ***TTL*** is time-to-lock, i.e., how many timesteps until recognition occurs. The subsets of games listed in this table are a representative sample. Results for all games are located in Appendix B.4.2.

† Standard Deviation (***σ***) for population (all episodes, all game models) this row only, all others sample.
†† Standard Deviation (***σ***) over all episodes for the thirty-six game models in each layer.
††† Standard Deviation (***σ***) over all episodes for each individual game in each group.

Gamelock by Preference, Observer Summary Q-Learning, RGS, 1k							
Model	**Episodes**	***gL* Count**	***gL* Rate**	**Mean TTL**	**TTL *σ***	**Min TTL**	**Max TTL**
All †	14400	14136	.982	140.32	170.25	5	1000
Layer 1 ††	3600	3577	.994	109.8	128.11	5	993
Layer 2	3600	3527	.98	171.01	197.03	6	999
Layer 3	3600	3460	.961	175.32	199.13	6	1000
Layer 4	3600	3572	.992	105.16	129.14	7	998
g111 †††	100	100	1	70.8	46.66	10	276
g112	100	100	1	81.45	52.21	13	339
g113	100	100	1	78.31	51.33	11	286
g114	100	100	1	81.07	50.94	9	267
g115	100	100	1	75.14	46.24	14	242
g116	100	100	1	77.08	40.54	13	202
g261	100	91	.91	249.94	246.72	23	942
g262	100	93	.93	264.42	242.61	14	994
g263	100	98	.98	278.72	264.51	19	922
g264	100	98	.98	251.34	233.81	18	864
g265	100	100	1	266.08	232.83	18	883
g266	100	95	.95	264.46	251.55	16	951
g361	100	90	.9	251.01	236.49	19	925
g362	100	87	.87	241.43	227.19	8	975
g363	100	92	.92	266.45	262.55	14	1000
g364	100	90	.9	216.15	213.55	11	870
g365	100	94	.94	264.3	257.51	11	977
g366	100	93	.93	283.54	263.92	13	921
g421	100	100	1	72.29	41.63	11	200
g422	100	100	1	72.67	38.81	8	200
g423	100	100	1	72.13	43.48	18	251
g424	100	100	1	72.1	44.42	15	219
g425	100	100	1	70.61	40.85	7	210
g426	100	100	1	75.32	41.04	8	236

Table 6.4: Interlock by Preference, Agent Summary. Strategy algorithm is *Q-Learning*, with asymmetric parameters (see §6.1). Each episode is 1000 timesteps. gL is gamelock count, i.e., sum of the individual acts of recognition. TTL is time-to-lock, i.e., how many timesteps until recognition occurs. The subsets of games listed in this table are a representative sample. Results for all games are located in Appendix B.4.2.

† Standard Deviation (σ) for population (all episodes, all game models) this row only, all others sample.
†† Standard Deviation (σ) over all episodes for the thirty-six game models in each layer (L).
††† Standard Deviation (σ) over all episodes for each individual game in each group.

Interlock by Preference, Agent Summary

Q-Learning, RGS, 1k

		iL Count		iL Rate		Mean TTL		TTL σ		Min TTL		Max TTL	
Model	**Episodes**	A_0	A_1	A_0	A_1	A_0	A_1	A_0	A_1	A_0	A_1	A_0	A_1
All †	14400	14139	14162	.982	.983	136.46	128.44	168.9	163.91	4	4	1000	1000
L1 ††	3600	3577	3591	.994	.998	101.83	83.28	125.28	96.74	4	4	991	993
L2	3600	3530	3527	.981	.98	164.17	170.68	193.91	197.24	6	5	999	999
L3	3600	3460	3460	.961	.961	175.32	175.32	199.13	199.13	6	6	1000	1000
L4	3600	3572	3584	.992	.996	104.91	84.47	129.27	107.79	5	6	998	998
g111†††	100	100	100	1	1	63.67	60.66	43.16	39.92	8	10	276	251
g112	100	100	100	1	1	74.4	71.87	48.22	51.08	13	9	317	339
g113	100	100	100	1	1	74.33	62.62	50.87	41.86	11	9	286	234
g114	100	100	100	1	1	74.09	70.27	49.8	45.96	4	9	246	267
g115	100	100	100	1	1	66.44	67.85	42.24	46.09	8	9	222	242
g116	100	100	100	1	1	67.88	67.41	35.23	37.46	13	13	202	187
g261	100	91	91	.91	.91	236.12	249.94	244.48	246.72	17	23	942	942
g262	100	93	93	.93	.93	250.94	264.22	234.17	242.18	12	12	929	994
g263	100	98	98	.98	.98	265.45	278.3	260.02	264.88	9	19	922	922
g264	100	98	98	.98	.98	243.08	250.95	236.39	234.2	18	6	864	864
g265	100	100	100	1	1	263.2	265.28	233.94	233.65	18	14	883	883
g266	100	95	95	.95	.95	263.15	264.18	252.27	251.82	16	10	951	951
g361	100	90	90	.9	.9	251.01	251.01	236.49	236.49	19	19	925	925
g362	100	87	87	.87	.87	241.43	241.43	227.19	227.19	8	8	975	975
g363	100	92	92	.92	.92	266.45	266.45	262.55	262.55	14	14	1000	1000
g364	100	90	90	.9	.9	216.15	216.15	213.55	213.55	11	11	870	870
g365	100	94	94	.94	.94	264.3	264.3	257.51	257.51	11	11	977	977
g366	100	93	93	.93	.93	283.54	283.54	263.92	263.92	13	13	921	921
g421	100	100	100	1	1	72.05	64.32	41.83	36.9	11	11	200	200
g422	100	100	100	1	1	72.49	66.39	39.06	36.22	8	8	200	188
g423	100	100	100	1	1	72.12	64.4	43.48	28.96	18	18	251	212
g424	100	100	100	1	1	71.75	61.03	44.76	37.2	11	9	219	219
g425	100	100	100	1	1	70.61	59.72	40.85	31.89	7	7	210	156
g426	100	100	100	1	1	75.27	67.4	41.1	36.15	8	6	236	185

6.3.3 Three Dilemmas and a Coordination Game

This section presents the results of applying the recognition algorithm to three social dilemma games: Prisoner's Dilemma, Stag Hunt, Chicken; and also, a normalised Coordination game. Discussion of these results is held over to the next section. Variations of the four game models are shown in **Figure 6.3**: in **a)** the canonical Prisoner's Dilemma (Axelrod & Hamilton, 1981; Hofstadter, 1983; Axelrod, 1984), in **b)** the normalised form of the Prisoner's Dilemma; in **c)** a canonical form of Stag Hunt (Fang et al. 2002), and in **d)** an alternative canonical form of the same game (Powers, 2010); in **e)** a canonical form of Chicken (Kollock, 1998), and in **f)** an alternative canonical Chicken form by Leibo et al. (2017); lastly, in **g)** the normalised form of the Coordination game model from Fang et al. (2002). The original canonical form is identical to the ordinal form **g311,** so the game model's values have been normalised to allow evaluation of an alternative variation. The performance of the preference mapping method is shown in **Table 6.5**.

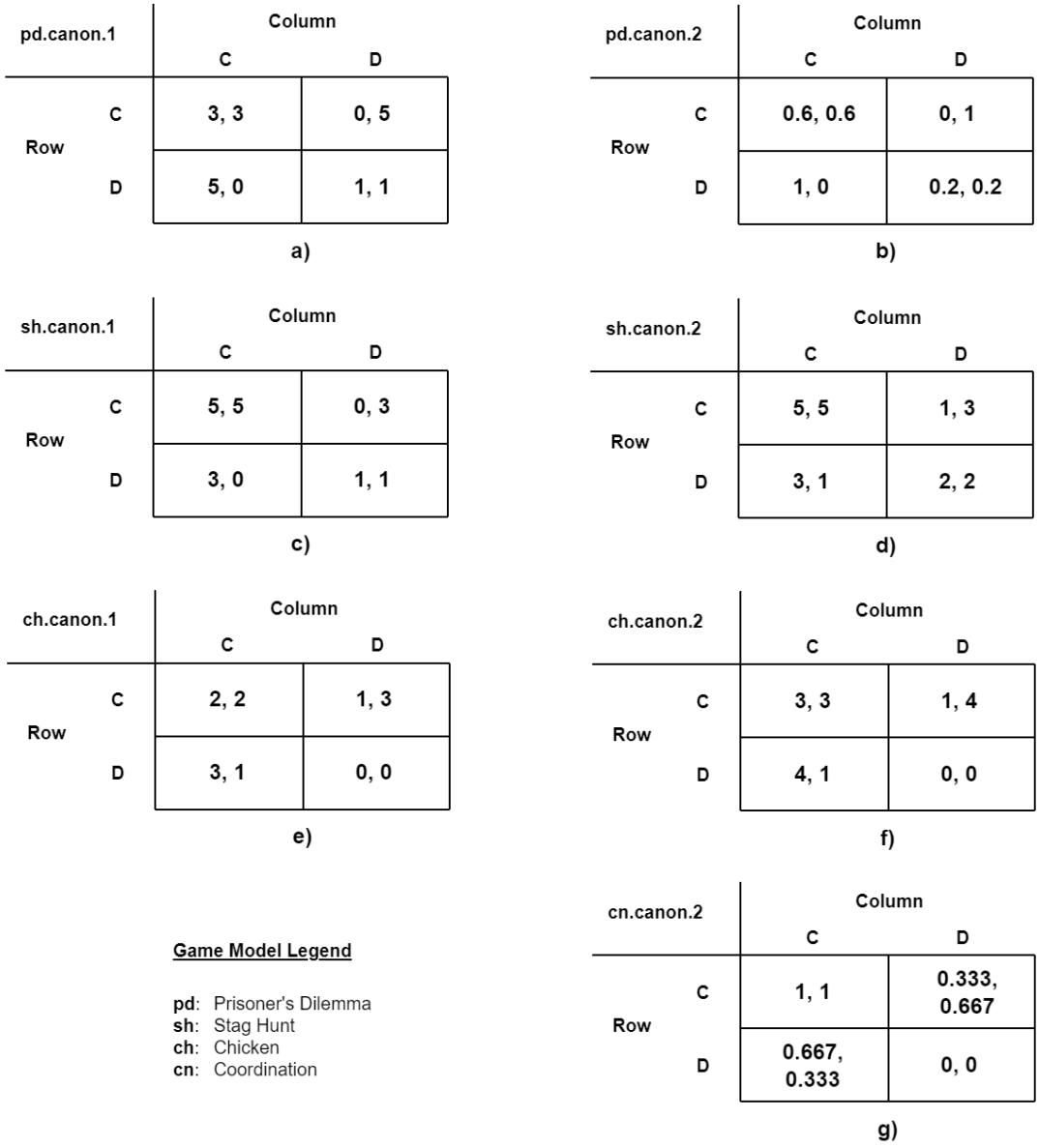


Figure 6.3: Seven canonical game models. Note that cn.canon.1 (which is not pictured), is identical to the ordinal form **g311**, so was not used in this experiment in favour of cn.canon.2 which is the normalised form of cn.canon.1 (**g311)**.

Table 6.5: Gamelock by Preference, Four Canonical Game Models.

† It is not evident in this table, but in episode 28 of the *ch.canon.1* experiment instance, the *Observer* failed to lock to a single candidate, instead offering two candidates—one of which was the correct ordinal model for Chicken, **g122,** and the other candidate was Chicken's neighbour **g121**. In **Table 6.6** this is seen as Agent One, in the same game model, failing to find an interlock, while Agent Zero did find an interlock candidate set for the same episode.

Gamelock by Preference, Observer Summary Q-Learning, Four Games, 1k						
Model	**Episodes**	***gL* Count**	***gL* Rate**	**Mean TTL**	**TTL σ**	**Min TTL**	**Max TTL**
pd.canon.1	100	100	1	70.67	52.75	9	244
pd.canon.2	100	100	1	74.02	43.64	11	221
sh.canon.1	100	100	1	79.41	53.74	10	307
sh.canon.2	100	100	1	96.78	56.29	18	250
ch.canon.1†	100	100	1	87.34	58.09	11	418
ch.canon.2	100	50	.5	94.54	60.24	8	260
cn.canon.2	100	100	1	114.73	72.58	15	325

Table 6.6: Interlock by Preference, Four Canonical Game Models.

† It is not evident in this table, but in episode 28 of the *ch.canon.1* experiment instance the *Observer* failed to lock to a single candidate, instead offering two candidates—one of which was the correct ordinal model for Chicken, **g122,** and the other candidate was Chicken's neighbour **g121**. This occurrence is evident in this table as Agent One, in the same game model, fails to find an interlock, while Agent Zero did find an interlock candidate set for the same episode.

Interlock by Preference, Agent Summary Q-Learning, Four Games, 1k													
		***iL* Count**		***iL* Rate**		**Mean TTL**		**TTL σ**		**Min TTL**		**Max TTL**	
Model	**Episodes**	A_0	A_1	A_0	A_1	A_0	A_1	A_0	A_1	A_0	A_1	A_0	A_1
pd.canon.1	100	100	100	1	1	69.97	60.59	52.75	45.01	6	6	244	227
pd.canon.2	100	100	100	1	1	73.33	64.42	43.97	39.15	10	9	221	221
sh.canon.1	100	100	100	1	1	78.51	78.6	53.68	53.18	10	10	307	307
sh.canon.2	100	100	100	1	1	95.1	94.4	57.43	55.6	18	18	250	250
ch.canon.1†	100	100	99	1	.99	74.32	73.69	45.13	55.28	10	10	239	418
ch.canon.2	100	50	50	.5	.5	87.46	67.24	58.05	41.44	8	6	260	143
cn.canon.2	100	100	100	1	1	107.27	105.58	70.7	66.77	15	10	325	306

6.4 Discussion

The aim of this experiment series was to evaluate two novel methods, *direct mapping*, and *preference mapping*, for the recognition of strategic dynamic equivalence referred to in this thesis as identifying cooperative intent. The first of these methods, the simple case, was described in §6.2.1, and the results of applying the method were given in §6.3.1. The second of the two methods extended the recognition process to map between non-ordinal game models and ordinal game models and was described in §6.2.2; the results of applying the second method were given in §6.3.2.

The results for the first method, presented in **Table 6.1**, indicate that *direct mapping* by the *Observer* component recognised the game model matrix in 98.7% of all episodes. The one-to-one mapping of ordinal game models to ordinal game models (the simple case) by the Observer gives a baseline result of **Mean TTL** = 103.62 timesteps, with $\boldsymbol{\sigma}$ = 149.15. These results confirm that the direct mapping method is functional and provides a baseline measure for comparison with the second method, preference mapping.

Under the preference mapping method, the Observer component's results, shown in **Table 6.3**, indicate a slightly reduced overall recognition rate of 98.2% of episodes resulted in a gamelock occurrence. The **Mean TTL** for this method, over all game models, was 140.32 timesteps, with $\boldsymbol{\sigma}$ = 170.25. The disparity between the two methods indicates that the second method takes longer to achieve a gamelock than does the first.

The *Agent Summaries* for each method, **Table 6.2** and **Table 6.4**, respectively, show that both agent's recognition components perform to an approximately equivalent standard. The increased times it takes both agents to complete an interlock in some sets of game models confirms that different regions of the *RGS* have different properties (Robinson & Goforth, 2005; Crandall et al., 2018a, 2018b). Perhaps counter-intuitively, it is the game models with a stronger cooperative bias (Crandall et al., 2018b; Bruns, 2012; Robinson & Goforth, 2005), such as the six game models from Layer Three (see **Table 6.4**), that take longer to find an interlock. The extended average time-to-lock under the second method may be a function of a lack of variance in agent actions—as discussed in §6.1, the agents need to undertake a certain amount of exploration to provide enough data to perform preference mapping. In game models where there is a dominant equilibrium with high payoffs for both participants there is incentive for them to become 'locked-in' (see §2.2.3.3**)**, and so not explore the action-space to the extent required for the preference mapping process to perform recognition.

An *interlock* is a set of candidate game models. For $\boldsymbol{A_0}$ the candidates will be the twelve games along the row that holds values in common in the same outcome position (or positions) as does the agent. For $\boldsymbol{A_1}$ the candidates are the column of twelve games with the same property of matching the agent's action preferences. In an experiment instance with the game model **g111** an interlock will consist of four sets of candidates, as shown in **Figure 6.4**. Each agent has two sets of candidates of six game models each. The intersection of both agent's candidate sets produces a gamelock for the Observer component. In the figure, the common outcome payoffs in each candidate set are marked in bold. As can be seen, for each agent, their two sets of candidates match on a different game outcome. However, the intersection of both agent's candidates does not rely on all four game outcomes to be matched. Further research into this process is a subject for future work.

Returning to the primary objective of game model recognition, §6.3.3 evaluated the preference mapping method by both agents and also an Observer component, on seven canonical game theory games, as shown in **Figure 6.3**. The results of the seven experiment instances were listed in **Table 6.5** and **Table 6.6**.

Except for the game of Chicken, both agents and the Observer have a 100% recognition rate of the correct ordinal game model. Chicken (**g122**) is the exception: the game model *ch.canon.1* (Kollock, 1998) is twin-identified by the Observer's gamelock as being either **g122**, *or* the neighbouring game model **g121**. The Observer can provide as many candidates as it finds over the course of an episode and this occurrence hints at shared

dynamics between the two game models **g122** and **g121**. In the episode (#28) that the Observer provided two candidates A_1 was not able to generate a set of candidates at all.

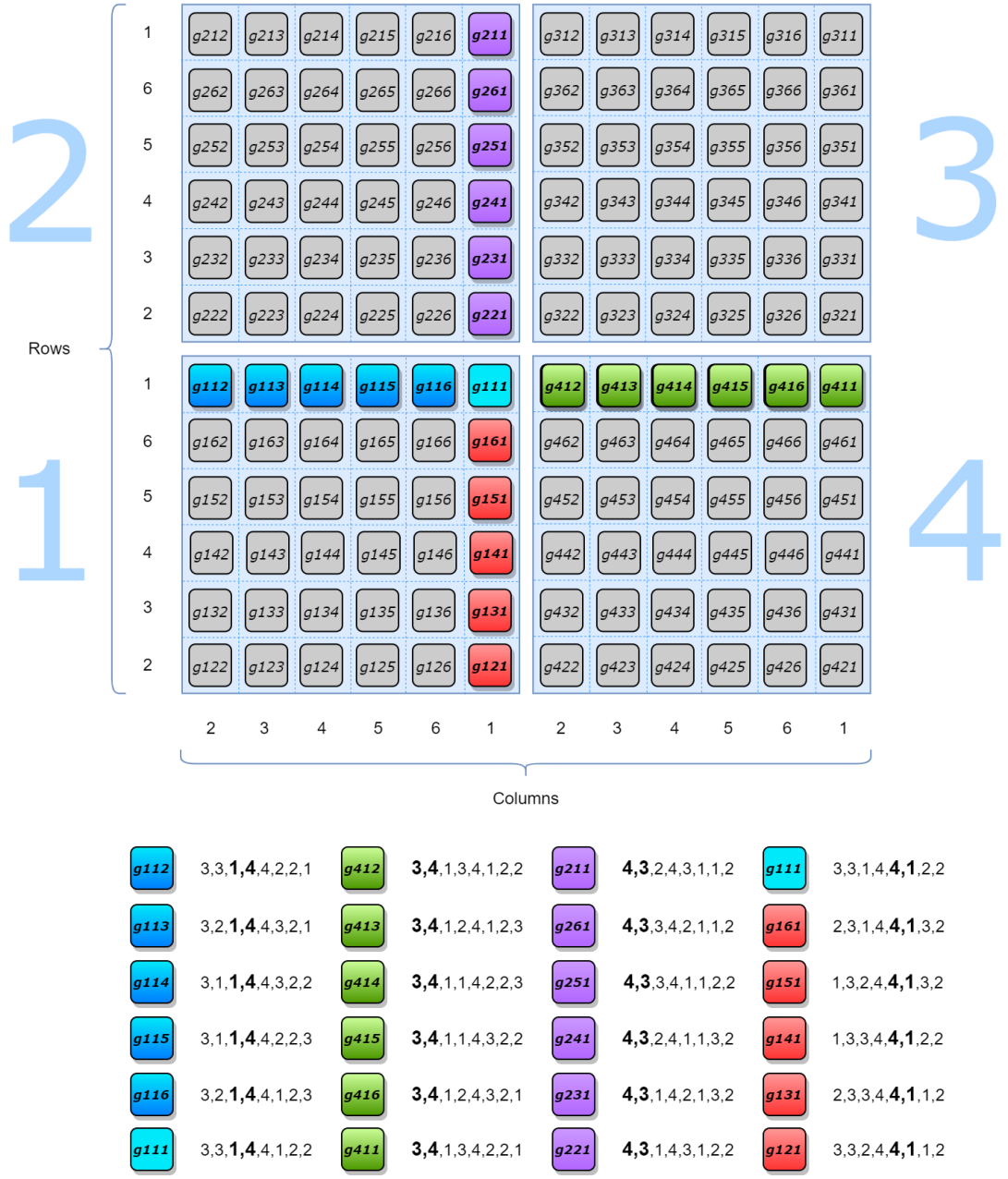


Figure 6.4: Interlock Candidate Sets.

Another case of interest, also in Chicken, was found using the game model *ch.canon.2*—the normalised form of *ch.canon.1* (see **Figure 6.3**). Here, both of the agents failed to generate an interlock in exactly half of the episodes. This flowed through to the Observer being unable to produce a gamelock in exactly half of the cases. The mechanism underlying this failure to lock, in either of the two Chicken cases, is as yet unclear, but initial thoughts are concentrated on the strictness (or not) of the encoded dynamics in the example game models. This question is also a matter for future work.

Notwithstanding these two issues, the overall rate of successful recognition using preference mapping on the set of canonical models is 92.71%. Further research to establish the boundaries of successful recognition is required to better understand the preference mapping process, and also to extend it to more complex domains such as sparse reward grid-world models. In such a setting, the recognition of a game model as a reflection of an agent's behaviour at a single point in time is a measure of the cooperative state of the algorithm at that time-instance. Reifying cooperative state by way of a recognition occurrence, to create a time-series data stream, provides for a first model of *cooperative intent*, as proposed in Chapter One.

Before closing this chapter, it is of practical interest to contrast the preference mapping procedure with the algorithm *Fictitious Play* (Brown, 1951; Robinson, 1951) which was introduced in §2.2.4 and discussed further in §4.1.2.1. On first impression they appear to be somewhat akin, partly due to their use of behaviour as a data stream in addition to reward, and partly due to the way 'belief-based' algorithms draw inference from observation of the *self*, as well as observation of the *other*.

The preference mapping recognition procedure differs significantly from *Fictitious Play* in that recognition orders the preferences for outcomes and also orders each outcome's constituent actions by preference for the reward those actions generate (as opposed to calculating the frequency of action and action-response, as in *Fictitious Play*). Therefore, the preference mapping method combines reward with behaviour by marrying the past actions of the agent with the results of those actions, to ordinally rank the agents' own preferences for the outcomes of those actions.

Extending preference mapping to model the external world (i.e., other agents) would enable application of the method as a strategy, or algorithm, however, the association of preference to action via reward, in others, would require visibility into the other participant's payoffs and this would violate the constraint of imperfect information over an agent's *state information visibility*.

The objective for future work incorporating a preference mapping method is to extend the method as a generalised mechanism for contextualising cooperative machinic state in environments with sparse rewards. A complex environment potentially contains more than one strategic dynamic. Further, an environment can be encoded with a game model's Nash Equilibria using Brouwer fixpoints (Daskalakis et al., 2009). Encoding an environment with an assertable strategic dynamic, as this method offers, would allow experiments to be conducted with a concretely-known 'ground-truth'. The goal then, for a generalised preference mapping method would be to identify Nash dynamics encoded in a complex, polymatrix (Janovskaja, 1968; Cai et al., 2016) game. For example, if a game based in a resource-gathering model has a number of players ($\boldsymbol{n > 2}$) and a variety of opportunities ($\boldsymbol{actions > 2}$) for an agent's actions to influence the resource dynamics of the game, then the potential for strategic dynamics in pairwise interactions to emerge will increase (Omidshafiei et al., 2020). In such a complex game, the compositional hierarchy (Chen et al., 2008) of pair-wise strategy dynamics generally requires an exponential number of bits to build a representation for all interaction pairs (Nguyen & Zick, 2018). *Response graphs* (Biggar & Shames, 2023) discard payoff values to represent a game model's strategic preference *structure*; aggregating these models with preference mapping may assist contextualised *SIV*-dependent observations on an environment.

To sum up, the principal objective of the thesis was to find a way to identify the strategic dynamics of an environment from an assessment of an agent's behaviour, on the belief that the pattern of behaviour that agents exhibit will be influenced in some way by the strategic dynamics in which they find themselves participating (see §1.2.1). This is demonstrated, within the constraints of the thesis (§1.3), by the development and evaluation of a method equivalent to an identity function, performing a process of preference mapping over the game models of the *rRGS*, in §6.3.2.

Finally, in §2.3.1.2, and also in §6.1, Rapoport and Chammah's (1965) proposition that a game model dynamic would be reflected in a participant's pattern of play was raised. Their proposition is confirmed by the findings of this chapter: the preference method operating over canonical game models reveals the preference an entity has for the outcomes resulting from their actions, and in a discrete preference space this reveals 'signature' patterns in behaviour that identify the strategic dynamic at hand.

Chapter Seven

In Conclusion

'If a word can heal, a word can wound,' the witch said. 'If a hand can kill, a hand can cure. It's a poor cart that goes only one direction.'

— U. K. Le Guin, *Dragonfly*[62]

This thesis has explored a single idea along two paths. The idea is to imbue an agent with a *relational* component based in *cooperative intent*, tied to the internal state of the machine. A relational component operates in the interface between an agent and other entities—be they human or machine. Developing a method to perform this task constitutes the primary research objective, specifically, to develop an algorithm for the observation and identification of cooperative intent (see §1.2.1.1), in the behaviour of an agent.

The first path this objective has been approached from is that of experimental proving; i.e., empirical iteration in a constrained modelling domain to isolate, identify, and iterate on the core requirements of the algorithm. The second path has taken a wider view and attempted to illustrate issues of human and AI co-existence in a societal context, with respect to our understanding of cooperation in principally the biological sciences, and secondly, through an exploration of relationality. Though separate, the two paths are intertwined, and they inform each other. This interplay is now summarised with a short account of the thesis chapter by chapter.

7.1 Empirical and Conceptual Findings

The empirical path was first introduced in Chapter One by way of presenting the research question underpinning and motivating this thesis. Conceived of in the AI space, this work started with investigating cooperation in the context of multi-objective reinforcement learning; however, the focus of the project quickly gravitated to game theory model spaces, via the Prisoner's Dilemma game model. The Prisoner's Dilemma can be reduced to a point in topological space—a space that consists of an array of other points that *are also* numerical game models; similar to the Prisoner's Dilemma, each having differing strategic properties.

[62] Ursula. K. Le Guin, *Dragonfly*; in *Tales of Earthsea* in *The Books of Earthsea*; *The Complete Illustrated Edition*. First Saga Edition. New York, Saga Press (2018).

The variety of strategic dynamics across game models is encoded in each model's numerical structure. Strategic dynamics are game-theoretic concepts that identify locus points around which participants of a game will be attracted (von Neumann, 1928b). The mechanisms for identifying these points are generally computationally expensive, so are not ideal for calculating on-the-fly (see §1.3.1 and §2.2.4). Nevertheless, these properties inform our understanding of the dynamic represented in the game model. For example, the existence of two Nash Bargaining Solutions in a game model may induce in two participants a cyclic alternation between the two solutions, such that over time each participant gains an equal share of the available reward (Crandall et al., 2018a).

Importantly, not all game models are actually dilemmas. The game models in the Robinson and Goforth (2005) topology span a range of dynamics, from representing conflict through to representing cooperation. However, the topology cannot represent all possible strategic dynamics; in either domain or gradation. In a discrete domain the latter of these, gradation, refers to the 'gaps' between models, as the topology does not model tied preferences, which immediately omits an entire class of dynamics involving equivalent utility. Therefore, each discrete point (game model) in the topology has a relation to another discrete point (game model), separated by strict ordinality[63].

The domain of representation that the topology inhabits is of a binary nature: given two options, which is preferred? Given two participants, each with two options, of the four possible outcomes, which is preferred? And what is the preferred ordering of the remaining outcomes? The complexity of decision-making in the real-world, on first presentation, rarely seems to reduce to a clear choice between two options, not least due to the complexity inherent in solving utility representations (§2.2.4). To make these various concerns extant in an experimental domain, Chapter One concludes by demarcating the scope and the constraints applied in each experiment series.

The second path that this thesis has attempted to follow is to try and understand how a relational component could assist with the issues of AI that are of importance and traction in the world today. The 'imagining' of AI is more than a recount of historical stories of automatons, and more than the mythology that emanates from cultures around the world; it is also the gathering of contemporary voices speaking on AI; saying that as a community we can build AI as we wish (Cave et al., 2023); furthermore, as Walsh et al., (2019, p. 187) conclude, we can "ensure that the development of AI does not come at the expense of human rights" by prioritising *equity*, that is, by centering a concern for "reducing inequity" in the development of AI (2019, p. 8).

In embracing a diversity of voices, we gain a pluralistic view of how AI can integrate into our daily lives. Among the diverse cultural imaginings and modern-day deployments of AI outside of the Western sphere, the conception of AI as *companion*, or *partner* (Katsuno & White, 2023), is a dominant theme in embodied AI and robotics; exemplified in South Korea, China, and Japan (see §1.1); but also present among cultures and societies in the Global South, where there is a thematic conception of AI as *helper* in a role that emanates from community, ritual, and ceremony (Lewis, 2023).

Chapter Two began with a placing of the concept of cooperation as a modality. Cooperation, as a modality, has been studied extensively in the biological sciences. Each

[63] In later work, Robinson, Goforth and Cargill (2007) extend their treatment to a model with tied preferences.

modality, whether it be an agreed upon theory, or hotly debated, offers to an ethological machine behaviour interpretation a possibility of a potential algorithmic *modus operandi*. A modality can be described in a biological setting, and its accuracy can be debated, endlessly (Herbers, 2009). For a machine, a modality is a potentially valid method of operation, as machines are not bound by biological evolution, nor are they bound by the constraints on cooperation that evolution places (notwithstanding second-order effects, i.e., the effects of evolution on us as creator of the machine). Reviewing cooperative modalities also provides insight into the dynamics of cooperation from the perspective of game theory, in no small part due to the extensive use of dilemma games in modelling real-world processes.

From this discussion and review of cooperative modalities, Chapter Two moved to a review of computational learning and its use in game theory models. The formal context of a Markov Game was introduced along with discussion of types of algorithms and the pros and cons of applying them in the experimental domain. Chapter Two concluded by introducing the Robinson and Goforth (2005) topology. The practical use to which the topology would be put in this thesis was outlined in §2.4.

Chapter Three isolated a subset of features of the full Robinson and Goforth topology to incorporate as the graph $\boldsymbol{G}$; to clearly specify what aspects of the topology were in play in the experiments to follow, and which aspects were not. Chapter Three defined this reduced form of the topology, an *rRGS* graph, and then illustrated and further explained concepts from the topological model such as 'swaps', i.e., a mechanism for the application of group-theoretic generators onto the points (game models) in the topological space, to produce edges between the points and form a connected graph.

Chapter Four accounts for a multi-model tournament experiment series in the style of Axelrod (1980a, 1980b, 1984) and Crandall et al. (2018a). The primary purpose of this experiment series was to get to know the game theory domain, by creating a platform-independent, HPC-friendly[64], experiment framework to run multiple experiment types (*one-on-one*, *symmetric selfplay*, *asymmetric selfplay*, *parameter study*, '*spacewalk*', and *single-model* and *multi-model tournaments*)[65]. The empirical work described in this chapter attempts to plot a straight course with respect to concentrating on the tournament and the issues that arise in the conduct of a tournament; despite the opportunities for exploring multiple avenues of inquiry due to both the wealth of data obtained from tournament experiments, and the many opportunities to pursue research threads within the tournament context. One outcome of this experiment series is a novel output: a synthesis of literature to produce a listing of the mutual cooperation locations in the game models that comprise the topology, along with both cartesian and canonical payoff vector orderings. Another outcome of this experiment series was the observation of unexpected variance in agent behaviour under certain conditions. This observation was judged unusual enough to warrant an experiment series of its own, the results of which were presented in the next chapter.

[64] High Performance Cluster; running PBS service allows concurrent execution of multiple independent instances, squishing a minimum 144-hour workflow down to one or two hours.

[65] Though implemented, the *spacewalk* experiment type was not used in the thesis. A spacewalk runs *symmetric selfplay* instances across the *rRGS* and obtains metrics regarding the make-up and character of the game models themselves, both individually, and as members of a tile or layer aggregation, as defined in the full RGS model, see §2.3.3.4 and Chapter Three.

Chapter Five is an account of the experiment series set up to pursue this observation of variance in agent behaviour. The question of whether agent behaviour is sensitive to the representation of the game model was answered in the affirmative as the previously observed effect of variance in agent behaviour under varying isomorphic representations of the game model was confirmed statistically by a Wilcoxon Signed-Rank test. The treatment applied to the experiment was the transformation of the representation of the game model such that the strategic dynamics of the game model structure are maintained, i.e., the representations are isomorphic. As a game model subject to the social dilemma inequalities the null hypothesis for the experiment series was that the behaviour that agents would exhibit under varying representations would correlate to an equivalent frequency of game outcomes, regardless of any isomorphic representation.

This finding, that agent behaviour is *not* equivalent under isomorphic representations (in this contextualised domain) forced a reconsideration of the direction of the empirical path. To this point the thought was that the structure of the game model was the crucial factor; that being able to identify the strategic dynamic(s) inherent to an environment would allow a mapping to the topology. As it turns out, this term, *game model recognition*, is something of a misnomer as its first use in the thesis referenced the game model constituting the environment, such that the process would operate over the agent's reward signals before the observation function $\boldsymbol{O_i}$ was involved. As a result of Chapter Four's findings, it can not be assured that identifying a strategic dynamic in the environment would hold with respect to an agent's behaviour. In light of this, the thesis adapted by performing game model recognition not on the environment, but on the agent's own behaviour. This would place the recognition process as a component of the agent's observation function $\boldsymbol{O_i}$.

This change in the thesis' focus is justified, for if an agent's behaviour varies, under the same game model but with a varying representation, then identifying the game model in the environment would not necessarily shed light on the dynamic(s) that the agent may exhibit in that environment. Given this was a turning point in the thesis, consideration for an investigation to explain the effect was deferred in favour of maintaining the original objective.

Chapter Six describes three experiments. The first experiment is conducted as a base case to demonstrate a simple procedure for identifying a game model dynamic from the reward signals in the environment in order to gain baseline data. The second experiment modifies the procedure to demonstrate operation of recognition from behaviour rather than reward. The third experiment is a series of tests of the modified procedure on canonical game theory models and demonstrates single-game-model resolution by an Observer entity. For an agent, the recognition resolution is one in twelve. That is, the best a single agent can do, under the constraint of perfect- and incomplete-information, is to identify a set of twelve candidate game models, of which one *is* a valid mapping to the environment's game model matrix, i.e., it is the same game model as constituting the strategy dynamics in the Markov Game $\boldsymbol{M}$. The second method, preference mapping, sources its input from the pattern of actions and outcomes resulting from the agent's behaviour. By considering this behaviour with respect to received reward a mapping is created to the preferences the agent has for its actions, given the strategic dynamic(s) inherent to the environment. Interestingly, one of the tests in the third experiment, performed on a variant of the Chicken game model, failed to map, i.e., did not perform

recognition, in 50% of its cases. Speculatively, this could be due to the exact values of different game models not reflecting the same dynamic as is held in the ordinal representation. Alternatively, the game model may be a variation of the ordinal 'archetype', fitting somewhere in the space between the discrete game models of the topology. Investigating this would be one avenue for future work; another would be to investigate the bounds of a positive linear transformation over a wider range of sample game models. To this point in the thesis, the social dilemma inequalities have enabled assertions over equivalence, but as the topology is, at best, eleven twelfths *not* social dilemmas (Robinson & Goforth, 2005), this approach to asserting strategic dynamics has to be integrated into something more comprehensive in order to adequately separate the archetypal game models from the potentially similar models that do not exhibit an exact match to the strategic dynamics of the archetype.

7.2 Contribution

Overall, in this thesis, Chapters One and Two contribute to a discussion about what AI can be if we choose to pro-actively build what we want, rather than allow 'market forces' to determine the evolution of AI. Along with this conceptual thread the series of three experiments presented in Chapters Four, Five, and Six contribute to computational game theory, to computational learning, to reinforcement learning, and to the study of algorithmic cooperation. The principal contributions of this thesis are now presented; this is followed by a compilation of novel outputs and other contributions.

Within the domain of computational learning this thesis is a contribution to modelling cooperation in a machine context; and within this domain the contribution of a measure of machine cooperation, i.e., *cooperative intent*. Cooperative intent, in an agent, is a representation of the internal state of the agent, manifested as an observable behaviour (see §1.2.1) and is *sans causal motivation*—it is the identification, or recognition, of a point in the topological space. Recognition is then the reflective act of reifying the *identity* group-operation.

Cooperative intent may have potential in the areas of intrinsic motivation (Barto et al., 2004; Oudeyer & Kaplan, 2007); curiosity-driven exploration (Pathak et al, 2017); the *relationality* models of human and AI as co-existing entities; by way of *cooperation-as-policy* to the policy component in actor/critic architectures; to the use of reflection in the context of 'self-aware' computing (Lewis, 2015), to self-referential and continual reinforcement learning (Smith, B. C., 1982; Khetarpal et al., 2022), and, to intention recognition research (Han, 2013).

To the study of cooperation and game theory this thesis contributes empirical work in the area of topological game theory, and in addition contributes to specific topics within the field, principally the findings presented in Chapter Six confirm Rapoport and Chammah's (1965) conjecture that participant behaviour would be influenced, and identifiable, by the strategic dynamics of the game being played.

The domain of application for this first iteration of the preference mapping method is a discrete space. By tapping into the semantic power of game theory models, which are by nature numerical devices, the overall contribution of this thesis is to first position a metric of cooperative intent as a relationality device that bridges human understanding

(in a continuous space) on the one hand, with AI computation (in a discrete space) on the other; and second, to have developed a relatively simple method for mapping preference from behaviour.

The work in this thesis has generated a number of novel outputs:

- A synthesis of strategic dynamics (*Nash Equilibria*, *Nash Bargaining Solutions*, *MaxiMin solution*) presented in both cartesian and canonical payoff vector ordering with locations of mutual cooperation (**MC**) outcomes for each game model in the *RGS* (see Table B.7 in Appendix B.2.2).
- An extant adjacency list of the graph $\boldsymbol{G}$; $\boldsymbol{G} = (\boldsymbol{V_{144}}, \boldsymbol{E_{336}})$; where edges are formed by the application of the generators $\boldsymbol{R_{12}}$, $\boldsymbol{R_{23}}$, $\boldsymbol{R_{34}}$, $\boldsymbol{C_{12}}$, $\boldsymbol{C_{23}}$, and $\boldsymbol{C_{34}}$ to every game model in the *RGS*. Note there are a total of 336 undirected edges in the graph $\boldsymbol{G}$, 480 when including the 144 identity edges $\boldsymbol{e_1}$ (see Table B.1 in Appendix B.1).
- Chapter Five demonstrates *algorithmic bias*, as opposed to *bias-in-data*, in *fRL* algorithms and is a contribution to computational learning principally with respect to the understanding of agent behaviour; and also, to game theory by providing empirical data and significant findings to the under-explored topic of representational stability.
- The experimental framework developed in this thesis for the running of multi-model tournaments is open-source (see Appendix B.5).
- The generalised normal-form matrix presented in **Figure 2.2**, see §2.3.1.

7.3 Future Work

Several items have been identified in the thesis as opportunities for further work and these are reproduced below in point-form. Overall, the very next thing is to explore possibilities in using *cooperation-as-policy* in a sparse-reward and competitive-resource model, using contemporary reinforcement learning algorithms. Moving away from an agent's preference for reward to instead map an externally applied metric on a behaviour would allow preference mapping to track that specific behaviour, for example, if *Farmer Joe* keeps getting too many chickens on a shared allotment, then the act of getting another chicken may be interpreted as an *act of defection*. Preference mapping may be able to decompose the dynamics of the public good game into its sub-game components—and then track the trajectories of agents, and the game, to see if these subgames vary, and if so, how they vary in relation to the outcomes of the game itself. In addition, the application of a target game model for an agent to aspire to would assess the application of the preference mapping method to *cooperation-as-policy*.

If it is found from a first sparse rewards model that sub-games are identifiable then a range of potential modelling opportunities become viable. Abstracting the notion of cooperation allows an agent to abstract the application of the preference mapping method; for example, it can be used to track the agent's relationship to the receipt of positive rewards (i.e., gain, or positive reinforcement) over negative rewards (i.e., harm, or negative reinforcement) so that the agent can reflect on it's own overall state in relation to the two variables through the discrete, prismatic lens of the *RGS*.

Therefore, a second avenue for future work in a sparse rewards paradigm is to design and evaluate an experiment based on a model of mammal reproductive strategies as described by Raihani (2021). In this model, Raihani explains that as gender ratios in a population change so do the preferences that males and females show for adopting polygyny or monogamy. Where the number of females to males is greater than half, the males will show an increased preference to polygyny, and conversely, when the number of females to males is less than half, the males will show an increased preference to monogamy. As this is essentially a binary action space it allows the modelling of the reproductive dynamic, such that an agent's actions (to be either polygynous or monogamous) can be translated to a preference structure. As a preference structure my conjecture is that the preferences that the agents would display under various ratios of males to females will map to a game model in the *rRGS*. Raihani details the preferences that males would exhibit under several ratios but does not explicitly detail the preferences that females would exhibit, providing an opportunity to confirm Raihani's description for male behaviour, and in addition, an opportunity to assess the predictive power of the preference mapping method by pre-registering expectations of preferences for both male and female participants.

A further possibility in moving from the pairwise-interaction models explored in this thesis to more complex, $\boldsymbol{n}$-person games would be the expanded scope for examining the generalisation of *cooperation-as-policy*, and the recognition of *cooperative intent*, in complex scenarios that more obviously mirror real-world situations. In the pairwise models investigated in this research the relationship between an agent and the external world (i.e., the other agent) is expressed as a *game-against-nature* (see §6.1), allowing a single model of cooperative intent, per agent. To generalise to $\boldsymbol{n}$-person games this expression can be reformulated, for example, each agent could model each and every pair-wise interaction it has with any other agent as a game-against-nature, and then use this set of models as input to downstream policy forming operations. Similarly, an agent could identify groups of agents as an entity with which it could seek to model cooperative intent. An important concern to be clear about with any expanded scope is to maintain clarity over the constraints in operation, particularly those of perfect-/imperfect- and complete-/incomplete-information (see 1.3).

Expanding the scope that the work is applied to, if tractable and otherwise found to be useful, may indicate that modelling cooperative intent in $\boldsymbol{n}$-person environments with sparse rewards, will have potential for direct application to current and near-future real-world applications, such as embodied robotics, assistive agents, or resource management; however, some of these application domains do remain speculative, given the research is of a fundamental nature, as outlined in Chapter One.

Additional possibilities for future work are:

- extend preference mapping to be a generalised discrete mechanism for contextualising cooperative machine state in environments with sparse rewards, where the environment potentially contains more than one strategic dynamic and investigate whether multiple dynamics can be identified in a given temporal interval (§6.4).

- investigate strict interpretation of dynamics and inequalities on structure of game model and consequent recognition, for example the boundaries of Chicken, raised in §6.4.
- investigate in more detail the set formations in the interlock process, §6.4.
- investigate effect, if any, of an agent's behavioural-variance on recognition time, §6.1.
- investigate the equivalence of behaviour exhibited by agents using non-linear function approximation and policy-gradient architectures, §5.4.2.
- in multi-model tournaments, empirically map the *rRGS MCR* for different algorithms; perform ablation; construct a generalised linear model of algorithm components; and introduce more algorithms, §4.3.
- translate a sparse rewards model to focus on coarse-grain consolidation, see §2.3.
- to expand and make publicly available an earlier iteration of the *agent model* (Appendix A), a web-application called *agent model explorer*, which was implemented using *Dash* (Plotly, 2015) and enabled in-browser single-episode matches between algorithms, with configurable hyperparameters.
- Apply deep learning methods to the game model recognition algorithm to potentially increase generalisation, transferability, and specificity, under varying levels of *SIV*.
- Create a visual interface to the *rRGS* such that game model points have visually-associative cues, so as to create a platform for investigating any affinity between semantic, i.e., human, understanding of a dynamic, and its potential correlates in the *rRGS*.
- investigate the *spacewalk* experiment type to gather empirical understanding of the *rRGS* topology, see Appendix A.

References

Abbot, P., Abe, J., Alcock, J., Alizon, S., Alpedrinha, J. A. C., Andersson, M., Andre, J.-B., van Baalen, M., Balloux, F., Balshine, S., Barton, N., Beukeboom, L. W., Biernaskie, J. M., Bilde, T., Borgia, G., Breed, M., Brown, S., Bshary, R., Buckling, A., … Zink, A. (2011). Inclusive fitness theory and eusociality. *Nature*, 471(7339), E1–E4. https://doi.org/10.1038/nature09831

Adams, R. (2023). Artificial Intelligence Elsewhere: The Case of the Ogbanje. In Cave S., & Dihal K. (Eds.), *Imagining AI: How the World Sees Intelligent Machines*. (pp. 261–274). Oxford University Press.

Allen, M., Poggiali, D., Whitaker, K., Marshall, T. R., van Langen, J., & Kievit, R. A. (2021). Raincloud plots: A multi-platform tool for robust data visualization. *Wellcome Open Research*, 4(63). https://doi.org/10.12688/wellcomeopenres.15191.2

Alpern, S., Gal, S., Lee, V., & Casas, J. (2019). A stochastic game model of searching predators and hiding prey. *Journal of the Royal Society Interface*, 16(153), 20190087. https://doi.org/10.1098/rsif.2019.0087

American Heritage Dictionary of the English Language, (2023) Intent. In American Heritage Dictionary of the English Language, Harper Collins. Retrieved March 23, 2023, from https://www. ahdictionary.com/word/search.html?q=intent

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). *Concrete Problems in AI Safety*. https://doi.org/10.48550/arXiv.1606.06565

Anderson, L. (1982). Walking and Falling. On *Big Science* [cassette]. New York City, Warner Bros.

Anderson, L. (1984). Walking and Falling. On *United States Live* [LP]. New York City, Warner Bros. (1983)

Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016, May 23). Machine Bias. *ProPublica*. Retrieved July 5, 2021, from https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Areco, M. (2023). Imaginaries of Technology and Subjectivity: Representations of AI in Recent Latin American Science Fiction. In Cave s., & Dihal K. (Eds.), *Imagining AI: How the World Sees Intelligent Machines*. (pp. 192-209). Oxford University Press.

Armentano, M. G., & Amandi, A. (2007). Plan recognition for interface agents: State of the art. *Artificial Intelligence Review*, 28(2), 131–162. https://doi.org/10.1007/s10462-009-9095-8

Armstrong, S., Bostrom, N., & Shulman, C. (2016). Racing to the precipice: A model of artificial intelligence development. *AI & Society*, 31(2), 201–206. https://doi.org/10.1007/s00146-015-0590-y

Ashlock, D., & Kim, E. (2008). *Fingerprinting: Visualization and Automatic Analysis of Prisoner's Dilemma Strategies*. IEEE Transactions on Evolutionary Computation, 12(5), 647–659. https://doi.org/10.1109/TEVC.2008.920675

Ashlock, D., Kim, E., & Ashlock, W. (2010). *A fingerprint comparison of different Prisoner's Dilemma payoff matrices*. Proceedings of the 2010 IEEE Conference on

Computational Intelligence and Games, 219–226. https://doi.org/10.1109/ITW.2010.5593352

Astrachan, O., Berry, G., Cox, L., & Mitchener, G. (1998). *Design patterns: An essential component of CS curricula*. SIGSCE 98. http://dx.doi.org/10.1145/273133.273182

Axelrod, R. (1980a). Effective Choice in the Prisoner's Dilemma. *The Journal of Conflict Resolution*, 24(1), 3–25. https://www.jstor.org/stable/173932

Axelrod, R. (1980b). More Effective Choice in the Prisoner's Dilemma. *The Journal of Conflict Resolution*, 24(3), 379–403. https://www.jstor.org/stable/173638

Axelrod, R. (1984). *The Evolution of Cooperation*. Basic Books.

Axelrod, R., & Hamilton, W. D. (1981). The Evolution of Cooperation. *Science*, 211(4489), 1390–1396. https://doi.org/10.1126/science.7466396

Axelrod-Python/Axelrod. (2015). [Python]. *Axelrod-Python*. Retrieved November 3, 2020, from https://github.com/Axelrod-Python

Axelrod-Python/TourExec. (2020). [Fortran]. *Axelrod-Python/TourExec*. Retrieved March 12, 2020 from https://github.com/Axelrod-Python/TourExec

Azar, O. H. (2019). The influence of psychological game theory. *Journal of Economic Behavior & Organization*, 167, 445–453. https://doi.org/10.1016/j.jebo.2018.09.009

Banerjee, D., & Sen, S. (2007). Reaching pareto-optimality in prisoner's dilemma using conditional joint action learning. *Autonomous Agents and Multi-Agent Systems*, 15(1), 91–108. https://doi.org/10.1007/s10458-007-0020-8

Barto, A. G., Singh, S., & Chentanez, N. (2004). *Intrinsically Motivated Learning of Hierarchical Collections of Skills*. Retrieved July 1, 2023 from https://web.eecs.umich.edu/~baveja/Papers/Barto-Singh-Chentanezfinal.pdf

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). *On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?* 🦜. Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 610–623. https://doi.org/10.1145/3442188.3445922

Benjamin, R. (2019). Race After Technology: Abolitionist Tools for the New Jim Code. Polity.

Bhattacharya, A. (2021). *The Man from the Future*. Allen Lane Penguin Random House.

Biggar, O., & Shames, I. (2023). The graph structure of two-player games. *Scientific Reports*, 13(1), Article 1. https://doi.org/10.1038/s41598-023-28627-8

Billard, E. A. (1996). Adaptation in a stochastic prisoner's dilemma with delayed information. *Biosystems*, 37(3), 211–227. https://doi.org/10.1016/0303-2647(95)01560-4

Binmore, K. (2007). *Playing for Real: Game Theory*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195300574.001.0001

Binmore, K., Rubinstein, A., & Wolinsky, A. (1986). The Nash Bargaining Solution in Economic Modelling. *The RAND Journal of Economics*, 17(2), 176. https://doi.org/10.2307/2555382

Birhane, A., Prabhu, V., Han, S., & Boddeti, V. N. (2023). *On Hate Scaling Laws for Data-Swamps* (arXiv:2306.13141). arXiv. Retrieved June 30, 2023 from https://doi.org/10.48550/arXiv.2306.13141

Birhane, A., Kalluri, P., Card, D., Agnew, W., Dotan, R., & Bao, M. (2022). *The Values Encoded in Machine Learning Research*. 2022 ACM Conference on Fairness, Accountability, and Transparency, (pp. 173–184). https://doi.org/10.1145/3531146.3533083

Blake, W., (1803; 1983) *Auguries of Innocence*. In Bronowski, J. (Ed.), *William Blake* (pp. 67–71). Penguin.

Bondi, E., Oh, H., Xu, H., Fang, F., Dilkina, B., & Tambe, M. (2019). *Using Game Theory in Real Time in the Real World: A Conservation Case Study*. Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, 2336–2338.

Bostrom, N. (2003). Are You Living in a Computer Simulation? *Philosophical Quarterly*, 53(211), 243–255.

Bostrom, N. (2012). The Superintelligent Will: Motivation and Instrumental Rationality in Advanced Artificial Agents. *Minds and Machines*, 22(2), 71–85. https://doi.org/10.1007/s11023-012-9281-3

Bowling, M., & Veloso, M. (2001). *Rational and convergent learning in stochastic games*. Proceedings of the 17th International Joint Conference on Artificial Intelligence, Volume 2, 1021–1026.

Boyd, R. (1988). Is the repeated prisoner's dilemma a good model of reciprocal altruism? *Ethology and Sociobiology*, 9(2–4), 211–222. https://doi.org/10.1016/0162-3095(88)90022-2

Boyd, R. (1989). Mistakes allow evolutionary stability in the repeated prisoner's dilemma game. *Journal of Theoretical Biology*, 136(1), 47–56. https://doi.org/10.1016/S0022-5193(89)80188-2

Boyd, R., & Lorberbaum, J. P. (1987). No pure strategy is evolutionarily stable in the repeated Prisoner's Dilemma game. *Nature*, 327(6117), 58. https://doi.org/10.1038/327058a0

Bradley, B. (2020). *Darwin's Psychology*. Oxford University Press. https://doi.org/10.1093/oso/9780198708216.001.0001

Brams, S. J. (1994). *Theory of Moves*. Cambridge University Press. https://nyuscholars.nyu.edu/en/publications/theory-of-moves

Brams, S. J. (1993). Theory of Moves. *American Scientist*, 81(November-December), 562–570. Retrieved November 2, 2019 from https://www.acsu.buffalo.edu/~fczagare/Game%20Theory/Theory%20of%20Moves.pdf

Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. (2021). *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. arXiv:2104.13478 [Cs, Stat]. Retrieved May 3, 2021 from http://arxiv.org/abs/2104.13478

Brooks, R. (2017, September 7). *The Seven Deadly Sins of Predicting the Future of AI. Robots, AI, and Other Stuff*. Retrieved May 20, 2023 from https://rodneybrooks.com/the-seven-deadly-sins-of-predicting-the-future-of-ai/

Brown, G. W. (1951). Iterative Solutions of Games by Fictitious Play, in Koopmans T. J. (Ed.) *Activity Analysis of Production and Allocation* (pp. 374–376). New York: Wiley.

Brown, J. S., Laundré, J. W., & Gurung, M. (1999). The Ecology of Fear: Optimal Foraging, Game Theory, and Trophic Interactions. *Journal of Mammalogy*, 80(2), 385–399. https://doi.org/10.2307/1383287

Bruns, B. (2010). Navigating the Topology of 2x2 Games: An Introductory Note on Payoff Families, Normalization, and Natural Order. ArXiv:1010.4727 [Cs]. Retrieved July 16, 2019 from http://arxiv. org/abs/1010.4727

Bruns, B. (2012). *Escaping Prisoner's Dilemmas: From Discord to Harmony in the Landscape of 2x2 Games*. ArXiv:1206.1880 [Cs]. Retrieved April 30, 2019 from http://arxiv.org/abs/1206.1880

Bruns, B. R. (2015). Names for Games: Locating 2x2 Games. *Games*, 6(4), 495–520. https://doi.org/10.3390/g6040495

Bryson, J. J. (2018). Patiency is not a virtue: The design of intelligent systems and systems of ethics. *Ethics and Information Technology*, 20(1), 15–26. https://doi.org/10.1007/s10676-018-9448-6

Bui, H. (2003). *A General Model for Online Probabilistic Plan Recognition*. IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003, 1309–1318.

Busch, M. L., & Reinhardt, E. R. (1993). Nice Strategies in a World of Relative Gains: The Problem of Cooperation under Anarchy. *The Journal of Conflict Resolution*, 37(3), 427–445.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). *Pattern-Oriented Software Architecture Volume 1: A System of Pattern*. Retrieved May 2, 2023 from http://software-pattern.org/Book/30

Bush, R. R., & Mosteller, F. (1951). A mathematical model for simple learning. *Psychological Review*, 58, 313–323. https://doi.org/10.1037/h0054388

Busoniu, L., Babuska, R., & De Schutter, B. (2008). *A Comprehensive Survey of Multiagent Reinforcement Learning*. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38(2), 156–172. https://doi.org/10.1109/TSMCC.2007.913919

Cai, Y., Candogan, O., Daskalakis, C., & Papadimitriou, C. (2016). Zero-Sum Polymatrix Games: A Generalization of Minmax. *Mathematics of Operations Research*, 41(2), 648–655. https://doi.org/10.1287/moor.2015.0745

Cave, S., Coughlan, K., & Dihal, K. (2019a). *'Scary Robots': Examining Public Responses to AI*. Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, 331–337. https://doi.org/10.1145/3306618.3314232

Cave, S., Nyrup, R., Vold, K., & Weller, A. (2019b). *Motivations and Risks of Machine Ethics*. Proceedings of the IEEE, 107(3), 562–574. https://doi.org/10.1109/JPROC.2018.2865996

Cave, S., Ryan, F., & Xu, V. X. (2019c). *Mapping more of China's tech giants: AI and surveillance*. Retrieved May 1, 2023 from http://www.aspi.org.au/ report/mapping-more-chinas-tech-giants

Cave, S., & Dihal, K. (Eds.). (2023). *Imagining AI: How the World Sees Intelligent Machines*. Oxford University Press

Cave, S., Dihal, K., & Dillon, S. (Eds.). (2020). *AI Narratives: A History of Imaginative Thinking about Intelligent Machines*. Oxford University Press.

Chang, H. (2004). *Inventing Temperature*. Oxford University Press. https://doi.org/10.1093/0195171276.003.0001

Charniak, E., & Goldman, R. P. (1993). A Bayesian model of plan recognition. *Artificial Intelligence*, 64(1), 53–79. https://doi.org/10.1016/0004-3702(93)90060-O

Chen C., Clack C. D., & Nagl S. B. (2010) Identifying Multi-Level Emergent Behaviors in Agent-Directed Simulations using Complex Event Type Specifications. *SIMULATION*, 86(1), 41–51. doi:10.1177/0037549709106692

Christian, B. (2020). *The Alignment Problem*. W. W. Norton & Company

Claus, C., & Boutilier, C. (1998). *The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems*. AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence July 1998, pp.746–752.

Collins English Dictionary. (2023) Intent. In *Collins English Dictionary*. Retrieved March 23, 2023, from https://www.collinsdictionary.com/dictionary/english/intent

Colman, A. M. (1995). *Game Theory and its Applications in the Social and Biological Sciences*. Psychology Press. https://doi.org/10.4324/9780203761335

Colman, A. M. (1998). Rationality assumptions of game theory and the backward induction paradox. In Oaksford M. & Chater N. (Eds.), *Rational Models of Cognition* (pp. 353–371). Oxford University Press.

Conlisk, J. (1996). Why Bounded Rationality? *Journal of Economic Literature*, 34(2), pp. 669–700

Cook, J., D. (2018, July 23). *Biased random number generation*. Retrieved June 28, 2023, from https://www.johndcook.com/blog/2018/07/23/biased-random-number-generation/

Copeland, J., Bowen, J., Sprevak, M., & Wilson, R. (2017). *The Turing Guide*. Oxford University Press. https://doi.org/10.1093/oso/9780198747826.001.0001

Corporation for Digital Scholarship. (2018). *Zotero* (6.0.26) [Computer software]. Vienna, Virginia, USA. https://www.zotero.org

Crandall, J. W. (2015). *Robust Learning for Repeated Stochastic Games via Meta-Gaming*. IJCAI15: Proceedings of the 24th International Conference on Artificial Intelligence, July 2015 pp. 3416–3422 https://dl.acm.org/doi/10.5555/2832581.2832725

Crandall, J. W., & Goodrich, M. A. (2005). *Learning to compete, compromise, and cooperate in repeated general-sum games*. Proceedings of the 22nd International Conference on Machine Learning, ICML'05, 161–168. https://doi.org/10.1145/1102351.1102372

Crandall, J. W., & Goodrich, M. A. (2011). Learning to compete, coordinate, and cooperate in repeated games using reinforcement learning. *Machine Learning*, 82(3), 281–314. https://doi.org/10.1007/s10994-010-5192-9

Crandall, J. W., Oudah, M., Tennom, Ishowo-Oloko, F., Abdallah, S., Bonnefon, J., Cebrian, M., Shariff, A., Goodrich, M. A., & Rahwan, I. (2018a). Cooperating with machines. *Nature Communications*, 9(1), 233. https://doi.org/10.1038/s41467-017-02597-8

Crandall, J. W., Oudah, M., Tennom, Ishowo-Oloko, F., Abdallah, S., Bonnefon, J., Cebrian, M., Shariff, A., Goodrich, M. A., & Rahwan, I. (2018b). Supplementary Material—Cooperating with Machines. *Nature Communications*, 9(1). Retrieved January 7, 2020, from https://static-content.springer.com/esm/art%3A10.1038%2Fs41467-017-02597-/MediaObjects/41467_2017_2597_MOESM1_ESM.pdf

Criado, N., Argente, E., & Botti, V. (2011). Open issues for normative multi-agent systems. *AI Communications*, 24(3), 233–264. https://doi.org/10.3233/AIC-2011-0502

Criminal Code Act 1995 (Cth) s 5.2 *Intention*. Retrieved June 1, 2023, from https://www.legislation.gov.au/

Csardi G, Nepusz T (2006). "The igraph software package for complex network research." InterJournal, Complex Systems, 1695. Retrieved July 11, 2023, from https://igraph.org.

Darwin, C. (1859). On the origin of species by means of natural selection, or, the preservation of favoured races in the struggle for life. J. Murray.

Daskalakis, C., Goldberg, P. W., & Papadimitriou, C. H. (2009). The Complexity of Computing a Nash Equilibrium. *SIAM Journal on Computing*, 39(1), 195–259. https://doi.org/10.1137/070699652

Datacenters.com, Technology, (2015). *Data Center Power Costs and Requirements*. Retrieved July 14, 2023, from https://www.datacenters.com/news/data-center-power-costs-and-requirements

Davis, E., & Marcus, G. (2021). *Insights for AI from the Human Mind*. Retrieved January 8, 2021, from https://cacm.acm.org/magazines/2021/1/249452-insights-for-ai-from-the-human-mind/fulltext

Dawkins, R. (1976). *The Selfish Gene* (New Ed. 1989). Oxford University Press.

Dell. (2021). *Latitude 5490 Owner's Manual*. Dell Technologies. Retrieved July 17, 2023, from https://dl.dell.com/topicspdf/latitude-14-5490-laptop_owners-manual_en-us.pdf

Di Stefano, A., Jayne, C., Angione, C., & Han, T. A. (2023, July 24). Recognition of Behavioural Intention in Repeated Games using Machine Learning. *ALIFE 2023: Ghost in the Machine: Proceedings of the 2023 Artificial Life Conference.* https://doi.org/10.1162/isal_a_00637

Diaconescu, A., & Pitt, J. (2017). Technological Impacts in Socio-Technical Communities: Values and Pathologies. *IEEE Technology and Society Magazine*, 36(3), 63–71. https://doi.org/10.1109/MTS.2017.2728780

Doebeli, M., & Hauert, C. (2005). Models of cooperation based on the Prisoner's Dilemma and the Snowdrift game. *Ecology Letters*, 8(7), 748–766. https://doi.org/10.1111/j.1461-0248.2005.00773.x

Dugatkin, L. A., & Dugatkin, L. A. (1997). *Cooperation Among Animals: An Evolutionary Perspective*. Oxford University Press.

Dugatkin, L. A., & Mesterton-Gibbons, M. (1996). Cooperation among unrelated individuals: Reciprocal altruism, by-product mutualism and group selection in fishes. *Biosystems*, 37(1), 19–30. https://doi.org/10.1016/0303-2647(95)01542-6

Eliot, T. S. (1943) *Four Quartets*. Harcourt.

Erev, I., & Roth, A. E. (1998). Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique, Mixed Strategy Equilibria. *American Economic Review*, 88(4), 848–881.

Eriksson K. & Strimling P. (2012) The Hard Problem of Cooperation. *PLoS ONE* 7(7): e40325. https://doi.org/10.1371/journal.pone.0040325

Fang, C., Kimbrough, S. O. K., & Zheng, Z. (2002). On Adaptive Emergence of Trust Behavior in the Game of Stag Hunt. *Group Decision and Negotiation* 11, 449–467 (2002). https://doi.org/10.1023/A:1020639132471

Fang, F. (2016). *Towards Addressing Spatio-Temporal Aspects in Security Games*. University of Southern California, Doctoral Thesis. Retrieved February 15, 2022 from https://teamcore.seas.harvard.edu/publications/towards-addressing-spatio-temporal-aspects-security-games-0

Fang, F. (2021). *Game Theoretic Models for Cyber Deception*. Proceedings of the 8th ACM Workshop on Moving Target Defense, 23–24. https://doi.org/10.1145/3474370.3485656

Fang, F., Ford, B., Yang, R., Tambe, M., & Lemieux, A. M. (2017). PAWS: Game Theory Based Protection Assistant for Wildlife Security. In *Conservation Criminology* (pp. 179–195). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119376866.ch10

Fang, F., Jiang, A. X., & Tambe, M. (2013). Protecting Moving Targets with Multiple Mobile Resources. *Journal of Artificial Intelligence Research*, 48, 583–634. https://doi.org/10.1613/jair.4027

Farrell, J., & Rabin M. (1996). Cheap talk. *Journal of Economic Perspectives*, 10(3), 103–118.

Fisac, J. F., Bronstein, E., Stefansson, E., Sadigh, D., Sastry, S. S., & Dragan, A. D. (2018). *Hierarchical Game-Theoretic Planning for Autonomous Vehicles*. ArXiv:1810.05766. Retrieved May 29, 2019 from http://arxiv.org/abs/1810.05766

Fisher, R. A. (1930). *The genetical theory of natural selection*. Oxford, Clarendon Press. https://doi.org/10.5962/bhl.title.27468

Fisher, R. A. (1958). *The genetical theory of natural selection*, (2d rev. ed.). Dover Publications.

Flack, J. C. (2012). Multiple time-scales and the developmental dynamics of social systems. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1597), 1802–1810. https://doi.org/10.1098/rstb.2011.0214

Fraser, N. M., & Kilgour, D. M. (1986). Non-strict ordinal 2 × 2 games: A comprehensive computer-assisted analysis of the 726 possibilities. *Theory and Decision*, 20(2), 99–121. https://doi.org/10.1007/BF00135087

Frey, S., & Sumner, R. W. (2019). Emergence of integrated institutions in a large population of self-governing communities. *PLoS ONE*, 14(7), e0216335. https://doi.org/10.1371/journal.pone.0216335

Frohock, F. M. (1987). *Rational Association*. Syracuse University Press.

Fudenberg, D., & Levine, D. K. (1998). *The Theory of Learning in Games*. MIT Press.

Galliott, J., & Wyatt, A. (2020). Risks and Benefits of Autonomous Weapon Systems: Perceptions among Future Australian Defence Force Officers. *Journal of Indo-Pacific Affairs*, Winter, 17–34.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc.

Geanakoplos, J., Pearce, D., & Stacchetti, E. (1989). Psychological games and sequential rationality. *Games and Economic Behavior*, 1(1), 60–79. https://doi.org/10.1016/0899-8256(89)90005-5

Geib, C. W., & Goldman, R. P. (2009). A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, 173(11), 1101–1132. https://doi.org/10.1016/j.artint.2009.01.003

Gonzalez, W. (2022, April 19). *Three Ways AI Is Impacting The Automobile Industry*. *Forbes*. Retrieved April 26, 2023 from https://www.forbes.com/sites/forbesbusinesscouncil/2022/04/19/three-ways-ai-is-impacting-the-automobile-industry/

Grim, P. (1996). Spatialization and greater generosity in the stochastic Prisoner's Dilemma. *Biosystems*, 37(1), 3–17. https://doi.org/10.1016/0303-2647(95)01541-8

Güth, W., Schmittberger, R., & Schwarze, B. (1982). An experimental analysis of ultimatum bargaining. *Journal of Economic Behavior & Organization*, 3(4), 367–388. https://doi.org/10.1016/0167-2681(82)90011-7

Guyer, M. J., & Rapoport, A. (1972). 2×2 Games Played Once. *The Journal of Conflict Resolution*, 16(3), 409–431.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*. ArXiv:1801.01290. Retrieved October 27, 2021 from http://arxiv.org/abs/1801.01290

Hamburger, H. (1973). N-person Prisoner's Dilemma. *The Journal of Mathematical Sociology*, 3(1), 27–48. https://doi.org/10.1080/0022250X.1973.9989822

Hamilton, W. D. (1963). The Evolution of Altruistic Behavior. *The American Naturalist*, 97(896), 354–356.

Hamilton, W. D. (1964a). The genetical evolution of social behaviour. I. *Journal of Theoretical Biology*, 7(1), 1–16. https://doi.org/10.1016/0022-5193(64)90038-4

Hamilton, W. D. (1964b). The genetical evolution of social behaviour. II. *Journal of Theoretical Biology*, 7(1), 17–52. https://doi.org/10.1016/0022-5193(64)90039-6

Han, T. A., Pereira, L. M., & Santos, F. C. (2011a). Intention recognition promotes the emergence of cooperation. Adaptive Behavior, 19(4), 264–279. https://doi.org/10.1177/1059712311410896

Han, T. A., Pereira, L. M., & Santos, F. C. (2011b). The role of intention recognition in the evolution of cooperative behavior. IJCAI 2011 - 22nd International Joint Conference on Artificial Intelligence, 1684–1689. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-283

Han, T. A., Pereira, L. M., & Santos, F. C. (2012). Corpus-Based Intention Recognition in Cooperation Dilemmas. *Artificial Life*, 18(4), 365–383. https://doi.org/10.1162/ARTL_a_00072

Han, T. A. (2013). Intention Recognition, Commitment and the Evolution of Cooperation. In T. A. Han (Ed.), Intention Recognition, Commitment and Their Roles in the Evolution of Cooperation: From Artificial Intelligence Techniques to Evolutionary Game Theory Models (pp. 123–132). Springer. https://doi.org/10.1007/978-3-642-37512-5_8

Han, T. A., & Pereira, L. M. (2013). State-of-the-art of intention recognition and its use in decision making. AI Communications, 26(2), 237–246. https://doi.org/10.3233/AIC-130559

Harsanyi, J. C., & Selten, R. (1972). A Generalized Nash Solution for Two-Person Bargaining Games with Incomplete Information. *Management Science*, 18(5), pp. 80-106.

Hawking, S., Tegmark, M., & Russell, S. (2014, April 19). *Transcending Complacency On Superintelligent Machines*. Huffington Post. Retrieved May 20, 2023 from https://www.huffpost.com/entry/ artificial-intelligence_b_5174265

Hazra, T., & Anjaria, K. (2022). Applications of game theory in deep learning: A survey. *Multimedia Tools and Applications*, 81(6), 8963–8994. https://doi.org/10.1007/ s11042-022-12153-2

Heinze, C. (2003). *Modelling intention recognition for intelligent agent systems.* Defence Science and Technology Organisation (Australia) Systems Sciences Laboratory. Doctoral Thesis

Henrich, J., Boyd, R., Bowles, S., Camerer, C., Fehr, E., Gintis, H., & McElreath, R. (2001). In Search of Homo Economicus: Behavioral Experiments in 15 Small-Scale Societies. *American Economic Review*, 91(2), 73–78. https://doi.org/10.1257/aer.91.2.73

Herbers, J. M. (2009). Darwin's 'one special difficulty': Celebrating Darwin 200. *Biology Letters*, 5(2), 214–217. https://doi.org/10.1098/rsbl.2009.0014

Herndon, H. (2021, July 14). *Holly+* 👫🗣️🦾. Retrieved April 26, 2022 from https://holly.mirror.xyz/54ds2Ii OnvthjGFkokFCoaI4EabytH9xjAYy1irHy94

Herzing, D. L. (2014). Profiling nonhuman intelligence: An exercise in developing unbiased tools for describing other "types" of intelligence on earth. *Acta Astronautica*, 94(2), 676–680. https://doi.org/10.1016/j.actaastro.2013.08.007

Ho, D. (2003). *Notepad++* [Computer software]. https://notepad-plus-plus.org

Hofbauer, J. & Sandholm, W. H. (2002), On the Global Convergence of Stochastic Fictitious Play. *Econometrica*, 70: 2265-2294. https://doi.org/10.1111/j.1468-0262.2002.00440.x

Hofstadter, D. (1983). The Prisoner's Dilemma, Computer Tournaments and the Evolution of Cooperation. *Scientific American*. https://doi.org/10.1038/scientificamerican0583-16

Hooker, S. (2021). Moving beyond "algorithmic bias is a data problem". *Patterns*, 2(4). https://doi.org/10.1016/j.patter.2021.100241

Hooker, S., Moorosi, N., Clark, G., Bengio, S., & Denton, E. (2020). *Characterising Bias in Compressed Models*. Retrieved June 28, 2021 from https://arxiv.org/abs/2010.03058v2

Huynh, T. D., Jennings, N. R., & Shadbolt, N. R. (2006). An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2), 119–154. https://doi.org/10.1007/s10458-005-6825-4

Janovskaja, E. (1968). Equilibrium points in polymatrix games. *Lithuanian Mathematical Journal*, 8(2), 381–384. https://doi.org/10.15388/LMJ.1968.20224

Jervis, R. (1978). Cooperation under the Security Dilemma. *World Politics*, 30(2), 167–214. https://doi.org/10.2307/2009958

JGraph. (2021). *Diagrams.net, draw.io* (15.5.2) [Computer software]. https://www.diagrams.net/

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Dai, W., Madotto, A., & Fung, P. (2023). Survey of Hallucination in Natural Language Generation. ACM Computing Surveys, 55(12), 1–38. https://doi.org/10.1145/3571730

Katsuno, H., & White, D. (2023). Engineering Robots with Heart in Japan: The Politics of Cultural Difference in Artificial Emotional Intelligence. In Cave S., & Dihal K. (Eds.), *Imagining AI: How the World Sees Intelligent Machines*. (pp. 295–317). Oxford University Press.

Khetarpal, K., Riemer, M., Rish, I., & Precup, D. (2022). *Towards Continual Reinforcement Learning: A Review and Perspectives* (arXiv:2012.13490). Retrieved March 30, 2023 from https://doi.org/10.48550/arXiv.2012.13490

Kilgour, D. M., & Fraser, N. M. (1988). A taxonomy of all ordinal 2 × 2 games. *Theory and Decision*, 24(2), 99–117. https://doi.org/10.1007/BF00132457

Kim, S. (2023). Development and Developmentalism of Artificial Intelligence: Decoding South Korean Policy Discourse on Artificial Intelligence. In Cave S. & Dihal K. (Eds.), *Imagining AI: How the World Sees Intelligent Machines*. (pp. 318–338). Oxford University Press.

Knight, V., & Campbell, J. (2017). *NashPy: A library with algorithms on 2 player games*. (0.0.35). Retrieved January 6, 2023 from https://nashpy.readthedocs.io/en/stable/

Knight, V., Campbell, O., Harper, M., Langner, K., Campbell, J., Campbell, T., Carney, A., Chorley, M., Davidson-Pilon, C., Glass, K., Glynatsi, N., Ehrlich, T., Jones, M., Koutsovoulos, G., Tibble, H., Müller, J., Palmer, G., Petunov, P., Slavin, P., … Molden, K. (2016). An Open Framework for the Reproducible Study of the Iterated Prisoner's Dilemma. *Journal of Open Research Software*, 4(1), Article 1. https://doi.org/10.5334/jors.125

Kollock, P. (1998). Social Dilemmas: The Anatomy of Cooperation. *Annual Review of Sociology*, 24, 183–214.

Konior, B. (2023). Automatic Gnosis: On Lem's Summa Technologiae. In Cave S. & Dihal K. (Eds.), *Imagining AI: How the World Sees Intelligent Machines*. (pp. 89–108). Oxford University Press.

Kreps, D. M., Milgrom, P., Roberts, J., & Wilson, R. (1982). Rational cooperation in the finitely repeated prisoners' dilemma. *Journal of Economic Theory*, 27(2), 245–252. https://doi.org/10.1016/0022-0531(82)90029-1

Kubí, A. (2003). Toward a Formalization of Emergence. *Artificial Life*, 9(1), 41–65. https://doi.org/10.1162/106454603321489518

Kubrick, S. (Director). (1968). *2001: A Space Odyssey* [Film]. Metro-Goldwyn-Mayer.

Kubrick, S., & Clarke, A. C. (1968). *2001: A Space Odyssey Transcript*. Retrieved December 31, 2022 from http://www.archiviokubrick.it/opere/film/2001/script/2001-originalscript.pdf

Kuhn, H. W., Harsanyi, J. C., Selten, R., Weibull, J. W., van Damme, E., Nash, J. F., & Hammerstein, P. (1996). The Work of John Nash in Game Theory. *Journal of Economic Theory*, 69(1), 153–185.

Kümmerli, R., Colliard, C., Fiechter, N., Petitpierre, B., Russier, F., & Keller, L. (2007). Human cooperation in social dilemmas: Comparing the Snowdrift game with the Prisoner's Dilemma. *Proceedings of the Royal Society B*, 274, 2965–2970. http://dx.doi.org/10.1098/rspb.2007.0793

Kutskir, I. (2013). *Photopea* [Computer software]. https://www.photopea.com/

Laland, K. N., & Brown, G. R. (2011). *Sense and nonsense: Evolutionary perspectives on human behaviour*. Oxford University Press.

Lanctot, M., Zambaldi, V., & Lazaridou, A. (2017). *A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning*. 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, US, 14.

Latour, B. (1999). Pandora's hope: Essays on the reality of science studies. Harvard University Press.

Laurent, G. J., Matignon, L., & Fort-Piat, N. L. (2011). The world of independent learners is not markovian. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 15(1), 55–64.

Lee, C. J., & Matthews, G. (2023). Intelligent Infrastructure, Humans as Resources, and Coevolutionary Futures: AI Narratives in Singapore. In Cave S. & Dihal K. (Eds.), *Imagining AI: How the World Sees Intelligent Machines*. (pp. 382–402). Oxford University Press.

Le Guin, U. K. (2018). Dragonfly; in Tales of Earthsea in The Books of Earthsea; The Complete Illustrated Edition, First Saga Edition. New York, Saga Press.

Leibo, J. Z., Zambaldi, V., Lanctot, M., Marecki, J., & Graepel, T. (2017). *Multi-agent Reinforcement Learning in Sequential Social Dilemmas*. Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems, 464–473.

Lemire, D. (2019). Fast Random Integer Generation in an Interval. *ACM Transactions on Modeling and Computer Simulation*, 29(1), 3:1-3:12. https://doi.org/10.1145/3230636

Lewis, H. M., & Dumbrell, A. J. (2013). Evolutionary games of cooperation: Insights through integration of theory and data. *Ecological Complexity*, 16, 20–30. https://doi.org/10.1016/j.ecocom.2013.02.007

Lewis, J. E. (2023). Imagining Indigenous AI. In Cave S., & Dihal K. (Eds.), *Imagining AI: How the World Sees Intelligent Machines*. (pp. 210-217). Oxford University Press.

Lewis, P. R., Chandra, A., Faniyi, F., Glette, K., Chen, T., Bahsoon, R., Torresen, J., & Yao, X. (2015). Architectural Aspects of Self-Aware and Self-Expressive Computing Systems: From Psychology to Engineering. *Computer*, 48(8), 62–70. https://doi.org/10.1109/MC.2015.235

Littman, M. L. (1994). *Markov games as a framework for multi-agent reinforcement learning*. In Machine Learning Proceedings 1994 (pp. 157–163). Elsevier. https://doi.org/10.1016/B978-1-55860-335-6.50027-1

Littman, M. L., Ajunwa, I., Berger, G., Boutilier, C., Currie, M., Doshi-Velez, F., Hadfield, G., Horowitz, M. C., Isbell, C., Kitano, H., Levy, K., Lyons, T., Mitchell, M., Shah, J., Sloman, S., Vallor, S., & Walsh, T. (2021). *Gathering Strength, Gathering Storms: The One Hundred Year Study on Artificial Intelligence (AI100) 2021 Study Panel Report*. Stanford University. Retrieved September 29, 2021 from https://ai100.stanford.edu/

Liu, N. F., Zhang, T., & Liang, P. (2023). *Evaluating Verifiability in Generative Search Engines* (arXiv:2304.09848). arXiv. Retrieved April 24, 2023 from https://doi.org/10.48550/arXiv.2304.09848

Luo, J., Paduraru, C., Voicu, O., Chervonyi, Y., Munns, S., Li, J., Qian, C., Dutta, P., Davis, J. Q., Wu, N., Yang, X., Chang, C., Li, T., Rose, R., Fan, M., Nakhost, H., Liu, T., Kirkman, B., Altamura, F., … Mankowitz, D. J. (2022). *Controlling Commercial Cooling Systems Using Reinforcement Learning* (arXiv:2211.07357). arXiv. Retrieved November 26, 2022 from https://doi. org/10.48550/arXiv.2211.07357

Macy, M. W., & Flache, A. (2002). *Learning dynamics in social dilemmas*. Proceedings of the National Academy of Sciences, 99(suppl 3), 7229–7236. https://doi.org/10.1073/pnas.092080099

Marcus, G. (2022a). Deep Learning Alone Isn't Getting Us to Human-Like AI. *Noema*. Retrieved August 12, 2022 from https://www.noemamag.com/deep-learning-alone-isnt-getting-us-to-human-like-ai

Marcus, G. (2022b). Deep Learning Is Hitting a Wall. *Nautilus*. Retrieved September 27, 2022 from https://nautil.us/deep-learning-is-hitting-a-wall-238440/

Marcus, G. (2022c). Sentience and AI: A Dialog between Gary Marcus and Blake Lemoine [Substack newsletter]. *The Road to AI We Can Trust*. Retrieved November 25, 2022 from https://garymarcus. substack.com/p/sentience-and-ai-a-dialog-between

Marwala, T. (2021). *Rational Machines and Artificial Intelligence*. Elsevier. https://doi.org/10.1016/C2019-0-02529-0

Maslej, N., Fattorini, L., Brynjolfsson, E., Etchemendy, J., Ligett, K., Lyons, T., Manyika, J., Ngo, H., Niebles, Parli, V., Shoham, Y., Wald, R., Clark, & Perrault, R. (2023). The AI Index 2023 Annual Report. Institute for Human-Centered AI, Stanford University, Stanford, CA. Retrieved June 26, 2023 from https://aiindex.stanford. edu/report/

Masuda, N., & Ohtsuki, H. (2009). A Theoretical Analysis of Temporal Difference Learning in the Iterated Prisoner's Dilemma Game. *Bulletin of Mathematical Biology*, 71(8), 1818–1850. https://doi.org/10.1007/s11538-009-9424-8

Maynard Smith, J. (1964). Group Selection and Kin Selection. *Nature*, 201(4924), 1145–1147. https://doi.org/10.1038/2011145a0

Maynard Smith, J. (1982). *Evolution and the Theory of Games*. Cambridge University Press.

McKelvey, R. D., McLennan, A. M., & Turocy, T. L. (2016). *Gambit: Software Tools for Game Theory*. Retrieved January 1, 2023 from https://www.gambit-project.org/

McNamara, J. M. (2022). Game Theory in Biology: Moving beyond Functional Accounts. *The American Naturalist*, 199(2), 179–193. https://doi.org/10.1086/717429

Merriam-Webster. (2023) Cooperation. In *Merriam-Webster*. Retrieved May 3, 2023 from https://www.merriam-webster.com/dictionary/cooperation

Minsky, M. (1961). *Steps toward Artificial Intelligence*. Proceedings of the IRE, 49(1), 8–30. https://doi.org/10.1109/JRPROC.1961.287775

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). *Playing Atari with Deep Reinforcement Learning* (arXiv:1312.5602). arXiv. Retrieved December 14, 2021 from https://doi.org/ 10.48550/arXiv.1312.5602

Moor, J. H. (2011). The Nature, Importance, and Difficulty of Machine Ethics. In M. Anderson & S. L. Anderson (Eds.), *Machine Ethics* (1st ed., pp. 13–20). Cambridge University Press. https://doi.org/10.1017/CBO9780511978036.003

Mossalam, H., Assael, Y. M., Roijers, D. M., & Whiteson, S. (2016). *Multi-Objective Deep Reinforcement Learning*. Retrieved January 15, 2019 from https://arxiv.org/ abs/1610.02707v1

Narayanan, A., & Kapoor, S. (2023). *Generative AI companies must publish transparency reports*. Retrieved June 27, 2023 from http://knightcolumbia.org/blog/generative-ai-companies-must-publish-transparency-reports

Nash, J. (1951). Non-Cooperative Games. *Annals of Mathematics*, 54, 286–295.

Nash, J. (1953). Two-Person Cooperative Games. *Econometrica*, 21(1), 128–140. https://doi.org/10.2307/1906951

Nash, J. F. (1950a). *Equilibrium Points in n-Person Games*. Proceedings of the National Academy of Sciences of the United States of America, 36(1), 48–49.

Nash, J. F. (1950b). The Bargaining Problem. *Econometrica*, 18(2), 155–162. https://doi.org/10.2307/1907266

Nguyen, T.D., & Zick, Y. (2018). Resource Based Cooperative Games: Optimization, Fairness and Stability. *Algorithmic Game Theory*.

Nowak, Krzysztof, Nielsen, Lars Holm, & Ioannidis Pantopikos, Alexandros Themistoklis. (2016, May 24). Zenodo, a free and open platform for preserving and sharing research output. Zenodo. https://doi.org/10.5281/zenodo.51902

Nowak, M. A. (2006). Five Rules for the Evolution of Cooperation. *Science*, 314(5805), 1560–1563. https://doi.org/10.1126/science.1133755

Nowak, M. A., & May, R. M. (1992). Evolutionary games and spatial chaos. *Nature*, 359(6398), 826–829. https://doi.org/10.1038/359826a0

Nowak, M. A., & Sigmund, K. (1998). Evolution of indirect reciprocity by image scoring. *Nature*, 393(6685), 573–577. https://doi.org/10.1038/31225

Nowak, M. A., & Sigmund, K. (2005). Evolution of indirect reciprocity. *Nature*, 437(7063), 1291–1298. https://doi.org/10.1038/nature04131

Nowak, M. A., Tarnita, C. E., & Wilson, E. O. (2010). The evolution of eusociality. *Nature*, 466(7310), 1057–1062. https://doi.org/10.1038/nature09205

Nowak, M., & Sigmund, K. (1993). A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. *Nature*, 364(6432), 56–58. https://doi.org/10.1038/364056a0

Odling-Smee, F. J., Laland, K. N., & Feldman, M. W. (2003). *Niche Construction: The Neglected Process in Evolution*. Princeton University Press.

Omidshafiei, S., Tuyls, K., Czarnecki, W. M., Santos, F. C., Rowland, M., Connor, J., Hennes, D., Muller, P., Pérolat, J., Vylder, B. D., Gruslys, A., & Munos, R. (2020). Navigating the landscape of multiplayer games. *Nature Communications*, 11(1), Article 1. https://doi.org/10.1038/s41467-020-19244-4

Omohundro, S. M. (2007). The Nature of Self-Improving Artificial Intelligence. *Self-Aware Systems*, 48.

Omohundro, S. M. (2008). *The Basic AI Drives*. Proceedings of the 2008 Conference on Artificial General Intelligence 2008: Proceedings of the First AGI Conference, 483–492.

OpenAI. (2022). *Introducing ChatGPT*. Retrieved April 26, 2023 from https://openai.com/blog/chatgpt

Orbell, J. M., & Dawes, R. M. (1993). Social Welfare, Cooperators' Advantage, and the Option of Not Playing the Game. *American Sociological Review*, 58(6), 787. https://doi.org/10.2307/2095951

Ostrom, E. (1990). Governing the Commons: The Evolution of Institutions for Collective Action. Cambridge University Press.

Ostrom, E. (2015). *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press. https://doi.org/10.1017/CBO9781316423936

Oudeyer, P., & Kaplan, K. (2007). What is intrinsic motivation? A typology of computational approaches. *Frontiers in Neurorobotics*. https://doi.org/10.3389/neuro.12.006.2007

Owen, G. (1982). *Game theory*. New York: Academic Press.

Oxford University Press. (2023). Cooperation. In *OED Online*, Oxford University Press. Retrieved October 27, 2019 from https://www.oed.com/view/Entry/41037.

Oxford University Press. (2023) Intent. In *OED Online*, Oxford University Press. Retrieved May 3, 2023 from https://www.oed.com/view/Entry/97483

Papadimitriou, C. H., & Tsitsiklis, J. N. (1987). The Complexity of Markov Decision Processes. *Mathematics of Operations Research*, 12(3), 441–450.

Paternotte, C., & Grose, J. (2013). Social Norms and Game Theory: Harmony or Discord? *The British Journal for the Philosophy of Science*, 64(3), 551–587.

Pathak, D., Agrawal, P., Efros, A. A., & Darrell, T. (2017). *Curiosity-Driven Exploration by Self-Supervised Prediction*. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 488–489. https://doi.org/10.1109/CVPRW.2017.70

Patton, D. U., Brunton, D., Dixon, A., Miller, R. J., Leonard, P., & Hackman, R. (2017). Stop and Frisk Online: Theorizing Everyday Racism in Digital Policing in the Use of Social Media for Identification of Criminal Conduct and Associations. *Social Media + Society*, 3(3), 2056305117733344. https://doi.org/10.1177/2056305117733344

Petruzzi, P. E., Pitt, J., & Busquets, D. (2017). Electronic Social Capital for Self-Organising Multi-Agent Systems. *ACM Transactions on Autonomous and Adaptive Systems*, 12(3), 13:1-13:25. https://doi.org/10.1145/3124642

Perlo-Freeman, S. (2006). *The Topology of Conflict and Co-operation* (No. 0609; Working Papers). Department of Accounting, Economics and Finance, Bristol Business School, University of the West of England, Bristol. https://ideas.repec.org/p/uwe/wpaper/0609.html

Pervushin, A. (2023). Boys from a Suitcase: AI Concepts in Science Fiction of the USSR Science Fiction: The Evil Robot and the Funny Robot. In Cave S., & Dihal K. (Eds.), *Imagining AI: How the World Sees Intelligent Machines*. (pp. 109-125). Oxford University Press.

Pitt, J. (2003). *Constitutive Rules for Agent Communication Languages*. IJCAI-03 Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI), pp. 691–698.

Pitt, J. (2017). Interactional Justice and Self-Governance of Open Self-Organising Systems. *2017 IEEE 11th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, 31–40. https://doi.org/10.1109/SASO.2017.12

Pitt, J. (2021). *Self-Organising Multi-Agent Systems: Algorithmic Foundations of Cyber-Anarcho-Socialism*. World Scientific (Europe). https://doi.org/10.1142/q0307

Pitt, J., Ramirez-Cano, D., Draief, M., & Artikis, A. (2011). Interleaving multi-agent systems and social networks for organized adaptation. *Computational and Mathematical Organization Theory*, 17(4), 344–378. https://doi.org/10.1007/s10588-011-9089-3

Pitt, J., Busquets, D., & Riveret, R. (2013). Procedural Justice and 'Fitness for Purpose' of Self-organising Electronic Institutions. In G. Boella, E. Elkind, B. T. R. Savarimuthu, F. Dignum, & M. K. Purvis (Eds.), *PRIMA 2013: Principles and Practice of Multi-Agent Systems* (pp. 260–275). Springer. https://doi.org/10.1007/978-3-642-44927-7_18

Pitt, J., Dryzek, J., & Ober, J. (2020). Algorithmic Reflexive Governance for Socio-Techno-Ecological Systems. *IEEE Technology and Society Magazine*, 39(2), pp. 52–59. https://doi.org/10.1109/MTS.2020.2991500

Plotly (2015). Collaborative data science [Computer software]. Plotly Technologies Inc. Retrieved September 12, 2019 from https://plot.ly

Poundstone, W. (1993). *Prisoner's dilemma*. Anchor Books.

Powers, Simon T. (2010) *Social niche construction: evolutionary explanations for cooperative group formation*. University of Southampton, School of Electronics and Computer Science, Doctoral Thesis.

Press, W. H., & Dyson, F. J. (2012). *Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent.* Proceedings of the National Academy of Sciences of the United States of America, 109(26), 10409–10413.

Proudfoot, D. (2011). Anthropomorphism and AI: Turing's much misunderstood imitation game. *Artificial Intelligence*, 175(5), 950–957. https://doi.org/10.1016/j.artint.2011.01.006

Qiao, H., Rozenblit, J., Szidarovszky, F., & Yang, L. (2006). *Multi-Agent Learning Model with Bargaining*. Proceedings of the 2006 Winter Simulation Conference, 934–940. https://doi.org/10.1109/WSC.2006.323178

Rabin, M. (1993). Incorporating Fairness into Game Theory and Economics. *The American Economic Review*, 83(5), 1281–1302.

Rahwan, I., Cebrian, M., Obradovich, N., Bongard, J., Bonnefon, J.-F., Breazeal, C., Crandall, J. W., Christakis, N. A., Couzin, I. D., Jackson, M. O., Jennings, N. R., Kamar, E., Kloumann, I. M., Larochelle, H., Lazer, D., McElreath, R., Mislove, A., Parkes, D. C., Pentland, A. 'Sandy', … Wellman, M. (2019). Machine behaviour. *Nature*, 568(7753), 477. https://doi.org/10.1038/s41586-019-1138-y

Raihani, N. (2021). The Social Instinct: How Cooperation Shaped the World. St. Martin's Publishing Group.

Rapoport, A. (1974). Prisoner's Dilemma—Recollections and Observations. In A. Rapoport (Ed.), *Game Theory as a Theory of a Conflict Resolution* (pp. 17–34). Springer Netherlands. https://doi.org/10.1007/978-94-010-2161-6_2

Rapoport, A., & Chammah, A. M. (1965). *Prisoner's dilemma; a study in conflict and cooperation*. University of Michigan Press.

Rapoport, A., & Guyer, M. (1966). *A Taxonomy of 2 X 2 Games*. Bobbs-Merrill.

Rapoport, A., Guyer, M., & Gordon, D. G. (1976). *The 2 X 2 game*. University of Michigan Press.

Rapoport, A., Seale, D. A., & Colman, A. M. (2015). Is Tit-for-Tat the Answer? On the Conclusions Drawn from Axelrod's Tournaments. *PLoS ONE*, 10(7), e0134128. https://doi.org/10.1371/journal.pone.0134128

Ratitch, B., & Precup, D. (2002). Characterizing Markov Decision Processes. In T. Elomaa, H. Mannila, & H. Toivonen (Eds.), *Machine Learning: ECML 2002* (Vol. 2430, pp. 391–404). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-36755-1_33

Ratnieks, F. L. W., Foster, K. R., & Wenseleers, T. (2011). Darwin's special difficulty: The evolution of "neuter insects" and current theory. *Behavioral Ecology and Sociobiology*, 65(3), 481–492. https://doi.org/10.1007/s00265-010-1124-8

R Core Team (2022). *R: A language and environment for statistical computing*. v4.2.2. [Software] R Foundation for Statistical Computing. Vienna, Austria. https://www.r-project.org/

Revell, T. (2019). *Caffeine* [Computer software]. https://www.zhornsoftware.co.uk

Rey, D., & Neuhäuser, M. (2011). Wilcoxon-Signed-Rank Test. In Lovric, M. (Eds.) *International Encyclopedia of Statistical Science*. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04898-2_616

Roberts, G. (2008). *Evolution of direct and indirect reciprocity*. Proceedings of the Royal Society B: Biological Sciences, 275(1631), 173–179. https://doi.org/10.1098/rspb.2007.1134

Robertson, G., & Watson, I. (2014). A review of real-time strategy game AI. *AI Magazine*, 35(4), 75–104. Retrieved November 15, 2018 from https://doi.org/10.1609/aimag.v35i4.2478

Robinson, D., & Goforth, D. (2005). *The Topology of the 2x2 Games: A New Periodic Table*. Routledge. https://doi.org/10.4324/9780203340271

Robinson, D., Goforth, D., & Cargill, M. (2007). *Toward a Topological Treatment of the Non-strictly Ordered 2 × 2 Games*. https://core.ac.uk/display/102367155

Robinson, J. (1951). An Iterative Method of Solving a Game. *Annals of Mathematics*, 54(2), 296–301. https://doi.org/10.2307/1969530

Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., & Bronstein, M. (2020). *Temporal Graph Networks for Deep Learning on Dynamic Graphs*. Retrieved on June 28, 2022 from https://doi.org/10.48550/arXiv.2006.10637

Rothstein, B. (2000). Trust, Social Dilemmas and Collective Memories. *Journal of Theoretical Politics*, 12, 477–501. https://doi.org/10.1177/0951692800012004007

Rubinstein, A. (1991). Comments on the Interpretation of Game Theory. *Econometrica*, 59(4), 909–924. https://doi.org/10.2307/2938166

Sadri, F. (2011). Logic-Based Approaches to Intention Recognition. *Handbook of Research on Ambient Intelligence and Smart Environments: Trends and Perspectives*. https://doi.org/10.4018/978-1-61692-857-5.ch018

Sally, D. (1995). Conversation and Cooperation in Social Dilemmas: A Meta-Analysis of Experiments from 1958 to 1992. *Rationality and Society*, 7(1), 58–92.

Sandholm, T. W., & Crites, R. H. (1996). Multiagent reinforcement learning in the Iterated Prisoner's Dilemma. *Biosystems*, 37(1), 147–166. https://doi.org/10.1016/ 0303-2647(95)01551-5

Schelling, T. C. (1981). *The Strategy of Conflict*. Harvard University Press.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal Policy Optimization Algorithms*. ArXiv:1707.06347 Retrieved June 27, 2020 from http://arxiv.org/abs/1707.06347

Scott, M., & Pitt, J. (2023). Interdependent Self-Organizing Mechanisms for Cooperative Survival. *Artificial Life*, 29(2), 198–234. https://doi.org/10.1162/artl_a_00403

Searle, J. R. (1980). Minds, brains, and programs. *The Behavioral and Brain Sciences*, 3, 417–457.

Selten, R. (1978). The chain store paradox. *Theory and Decision*, 9(2), 127–159. https://doi.org/10.1007/BF00131770

Shanahan, M. (2018). Artificial intelligence. In Colombo, M. S., Matteo (Ed.). (2018). *The Routledge Handbook of the Computational Mind*. Routledge. https://doi.org/10.4324/9781315643670

Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3–4), 591–611. https://doi.org/10.1093/biomet/52.3-4.591

Shapley, L. S. (1952). *A Value for N-Person Games*. Rand Corporation. https://doi.org/10.7249/P0295

Shapley, L. S. (1953). *Stochastic Games*. Proceedings of the National Academy of Sciences of the United States of America, 39(10), 1095–1100.

Shi, Z. R., Wang, C., & Fang, F. (2020). *Artificial Intelligence for Social Good: A Survey*. https://doi.org/10.1609/aimag.v35i4.2478 Retrieved February 4, 2022 from https://arxiv.org/abs/2001.01818v1

Shubik, M. (1970). Game theory, behavior, and the paradox of the Prisoner's Dilemma: Three solutions. *Journal of Conflict Resolution*, 14(2), 181–193. https://doi.org/10.1177/002200277001400204

Simon, H. A. (1955). A Behavioral Model of Rational Choice. *The Quarterly Journal of Economics*, 69(1), 99–118. https://doi.org/10.2307/1884852

Sir, R. A. F., & Fisher, R. A. (1999). The Genetical Theory of Natural Selection: A Complete Variorum Edition. OUP Oxford.

Skyrms, B. (2001). *The Stag Hunt*. Proceedings and Addresses of the American Philosophical Association, 75(2), 31–41. https://doi.org/10.2307/3218711

Skyrms, B. (2003). *The Stag Hunt and the Evolution of Social Structure*. Cambridge University Press. https://doi.org/10.1017/CBO9781139165228

Skyrms, B. (2016). Evolution, Norms, and the Social Contract. *Arizona State Law Journal*, 48(4), 1087–1100.

Smith, B. C. (1982). *Procedural reflection in programming languages*. Massachusetts Institute of Technology, Doctoral Thesis.

Soares, N., Fallenstein, B., Armstrong, S., & Yudkowsky, E. (2015, April 1). *Corrigibility*. Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence. https://www.aaai.org/ocs/index.php/WS/AAAIW15/paper/view/10124

Sohrabi, S., Riabov, A. V., & Udrea, O. (2016). *Plan recognition as planning revisited*. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 3258–3264.

Song, B. (2023). How Chinese Philosophy Impacts AI Narratives and Imagined AI Futures. In Cave S., & Dihal K. (Eds.), *Imagining AI: How the World Sees Intelligent Machines*. (pp. 338-352). Oxford University Press.

Spence, A. M. (1974). Market signaling: Informational transfer in hiring and related screening processes. Cambridge, Harvard University Press.

Stanton, S. C., Dermoudy, J., & Ollington, R. (2022). *Representation-Induced Algorithmic Bias*. In G. Long, X. Yu, & S. Wang (Eds.), AI 2021: Advances in Artificial Intelligence (pp. 103–116). Springer International Publishing. https://doi.org/10.1007/978-3-030-97546-3_9

Stewart, A. J., & Plotkin, J. B. (2012). *Extortion and cooperation in the Prisoner's Dilemma*. Proceedings of the National Academy of Sciences, 109(26), 10134–10135. https://doi.org/10.1073/pnas.1208087109

Stone, P., Brynjolfsson, E., Calo, R., Etzioni, O., Hager, G., Hirschberg, J., Kalyanakrishnan, S., Kamar, E., Kraus, S., Leyton-Brown, K., Parkes, D., Press, W., Saxenian, A. L., Shah, J., Tambe, M., & Teller, A. (2016). *"Artificial Intelligence and Life in 2030." One Hundred Year Study on Artificial Intelligence: Report of the 2015-2016 Study Panel, Stanford University*. Stanford, CA. Retrieved January 18, 2019 from https://ai100.stanford.edu/2016-report

Sung, J., Guo, L., Grinter, R., & Christensen, H. (2007). *"My Roomba Is Rambo": Intimate Home Appliances*. In Lect. Note. Comput. Sci. (Vol. 4717). https://doi.org/10.1007/978-3-540-74853-3_9

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. The MIT Press.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction (2nd ed.)*. The MIT Press.

Sutton, R. S., Barto, A. G., & Williams, R. J. (1992). Reinforcement learning is direct adaptive optimal control. *IEEE Control Systems Magazine*, 12(2), 19–22. https://doi.org/10.1109/37.126844

Szepesvári, C. (2010). *Algorithms for Reinforcement Learning*. Morgan and Claypool Publishers.

Tafidis, P., Farah, H., Brijs, T., & Pirdavani, A. (2022). Safety implications of higher levels of automated vehicles: A scoping review. *Transport Reviews*, 42(2), 245–267. https://doi.org/10.1080/01441647.2021.1971794

Teófilo, L. F., Passos, N., Reis, L. P., & Cardoso, H. L. (2012). Adapting Strategies to Opponent Models in Incomplete Information Games: A Reinforcement Learning Approach for Poker. In M. Kamel, F. Karray, & H. Hagras (Eds.), *Autonomous and Intelligent Systems* (Vol. 7326, pp. 220–227). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-31368-4_26

The Commonwealth Criminal Code: A Guide for Practitioners. (2002). *Attorney-General's Department*. Retrieved June 1, 2023 from https://www.ag.gov.au/crime/publications/commonwealth-criminal-code-guide-practitioners

The Free Dictionary. (2023) *Intent*. In The Free Dictionary, Farlex. Retrieved May 3, 2023 from https://www.thefreedictionary.com/intent

Thill, S., Padó, S., & Ziemke, T. (2014). On the importance of a rich embodiment in the grounding of concepts: Perspectives from embodied cognitive science and computational linguistics. *Topics in Cognitive Science*, 6(3), 545–558. https://doi.org/10.1111/tops.12093

Tinbergen, N. (1963). On aims and methods of Ethology. *Ethology*, 20(4), 410–433. https://doi.org/10.1111/j.1439-0310.1963.tb01161.x

Totschnig, W. (2019). The problem of superintelligence: Political, not technological. *AI & Society*, 34(4), 907–920. https://doi.org/10.1007/s00146-017-0753-0

Tresch, J. (2021). The Reason for the Darkness of the Night: Edgar Allan Poe and the Forging of American Science. Farrar, Straus, and Giroux.

Trivers, R. L. (1971). The Evolution of Reciprocal Altruism. *The Quarterly Review of Biology*, 46(1), 35–57.

Trivers, R. L. (1974). Parent-Offspring Conflict. *American Zoologist*, 14(1), 249–264.

Tucker, A. (1950). *A Two-Person Dilemma*. Retrieved October 28, 2019 from http://www.rasmusen.org/x/images/pd.jpg

Tucker, A. W. (1983). The Mathematics of Tucker: A Sampler. *The Two-Year College Mathematics Journal*, 14(3), 228–232. https://doi.org/10.2307/3027092

Turing (1948). *Intelligent Machinery*. Report for National Physical Laboratory Universal Turing Machine. In Copeland, J., Bowen, J., Sprevak, M., & Wilson, R. (2017). The Turing Guide. Oxford University Press. https://doi.org/10.1093/oso/9780198747826.

001.0001, and available from https://turingarchive.kings.cam.ac.uk/unpublished-manuscripts-and-drafts-amtc/amt-c-11 (accessed 11/12/2023).

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59, 433–460. https://doi.org/10.1093/mind/LIX.236.433

Turing, A. M. (1951 [1996]). Intelligent Machinery - a Heretical Theory. Philosophia *Mathematica*, 4(3), 256–260.

Tversky, A., & Kahneman, D. (1974). Judgment under Uncertainty: Heuristics and Biases. *Science*, 185(4157), 1124–1131. https://doi.org/10.1126/science.185.4157. 1124

Ullman, T. (2023). *Large Language Models Fail on Trivial Alterations to Theory-of-Mind Tasks* (arXiv:2302.08399). arXiv. https://doi.org/10.48550/arXiv.2302.08399

van Damme, E. (1986). The Nash bargaining solution is optimal. *Journal of Economic Theory*, 38(1), 78–100. https://doi.org/10.1016/0022-0531(86)90089-X

van der Wal, J. (1980). Stochastic dynamic programming: successive approximations and nearly optimal strategies for markov decision processes and markov games. Eindhoven University of Technology. Stichting Mathematisch Centrum. Doctoral Thesis. https://doi.org/10.6100/IR144733

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. Retrieved March 5, 2019 from https://arxiv.org/abs/1706.03762v5

von Neumann, J. (1928a). Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1), 295–320. https://doi.org/10.1007/BF01448847

von Neumann, J., & Bargmann, S. (1928b, 1959). English Translation of Zur Theorie der Gesellschaftsspiele. In Tucker A. W., & Luce R. D. (Eds.), *Contributions to the Theory of Games* (AM-40), Volume IV, 1959. Princeton University Press.

von Neumann, J., & Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press.

von Neumann, J., & Morgenstern, O. (1953). *Theory of Games and Economic Behavior (Third Edition)*. Princeton University Press.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph Attention Networks. Retrieved July 12 2023 from https://doi.org/10.48550/arXiv.1710.10903

Wagner, T., Shapiro, J., Xuan, P., & Lesser, V. (1999). *MultiLevel Conflict in Multi-Agent Systems*. Proceedings of the 1999 AAAI Workshop on Negotiation in Multi-Agent Systems.

Walliser, B. (1988). A simplified taxonomy of 2 × 2 games. *Theory and Decision*, 25(2), 163–191. https://doi.org/10.1007/BF00134158

Walsh, T., Levy, N., Bell, G., Elliott, A., Maclaurin, J., Mareels, I.M.Y., Wood, F.M., (2019) The effective and ethical development of artificial intelligence: An opportunity to improve our wellbeing. Report for the Australian Council of Learned Academies, www.acola.org. Retrieved July 19, 2023 from https://acola.org/hs4-artificial-intelligence-australia/

Wang, J., & Guo, J. (2019). A synergy of punishment and extortion in cooperation dilemmas driven by the leader. *Chaos, Solitons & Fractals*, 119, 263–268. https://doi.org/10.1016/j.chaos.2019.01.004

Wang, W., Hao, J., Wang, Y., & Taylor, M. (2018). *Towards Cooperation in Sequential Prisoner's Dilemmas: A Deep Multiagent Reinforcement Learning Approach.* Retrieved February 4, 2022 from https://arxiv.org/abs/1803.00162

Wang, X., & Sandholm, T. (2002). Reinforcement Learning to Play an Optimal Nash Equilibrium in Team Markov Games. In Becker, S. Thrun S., Obermayer, K. (Eds.), *Advances in Neural Information Processing Systems*. (pp. 1571–1578). MIT Press.

Wang, Y., Shi, Z. R., Yu, L., Wu, Y., Singh, R., Joppa, L., & Fang, F. (2019). *Deep Reinforcement Learning for Green Security Games with Real-Time Information*. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), Article 01. https://doi.org/10.1609/aaai.v33i01.33011401

Watkins, C. (1989). *Learning from Delayed Rewards*. King's College, Cambridge, Doctoral Thesis

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., & Fedus, W. (2022). Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*.

Weinberg, M., & Rosenschein, J. S. (2004). *Best-response multiagent learning in non-stationary environments*. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004., 506–513.

Weiner, M. (Writer), & Getzinger, J. (Director) (2010, September 5). The Suitcase. (Season 4, Episode 7) [TV Series Episode]. Weiner, M. (Executive Producer), *Mad Men*. Weiner Bros. Productions, Lionsgate Television, & AMC Original Productions.

West-Eberhard, M. J. (2003). *Developmental plasticity and evolution*. Oxford University Press.

Wilcoxon, F. (1945) Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, Vol. 1, No. 6. (Dec., 1945), pp. 80-83.

Williams, G. C. (1966). Adaptation and natural selection: A critique of some current evolutionary thought. Princeton University Press.

Wilson, E. O. (1975). *Sociobiology: The new synthesis*. Belknap Press, Harvard University Press.

Winfield, A. F., Michael, K., Pitt, J., & Evers, V. (2019). *Machine Ethics: The Design and Governance of Ethical AI and Autonomous Systems [Scanning the Issue]*. Proceedings of the IEEE, 107(3), 509–517. https://doi.org/10.1109/JPROC.2019.2900622

Wong, A., Bäck, T., Kononova, A. V., & Plaat, A. (2023). Deep multiagent reinforcement learning: Challenges and directions. *Artificial Intelligence Review*, 56(6), 5023–5056. https://doi.org/10.1007/s10462-022-10299-x

Wooldridge, M. (2012a). Computation and the prisoner's dilemma. *IEEE Intelligent Systems*, 27(2), 75–80. https://doi.org/10.1109/MIS.2012.30

Wooldridge, M. (2012b). Does Game Theory Work? *IEEE Intelligent Systems*, 27(6), 76–80. https://doi.org/10.1109/MIS.2012.108

Wu, Y. (2023). Artificial Intelligence in Chinese Science Fiction: From the Spring and Autumn and Warring States Periods to the Era of Deng Xiaoping. In Cave S., & Dihal K. (Eds.), *Imagining AI: How the World Sees Intelligent Machines*. (pp. 361-372). Oxford University Press.

Wynne-Edwards, V. C. (1963). *Animal dispersion in relation to social behaviour*. Oliver and Boyd, Edinburgh.

Xu, L., Bondi, E., Fang, F., Perrault, A., Wang, K., & Tambe, M. (2020). *Dual-Mandate Patrols: Multi-Armed Bandits for Green Security*. Retrieved February 4, 2022 from https://arxiv.org/abs/2009.06560v2

Yang, X., Zhang, F., & Wang, W. (2019). Predation promotes cooperation in Prisoner's dilemma games. *Physica A: Statistical Mechanics and Its Applications*, 514, 20–24. https://doi.org/10.1016/j.physa.2018.09.054

Yao, X. (1996). Evolutionary stability in the n-person iterated prisoner's dilemma. *Biosystems*, 37(3), 189–197. https://doi.org/10.1016/0303-2647(95)01558-2

Young, H. P. (1993). The Evolution of Conventions. *Econometrica*, 61(1), 57. https://doi.org/10.2307/2951778

Yudkowsky, E. (2008). Artificial Intelligence as a Positive and Negative Factor in Global Risk. In Bostrom, N., & Ćirković, M. M. (Eds.), *Global Catastrophic Risks*, (pp. 308-345). Oxford University Press.

Zuboff, S. (2019). *The Age of Surveillance Capitalism*. Public Affairs.

Appendix A

Agent Model Overview

0000, 0001, 0010, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 0000, 0001, 0010, 0100, 0101, 0110

— Agent One

The *agent model* is a software tool for allowing two agents to execute various strategies within 2x2 Markov Games of canonical and topological orientation. Canonical games are the scalar representations of social dilemmas such as Prisoner's Dilemma, Stag Hunt, and Chicken. Topological games are derived from the Robinson and Goforth (2005) $\mathbf{2} \times \mathbf{2}$ topology. Strategies need only conform to firstly, their sole input being the reward signal from the last timestep (and possibly, if the constraints are relaxed, then the representation of the game state (e.g., *CC*) at the last timestep may also be available; and secondly, the only output from the strategy is a representation of an action—either 0 or 1.

Game state is represented as the concatenation of the actions of both agents at the last timestep; e.g., '10' represents agent 0 action of 1 (defect), and agent 1 action of 0 (cooperate). The model allows the definition of an experiment as one of the following modes:

- *symmetric selfplay*
- *selfplay parameter study*
- *spacewalk*
- *asymmetric selfplay*
- *round-robin*
- *single-model tournament*
- *multi-model tournament*

Each mode accepts parameters to define the operation of the experiment:

- Game model matrix
- $\boldsymbol{T}$, or episode length
- $\boldsymbol{e}$, or number of episodes
- Agent strategy and strategy options including hyperparameters.
- Game model reward type (for example, scalar or ordinal)
- Options to control execution e.g., whether to write agent state at each timestep.

The framework is platform agnostic, evaluated on Windows and Linux on HPC with PBS. All experiment modes runs under Python 3.7+. There are some caveats to this: parallelization handling in HPC/PBS environments allows some experiment modes that are impractical on a single core/few-core system. However, each experiment type will run standalone in its atomic mode on either platform. The platform-agnosticism allows moving data generated on one platform to the other to run other analysis. The experiment framework writes metadata at both experiment and observation stages: paths are stored in the metadata in a platform-specific manner for archival metadata and subsequently managed in code in a platform-agnostic way to ensure data and metadata are retained and accessible. The experiment framework writes most datafiles on an episodic basis. Each episode can generate, per timestep:

- an action history,
- a reward history,
- a game outcome history,
- each agent's internal state,
- each agent's strategy's internal state.

Each of these is written to file at the termination of each episode, flushed, then the next episode is started, re-instantiating all objects (agents, strategies). This allows the management of CPU, memory, and space resources, and ensures each episode is *i.i.d*.

For very large experiments (e.g., where $\boldsymbol{T}$ is large) agent and strategy state-writes can be opted-out. Where an observation only requires data from histories, not internal state, then this reduces overall resource use.

Another option allows whether to compress writes (at the end of each episode) for each of the types of data, i.e., *action_history*, or *reward_history*. This has two benefits: first, the data is highly compressible, on the order of two orders of magnitude, so has resource use benefit. Secondly, in an HPC environment it allows expedited transfer of moving one large file rather than thousands of small files.

In addition, every atomic game mode experiment instance[66] has a set of metadata that is collected on the computational aspects of the specific experiment instance. This set of metadata includes CPU time usage, memory usage, process time (to the nanosecond, $\mathbf{10^{09}}$) for execution of the algorithm under study, and data storage usage. These metrics provide insight into the performance and computational cost of an algorithm.

The data from each experiment instance is analysed in a two-stage process. The first stage takes the raw episodic data and calculates aggregated and summary statistics: on episode-and timestep-level actions and rewards; on game outcomes; and also, CPU-time and memory usage metadata. Subsequent analysis performs statistical tests on the outputs of the first pass. For example, in Experiment Series Two (Chapter Five) each outcome distribution is tested for normality via the Shapiro-Wilk normality test (Shapiro & Wilk, 1953), and then tested for equivalence via the Wilcoxon Signed Rank (Paired Treatment) test (Wilcoxon, 1945).

[66] As identified by experiment ID, see Appendix B.1.

A.1 Initial Hyperparameter Values

Mutable, or 'tuneable', hyperparameters such as $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ set the learning rate ($\boldsymbol{\alpha}$) and the discount on expected return ($\boldsymbol{\gamma}$) respectively. Before assessing the performance of algorithms in *symmetric selfplay* it is necessary to assess, for each algorithm that has hyperparameters, the optimal[67] parameter values for that algorithm. This chicken-or-egg problem can be resolved either by:

1. relying on accepted norms as found in the literature;
2. running initial scoping experiments;
3. allowing agents to modify their own parameters during execution; or,
4. by running parameter sweeps (aka grid-search) of a given granularity (e.g., 0.1 increments) to identify optimal settings.

In the case of (1), the literature suggests that the values for these parameters should be set within certain bounds depending on the character of the algorithm. For example, to learn slowly, or conversely to learn quickly, the learning rate α can be adjusted from low to high in the range (0,1]. For the results presented in Chapter Four, the parameters for the learning algorithms have been selected through the combination of (1), (2) and (4).

For other algorithms, the initial parameter values are assigned on the basis of (1) and (2). The third experiment series (ES3), presented in Chapter Six actually benefits from increased agent exploration, even to the point of poor agent rewards, as the focus is very specific, so the reward value is not a response variable in this experiment's context, only. Nevertheless, hyperparameter values are assigned to the algorithm *Q-Learning* in ES3 on the basis of (4).

A.2 Sources of Variance

A potential source of bias is identified in the implementation of the study in the ordering of the execution of the agent actions. While the model conforms to the constraints (§1.3) of imperfect information (agents cannot see other agent's actions) and also incomplete information (agents cannot see other agent's payoffs); additionally, neither agent becomes aware of the other's action in the current timestep, at the time of the subsequent timestep, except for those few already highlighted in §1.3.

The implementation of the agent class orders the returned value from the execution of each agent's *action()* method, orthogonally to the other agent, in a tuple (named *previous_outcome*) such that *agent_zero's* action() method is called first, and *agent_one's* action() method is called second, as tuples are ordered[68]. Without the ability for the agents to be aware of the other agent's action in the current timestep it is hard to see how this could affect their behaviour, and secondly, if there is an influence, or some bias at a deeper level (in the Python environment, for example, or in random number generation), it is not clear how that bias could emerge non-uniformly.

[67] *Optimal* is defined contextually, as either that setting of the parameter(s) that leads to the highest frequency of **mutual cooperation** for an agent; or that setting of the parameter that results in the highest **total reward**, over an episode, for an agent.

[68] https://docs.python.org/3/library/stdtypes.html#tuple

A.3 Memory Depth

The restriction on memory depth has implications to the performance of optimal mutual cooperation. Initial scoping showed that a memory depth of two, for some algorithms, substantially improved performance. However, the objectives of these experiments is not to optimize all available response variables—nor, specifically, to necessarily obtain the absolute optimal frequency of mutual cooperation, or reward—but instead, to examine the behaviour of algorithms over a subset of their possible states. In each experiment instance each and every algorithm is working to maximise its reward as its singular objective.

A.4 Energy Use

The total HPC energy use for the experiments presented in this thesis is **15.33kWh**, as summarised in **Table A.4** This is a lower bound on overall use, as it does not consider scoping experiments, developmental iteration, or testing.

A.4.1 Energy Use Formula

Energy use for the HPC cluster is the product of *time executing per CPU* by *the power draw of the CPU*. The per CPU energy use is **13.75W/CPU**[69], calculated as:

$$\mathbf{350}\boldsymbol{W\ per\ node}, \mathbf{28}\ \boldsymbol{CPUs\ per\ node}, \boldsymbol{PUE} = \mathbf{1.1}, \boldsymbol{gives}\ \mathbf{350\ /\ 28\ x\ 1.1\ =\ 13.75}\boldsymbol{W/cpu}.$$

using the formula for power usage (DataCenters.com, 2015):

$$\boldsymbol{E(kWh)} = \boldsymbol{P(W)} \times \boldsymbol{t}\ \mathbf{/\ 1000}\boldsymbol{(W/kW)}\ \boldsymbol{where\ t\ =\ CPU\ Hours}$$

In Experiment Series Three the instances with a 2-digit identifier are experiments performed on 'localhost', i.e., my laptop, not the HPC system. The formula is applied to the 'laptop' experiment instances using $\boldsymbol{P(W)} = \mathbf{65}\boldsymbol{W}$, being max draw (Dell, 2021).

A.4.2 Energy Use Experiment Series One

Table A.1: Energy Use for Experiment Series One. Experiment instance execution time recorded by python's *process_time_ns()* calls at start and end of python script execution. Energy use of HPC cluster calculated at **13.75W/CPU** (see §A.4.1).

Experiment Series One Energy Use				
Job ID	**CPU Minutes**	**CPU Hours**	**E(Wh)**	**E(kWh)**
398741	63.6	1.06	14.58	0.01458
398848	530.47	8.84	121.55	0.12155
398908	437.69	7.3	100.38	0.10038
478692	328.938	5.48	75.35	0.07535
Total	1360.698	22.68	311.86	**0.31186**

[69] Miezitis, J., personal communication, Digital Research Services, University of Tasmania, 1 Oct 2021.

A.4.3 Energy Use Experiment Series Two

Table A.2: Energy Use Experiment Series Two. Experiment instance execution time recorded by python's process_time_ns calls at start and end of python script execution. Energy use of HPC cluster calculated at **13.75W/CPU**, (see §A.4.1).

Experiment Series Two Energy Use				
Job ID	**CPU Minutes**	**CPU Hours**	**E(Wh)**	**E(kWh)**
180217	1322.91	22.04849167	303.1667604	0.303167
133414	1139.34	18.98907	261.0997125	0.2611
133440	1156.23	19.27053	264.9697875	0.26497
127288	892.39	14.87308333	204.5048958	0.204505
127612	1001.28	16.68796867	229.4595692	0.22946
127633	896.82	14.94704	205.5218	0.205522
127812	871.24	14.52061	199.6583875	0.199658
127850	916.43	15.27374167	210.0139479	0.210014
127910	940.29	15.67152833	215.4835146	0.215484
128039	980.22	16.33696167	224.6332229	0.224633
128198	952.65	15.877515	218.3158313	0.218316
128253	966.43	16.107195	221.4739313	0.221474
128327	968.49	16.14146	221.945075	0.221945
128384	922.11	15.36855	211.3175625	0.211318
180292	1263.1	21.05172167	289.4611729	0.289461
133454	1122.67	18.711215	257.2792063	0.257279
133455	1123.72	18.72864	257.5188	0.257519
129635	920.03	15.33374333	210.8389708	0.210839
129642	1039.95	17.33246	238.321325	0.238321
129656	911.54	15.19232833	208.8945146	0.208895
129683	904.86	15.08105667	207.3645292	0.207365
129710	945.83	15.76383	216.7526625	0.216753
129718	983.21	16.38676667	225.3180417	0.225318
129723	984.68	16.41125167	225.6547104	0.225655
129726	985.83	16.43046	225.918825	0.225919
129727	987.27	16.45443667	226.2485042	0.226249
129757	990.77	16.512865	227.0518938	0.227052
129758	941.92	15.69870833	215.8572396	0.215857
180263	1324.53	22.07550167	303.5381479	0.303538
133442	1161.64	19.36064833	266.2089146	0.266209
133451	1146.06	19.101025	262.6390938	0.262639
132060	881.1	14.68501833	201.9190021	0.201919
132090	1010.83	16.84720667	231.6490917	0.231649
132114	909.99	15.16645667	208.5387792	0.208539
132520	889.7	14.82826667	203.8886667	0.203889
132138	939.95	15.66576167	215.4042229	0.215404
132161	974.38	16.23959833	223.2944771	0.223294
132191	980.99	16.34977	224.8093375	0.224809
132355	964.22	16.07031	220.9667625	0.220967
132376	982.96	16.382585	225.2605438	0.225261
132414	987.94	16.465585	226.4017938	0.226402
132416	929.59	15.493235	213.0319813	0.213032
180380	1080	17.99996667	247.4995417	0.2475
133460	1137.04	18.95062667	260.5711167	0.260571
133462	1143.95	19.06576833	262.1543146	0.262154
133161	888.2	14.80334167	203.5459479	0.203546
133163	1003.68	16.727955	230.0093813	0.230009
133166	902.7	15.04503333	206.8692083	0.206869
133087	875.95	14.59912667	200.7379917	0.200738
133169	950.61	15.84349167	217.8480104	0.217848
133170	981.4	16.35671	224.9047625	0.224905
133173	973.4	16.22324833	223.0696646	0.22307
133177	968.78	16.14627167	222.0112354	0.222011

Experiment Series Two Energy Use				
Job ID	**CPU Minutes**	**CPU Hours**	**E(Wh)**	**E(kWh)**
133205	963.26	16.05436667	220.7475417	0.220748
133211	962.66	16.04425667	220.6085292	0.220609
133215	913.3	15.22160833	209.2971146	0.209297
Total	55860.96	931.02	12801.47	**12.8**

A.4.4 Energy Use Experiment Series Three

Table A.3: Energy Use Experiment Series Three. Experiment instance execution time recorded by python's process_time_ns calls at start and end of python script execution. Energy use of HPC cluster calculated at **13.75W/CPU**, (see §A.4.1). Note that the Job IDs with a 2-digit identifier were run locally, not on HPC, but same energy use formula has been applied for consistency.

Experiment Series Three Energy Use				
Job ID	**CPU Minutes**	**CPU Hours**	**E(Wh)**	**E(kWh)**
477295	7777.13	129.62	1782.26	1.78
14	26.99	0.45	29.24	0.03
15	26.43	0.44	28.63	0.03
16	26.31	0.44	28.5	0.03
18	122.6	2.04	132.81	0.13
19	70	1.17	75.79	0.08
20	102	1.7	110.26	0.11
21	29.27	0.5	31.71	0.03
Total	8180.46	136.34	2219.2	**2.22**

A.4.5 Energy Use Total

Table A.4: Energy Use Total – All Experiment Series. Experiment instance execution time recorded by python's process_time_ns calls at start and end of python script execution. Energy use of HPC cluster calculated at **13.75W/CPU**, energy use of laptop at **65W** (see §A.4.1).

Experiment Series - ALL Energy Use				
Exp Series	**CPU Minutes**	**CPU Hours**	**E(Wh)**	**E(kWh)**
One	1360.698	22.68	311.86	0.31
Two	55860.96	931.02	12801.47	12.8
Three	8180.46	136.34	2219.2	2.22
Total	65402.12	1090.035	15332.5	**15.33**

Appendix B

Supplementary Material

0000, 0001, 0010, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 0000, 0001, 0010, 0100, 0101, 0110

—Agent Zero

B.1 *rRGS* Graph Adjacency List

The reduced Robinson-Goforth Space (*rRGS*) is obtained from the atomic application of the generators $\boldsymbol{R_{12}}$, $\boldsymbol{R_{23}}$, $\boldsymbol{R_{34}}$, $\boldsymbol{C_{12}}$, $\boldsymbol{C_{23}}$, and $\boldsymbol{C_{34}}$ to any strictly ordinal $\mathbf{2 \times 2}$ game model. The *rRGS* is closed under this method. An adjacency list representation for the *rRGS* is given in **Table B.1**. This list identifies the game models reached via the edges formed by each generator. An adjacency list as a plain csv, and in *igraph* (Csardi et al., 2006) format, is detailed in Appendix B.5.

Table B.1: Adjacency table for *rRGS*.

g_i	R_{12}	R_{23}	R_{34}	C_{12}	C_{23}	C_{34}
g111	g121	g161	g221	g112	g116	g412
g112	g122	g162	g222	g111	g113	g411
g113	g123	g163	g223	g114	g112	
g114	g124	g164	g224	g113	g115	
g115	g125	g165	g225	g116	g114	
g116	g126	g166	g226	g115	g111	
g121	g111	g131	g211	g122	g126	g422
g122	g112	g132	g212	g121	g123	g421
g123	g113	g133	g213	g124	g122	
g124	g114	g134	g214	g123	g125	
g125	g115	g135	g215	g126	g124	
g126	g116	g136	g216	g125	g121	
g131	g141	g121		g132	g136	g432
g132	g142	g122		g131	g133	g431
g133	g143	g123		g134	g132	
g134	g144	g124		g133	g135	
g135	g145	g125		g136	g134	
g136	g146	g126		g135	g131	
g141	g131	g151		g142	g146	g442
g142	g132	g152		g141	g143	g441
g143	g133	g153		g144	g142	
g144	g134	g154		g143	g145	
g145	g135	g155		g146	g144	
g146	g136	g156		g145	g141	
g151	g161	g141		g152	g156	g452
g152	g162	g142		g151	g153	g451

g_i	R_{12}	R_{23}	R_{34}	C_{12}	C_{23}	C_{34}
g153	g163	g143		g154	g152	
g154	g164	g144		g153	g155	
g156	g166	g146		g155	g151	
g161	g151	g111		g162	g166	g462
g162	g152	g112		g161	g163	g461
g163	g153	g113		g164	g162	
g164	g154	g114		g163	g165	
g165	g155	g115		g166	g164	
g166	g156	g116		g165	g161	
g211	g221	g261	g121	g212	g216	g312
g212	g222	g262	g122	g211	g213	g311
g213	g223	g263	g123	g214	g212	
g214	g224	g264	g124	g213	g215	
g215	g225	g265	g125	g216	g214	
g216	g226	g266	g126	g215	g211	
g221	g211	g231	g111	g222	g226	g322
g222	g212	g232	g112	g221	g223	g321
g223	g213	g233	g113	g224	g222	
g224	g214	g234	g114	g223	g225	
g225	g215	g235	g115	g226	g224	
g226	g216	g236	g116	g225	g221	
g231	g241	g221		g232	g236	g332
g232	g242	g222		g231	g233	g331
g233	g243	g223		g234	g232	
g234	g244	g224		g233	g235	
g235	g245	g225		g236	g234	
g236	g246	g226		g235	g231	
g241	g231	g251		g242	g246	g342
g242	g232	g252		g241	g243	g341
g243	g233	g253		g244	g242	
g244	g234	g254		g243	g245	
g246	g236	g256		g245	g241	
g251	g261	g241		g252	g256	g352
g252	g262	g242		g251	g253	g351
g253	g263	g243		g254	g252	
g254	g264	g244		g253	g255	
g255	g265	g245		g256	g254	
g256	g266	g246		g255	g251	
g261	g251	g211		g262	g266	g362
g262	g252	g212		g261	g263	g361
g263	g253	g213		g264	g262	
g264	g254	g214		g263	g265	
g265	g255	g215		g266	g264	
g266	g256	g216		g265	g261	
g311	g321	g361	g421	g312	g316	g212
g312	g322	g362	g422	g311	g313	g211
g313	g323	g363	g423	g314	g312	
g314	g324	g364	g424	g313	g315	
g315	g325	g365	g425	g316	g314	
g316	g326	g366	g426	g315	g311	
g321	g311	g331	g411	g322	g326	g222
g322	g312	g332	g412	g321	g323	g221
g323	g313	g333	g413	g324	g322	
g324	g314	g334	g414	g323	g325	
g325	g315	g335	g415	g326	g324	
g326	g316	g336	g416	g325	g321	
g331	g341	g321		g332	g336	g232
g332	g342	g322		g331	g333	g231
g333	g343	g323		g334	g332	
g334	g344	g324		g333	g335	
g335	g345	g325		g336	g334	

g_i	R_{12}	R_{23}	R_{34}	C_{12}	C_{23}	C_{34}
g336	g346	g326		g335	g331	
g341	g331	g351		g342	g346	g242
g342	g332	g352		g341	g343	g241
g343	g333	g353		g344	g342	
g344	g334	g354		g343	g345	
g345	g335	g355		g346	g344	
g346	g336	g356		g345	g341	
g351	g361	g341		g352	g356	g252
g352	g362	g342		g351	g353	g251
g353	g363	g343		g354	g352	
g354	g364	g344		g353	g355	
g355	g365	g345		g356	g354	
g356	g366	g346		g355	g351	
g361	g351	g311		g362	g366	g262
g362	g352	g312		g361	g363	g261
g363	g353	g313		g364	g362	
g364	g354	g314		g363	g365	
g365	g355	g315		g366	g364	
g366	g356	g316		g365	g361	
g411	g421	g461	g321	g412	g416	g112
g412	g422	g462	g322	g411	g413	g111
g413	g423	g463	g323	g414	g412	
g414	g424	g464	g324	g413	g415	
g415	g425	g465	g325	g416	g414	
g416	g426	g466	g326	g415	g411	
g421	g411	g431	g311	g422	g426	g122
g422	g412	g432	g312	g421	g423	g121
g423	g413	g433	g313	g424	g422	
g424	g414	g434	g314	g423	g425	
g425	g415	g435	g315	g426	g424	
g426	g416	g436	g316	g425	g421	
g431	g441	g421		g432	g436	g132
g432	g442	g422		g431	g433	g131
g433	g443	g423		g434	g432	
g434	g444	g424		g433	g435	
g435	g445	g425		g436	g434	
g436	g446	g426		g435	g431	
g441	g431	g451		g442	g446	g142
g442	g432	g452		g441	g443	g141
g443	g433	g453		g444	g442	
g444	g434	g454		g443	g445	
g445	g435	g455		g446	g444	
g446	g436	g456		g445	g441	
g451	g461	g441		g452	g456	g152
g452	g462	g442		g451	g453	g151
g453	g463	g443		g454	g452	
g454	g464	g444		g453	g455	
g455	g465	g445		g456	g454	
g456	g466	g446		g455	g451	
g461	g451	g411		g462	g466	g162
g462	g452	g412		g461	g463	g161
g463	g453	g413		g464	g462	
g464	g454	g414		g463	g465	
g465	g455	g415		g466	g464	
g466	g456	g416		g465	g461	

B.2 Experiment Series One

B.2.1 Experiment IDs & Analysis Datasheets

Table B.2: Experiment Series One Experiment IDs.

Experiment Series One Experiment IDs				
Job ID	**Description**	**Representation**	**Episodes**	**Timesteps**
398741	Game-theoretic	Ordinal	1	1000
398848	Binary Bandits	Ordinal	1	1000
398908	Foundational-RL	Ordinal	1	1000
478692	Final Round	Ordinal	1	1000

Table B.3: Experiment Series One Datasheets. Datasheets are packaged with a data release as per Appendix B.5.
† igraph software (Csardi et al., 2006).

Experiment Series One Datasheets		
Item	**Filename**	**Description**
1	correlation-summary.xlsx	Calculated in R Studio (); Summarised in this file.
2	adjacency-list-rgs-ordering.txt	Adjacency List in arrow/csv format.
3	adjacency-list-rgs-ordering--igraph.txt	Adjacency list for rRGS in igraph† format.
4	ES1_tournament.xlsx	Match IDs, All Pairings, TR, and MCR summaries.

Table B.4: Experiment Series One Datasets.

Experiment Series One; Datasets BasePath="/ES1/${Dir}/${Dir2}"					
Item	**Description**	**Data Type**	**Dir**	**Dir2**	**Size (MB)**
1	Game-theoretic Algorithms	Raw Episode/ Timestep	exp_data	398741	9.5
2	Binary Bandits Algorithms	Raw Episode/ Timestep	exp_data	398848	111
3	foundational-RL Algorithms	Raw Episode/ Timestep	exp_data	398908	72.5
4	Final Round Algorithms	Raw Episode/ Timestep	exp_data	398945	543
5	Game-theoretic Algorithms	Orderset, Summaries	obs_data	398741	2.6
6	Binary Bandits Algorithms	Orderset, Summaries	obs_data	398848	24.1
7	foundational-RL Algorithms	Orderset, Summaries	obs_data	398908	20.8
8	Final Round Algorithms	Orderset, Summaries	obs_data	398945	21.2

B.2.2 Supplementary Data

- **Table B.5** lists default hyperparameter values for *Binary Bandit* algorithms.
- **Table B.6** lists default hyperparameter values for *fRL* algorithms.
- **Table B.7** shows mutual cooperation locations for every game model in the *rRGS*.

Table B.5: Default hyperparameter values for *Binary Bandit* algorithms. MCR was not recorded in these early ad-hoc *symmetric selfplay* experiment instances. Cross-representation comparisons were not formally conducted for this class of algorithms. All algorithms have a memory depth of one (1). Mutual cooperation rate was not recorded in these instances. **Initial Action:** Probability weighting to 0 (cooperate) or 1 (defect). $\boldsymbol{\alpha}$**:** alpha, learning rate parameter. $\boldsymbol{\beta}$**:** beta, learning rate parameter. $\boldsymbol{\gamma}$**:** gamma, discount on expected return rate parameter. $\boldsymbol{\tau}$ is temperature. $\boldsymbol{\lambda}$ $\boldsymbol{Critic}$**:** lambda parameter for critic. $\boldsymbol{\lambda}$ $\boldsymbol{Actor}$**:** lambda parameter for actor. $\boldsymbol{\lambda}$**:** lambda parameter. $\boldsymbol{e}$**:** epsilon, $\boldsymbol{e-greedy}$ action-selection. A dash, -, indicates default value used.

Algorithm	Parameter	Default	Representation	
			Scalar	Ordinal
Incremental, Action Preferences, Softmax				
bandit_inc_softmax_ap_2ed				
	$\boldsymbol{\alpha}$	0.1	-	-
	$\boldsymbol{\tau}$	10	-	-
Non-Incremental, Action Preferences, Softmax				
bandit_noninc_softmax_ap_2ed				
	$\boldsymbol{\alpha}$	0.1	-	-
	$\boldsymbol{\tau}$	10	-	-
Pursuit, Sample-Average				
bandit_pursuit_sav				
	Bias to 0	5	-	-
	Bias to 1	5	-	-
	$\boldsymbol{\beta}$	0.1	-	-
Reinforcement Comparison				
bandit_reinfcomp				
S&B do not use temperature parameter; ref_reward set to 4, as 5 would break ordinal	Initial Action	0.5	-	-
	$\boldsymbol{\alpha}$	0.1	-	-
	$\boldsymbol{\beta}$	0.1	-	-
	Ref Reward	4	-	-
Sample-Average Incremental				
bandit_sav_inc				
	Initial Action	0.5	-	-
	Bias to 0	5	-	-
	Bias to 1	5	-	-
	$\boldsymbol{e}$	0.1	-	-
Sample-Average, Incremental, Optimistic, Greedy				
bandit_sav_inc_optimistic_greedy				
	Bias to 0	5	-	-
	Bias to 1	5	-	-
Sample-Average, Incremental, Softmax				
bandit_sav_inc_softmax				
	$\boldsymbol{\tau}$	10	-	-
Sample-Average, Non-Incremental				
bandit_sav_noninc				
	Initial Action	0.5	-	-
	$\boldsymbol{e}$	0.1	-	-

Algorithm	Parameter	Default	Representation	
			Scalar	Ordinal
Sample-Average, Non-Incremental, Softmax				
bandit_sav_noninc_softmax				
	Initial Action	0.5	-	-
	τ	10	-	-
Supervised Learning, Direct				
bandit_sl_direct				
	Initial Action	0.5	-	-
Supervised Learning, Learning Automata, Linear-Reward-Inaction				
bandit_sl_la_lri				
	Initial Action	0.5	-	-
	α	0.3	-	-
Supervised Learning, Learning Automata, Linear-Reward-Penalty				
bandit_sl_la_lrp				
	Initial Action	0.5	-	-
	α	0.1	-	-
Weighted Average				
Tag: *bandit_wa*				
	Initial Action	0.5	-	-
	α	0.1	-	-
	e	0.1	-	-
Weighted-Average, Optimistic, Greedy				
bandit_wa_optimistic_greedy				
	α	0.1	-	-
	Bias to 0	10	-	-
	Bias to 1	10	-	-
Weighted Average, Softmax				
bandit_wa_softmax				
	α	0.1	-	-
	Bias to 0	5	-	-
	Bias to 1	5	-	-
	τ	10	-	-
Weighted Average, Softmax, Action Preferences				
bandit_wa_softmax_ap_2ed				
	α	0.1	-	-
	τ	10	-	-
Weighted Average, Upper Confidence Bounds				
bandit_wa_ucb				
	Initial Action	0.5	-	-
	α	0.1	-	-
	c	1	-	-
	Bias to 0	0	-	-
	Bias to 1	0	-	-

Table B.6: Default hyperparameter values for *fRL* algorithms. Values were observed in ad-hoc *symmetric selfplay* experiment instances prior to Multi-Model Tournament experiment series. All algorithms have a memory depth of one (1). **Best MCR:** Peak observed mutual cooperation rate. **Initial Action:** Probability weighting to 0 (cooperate) or 1 (defect). $\boldsymbol{\alpha}$**:** alpha, learning rate parameter. $\boldsymbol{\beta}$**:** beta, learning rate parameter. $\boldsymbol{\gamma}$**:** gamma, discount on expected return rate parameter. $\boldsymbol{\lambda\ Critic}$**:** lambda parameter for critic. $\boldsymbol{\lambda\ Actor}$**:** lambda parameter for actor. $\boldsymbol{\lambda}$**:** lambda parameter. $\boldsymbol{e}$**:** epsilon, $\boldsymbol{e-greedy}$ action-selection. A dash, -, indicates default value used. S&B is Sutton and Barto (1998, 2018).

Algorithm	Parameter	Default	Representation	
			Scalar	Ordinal
Actor/Critic				
actor_critic_1ed				
	Best MCR		0.30445	0.27602
S&B do not use temperature parameter	Initial Action	0.5	-	-
	$\boldsymbol{\alpha}$	0.9	0.6	0.9
	$\boldsymbol{\beta}$	0.9	0.9	0.9
	$\boldsymbol{\gamma}$	0.9	-	-
Actor/Critic, Eligibility Traces				
actor_critic_1ed_eligibility_traces				
	Best MCR		0.50128	0.46677
S&B do not use temperature parameter	Initial Action	0.5	-	-
	$\boldsymbol{\alpha}$	0.9	0.8	1
	$\boldsymbol{\beta}$	0.9	1	1
	$\boldsymbol{\gamma}$	0.9	-	-
	$\boldsymbol{\lambda\ Critic}$	0.9	-	-
	$\boldsymbol{\lambda\ Actor}$	0.9	-	-
Actor-Critic, Replacing Traces				
actor_critic_1ed_replacetrace				
	Best MCR		0.54141	0.44631
S&B do not use temperature parameter	Initial Action	0.5	-	-
	$\boldsymbol{\alpha}$	0.4	0.3	0.4
	$\boldsymbol{\beta}$	0.5	0.7	0.5
	$\boldsymbol{\gamma}$	0.9	-	-
	$\boldsymbol{\lambda\ Critic}$	0.9	-	-
	$\boldsymbol{\lambda\ Actor}$	0.9	-	-
Q-Learning				
qlearning				
	Best MCR		0.80064	0.81487
	Initial Action	0.5	-	-
	$\boldsymbol{\alpha}$	0.1	0.1	0.1
	$\boldsymbol{\gamma}$	0.9	1	0.9
	$\boldsymbol{e}$	0.1	-	-
Double Q-Learning				
double_qlearning				
	Best MCR		0.68897	0.6839
	Initial Action	0.5	-	-
	$\boldsymbol{\alpha}$	0.1	0.1	0.1
	$\boldsymbol{\gamma}$	0.8	0.9	0.8
	$\boldsymbol{e}$	0.1	-	-
Expected SARSA				
expected_sarsa				
	Best MCR		0.75725	0.77295
	Initial Action	0.5	-	-

Algorithm	Parameter	Default	Representation	
			Scalar	Ordinal
	α	0.1	0.1	0.1
	γ	0.9	1	1
	e	0.1	-	-
R Learning				
rlearning				
	Best MCR		0.25553	0.28276
	Initial Action	0.5	-	-
	α	0.1	1	0.1
	β	0.1	1	0.1
	e	0.1	-	-
SARSA				
sarsa				
	Best MCR		0.79898	0.80269
	Initial Action	0.5	-	-
	α	0.1	0.1	0.1
	γ	0.9	0.9	1
	e	0.1	-	-
SARSA Lambda				
sarsa_lambda				
	Best MCR		0.68815	0.86669
	Initial Action	0.5	-	-
	α	0.2	0.1	0.2
	γ	0.9	1	0.9
	λ	0.9		
	e	0.1	-	-
SARSA Lambda, Replacing Traces				
sarsa_lambdareplacetrace				
	Best MCR		0.88995	0.89
	Initial Action	0.5	-	-
	α	0.1	0.1	0.1
	γ	0.9	0.7	0.9
	λ	0.9		
	e	0.1	-	-
Watkins (naive) Q, Lambda				
watkins_naïve_q_lambda				
	Best MCR		0.78411	0.88067
	Initial Action	0.5	-	-
	α	0.3	0.2	0.3
	γ	0.8	0.1	0.8
	λ	0.9		
	e	0.1	-	-
Watkins (naive) Q, Lambda, Replacing Traces				
watkins_naïve_q_lambda_replacetrace				
	Best MCR		0.76433	0.89065
	Initial Action	0.5	-	-
	α	0.1	0.1	0.1
	γ	0.9	1	1
	λ	0.9		
	e	0.1	-	-

Algorithm	Parameter	Default	Representation	
			Scalar	Ordinal
Watkins Q, Lambda				
watkins_q_lambda				
	Best MCR		0.87174	0.87771
	Initial Action	0.5	-	-
	α	0.1	0.2	0.1
	γ	0.9	0.9	1
	λ	0.9		
	e	0.1	-	-
Watkins Q, Linear Function Approximation				
watkins_q_lfa				
	Best MCR		0.47126	0.46172
	Initial Action	0.5	-	-
	α	0.7	0.5	0.7
	γ	0.9	0.1	1
	e	0.1	-	-

Table B.7: RGS mutual cooperation locations. Nash Equilibria (**NE**), Nash Bargaining Solutions (**NBS**), MaxiMin (**MaxMin**), and Mutual Cooperation (**MC1, MC2**) cell locations are indicated using cartesian layout (**Cartesian, C**) and canonical (**RGS**) cell indexing. Mapping of cell locations to canonical semantic cell outcomes is as follows: CC ↔ (0,0), CD ↔ (0,1), DC ↔ (1,0), DD ↔ (1,1). Outcome payoff values indicated by **V**. Note lexical ordering of game models. **Sources: Cartesian**: payoff vector ordering (CD, CC, DD, DC) by Robinson and Goforth (2005), adopted by Bruns (2010), and continued with by Crandall et al. (2018a; 2018b). **Canonical:** payoff vector ordering (CC, CD, DC, DD) per canonical game theory (see §2.3.1). **NE1:** Nash Equilibrium One (Robinson & Goforth, 2005; Bruns 2015; Crandall et al, 2018b). **NE2:** Nash Equilibrium Two (Robinson & Goforth, 2005; Crandall et al, 2018b). **NBS1:** Nash Bargaining Solution One (Crandall et al., 2018b). **NBS2:** Nash Bargaining Solution Two (Crandall et al., 2018b). **MaxMin:** Maxi-min solution (Robinson & Goforth, 2005). **MC:** Synthesis: Translation from cartesian payoff vector ordering to canonical payoff vector ordering, see §2.3.1; Ordering of **MC #1** and **MC #2** as per Crandall et al. (2018b); where no NBS or NE solution (**g443**, and **g234**) select maxi-min for mutual cooperation (**MC #1**).

Game	Cartesian	RGS	NE1			NE2			NBS1			NBS2			MaxMin			MC	
Model	Payoff Vector	Payoff Vector	C	RGS	V	C	RGS	V	C	RGS	V	C	RGS	V	C	RGS	V	#1	#2
g111	1,4,3,3,2,2,4,1	3,3,1,4,4,1,2,2	1,0	1,1	2,2	-	-	-	0,1	0,0	3,3	-	-	-	1,0	1,1	2,2	0,0	-
g112	1,4,3,3,2,1,4,2	3,3,1,4,4,2,2,1	1,1	1,0	4,2	-	-	-	1,1	1,0	4,2	-	-	-	1,1	1,0	4,2	1,0	-
g113	1,4,3,2,2,1,4,3	3,2,1,4,4,3,2,1	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	0,0	0,1	1,4	1,1	1,0	4,3	1,0	0,1
g114	1,4,3,1,2,2,4,3	3,1,1,4,4,3,2,2	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	0,0	0,1	1,4	1,0	1,1	2,2	1,0	0,1
g115	1,4,3,1,2,3,4,2	3,1,1,4,4,2,2,3	1,0	1,1	2,3	-	-	-	1,1	1,0	4,2	0,0	0,1	1,4	1,0	1,1	2,3	1,0	0,1
g116	1,4,3,2,2,3,4,1	3,2,1,4,4,1,2,3	1,0	1,1	2,3	-	-	-	1,0	1,1	2,3	-	-	-	1,0	1,1	2,3	1,1	-
g121	2,4,3,3,1,2,4,1	3,3,2,4,4,1,1,2	0,0	0,1	2,4	-	-	-	0,0	0,1	2,4	-	-	-	0,0	0,1	2,4	0,1	-
g122	2,4,3,3,1,1,4,2	3,3,2,4,4,2,1,1	1,1	1,0	4,2	0,0	0,1	2,4	0,1	0,0	3,3	-	-	-	0,1	0,0	3,3	0,0	-
g123	2,4,3,2,1,1,4,3	3,2,2,4,4,3,1,1	1,1	1,0	4,3	0,0	0,1	2,4	1,1	1,0	4,3	0,0	0,1	2,4	0,1	0,0	3,2	1,0	0,1
g124	2,4,3,1,1,2,4,3	3,1,2,4,4,3,1,2	1,1	1,0	4,3	0,0	0,1	2,4	1,1	1,0	4,3	0,0	0,1	2,4	0,0	0,1	2,4	1,0	0,1
g125	2,4,3,1,1,3,4,2	3,1,2,4,4,2,1,3	0,0	0,1	2,4	-	-	-	1,1	1,0	4,2	0,0	0,1	2,4	0,0	0,1	2,4	1,0	0,1
g126	2,4,3,2,1,3,4,1	3,2,2,4,4,1,1,3	0,0	0,1	2,4	-	-	-	1,1	1,0	4,1	0,0	0,1	2,4	0,0	0,1	2,4	1,0	0,1
g131	3,4,2,3,1,2,4,1	2,3,3,4,4,1,1,2	0,0	0,1	3,4	-	-	-	1,0	1,1	4,1	0,0	0,1	3,4	0,0	0,1	3,4	1,1	0,1
g132	3,4,2,3,1,1,4,2	2,3,3,4,4,2,1,1	1,1	1,0	4,2	0,0	0,1	3,4	1,1	1,0	4,2	0,0	0,1	3,4	0,1	0,0	2,3	1,0	0,1
g133	3,4,2,2,1,1,4,3	2,2,3,4,4,3,1,1	1,1	1,0	4,3	0,0	0,1	3,4	1,1	1,0	4,3	0,0	0,1	3,4	0,1	0,0	2,2	1,0	0,1
g134	3,4,2,1,1,2,4,3	2,1,3,4,4,3,1,2	1,1	1,0	4,3	0,0	0,1	3,4	1,1	1,0	4,3	0,0	0,1	3,4	0,0	0,1	3,4	1,0	0,1
g135	3,4,2,1,1,3,4,2	2,1,3,4,4,2,1,3	0,0	0,1	3,4	-	-	-	0,0	0,1	3,4	-	-	-	0,0	0,1	3,4	0,1	-
g136	3,4,2,2,1,3,4,1	2,2,3,4,4,1,1,3	0,0	0,1	3,4	-	-	-	0,0	0,1	3,4	-	-	-	0,0	0,1	3,4	0,1	-
g141	3,4,1,3,2,2,4,1	1,3,3,4,4,1,2,2	0,0	0,1	3,4	-	-	-	1,1	1,0	4,1	0,0	0,1	3,4	1,0	1,1	2,2	1,0	0,1
g142	3,4,1,3,2,1,4,2	1,3,3,4,4,2,2,1	1,1	1,0	4,2	0,0	0,1	3,4	1,1	1,0	4,2	0,0	0,1	3,4	1,1	1,0	4,2	1,0	0,1
g143	3,4,1,2,2,1,4,3	1,2,3,4,4,3,2,1	1,1	1,0	4,3	0,0	0,1	3,4	1,1	1,0	4,3	0,0	0,1	3,4	1,1	1,0	4,3	1,0	0,1
g144	3,4,1,1,2,2,4,3	1,1,3,4,4,3,2,2	1,1	1,0	4,3	0,0	0,1	3,4	1,1	1,0	4,3	0,0	0,1	3,4	1,0	1,1	2,2	1,0	0,1
g145	3,4,1,1,2,3,4,2	1,1,3,4,4,2,2,3	0,0	0,1	3,4	-	-	-	0,0	0,1	3,4	-	-	-	1,0	1,1	2,3	0,1	-
g146	3,4,1,2,2,3,4,1	1,2,3,4,4,1,2,3	0,0	0,1	3,4	-	-	-	0,0	0,1	3,4	-	-	-	1,0	1,1	2,3	0,1	-

Game	Cartesian	RGS	NE1			NE2			NBS1			NBS2			MaxMin			MC	
Model	Payoff Vector	Payoff Vector	C	RGS	V	C	RGS	V	C	RGS	V	C	RGS	V	C	RGS	V	#1	#2
g151	2,4,1,3,3,2,4,1	1,3,2,4,4,1,3,2	1,0	1,1	3,2	-	-	-	1,1	1,0	4,1	0,0	0,1	2,4	1,0	1,1	3,2	1,0	0,1
g152	2,4,1,3,3,1,4,2	1,3,2,4,4,2,3,1	1,1	1,0	4,2	-	-	-	1,1	1,0	4,2	0,0	0,1	2,4	1,1	1,0	4,2	1,0	0,1
g153	2,4,1,2,3,1,4,3	1,2,2,4,4,3,3,1	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	1,0	-
g154	2,4,1,1,3,2,4,3	1,1,2,4,4,3,3,2	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	1,0	-
g155	2,4,1,1,3,3,4,2	1,1,2,4,4,2,3,3	1,0	1,1	3,3	-	-	-	1,0	1,1	3,3	-	-	-	1,0	1,1	3,3	1,1	-
g156	2,4,1,2,3,3,4,1	1,2,2,4,4,1,3,3	1,0	1,1	3,3	-	-	-	1,0	1,1	3,3	-	-	-	1,0	1,1	3,3	1,1	-
g161	1,4,2,3,3,2,4,1	2,3,1,4,4,1,3,2	1,0	1,1	3,2	-	-	-	1,0	1,1	3,2	-	-	-	1,0	1,1	3,2	1,1	-
g162	1,4,2,3,3,1,4,2	2,3,1,4,4,2,3,1	1,1	1,0	4,2	-	-	-	1,1	1,0	4,2	0,0	0,1	1,4	1,1	1,0	4,2	1,0	0,1
g163	1,4,2,2,3,1,4,3	2,2,1,4,4,3,3,1	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	1,0	-
g164	1,4,2,1,3,2,4,3	2,1,1,4,4,3,3,2	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	-	-	-	1,0	1,1	3,2	1,0	-
g165	1,4,2,1,3,3,4,2	2,1,1,4,4,2,3,3	1,0	1,1	3,3	-	-	-	1,0	1,1	3,3	-	-	-	1,0	1,1	3,3	1,1	-
g166	1,4,2,2,3,3,4,1	2,2,1,4,4,1,3,3	1,0	1,1	3,3	-	-	-	1,0	1,1	3,3	-	-	-	1,0	1,1	3,3	1,1	-
g211	2,4,4,3,1,2,3,1	4,3,2,4,3,1,1,2	0,0	0,1	2,4	-	-	-	0,1	0,0	4,3	-	-	-	0,0	0,1	2,4	0,0	-
g212	2,4,4,3,1,1,3,2	4,3,2,4,3,2,1,1	0,0	0,1	2,4	-	-	-	0,1	0,0	4,3	-	-	-	0,1	0,0	4,3	0,0	-
g213	2,4,4,2,1,1,3,3	4,2,2,4,3,3,1,1	0,0	0,1	2,4	-	-	-	1,1	1,0	3,3	-	-	-	0,1	0,0	4,2	1,0	-
g214	2,4,4,1,1,2,3,3	4,1,2,4,3,3,1,2	0,0	0,1	2,4	-	-	-	1,1	1,0	3,3	-	-	-	0,0	0,1	2,4	1,0	-
g215	2,4,4,1,1,3,3,2	4,1,2,4,3,2,1,3	0,0	0,1	2,4	-	-	-	0,1	0,0	4,1	0,0	0,1	2,4	0,0	0,1	2,4	0,0	0,1
g216	2,4,4,2,1,3,3,1	4,2,2,4,3,1,1,3	0,0	0,1	2,4	-	-	-	0,1	0,0	4,2	0,0	0,1	2,4	0,0	0,1	2,4	0,0	0,1
g221	1,4,4,3,2,2,3,1	4,3,1,4,3,1,2,2	1,0	1,1	2,2	-	-	-	0,1	0,0	4,3	-	-	-	1,0	1,1	2,2	0,0	-
g222	1,4,4,3,2,1,3,2	4,3,1,4,3,2,2,1	-	-	-	-	-	-	0,1	0,0	4,3	-	-	-	1,1	1,0	3,2	0,0	-
g223	1,4,4,2,2,1,3,3	4,2,1,4,3,3,2,1	-	-	-	-	-	-	1,1	1,0	3,3	-	-	-	1,1	1,0	3,3	1,0	-
g224	1,4,4,1,2,2,3,3	4,1,1,4,3,3,2,2	-	-	-	-	-	-	1,1	1,0	3,3	-	-	-	1,0	1,1	2,2	1,0	-
g225	1,4,4,1,2,3,3,2	4,1,1,4,3,2,2,3	1,0	1,1	2,3	-	-	-	1,0	1,1	2,3	-	-	-	1,0	1,1	2,3	1,1	-
g226	1,4,4,2,2,3,3,1	4,2,1,4,3,1,2,3	1,0	1,1	2,3	-	-	-	0,1	0,0	4,2	0,0	0,1	1,4	1,0	1,1	2,3	0,0	0,1
g231	1,4,4,3,3,2,2,1	4,3,1,4,2,1,3,2	1,0	1,1	3,2	-	-	-	0,1	0,0	4,3	-	-	-	1,0	1,1	3,2	0,0	-
g232	1,4,4,3,3,1,2,2	4,3,1,4,2,2,3,1	-	-	-	-	-	-	0,1	0,0	4,3	-	-	-	1,1	1,0	2,2	0,0	-
g233	1,4,4,2,3,1,2,3	4,2,1,4,2,3,3,1	-	-	-	-	-	-	0,1	0,0	4,3	0,0	0,1	1,4	1,1	1,0	2,3	0,0	0,1
g234	1,4,4,1,3,2,2,3	4,1,1,4,2,3,3,2	-	-	-	-	-	-	-	-	-	-	-	-	1,0	1,1	3,2	1,1	-
g235	1,4,4,1,3,3,2,2	4,1,1,4,2,2,3,3	1,0	1,1	3,3	-	-	-	1,0	1,1	3,3	0,0	0,1	1,4	1,0	1,1	3,3	1,1	0,1
g236	1,4,4,2,3,3,2,1	4,2,1,4,2,1,3,3	1,0	1,1	3,3	-	-	-	1,0	1,1	3,3	0,0	0,1	1,4	1,0	1,1	3,3	1,1	0,1
g241	2,4,4,3,3,2,1,1	4,3,2,4,1,1,3,2	1,0	1,1	3,2	-	-	-	0,1	0,0	4,3	1,0	1,1	3,2	0,0	0,1	2,4	0,0	1,1
g242	2,4,4,3,3,1,1,2	4,3,2,4,1,2,3,1	-	-	-	-	-	-	0,1	0,0	4,3	-	-	-	0,1	0,0	4,3	0,0	-

Game	Cartesian	RGS	NE1			NE2			NBS1			NBS2			MaxMin			MC	
Model	Payoff Vector	Payoff Vector	C	RGS	V	C	RGS	V	C	RGS	V	C	RGS	V	C	RGS	V	#1	#2
g243	2,4,4,2,3,1,1,3	4,2,2,4,1,3,3,1	-	-	-	-	-	-	0,1	0,0	4,2	0,0	0,1	2,4	0,1	0,0	4,2	0,0	0,1
g244	2,4,4,1,3,2,1,3	4,1,2,4,1,3,3,2	-	-	-	-	-	-	0,1	0,0	4,1	0,0	0,1	2,4	0,0	0,1	2,4	0,0	0,1
g245	2,4,4,1,3,3,1,2	4,1,2,4,1,2,3,3	1,0	1,1	3,3	-	-	-	1,0	1,1	3,3	0,0	0,1	2,4	0,0	0,1	2,4	1,1	0,1
g246	2,4,4,2,3,3,1,1	4,2,2,4,1,1,3,3	1,0	1,1	3,3	-	-	-	1,0	1,1	3,3	0,0	0,1	2,4	0,0	0,1	2,4	1,1	0,1
g251	3,4,4,3,2,2,1,1	4,3,3,4,1,1,2,2	0,0	0,1	3,4	-	-	-	0,1	0,0	4,3	-	-	-	0,0	0,1	3,4	0,0	-
g252	3,4,4,3,2,1,1,2	4,3,3,4,1,2,2,1	0,0	0,1	3,4	-	-	-	0,1	0,0	4,3	-	-	-	0,1	0,0	4,3	0,0	-
g253	3,4,4,2,2,1,1,3	4,2,3,4,1,3,2,1	0,0	0,1	3,4	-	-	-	0,1	0,0	4,2	0,0	0,1	3,4	0,1	0,0	4,2	0,0	0,1
g254	3,4,4,1,2,2,1,3	4,1,3,4,1,3,2,2	0,0	0,1	3,4	-	-	-	0,1	0,0	4,1	0,0	0,1	3,4	0,0	0,1	3,4	0,0	0,1
g255	3,4,4,1,2,3,1,2	4,1,3,4,1,2,2,3	0,0	0,1	3,4	-	-	-	0,1	0,0	4,1	0,0	0,1	3,4	0,0	0,1	3,4	0,0	0,1
g256	3,4,4,2,2,3,1,1	4,2,3,4,1,1,2,3	0,0	0,1	3,4	-	-	-	0,1	0,0	4,2	0,0	0,1	3,4	0,0	0,1	3,4	0,0	0,1
g261	3,4,4,3,1,2,2,1	4,3,3,4,2,1,1,2	0,0	0,1	3,4	-	-	-	0,1	0,0	4,3	-	-	-	0,0	0,1	3,4	0,0	-
g262	3,4,4,3,1,1,2,2	4,3,3,4,2,2,1,1	0,0	0,1	3,4	-	-	-	0,1	0,0	4,3	-	-	-	0,1	0,0	4,3	0,0	-
g263	3,4,4,2,1,1,2,3	4,2,3,4,2,3,1,1	0,0	0,1	3,4	-	-	-	0,1	0,0	4,2	0,0	0,1	3,4	0,1	0,0	4,2	0,0	0,1
g264	3,4,4,1,1,2,2,3	4,1,3,4,2,3,1,2	0,0	0,1	3,4	-	-	-	0,1	0,0	4,1	0,0	0,1	3,4	0,0	0,1	3,4	0,0	0,1
g265	3,4,4,1,1,3,2,2	4,1,3,4,2,3,1,2	0,0	0,1	3,4	-	-	-	0,1	0,0	4,1	0,0	0,1	3,4	0,0	0,1	3,4	0,0	0,1
g266	3,4,4,2,1,3,2,1	4,2,3,4,2,1,1,3	0,0	0,1	3,4	-	-	-	0,1	0,0	4,2	0,0	0,1	3,4	0,0	0,1	3,4	0,0	0,1
g311	2,3,4,4,1,1,3,2	4,4,2,3,3,2,1,1	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g312	2,3,4,4,1,2,3,1	4,4,2,3,3,1,1,2	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,0	0,1	2,3	0,0	-
g313	2,2,4,4,1,3,3,1	4,4,2,2,3,1,1,3	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,0	0,1	2,2	0,0	-
g314	2,1,4,4,1,3,3,2	4,4,2,1,3,2,1,3	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g315	2,1,4,4,1,2,3,3	4,4,2,1,3,3,1,2	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g316	2,2,4,4,1,2,3,3	4,4,2,2,3,3,1,1	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g321	1,3,4,4,2,1,3,2	4,4,1,3,3,2,2,1	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	1,1	1,0	3,2	0,0	-
g322	1,3,4,4,2,2,3,1	4,4,1,3,3,1,2,2	0,1	0,0	4,4	1,0	1,1	2,2	0,1	0,0	4,4	-	-	-	1,0	1,1	2,2	0,0	-
g323	1,2,4,4,2,3,3,1	4,4,1,2,3,1,2,3	0,1	0,0	4,4	1,0	1,1	2,3	0,1	0,0	4,4	-	-	-	1,0	1,1	2,3	0,0	-
g324	1,1,4,4,2,3,3,2	4,4,1,1,3,2,2,3	0,1	0,0	4,4	1,0	1,1	2,3	0,1	0,0	4,4	-	-	-	1,1	1,0	3,2	0,0	-
g325	1,1,4,4,2,2,3,3	4,4,1,1,3,3,2,2	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	1,1	1,0	3,3	0,0	-
g326	1,2,4,4,2,1,3,3	4,4,1,2,3,3,2,1	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	1,1	1,0	3,3	0,0	-
g331	1,3,4,4,3,1,2,2	4,4,1,3,2,2,3,1	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	1,1	1,0	2,2	0,0	-
g332	1,3,4,4,3,2,2,1	4,4,1,3,2,1,3,2	0,1	0,0	4,4	1,0	1,1	3,2	0,1	0,0	4,4	-	-	-	1,0	1,1	3,2	0,0	-
g333	1,2,4,4,3,3,2,1	4,4,1,2,2,1,3,3	0,1	0,0	4,4	1,0	1,1	3,3	0,1	0,0	4,4	-	-	-	1,0	1,1	3,3	0,0	-
g334	1,1,4,4,3,3,2,2	4,4,1,1,2,2,3,3	0,1	0,0	4,4	1,0	1,1	3,3	0,1	0,0	4,4	-	-	-	1,1	1,0	2,2	0,0	-

Game	Cartesian	RGS	NE1			NE2			NBS1			NBS2			MaxMin			MC	
Model	Payoff Vector	Payoff Vector	C	RGS	V	C	RGS	V	C	RGS	V	C	RGS	V	C	RGS	V	#1	#2
g335	1,1,4,4,3,2,2,3	4,4,1,1,2,3,3,2	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	1,1	1,0	2,3	0,0	-
g336	1,2,4,4,3,1,2,3	4,4,1,2,2,3,3,1	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	1,1	1,0	2,3	0,0	-
g341	2,3,4,4,3,1,1,2	4,4,2,3,1,2,3,1	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g342	2,3,4,4,3,2,2,1	4,4,2,3,1,1,3,2	0,1	0,0	4,4	1,0	1,1	3,2	0,1	0,0	4,4	-	-	-	0,0	0,1	2,3	0,0	-
g343	2,2,4,4,3,3,1,1	4,4,2,2,1,1,3,3	0,1	0,0	4,4	1,0	1,1	3,3	0,1	0,0	4,4	-	-	-	0,0	0,1	2,2	0,0	-
g344	2,1,4,4,3,3,1,2	4,4,2,1,1,2,3,3	0,1	0,0	4,4	1,0	1,1	3,3	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g345	2,1,4,4,3,2,1,3	4,4,2,1,1,3,3,2	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g346	2,2,4,4,3,1,1,3	4,4,2,2,1,3,3,1	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g351	3,3,4,4,2,1,1,2	4,4,3,3,1,2,2,1	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g352	3,3,4,4,2,2,1,1	4,4,3,3,1,1,2,2	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,0	0,1	3,3	0,0	-
g353	3,2,4,4,2,3,1,1	4,4,3,2,1,1,2,3	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,0	0,1	3,2	0,0	-
g354	3,1,4,4,2,3,1,2	4,4,3,1,1,2,2,3	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g355	3,1,4,4,2,2,1,3	4,4,3,1,1,3,2,2	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g356	3,2,4,4,2,1,1,3	4,4,3,2,1,3,2,1	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g361	3,3,4,4,1,1,2,2	4,4,3,3,2,2,1,1	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g362	3,3,4,4,1,2,2,1	4,4,3,3,2,1,1,2	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,0	0,1	3,3	0,0	-
g363	3,2,4,4,1,3,2,1	4,4,3,2,2,1,1,3	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,0	0,1	3,2	0,0	-
g364	3,1,4,4,1,3,2,2	4,4,3,1,2,2,1,3	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g365	3,1,4,4,1,2,2,3	4,4,3,1,2,3,1,2	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g366	3,1,4,4,1,2,2,3	4,4,3,2,2,3,1,1	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	-	-	-	0,1	0,0	4,4	0,0	-
g411	1,3,3,4,2,1,4,2	3,4,1,3,4,2,2,1	1,1	1,0	4,2	-	-	-	0,1	0,0	3,4	-	-	-	1,1	1,0	4,2	0,0	-
g412	1,3,3,4,2,2,4,1	3,4,1,3,4,1,2,2	1,0	1,1	2,2	-	-	-	0,1	0,0	3,4	-	-	-	1,0	1,1	2,2	0,0	-
g413	1,2,3,4,2,3,4,1	3,4,1,2,4,1,2,3	1,0	1,1	2,3	-	-	-	0,1	0,0	3,4	-	-	-	1,0	1,1	2,3	0,0	-
g414	1,1,3,4,2,3,4,2	3,4,1,1,4,2,2,3	1,0	1,1	2,3	-	-	-	0,1	0,0	3,4	-	-	-	1,1	1,0	4,2	0,0	-
g415	1,1,3,4,2,2,4,3	3,4,1,1,4,3,2,2	1,1	1,0	4,3	-	-	-	0,1	0,0	3,4	-	-	-	1,1	1,0	4,3	0,0	-
g416	1,2,3,4,2,1,4,3	3,4,1,2,4,3,2,1	1,1	1,0	4,3	-	-	-	0,1	0,0	3,4	-	-	-	1,1	1,0	4,3	0,0	-
g421	2,3,3,4,1,1,4,2	3,4,2,3,4,2,1,1	1,1	1,0	4,2	-	-	-	0,1	0,0	3,4	-	-	-	0,1	0,0	3,4	0,0	-
g422	2,3,3,4,1,2,4,1	3,4,2,3,4,1,1,2	-	-	-	-	-	-	0,1	0,0	3,4	-	-	-	0,0	0,1	2,3	0,0	-
g423	2,2,3,4,1,3,4,1	3,4,2,2,4,1,1,3	-	-	-	-	-	-	0,1	0,0	3,4	-	-	-	0,0	0,1	2,2	0,0	-
g424	2,1,3,4,1,3,4,2	3,4,2,1,4,2,1,3	-	-	-	-	-	-	0,1	0,0	3,4	-	-	-	0,1	0,0	3,4	0,0	-
g425	2,1,3,4,1,2,4,3	3,4,2,1,4,3,1,2	1,1	1,0	4,3	-	-	-	0,1	0,0	3,4	-	-	-	0,1	0,0	3,4	0,0	-
g426	2,2,3,4,1,1,4,3	3,4,2,2,4,3,1,1	1,1	1,0	4,3	-	-	-	0,1	0,0	3,4	-	-	-	0,1	0,0	3,4	0,0	-

Game	Cartesian	RGS	NE1			NE2			NBS1			NBS2			MaxMin			MC	
Model	Payoff Vector	Payoff Vector	C	RGS	V	C	RGS	V	C	RGS	V	C	RGS	V	C	RGS	V	#1	#2
g431	3,3,2,4,1,1,4,2	2,4,3,3,4,2,1,1	1,1	1,0	4,2	-	-	-	1,1	1,0	4,2	0,0	0,1	3,3	0,1	0,0	2,4	1,0	0,1
g432	3,3,2,4,1,2,4,1	2,4,3,3,4,1,1,2	-	-	-	-	-	-	0,0	0,1	3,3	-	-	-	0,0	0,1	3,3	0,1	-
g433	3,2,2,4,1,3,4,1	2,4,3,2,4,1,1,3	-	-	-	-	-	-	1,1	1,0	4,1	0,1	0,0	2,4	0,0	0,1	3,2	1,0	0,0
g434	3,1,2,4,1,3,4,2	2,4,3,1,4,2,1,3	-	-	-	-	-	-	1,1	1,0	4,2	0,1	0,0	2,4	0,1	0,0	2,4	1,0	0,0
g435	3,1,2,4,1,3,4,2	2,4,3,1,4,3,1,2	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	0,1	0,0	2,4	0,1	0,0	2,4	1,0	0,0
g436	3,2,2,4,1,1,4,3	2,4,3,2,4,3,1,1	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	0,1	0,0	2,4	0,1	0,0	2,4	1,0	0,0
g441	3,3,1,4,2,1,4,2	1,4,3,3,4,2,2,1	1,1	1,0	4,2	-	-	-	1,1	1,0	4,2	0,0	0,1	3,3	1,1	1,0	4,2	1,0	0,1
g442	3,3,1,4,2,2,4,1	1,4,3,3,4,1,2,2	-	-	-	-	-	-	0,1	0,0	3,4	-	-	-	1,0	1,1	2,2	0,0	-
g443	3,2,1,4,2,3,4,1	1,4,3,2,4,1,2,3	-	-	-	-	-	-	-	-	-	-	-	-	1,0	1,1	2,3	1,1	-
g444	3,1,1,4,2,3,4,2	1,4,3,1,4,2,2,3	-	-	-	-	-	-	1,1	1,0	4,2	0,1	0,0	1,4	1,1	1,0	4,2	1,0	0,0
g445	3,1,1,4,2,2,4,3	1,4,3,1,4,3,2,2	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	0,1	0,0	1,4	1,1	1,0	4,3	1,0	0,0
g446	3,2,1,4,2,1,4,3	1,4,3,2,4,3,2,1	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	0,1	0,0	1,4	1,1	1,0	4,3	1,0	0,0
g451	2,3,1,4,3,1,4,2	1,4,2,3,4,2,3,1	1,1	1,0	4,2	-	-	-	1,1	1,0	4,2	0,1	0,0	1,4	1,1	1,0	4,2	1,0	0,0
g452	2,3,1,4,3,2,4,1	1,4,2,3,4,1,3,2	1,0	1,1	3,2	-	-	-	1,0	1,1	3,2	-	-	-	1,0	1,1	3,2	1,1	-
g453	2,2,1,4,3,3,4,1	1,4,2,2,4,1,3,3	1,0	1,1	3,3	-	-	-	1,1	1,0	4,1	1,0	1,1	3,3	1,0	1,1	3,3	1,0	1,1
g454	2,1,1,4,3,3,4,2	1,4,2,1,4,2,3,3	1,0	1,1	3,3	-	-	-	1,1	1,0	4,2	1,0	1,1	3,3	1,1	1,0	4,2	1,0	1,1
g455	2,1,1,4,3,2,4,3	1,4,2,1,4,3,3,2	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	0,1	0,0	1,4	1,1	1,0	4,3	1,0	0,0
g456	2,2,1,4,3,1,4,3	1,4,2,2,4,3,3,1	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	0,1	0,0	1,4	1,1	1,0	4,3	1,0	0,0
g461	1,3,2,4,3,1,4,2	2,4,1,3,4,2,3,1	1,1	1,0	4,2	-	-	-	1,1	1,0	4,2	0,1	0,0	2,4	1,1	1,0	4,2	1,0	0,0
g462	1,3,2,4,3,2,4,1	2,4,1,3,4,1,3,2	1,0	1,1	3,2	-	-	-	1,1	1,0	4,1	0,1	0,0	2,4	1,0	1,1	3,2	1,0	0,0
g463	1,2,2,4,3,3,4,1	2,4,1,2,4,1,3,3	1,0	1,1	3,3	-	-	-	1,1	1,0	4,1	1,0	1,1	3,3	1,0	1,1	3,3	1,0	1,1
g464	1,1,2,4,3,3,4,2	2,4,1,1,4,2,3,3	1,0	1,1	3,3	-	-	-	1,1	1,0	4,2	1,0	1,1	3,3	1,1	1,0	4,2	1,0	1,1
g465	1,1,2,4,3,2,4,3	2,4,1,1,4,3,3,2	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	0,1	0,0	2,4	1,1	1,0	4,3	1,0	0,0
g466	1,2,2,4,3,1,4,3	2,4,1,2,4,3,3,1	1,1	1,0	4,3	-	-	-	1,1	1,0	4,3	0,1	0,0	2,4	1,1	1,0	4,3	1,0	0,0

B.2.3 Tournament Framework Validation

The matches for the six Game-Theoretic algorithms are listed in **Table B.8**. The number of matches in a Game-Theoretic round-robin tournament, over a single game model, is 15. Each match is independent. **Table B.9** summarises results over the game model **g111**. A plot of all matches in the single-model tournament over **g111** is shown in **Figure B.1**.

Table B.8: Multi-Model Tournament Algorithm Set: Game Theoretic. **Match ID** is for reference in-text. **Exp_Match_ID** identifies experimental instance, dataset, and analysis outputs (see Appendix A for details of the agent model software framework and architecture). Exp_ID: exp_398741_sj_0 g111.

Multi-Model Tournament Algorithm Set Game Theoretic			
Match ID	**Exp_Match_ID**	**Agent Zero**	**Agent One**
0	1	allc	alld
1	2	allc	bully_naïve
2	3	allc	fictitious_play
3	4	allc	tft
4	5	allc	random
5	8	alld	bully_naïve
6	9	alld	fictitious_play
7	10	alld	tft
8	11	alld	random
9	15	bully_naïve	fictitious_play
10	16	bully_naïve	tft
11	17	bully_naïve	random
12	22	fictitious_play	tft
13	23	fictitious_play	random
14	29	tft	random

Table B.9: Single-Model Tournament Match Summary: Game Theoretic Match Results. All matches are over single game model **g111**. **MC**, as defined in §4.1.3, is the total number of timesteps that result in an outcome that maps to one of the game model's **MC** locations. Exp_ID: exp_398741_sj_0_g111.

Single-Model Tournament Summary Match Results -- Game Theoretic, g111, 1k				
Agent Zero	**Agent One**	**Reward Agent Zero**	**Reward Agent One**	**MC**
allc	alld	1000	4000	0
allc	bully_naïve	1000	4000	0
allc	fictitious_play	3000	3000	1000
allc	tft	3000	3000	1000
allc	random	2012	3494	506
alld	bully_naïve	3998	1001	0
alld	fictitious_play	2002	1999	0
alld	tft	2002	1999	0
alld	random	2960	1520	0
bully_naïve	fictitious_play	1001	3998	0
bully_naïve	tft	2500	2500	250
bully_naïve	random	2572	2428	256
fictitious_play	tft	3000	3000	1000
fictitious_play	random	1986	3501	489
tft	random	2521	2524	289

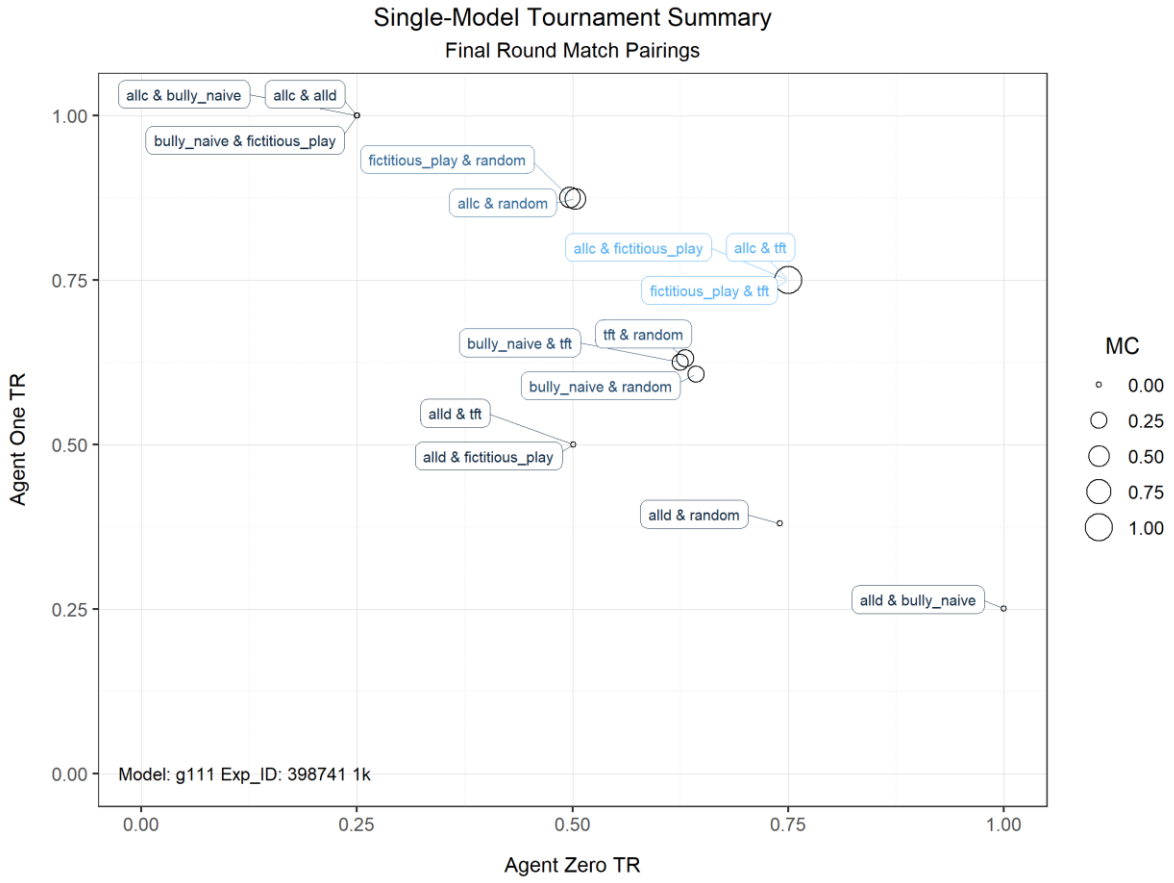


Figure B.1: Single-Model Tournament Summary: All Game-Theoretic Match Pairings, game model **g111**. Axes indicate proportion of available reward (**TR**) gained by each agent. The colour of the labels for each pairing lightens in hue as the amount of mutual cooperation (**MC**) in each pairing increases. This is also indicated by the size of the data point.

The performance of the static—i.e., deterministic automaton—algorithms is as expected. The algorithm *Always Defect* plays only the defection action. The mirror-image algorithm *Always Cooperate* only ever plays the cooperation action. In a match paring these two algorithms, *alld* gains a reward of 4000 and *Always Cooperate* gains a reward of 1000, of a possible maximum of 4000. For every timestep these two algorithms play, the outcome is the same: *Cooperate-Defect* (CD, or 01). Given this, the total amount of mutual cooperation (CC, or 00) in the joint play of these two algorithms, is zero (0).

B.3 Experiment Series Two

B.3.1 Experiment IDs & Analysis Datasheets

Table B.10: Experiment Series Two Experiment IDs.

Experiment Series Two Experiment IDs				
Exp ID	**Algorithm**	**Representation**	**Episodes**	**Timesteps**
180217	Actor/Critic	Scalar	500	1000
133414	Actor/Critic with Eligibility Traces	Scalar	500	1000
133440	Actor/Critic with Replacing Traces	Scalar	500	1000
127288	Q-Learning	Scalar	500	1000
127612	Double Q-Learning	Scalar	500	1000
127633	Expected SARSA	Scalar	500	1000
127812	R Learning	Scalar	500	1000
127850	SARSA	Scalar	500	1000
127910	SARSA Lambda	Scalar	500	1000
128039	SARSA Lambda, with Replacing Traces	Scalar	500	1000
128198	Watkins (naïve) Q, Lambda	Scalar	500	1000
128253	Watkins (naïve) Q, Lambda, Replacing Traces	Scalar	500	1000
128327	Watkins Q, Lambda	Scalar	500	1000
128384	Watkins Q, Linear Function Approximation	Scalar	500	1000
180292	Actor/Critic	Ordinal	500	1000
133454	Actor/Critic with Eligibility Traces	Ordinal	500	1000
133455	Actor/Critic with Replacing Traces	Ordinal	500	1000
129635	Q-Learning	Ordinal	500	1000
129642	Double Q-Learning	Ordinal	500	1000
129656	Expected SARSA	Ordinal	500	1000
129683	R Learning	Ordinal	500	1000
129710	SARSA	Ordinal	500	1000
129718	SARSA Lambda	Ordinal	500	1000
129723	SARSA Lambda, with Replacing Traces	Ordinal	500	1000
129726	Watkins (naïve) Q, Lambda	Ordinal	500	1000
129727	Watkins (naïve) Q, Lambda, Replacing Traces	Ordinal	500	1000
129757	Watkins Q, Lambda	Ordinal	500	1000
129758	Watkins Q, Linear Function Approximation	Ordinal	500	1000
180263	Actor/Critic	Normalised Scalar	500	1000
133442	Actor/Critic with Eligibility Traces	Normalised Scalar	500	1000
133451	Actor/Critic with Replacing Traces	Normalised Scalar	500	1000
132060	Q-Learning	Normalised Scalar	500	1000
132090	Double Q-Learning	Normalised Scalar	500	1000
132114	Expected SARSA	Normalised Scalar	500	1000
132520	R Learning	Normalised Scalar	500	1000
132138	SARSA	Normalised Scalar	500	1000
132161	SARSA Lambda	Normalised Scalar	500	1000
132191	SARSA Lambda, with Replacing Traces	Normalised Scalar	500	1000
132355	Watkins (naïve) Q, Lambda	Normalised Scalar	500	1000
132376	Watkins (naïve) Q, Lambda, Replacing Traces	Normalised Scalar	500	1000
132414	Watkins Q, Lambda	Normalised Scalar	500	1000
132416	Watkins Q, Linear Function Approximation	Normalised Scalar	500	1000
180380	Actor/Critic	Normalised Ordinal	500	1000
133460	Actor/Critic with Eligibility Traces	Normalised Ordinal	500	1000
133462	Actor/Critic with Replacing Traces	Normalised Ordinal	500	1000
133161	Q-Learning	Normalised Ordinal	500	1000
133163	Double Q-Learning	Normalised Ordinal	500	1000
133166	Expected SARSA	Normalised Ordinal	500	1000

Experiment Series Two Experiment IDs				
Exp ID	**Algorithm**	**Representation**	**Episodes**	**Timesteps**
133087	R Learning	Normalised Ordinal	500	1000
133169	SARSA	Normalised Ordinal	500	1000
133170	SARSA Lambda	Normalised Ordinal	500	1000
133173	SARSA Lambda, with Replacing Traces	Normalised Ordinal	500	1000
133177	Watkins (naïve) Q, Lambda	Normalised Ordinal	500	1000
133205	Watkins (naïve) Q, Lambda, Replacing Traces	Normalised Ordinal	500	1000
133211	Watkins Q, Lambda	Normalised Ordinal	500	1000
133215	Watkins Q, Linear Function Approximation	Normalised Ordinal	500	1000

Table B.11: Experiment Series Two Datasheets. Datasheets are packaged with a data release as per Appendix B.5

Experiment Series Two Datasheets / Analysis		
Item	**Filename**	**Description**
1	ES2_equivalence.xlsx	All Wilcoxon test results from R Studio

Table B.12: Experiment Series Two Datasets.

Experiment Series One BasePath="/ES2/${Dir}/${Dir2}"					
Item	**Description**	**Data Type**	**Dir**	**Dir2**	**Size (MB)**
1	experiment data	Raw Episode/ Timestep	exp_data		8750
2	observation data	Orderset, Summaries	obs_data		293
3	pbs output files: .OU, .ER	Text	obs_exp	pbs_output	197
4	exp journal	JSON	obs_exp	journal	53.4
5	R PDF output	Plots and statistical tests	R		10.5

B.3.2 Normality Test Data

Table B.13: Shapiro-Wilk Normality Test, Scalar Transform. Aggregate Outcome.

Algorithm	**Exp ID**	**Shapiro-Wilk**	
		W	**p-value**
Actor/Critic	180217	0.80962	$< 2.2\times10^{-16}$
Actor/Critic with Eligibility Traces	133414	0.78715	$< 2.2\times10^{-16}$
Actor/Critic with Replacing Traces	133440	0.88111	$< 2.2\times10^{-16}$
Q-Learning	127288	0.7517	$< 2.2\times10^{-16}$
Double Q-Learning	127612	0.72745	$< 2.2\times10^{-16}$
Expected SARSA	127633	0.68748	$< 2.2\times10^{-16}$
R Learning	127812	0.74557	$< 2.2\times10^{-16}$
SARSA	127850	0.75012	$< 2.2\times10^{-16}$
SARSA Lambda	127910	0.80065	$< 2.2\times10^{-16}$
SARSA Lambda, with Replacing Traces	128039	0.78611	$< 2.2\times10^{-16}$
Watkins (naive) Q, Lambda	128198	0.85396	$< 2.2\times10^{-16}$
Watkins (naive) Q, Lambda with Replacing Traces	128253	0.78252	$< 2.2\times10^{-16}$
Watkins Q, Lambda	128327	0.81204	$< 2.2\times10^{-16}$
Watkins Q, Linear Function Approximation	128384	0.95401	7.69×10^{-10}

Table B.14: Shapiro-Wilk Normality Test, Scalar Transform. CC Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180217	0.78361	8.01×10^{-11}
Actor/Critic with Eligibility Traces	133414	0.91779	1.08×10^{-05}
Actor/Critic with Replacing Traces	133440	0.92497	2.61×10^{-05}
Q-Learning	127288	0.81108	5.50×10^{-10}
Double Q-Learning	127612	0.78597	9.39×10^{-11}
Expected SARSA	127633	0.7701	3.30×10^{-11}
R Learning	127812	0.88801	4.02×10^{-07}
SARSA	127850	0.78906	1.16×10^{-10}
SARSA Lambda	127910	0.85967	2.75×10^{-08}
SARSA Lambda, with Replacing Traces	128039	0.82949	2.22×10^{-09}
Watkins (naive) Q, Lambda	128198	0.88697	3.62×10^{-07}
Watkins (naive) Q, Lambda with Replacing Traces	128253	0.86046	2.95×10^{-08}
Watkins Q, Lambda	128327	0.90162	1.69×10^{-06}
Watkins Q, Linear Function Approximation	128384	0.73861	4.73×10^{-12}

Table B.15: Shapiro-Wilk Normality Test, Scalar Transform. CD Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180217	0.91797	1.1×10^{-05}
Actor/Critic with Eligibility Traces	133414	0.86837	6.03×10^{-08}
Actor/Critic with Replacing Traces	133440	0.88403	2.70×10^{-07}
Q-Learning	127288	0.90829	3.54×10^{-06}
Double Q-Learning	127612	0.88389	2.66×10^{-07}
Expected SARSA	127633	0.77689	5.13×10^{-11}
R Learning	127812	0.92694	3.35×10^{-05}
SARSA	127850	0.88244	2.31×10^{-07}
SARSA Lambda	127910	0.87434	1.05×10^{-07}
SARSA Lambda, with Replacing Traces	128039	0.81523	7.48×10^{-10}
Watkins (naive) Q, Lambda	128198	0.86642	5.04×10^{-08}
Watkins (naive) Q, Lambda with Replacing Traces	128253	0.90184	1.73×10^{-06}
Watkins Q, Lambda	128327	0.80338	3.15×10^{-10}
Watkins Q, Linear Function Approximation	128384	0.79932	2.36×10^{-10}

Table B.16: Shapiro-Wilk Normality Test, Scalar Transform. DC Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180217	0.9232	2.09×10^{-05}
Actor/Critic with Eligibility Traces	133414	0.88167	2.14×10^{-07}
Actor/Critic with Replacing Traces	133440	0.8884	4.19×10^{-07}
Q-Learning	127288	0.90448	2.31×10^{-06}
Double Q-Learning	127612	0.88371	2.61×10^{-07}
Expected SARSA	127633	0.75682	1.42×10^{-11}
R Learning	127812	0.9134	6.39×10^{-06}
SARSA	127850	0.88405	2.71×10^{-07}
SARSA Lambda	127910	0.87513	1.14×10^{-07}
SARSA Lambda, with Replacing Traces	128039	0.82008	1.08×10^{-09}
Watkins (naive) Q, Lambda	128198	0.87024	7.17×10^{-08}
Watkins (naive) Q, Lambda with Replacing Traces	128253	0.9061	2.77×10^{-06}
Watkins Q, Lambda	128327	0.80085	2.63×10^{-10}
Watkins Q, Linear Function Approximation	128384	0.85936	2.67×10^{-08}

Table B.17: Shapiro-Wilk Normality Test, Scalar Transform. DD Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180217	0.91751	1.04×10^{-05}
Actor/Critic with Eligibility Traces	133414	0.92275	1.98×10^{-05}
Actor/Critic with Replacing Traces	133440	0.92448	2.45×10^{-05}
Q-Learning	127288	0.86937	6.61×10^{-08}
Double Q-Learning	127612	0.83178	2.66×10^{-09}
Expected SARSA	127633	0.79496	1.74×10^{-10}
R Learning	127812	0.92431	2.40×10^{-05}
SARSA	127850	0.87463	1.08×10^{-07}
SARSA Lambda	127910	0.89227	6.23×10^{-07}
SARSA Lambda, with Replacing Traces	128039	0.90506	2.47×10^{-06}
Watkins (naive) Q, Lambda	128198	0.96088	.004
Watkins (naive) Q, Lambda with Replacing Traces	128253	0.90503	2.46×10^{-06}
Watkins Q, Lambda	128327	0.93885	1.64×10^{-04}
Watkins Q, Linear Function Approximation	128384	0.91015	4.38×10^{-06}

Table B.18: Shapiro-Wilk Normality Test, Ordinal Transform. Aggregate Outcome

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180292	0.83301	$< 2.2\times10^{-16}$
Actor/Critic with Eligibility Traces	133454	0.79735	$< 2.2\times10^{-16}$
Actor/Critic with Replacing Traces	133455	0.8268	$< 2.2\times10^{-16}$
Q-Learning	129635	0.74157	$< 2.2\times10^{-16}$
Double Q-Learning	129642	0.75569	$< 2.2\times10^{-16}$
Expected SARSA	129656	0.77098	$< 2.2\times10^{-16}$
R Learning	129683	0.69208	$< 2.2\times10^{-16}$
SARSA	129710	0.75139	$< 2.2\times10^{-16}$
SARSA Lambda	129718	0.79987	$< 2.2\times10^{-16}$
SARSA Lambda, with Replacing Traces	129723	0.77676	$< 2.2\times10^{-16}$
Watkins (naive) Q, Lambda	129726	0.66681	$< 2.2\times10^{-16}$
Watkins (naive) Q, Lambda with Replacing Traces	129727	0.73427	$< 2.2\times10^{-16}$
Watkins Q, Lambda	129757	0.79114	$< 2.2\times10^{-16}$
Watkins Q, Linear Function Approximation	129758	0.97079	3.53×10^{-07}

Table B.19: Shapiro-Wilk Normality Test, Ordinal Transform. CC Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180292	0.74962	9.15×10^{-12}
Actor/Critic with Eligibility Traces	133454	0.92164	1.72×10^{-05}
Actor/Critic with Replacing Traces	133455	0.89026	5.07×10^{-07}
Q-Learning	129635	0.83001	2.32×10^{-09}
Double Q-Learning	129642	0.84165	5.90×10^{-09}
Expected SARSA	129656	0.8317	2.65×10^{-09}
R Learning	129683	0.842	6.07×10^{-09}
SARSA	129710	0.82407	1.46×10^{-09}
SARSA Lambda	129718	0.87927	1.69×10^{-07}
SARSA Lambda, with Replacing Traces	129723	0.84533	8.00×10^{-09}
Watkins (naive) Q, Lambda	129726	0.58076	1.88×10^{-15}
Watkins (naive) Q, Lambda with Replacing Traces	129727	0.86378	3.97×10^{-08}
Watkins Q, Lambda	129757	0.88626	3.37×10^{-11}
Watkins Q, Linear Function Approximation	129758	0.77158	3.63×10^{-11}

Table B.20: Shapiro-Wilk Normality Test, Ordinal Transform. CD Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180292	0.86895	6.36×10^{-08}
Actor/Critic with Eligibility Traces	133454	0.75909	1.64×10^{-11}
Actor/Critic with Replacing Traces	133455	0.88481	2.92×10^{-07}
Q-Learning	129635	0.81651	8.22×10^{-10}
Double Q-Learning	129642	0.85147	1.35×10^{-08}
Expected SARSA	129656	0.85121	1.32×10^{-08}
R Learning	129683	0.85586	1.96×10^{-08}
SARSA	129710	0.81617	8.02×10^{-10}
SARSA Lambda	129718	0.72506	2.16×10^{-12}
SARSA Lambda, with Replacing Traces	129723	0.71418	1.17×10^{-12}
Watkins (naive) Q, Lambda	129726	0.54811	4.92×10^{-16}
Watkins (naive) Q, Lambda with Replacing Traces	129727	0.80036	2.54×10^{-10}
Watkins Q, Lambda	129757	0.73191	3.20×10^{-12}
Watkins Q, Linear Function Approximation	129758	0.81363	6.64×10^{-10}

Table B.21: Shapiro-Wilk Normality Test, Ordinal Transform. DC Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180292	0.88574	3.20×10^{-07}
Actor/Critic with Eligibility Traces	133454	0.75881	1.61×10^{-11}
Actor/Critic with Replacing Traces	133455	0.87884	1.62×10^{-07}
Q-Learning	129635	0.83233	2.78×10^{-09}
Double Q-Learning	129642	0.88296	2.43×10^{-07}
Expected SARSA	129656	0.86087	3.06×10^{-08}
R Learning	129683	0.85609	2.01×10^{-08}
SARSA	129710	0.82742	1.89×10^{-09}
SARSA Lambda	129718	0.7232	1.94×10^{-12}
SARSA Lambda, with Replacing Traces	129723	0.71501	1.22×10^{-12}
Watkins (naive) Q, Lambda	129726	0.54914	5.13×10^{-16}
Watkins (naive) Q, Lambda with Replacing Traces	129727	0.80698	4.08×10^{-10}
Watkins Q, Lambda	129757	0.72448	2.09×10^{-12}
Watkins Q, Linear Function Approximation	129758	0.88318	2.48×10^{-07}

Table B.22: Shapiro-Wilk Normality Test, Ordinal Transform. DD Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180292	0.85307	1.54×10^{-08}
Actor/Critic with Eligibility Traces	133454	0.88378	2.63×10^{-07}
Actor/Critic with Replacing Traces	133455	0.92766	3.68×10^{-05}
Q-Learning	129635	0.84969	1.16×10^{-08}
Double Q-Learning	129642	0.85779	2.33×10^{-08}
Expected SARSA	129656	0.87861	1.59×10^{-07}
R Learning	129683	0.86936	6.61×10^{-08}
SARSA	129710	0.87314	9.40×10^{-08}
SARSA Lambda	129718	0.9244	2.43×10^{-05}
SARSA Lambda, with Replacing Traces	129723	0.86666	5.15×10^{-08}
Watkins (naive) Q, Lambda	129726	0.68956	3.10×10^{-13}
Watkins (naive) Q, Lambda with Replacing Traces	129727	0.8473	9.44×10^{-09}
Watkins Q, Lambda	129757	0.91401	6.86×10^{-06}
Watkins Q, Linear Function Approximation	129758	0.92416	2.36×10^{-05}

Table B.23: Shapiro-Wilk Normality Test, Normalised Scalar Transform. Aggregate Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180263	0.63534	$< 2.2\times10^{-16}$
Actor/Critic with Eligibility Traces	133442	0.69701	$< 2.2\times10^{-16}$
Actor/Critic with Replacing Traces	133451	0.78516	$< 2.2\times10^{-16}$
Q-Learning	132060	0.75227	$< 2.2\times10^{-16}$
Double Q-Learning	132090	0.72783	$< 2.2\times10^{-16}$
Expected SARSA	132114	0.69119	$< 2.2\times10^{-16}$
R Learning	132520	0.74555	$< 2.2\times10^{-16}$
SARSA	132138	0.75008	$< 2.2\times10^{-16}$
SARSA Lambda	132161	0.80904	$< 2.2\times10^{-16}$
SARSA Lambda, with Replacing Traces	132191	0.78693	$< 2.2\times10^{-16}$
Watkins (naive) Q, Lambda	132355	0.67434	$< 2.2\times10^{-16}$
Watkins (naive) Q, Lambda with Replacing Traces	132376	0.74143	$< 2.2\times10^{-16}$
Watkins Q, Lambda	132414	0.81226	$< 2.2\times10^{-16}$
Watkins Q, Linear Function Approximation	132416	0.95425	8.32×10^{-10}

Table B.24: Shapiro-Wilk Normality Test, Normalised Scalar Transform. CC Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180263	0.74022	5.20×10^{-12}
Actor/Critic with Eligibility Traces	133442	0.80929	4.82×10^{-10}
Actor/Critic with Replacing Traces	133451	0.79237	1.45×10^{-10}
Q-Learning	132060	0.80918	4.79×10^{-10}
Double Q-Learning	132090	0.78697	1.01×10^{-10}
Expected SARSA	132114	0.77067	3.42×10^{-11}
R Learning	132520	0.88836	4.17×10^{-07}
SARSA	132138	0.78977	1.22×10^{-10}
SARSA Lambda	132161	0.89397	7.44×10^{-07}
SARSA Lambda, with Replacing Traces	132191	0.82929	2.19×10^{-09}
Watkins (naive) Q, Lambda	132355	0.64069	2.73×10^{-14}
Watkins (naive) Q, Lambda with Replacing Traces	132376	0.85617	2.02×10^{-08}
Watkins Q, Lambda	132414	0.90138	1.64×10^{-06}
Watkins Q, Linear Function Approximation	132416	0.73034	2.92×10^{-12}

Table B.25: Shapiro-Wilk Normality Test, Normalised Scalar Transform. CD Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180263	0.75968	1.40×10^{-11}
Actor/Critic with Eligibility Traces	133442	0.64772	3.82×10^{-14}
Actor/Critic with Replacing Traces	133451	0.85295	1.53×10^{-08}
Q-Learning	132060	0.90679	2.99×10^{-06}
Double Q-Learning	132090	0.88486	2.93×10^{-07}
Expected SARSA	132114	0.7911	1.33×10^{-10}
R Learning	132520	0.92624	3.07×10^{-05}
SARSA	132138	0.88823	4.11×10^{-07}
SARSA Lambda	132161	0.78144	6.93×10^{-11}
SARSA Lambda, with Replacing Traces	132191	0.82059	1.12×10^{-09}
Watkins (naive) Q, Lambda	132355	0.57428	1.43×10^{-15}
Watkins (naive) Q, Lambda with Replacing Traces	132376	0.86993	6.97×10^{-08}
Watkins Q, Lambda	132414	0.80282	3.03×10^{-10}
Watkins Q, Linear Function Approximation	132416	0.80164	2.78×10^{-02}

Table B.26: Shapiro-Wilk Normality Test, Normalised Scalar Transform. DC Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180263	0.76476	2.34×10^{-11}
Actor/Critic with Eligibility Traces	133442	0.64952	4.16×10^{-14}
Actor/Critic with Replacing Traces	133451	0.84813	1.01×10^{-08}
Q-Learning	132060	0.90939	4.02×10^{-06}
Double Q-Learning	132090	0.89174	5.90×10^{-07}
Expected SARSA	132114	0.80226	2.91×10^{-10}
R Learning	132520	0.91352	6.48×10^{-06}
SARSA	132138	0.88587	3.24×10^{-07}
SARSA Lambda	132161	0.77666	5.05×10^{-11}
SARSA Lambda, with Replacing Traces	132191	0.81899	9.91×10^{-10}
Watkins (naive) Q, Lambda	132355	0.57318	1.37×10^{-15}
Watkins (naive) Q, Lambda with Replacing Traces	132376	0.87336	9.60×10^{-08}
Watkins Q, Lambda	132414	0.8058	3.75×10^{-10}
Watkins Q, Linear Function Approximation	132416	0.86635	5.01×10^{-08}

Table B.27: Shapiro-Wilk Normality Test, Normalised Scalar Transform. DD Outcome

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180263	0.77707	5.19×10^{-11}
Actor/Critic with Eligibility Traces	133442	0.77467	4.43×10^{-11}
Actor/Critic with Replacing Traces	133451	0.8316	2.63×10^{-09}
Q-Learning	132060	0.87095	7.66×10^{-08}
Double Q-Learning	132090	0.83025	2.36×10^{-09}
Expected SARSA	132114	0.79572	1.83×10^{-10}
R Learning	132520	0.92336	2.13×10^{-05}
SARSA	132138	0.87353	9.76×10^{-08}
SARSA Lambda	132161	0.92792	3.80×10^{-05}
SARSA Lambda, with Replacing Traces	132191	0.90543	2.57×10^{-06}
Watkins (naive) Q, Lambda	132355	0.75515	1.28×10^{-11}
Watkins (naive) Q, Lambda with Replacing Traces	132376	0.87061	7.42×10^{-08}
Watkins Q, Lambda	132414	0.93978	1.86×10^{-04}
Watkins Q, Linear Function Approximation	132416	0.91343	6.41×10^{-06}

Table B.28: Shapiro-Wilk Normality Test, Normalised Ordinal Transform. Aggregate Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180380	0.62532	$< 2.2\times10^{-16}$
Actor/Critic with Eligibility Traces	133460	0.69426	$< 2.2\times10^{-16}$
Actor/Critic with Replacing Traces	133462	0.73848	$< 2.2\times10^{-16}$
Q-Learning	133161	0.72104	$< 2.2\times10^{-16}$
Double Q-Learning	133163	0.71126	$< 2.2\times10^{-16}$
Expected SARSA	133166	0.67669	$< 2.2\times10^{-16}$
R Learning	133087	0.67962	$< 2.2\times10^{-16}$
SARSA	133169	0.72547	$< 2.2\times10^{-16}$
SARSA Lambda	133170	0.80399	$< 2.2\times10^{-16}$
SARSA Lambda, with Replacing Traces	133173	0.7735	$< 2.2\times10^{-16}$
Watkins (naive) Q, Lambda	133177	0.67517	$< 2.2\times10^{-16}$
Watkins (naive) Q, Lambda with Replacing Traces	133205	0.7232	$< 2.2\times10^{-16}$
Watkins Q, Lambda	133211	0.79992	$< 2.2\times10^{-16}$
Watkins Q, Linear Function Approximation	133215	0.96486	3.33×10^{-08}

Table B.29: Shapiro-Wilk Normality Test, Normalised Ordinal Transform. CC Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180380	0.77955	6.11×10^{-11}
Actor/Critic with Eligibility Traces	133460	0.81312	6.39×10^{-10}
Actor/Critic with Replacing Traces	133462	0.77629	4.93×10^{-11}
Q-Learning	133161	0.79224	1.44×10^{-10}
Double Q-Learning	133163	0.78802	1.08×10^{-10}
Expected SARSA	133166	0.74248	5.95×10^{-12}
R Learning	133087	0.83694	4.02×10^{-09}
SARSA	133169	0.78054	6.52×10^{-11}
SARSA Lambda	133170	0.89322	6.88×10^{-07}
SARSA Lambda, with Replacing Traces	133173	0.82401	1.45×10^{-09}
Watkins (naive) Q, Lambda	133177	0.64589	3.50×10^{-14}
Watkins (naive) Q, Lambda with Replacing Traces	133205	0.84969	1.16×10^{-08}
Watkins Q, Lambda	133211	0.89691	1.02×10^{-06}
Watkins Q, Linear Function Approximation	133215	0.74608	7.39×10^{-12}

Table B.30: Shapiro-Wilk Normality Test, Normalised Ordinal Transform. CD Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180380	0.61081	6.94×10^{-15}
Actor/Critic with Eligibility Traces	133460	0.64656	3.61×10^{-14}
Actor/Critic with Replacing Traces	133462	0.75037	9.58×10^{-12}
Q-Learning	133161	0.82376	1.42×10^{-09}
Double Q-Learning	133163	0.84062	5.42×10^{-09}
Expected SARSA	133166	0.8232	1.37×10^{-09}
R Learning	133087	0.84038	5.32×10^{-09}
SARSA	133169	0.7941	1.64×10^{-10}
SARSA Lambda	133170	0.74914	8.89×10^{-12}
SARSA Lambda, with Replacing Traces	133173	0.7768	5.10×10^{-11}
Watkins (naive) Q, Lambda	133177	0.55514	6.53×10^{-16}
Watkins (naive) Q, Lambda with Replacing Traces	133205	0.82559	1.64×10^{-09}
Watkins Q, Lambda	133211	0.75693	1.43×10^{-11}
Watkins Q, Linear Function Approximation	133215	0.8073	4.18×10^{-10}

Table B.31: Shapiro-Wilk Normality Test, Normalised Ordinal Transform. DC Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180380	0.62083	1.09×10^{-14}
Actor/Critic with Eligibility Traces	133460	0.65266	4.84×10^{-14}
Actor/Critic with Replacing Traces	133462	0.7553	1.30×10^{-11}
Q-Learning	133161	0.8237	1.42×10^{-09}
Double Q-Learning	133163	0.8351	3.47×10^{-09}
Expected SARSA	133166	0.83363	3.08×10^{-09}
R Learning	133087	0.84065	5.44×10^{-09}
SARSA	133169	0.7844	8.45×10^{-11}
SARSA Lambda	133170	0.75221	1.07×10^{-11}
SARSA Lambda, with Replacing Traces	133173	0.77525	4.60×10^{-11}
Watkins (naive) Q, Lambda	133177	0.55269	5.91×10^{-16}
Watkins (naive) Q, Lambda with Replacing Traces	133205	0.81366	6.65×10^{-10}
Watkins Q, Lambda	133211	0.75405	1.20×10^{-11}
Watkins Q, Linear Function Approximation	133215	0.88426	2.76×10^{-07}

Table B.32: Shapiro-Wilk Normality Test, Normalised Ordinal Transform. DD Outcome.

Algorithm	Exp ID	Shapiro-Wilk	
		W	p-value
Actor/Critic	180380	0.74911	8.87×10^{-12}
Actor/Critic with Eligibility Traces	133460	0.78292	7.65×10^{-11}
Actor/Critic with Replacing Traces	133462	0.77783	5.45×10^{-11}
Q-Learning	133161	0.81324	6.45×10^{-10}
Double Q-Learning	133163	0.80562	3.70×10^{-10}
Expected SARSA	133166	0.75807	1.54×10^{-11}
R Learning	133087	0.84079	5.50×10^{-09}
SARSA	133169	0.82356	1.40×10^{-09}
SARSA Lambda	133170	0.92275	1.98×10^{-05}
SARSA Lambda, with Replacing Traces	133173	0.89123	5.60×10^{-07}
Watkins (naive) Q, Lambda	133177	0.74594	7.32×10^{-12}
Watkins (naive) Q, Lambda with Replacing Traces	133205	0.85714	2.20×10^{-08}
Watkins Q, Lambda	133211	0.92729	3.51×10^{-05}
Watkins Q, Linear Function Approximation	133215	0.91958	1.34×10^{-05}

B.3.3 Wilcoxon Signed Rank Test

Table B.33: Experiment Group One: Scalar ~ Ordinal Wilcoxon, aggregated outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Ordinal	V	p-value	CI L	CI U
Actor/Critic	180217	180292	35742	.071	-0.017597941	0.000633154
Actor/Critic, ET	133414	133454	40978	.705	-0.00362061	0.004277553
Actor/Critic, RT	133440	133455	45337	.018	0.001938575	0.020033241
Q-Learning	127288	129635	46086	.01	0.0009571576	0.0060980848
Double Q-Learning	127612	129642	34431	.014	-0.0059096451	-0.0006184118
Expected SARSA	127633	129656	27735	9.08×10^{-08}	-0.018479492	-0.006970142
R Learning	127812	129683	45520	.019	0.002401994	0.017690573
SARSA	127850	129710	46848	.004	0.001172529	0.005303643
SARSA Lambda	127910	129718	49435	5.47×10^{-05}	0.01710091	0.04110439
SARSA Lambda, RT	128039	129723	45731	.015	0.0005421006	0.005718562
Watkins (naive) Q, Lambda	128198	129726	50043	1.73×10^{-05}	0.02699262	0.05050426
Watkins (naive) Q, Lambda, RT	128253	129727	49320	6.76×10^{-05}	0.01932345	0.04874353
Watkins Q, Lambda	128327	129757	49355	6.35×10^{-05}	0.002871179	0.008939323
Watkins Q, LFA	128384	129758	38262	.478	-0.0012568376	0.0005626993

Table B.34: Experiment Group One: Scalar ~ Ordinal Wilcoxon, CC outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Ordinal	V	p-value	CI L	CI U
Actor/Critic	180217	180292	3165	.028	0.001046142	0.017499989
Actor/Critic, ET	133414	133454	2530	.988	-0.01172751	0.01096752
Actor/Critic, RT	133440	133455	3389	.001	0.005885722	0.026827863
Q-Learning	127288	129635	139	2.36×10^{-16}	-0.04049645	-0.0234118
Double Q-Learning	127612	129642	71	$< 2.2\times10^{-16}$	-0.04772838	-0.03115371
Expected SARSA	127633	129656	36	$< 2.2\times10^{-16}$	-0.06015317	-0.03623857
R Learning	127812	129683	3876	3.43×10^{-6}	0.005112172	0.010322807
SARSA	127850	129710	42	$< 2.2\times10^{-16}$	-0.04476904	-0.02646374
SARSA Lambda	127910	129718	70	$< 2.2\times10^{-16}$	-0.4302922	-0.3211401
SARSA Lambda, RT	128039	129723	469	1.58×10^{-12}	-0.06610901	-0.03297688
Watkins (naive) Q, Lambda	128198	129726	52	$< 2.2\times10^{-16}$	-0.4622213	-0.3523123
Watkins (naive) Q, Lambda, RT	128253	129727	12	$< 2.2\times10^{-16}$	-0.4472585	-0.3470883
Watkins Q, Lambda	128327	129757	182	8.00×10^{-16}	-0.08929644	-0.05713452
Watkins Q, LFA	128384	129758	674	3.29×10^{-16}	-0.009650418	-0.005429281

Table B.35: Experiment Group One: Scalar ~ Ordinal Wilcoxon, CD outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Ordinal	V	p-value	CI L	CI U
Actor/Critic	180217	180292	142	2.57×10^{-16}	-0.08427898	-0.06201411
Actor/Critic, ET	133414	133454	2353	.555	-0.019415312	0.002627167
Actor/Critic, RT	133440	133455	3843	5.90×10^{-6}	0.02425489	0.05497188
Q-Learning	127288	129635	4759	1.6×10^{-14}	0.01087203	0.01635679
Double Q-Learning	127612	129642	1588.5	.001	-0.008437723	-0.001569422
Expected SARSA	127633	129656	457	1.17×10^{-12}	-0.05398504	-0.01853972
R Learning	127812	129683	5050	$< 2.2\times10^{-16}$	0.0357288	0.04071659
SARSA	127850	129710	4579	1.66×10^{-12}	0.006620799	0.010523249
SARSA Lambda	127910	129718	4644	3.24×10^{-13}	0.03279173	0.05428426
SARSA Lambda, RT	128039	129723	3637.5	.0001316	0.002117421	0.011029979
Watkins (naive) Q, Lambda	128198	129726	4877	6.21×10^{-16}	0.04521073	0.0667187
Watkins (naive) Q, Lambda, RT	128253	129727	4915	$< 2.2\times10^{-16}$	0.046612	0.06892483
Watkins Q, Lambda	128327	129757	4515	7.89×10^{-12}	0.006092654	0.012466797
Watkins Q, LFA	128384	129758	1273.5	1.70×10^{-05}	-0.004028236	-0.001343551

Table B.36: Experiment Group One: Scalar ~ Ordinal Wilcoxon, DC outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Ordinal	V	p-value	CI L	CI U
Actor/Critic	180217	180292	138.5	2.33×10^{-16}	-0.08329102	-0.06112769
Actor/Critic, ET	133414	133454	2305	.450	-0.019853802	0.002777826
Actor/Critic, RT	133440	133455	3878	3.31×10^{-6}	0.02559451	0.05659649
Q-Learning	127288	129635	4683	1.19×10^{-13}	0.009981403	0.015328663
Double Q-Learning	127612	129642	1858	.022	-0.0069959579	-0.0004933201
Expected SARSA	127633	129656	542	9.33×10^{-12}	-0.05218497	-0.01829719
R Learning	127812	129683	5050	$< 2.2\times10^{-16}$	0.03633872	0.04152425
SARSA	127850	129710	4630	4.62×10^{-13}	0.007269037	0.01053125
SARSA Lambda	127910	129718	4612	7.28×10^{-13}	0.03309483	0.0543581
SARSA Lambda, RT	128039	129723	3390	.003	0.001241412	0.008979833
Watkins (naive) Q, Lambda	128198	129726	4901	3.14×10^{-16}	0.04574427	0.06671179
Watkins (naive) Q, Lambda, RT	128253	129727	4923	$< 2.2\times10^{-16}$	0.04644912	0.06821251
Watkins Q, Lambda	128327	129757	4415	8.21×10^{-11}	0.005442951	0.012466836
Watkins Q, LFA	128384	129758	4801.5	5.05×10^{-15}	0.006533091	0.013194777

Table B.37: Experiment Group One: Scalar ~ Ordinal Wilcoxon, DD outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Ordinal	V	p-value	CI L	CI U
Actor/Critic	180217	180292	4912	$< 2.2\times10^{-16}$	0.1157966	0.1587742
Actor/Critic, ET	133414	133454	3341	.005	0.00629348	0.03957437
Actor/Critic, RT	133440	133455	928	4.04×10^{-08}	-0.12436826	-0.06090153
Q-Learning	127288	129635	3176	.025	0.0008038397	0.0141740182
Double Q-Learning	127612	129642	4828	2.44×10^{-15}	0.04012704	0.06254666
Expected SARSA	127633	129656	4938	$< 2.2\times10^{-16}$	0.08183045	0.15898237
R Learning	127812	129683	9	$< 2.2\times10^{-16}$	-0.09081959	-0.07631891
SARSA	127850	129710	4125	3.81×10^{-08}	0.0115374	0.02533338
SARSA Lambda	127910	129718	4987	$< 2.2\times10^{-16}$	0.2350775	0.314678
SARSA Lambda, RT	128039	129723	4318	7.13×10^{-10}	0.03442807	0.06392223
Watkins (naive) Q, Lambda	128198	129726	5010	$< 2.2\times10^{-16}$	0.2442516	0.3253624
Watkins (naive) Q, Lambda, RT	128253	129727	4947	$< 2.2\times10^{-16}$	0.2304141	0.3113523
Watkins Q, Lambda	128327	129757	4794	6.20×10^{-15}	0.04167659	0.06326124
Watkins Q, LFA	128384	129758	2855	.257	-0.0007568552	0.0020948665

Table B.38: Experiment Two: Scalar ~ Scalar-Normalised Wilcoxon, Aggregate Outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Scalar Norm	V	p-value	CI L	CI U
Actor/Critic	180217	180263	41239	.623	-0.00485057	0.00743134
Actor/Critic, ET	133414	133442	46744	.004	0.00264998	0.01195780
Actor/Critic, RT	133440	133451	39599	.829	-0.01095706	0.00908854
Q-Learning	127288	132060	40091	.997	-0.00037674	0.00036412
Double Q-Learning	127612	132090	39339	.742	-0.00049893	0.00035731
Expected SARSA	127633	132114	37225	.214	-0.00063368	0.00013842
R Learning	127812	132520	39866	.92	-0.0004254	0.00037326
SARSA	127850	132138	40180	.904	-0.00033934	0.00038647
SARSA Lambda	127910	132161	49945	2.09×10^{-05}	0.01235475	0.03082885
SARSA Lambda, RT	128039	132191	39177	.690	-0.00048423	0.00031857
Watkins (naive) Q, Lambda	128198	132355	49853	2.50×10^{-05}	0.02731523	0.04882586
Watkins (naive) Q, Lambda, RT	128253	132376	48331	2.55×10^{-05}	0.0119681	0.03821075
Watkins Q, Lambda	128327	132414	39692	.928	-0.00046913	0.00043413
Watkins Q, LFA	128384	132416	39939	.945	-0.00039688	0.0003856

Table B.39: Experiment Group Two: Scalar ~ Scalar-Normalised Wilcoxon, CC outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Scalar Norm	V	p-value	CI L	CI U
Actor/Critic	180217	180263	5050	$< 2.2\times10^{-16}$	0.1460644	0.1867654
Actor/Critic, ET	133414	133442	4962	$< 2.2\times10^{-16}$	0.06135133	0.09392006
Actor/Critic, RT	133440	133451	4664	1.94×10^{-13}	0.06701938	0.1376999
Q-Learning	127288	132060	2744	.453	-0.00057568	0.0014793656
Double Q-Learning	127612	132090	2557.5	.912	-0.00090247	0.001031481
Expected SARSA	127633	132114	3935.5	1.25×10^{-06}	0.00149058	0.005940277
R Learning	127812	132520	2413	.701	-0.00057185	0.0004191055
SARSA	127850	132138	2163.5	.278	-0.00144965	0.0004194087
SARSA Lambda	127910	132161	80	$< 2.2\times10^{-16}$	-0.3471416	-0.2605388
SARSA Lambda, RT	128039	132191	2702	.544	-0.00094816	0.0015716208
Watkins (naive) Q, Lambda	128198	132355	54	$< 2.2\times10^{-16}$	-0.4387156	-0.3318402
Watkins (naive) Q, Lambda, RT	128253	132376	132	2.93×10^{-16}	-0.340838	-0.2628455
Watkins Q, Lambda	128327	132414	2284	.408	-0.00251059	0.001062267
Watkins Q, LFA	128384	132416	2713.5	.518	-0.00057447	0.0010860883

Table B.40: Experiment Group Two: Scalar ~ Scalar-Normalised Wilcoxon, CD outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Scalar Norm	V	p-value	CI L	CI U
Actor/Critic	180217	180263	2350	.549	-0.00621418	0.00344177
Actor/Critic, ET	133414	133442	3444	.002	0.003816918	0.01353804
Actor/Critic, RT	133440	133451	2385	.632	-0.01763602	0.0115929
Q-Learning	127288	132060	2557	.914	-0.00057156	0.00065229
Double Q-Learning	127612	132090	2122	.166	-0.00138269	0.0002186
Expected SARSA	127633	132114	1641.5	.002	-0.0014584	-0.00030373
R Learning	127812	132520	2266	.374	-0.00125935	0.00048369
SARSA	127850	132138	2281	.403	-0.00071761	0.00032514
SARSA Lambda	127910	132161	4692	9.40×10^{-14}	0.0227625	0.03960166
SARSA Lambda, RT	128039	132191	2236	.321	-0.00080193	0.00026532
Watkins (naive) Q, Lambda	128198	132355	4950	$< 2.2\times10^{-16}$	0.04680898	0.06634258
Watkins (naive) Q, Lambda, RT	128253	132376	4454	3.34×10^{-11}	0.0354318	0.06068024
Watkins Q, Lambda	128327	132414	2646	.552	-0.00032021	0.00056773
Watkins Q, LFA	128384	132416	2252	.349	-0.00106307	0.00041883

Table B.41: Experiment Group Two: Scalar ~ Scalar-Normalised Wilcoxon, DC outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Scalar Norm	V	p-value	CI L	CI U
Actor/Critic	180217	180263	2303	.446	-0.00671885	0.00243197
Actor/Critic, ET	133414	133442	3242	.014	0.00146496	0.01329975
Actor/Critic, RT	133440	133451	2384	.629	-0.0177312	0.01128454
Q-Learning	127288	132060	2383	.627	-0.00081176	0.00040649
Double Q-Learning	127612	132090	2791.5	.360	-0.00036303	0.00112255
Expected SARSA	127633	132114	1807	.014	-0.00118328	-0.00014934
R Learning	127812	132520	2660	.644	-0.00053404	0.00085804
SARSA	127850	132138	3169	.027	0.00004889	0.001253395
SARSA Lambda	127910	132161	4706	6.52×10^{-14}	0.02246063	0.03969705
SARSA Lambda, RT	128039	132191	2346.5	.540	-0.00066182	0.00036743
Watkins (naive) Q, Lambda	128198	132355	4923	$< 2.2\times10^{-16}$	0.04706199	0.06644825
Watkins (naive) Q, Lambda , RT	128253	132376	4454	3.34×10^{-11}	0.03484901	0.06081857
Watkins Q, Lambda	128327	132414	2510	.960	-0.00045623	0.00045270
Watkins Q, LFA	128384	132416	2841	.278	-0.00049222	0.00159247

Table B.42: Experiment Group Two: Scalar ~ Scalar-Normalised Wilcoxon, DD outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Scalar Norm	V	p-value	CI L	CI U
Actor/Critic	180217	180263	140	2.43×10^{-16}	-0.1854732	-0.1421832
Actor/Critic, ET	133414	133442	458	1.20×10^{-12}	-0.12238195	-0.07212133
Actor/Critic, RT	133440	133451	832	5.91×10^{-09}	-0.15050901	-0.07509804
Q-Learning	127288	132060	2403.5	.677	-0.00139903	0.00086494
Double Q-Learning	127612	132090	2466	.841	-0.00122332	0.00115222
Expected SARSA	127633	132114	1682	.004	-0.00546265	-0.00071509
R Learning	127812	132520	2596	.809	-0.00110371	0.00146423
SARSA	127850	132138	2576.5	.861	-0.00095000	0.00109516
SARSA Lambda	127910	132161	4946	$< 2.2\times10^{-16}$	0.1931609	0.2702887
SARSA Lambda, RT	128039	132191	2511	.963	-0.00122882	0.00117964
Watkins (naive) Q, Lambda	128198	132355	4898	3.42×10^{-16}	0.225921	0.3037625
Watkins (naive) Q, Lambda, RT	128253	132376	4639	3.68×10^{-13}	0.1566952	0.2433885
Watkins Q, Lambda	128327	132414	2671	.617	-0.00137959	0.00233344
Watkins Q, LFA	128384	132416	2113	.157	-0.00105562	0.00013636

Table B.43: Experiment Group Three: Scalar ~ Ordinal-Normalised Wilcoxon, aggregate outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Ordinal Norm	V	p-value	CI L	CI U
Actor/Critic	180217	180380	47086	.003	0.004814306	0.01950665
Actor/Critic, ET	133414	133460	48229	3.03×10^{-04}	0.00475854	0.01633408
Actor/Critic, RT	133440	133462	44761	.044	0.00028527	0.0253372
Q-Learning	127288	133161	47276	.002	0.00232471	0.00749736
Double Q-Learning	127612	133163	43233	.176	-0.00060221	0.00339996
Expected SARSA	127633	133166	49542	2.88×10^{-05}	0.00212332	0.00485067
R Learning	127812	133087	45745	.015	0.00892328	0.0241554
SARSA	127850	133169	47300	.002	0.00186892	0.0068467
SARSA Lambda	127910	133170	49205	8.33×10^{-05}	0.01219816	0.03113067
SARSA Lambda, RT	128039	133173	48050	5.91×10^{-04}	0.00098683	0.00326679
Watkins (naive) Q, Lambda	128198	133177	49985	1.94×10^{-05}	0.02687254	0.04959825
Watkins (naive) Q, Lambda, RT	128253	133205	49091	1.02×10^{-04}	0.01537017	0.04041253
Watkins Q, Lambda	128327	133211	50420	5.03×10^{-06}	0.00200162	0.00465533
Watkins Q, LFA	128384	133215	38471	.482	-0.00070436	0.0003536

Table B.44: Experiment Group Three: Scalar ~ Ordinal-Normalised Wilcoxon, CC outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Ordinal Norm	V	p-value	CI L	CI U
Actor/Critic	180217	180380	5050	$< 2.22\times10^{-16}$	0.1423645	0.1811838
Actor/Critic, ET	133414	133460	4992	$< 2.22\times10^{-16}$	0.06226982	0.09733796
Actor/Critic, RT	133440	133462	4917	$< 2.22\times10^{-16}$	0.08336978	0.15199963
Q-Learning	127288	133161	3041	.076	-0.00030305	0.00504607
Double Q-Learning	127612	133163	461.5	1.31×10^{-12}	-0.01064168	-0.00602208
Expected SARSA	127633	133166	4635.5	4.02×10^{-13}	0.00884707	0.01546753
R Learning	127812	133087	5040	$< 2.2\times10^{-16}$	0.01395436	0.01723121
SARSA	127850	133169	2796	.352	-0.0015894	0.00333968
SARSA Lambda	127910	133170	134	$< 2.2\times10^{-16}$	-0.3255981	-0.2410206
SARSA Lambda, RT	128039	133173	2949.5	.145	-0.00064813	0.00446444
Watkins (naive) Q, Lambda	128198	133177	60	$< 2.2\times10^{-16}$	-0.4315277	-0.324484
Watkins (naive) Q, Lambda, RT	128253	133205	234	3.39×10^{-15}	-0.3317842	-0.2553133
Watkins Q, Lambda	128327	133211	3728	3.56×10^{-05}	0.00673951	0.02292895
Watkins Q, LFA	128384	133215	708	4.22×10^{-10}	-0.00505373	-0.00281698

Table B.45: Experiment Group Three: Scalar ~ Ordinal-Normalised Wilcoxon, CD outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Ordinal Norm	V	p-value	CI L	CI U
Actor/Critic	180217	180380	3988	4.94×10^{-07}	0.01010631	0.02119447
Actor/Critic, ET	133414	133460	3885	2.95×10^{-06}	0.00972732	0.020776
Actor/Critic, RT	133440	133462	3375	.004	0.00809536	0.05176858
Q-Learning	127288	133161	4928	$< 2.2\times10^{-16}$	0.0178863	0.02355003
Double Q-Learning	127612	133163	4941	$< 2.2\times10^{-16}$	0.01092459	0.01553513
Expected SARSA	127633	133166	4810.5	3.64×10^{-16}	0.0058774	0.00832991
R Learning	127812	133087	5050	$< 2.2\times10^{-16}$	0.04031784	0.04477544
SARSA	127850	133169	4988	$< 2.2\times10^{-16}$	0.01578977	0.02092081
SARSA Lambda	127910	133170	4693.5	9.04×10^{-14}	0.03006055	0.04901389
SARSA Lambda, RT	128039	133173	4598	1.03×10^{-12}	0.00429334	0.00859095
Watkins (naive) Q, Lambda	128198	133177	4944	$< 2.2\times10^{-16}$	0.04824833	0.06863997
Watkins (naive) Q, Lambda, RT	128253	133205	4728	3.65×10^{-14}	0.04518684	0.06776545
Watkins Q, Lambda	128327	133211	4496	1.76×10^{-12}	0.00381331	0.00731213
Watkins Q, LFA	128384	133215	2935	.159	-0.00026088	0.00131079

Table B.46: Experiment Group Three: Scalar ~ Ordinal-Normalised Wilcoxon, DC outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Ordinal Norm	V	p-value	CI L	CI U
Actor/Critic	180217	180380	4042	1.85×10^{-07}	0.01068937	0.0207921
Actor/Critic, ET	133414	133460	3945	2.91×10^{-07}	0.01032237	0.02297579
Actor/Critic, RT	133440	133462	3492	8.9×10^{-04}	0.01142545	0.05187044
Q-Learning	127288	133161	4966	$< 2.2\times10^{-16}$	0.01716379	0.02319494
Double Q-Learning	127612	133163	5043	$< 2.2\times10^{-16}$	0.0129798	0.0177149
Expected SARSA	127633	133166	4933	$< 2.2\times10^{-16}$	0.00570858	0.00809393
R Learning	127812	133087	5049	$< 2.2\times10^{-16}$	0.04046034	0.0446461
SARSA	127850	133169	5027	$< 2.2\times10^{-16}$	0.01615538	0.02119366
SARSA Lambda	127910	133170	4695	8.69×10^{-14}	0.02995492	0.0488764
SARSA Lambda, RT	128039	133173	4563.5	2.43×10^{-12}	0.00451862	0.00826275
Watkins (naive) Q, Lambda	128198	133177	4934	$< 2.2\times10^{-16}$	0.04857597	0.06874454
Watkins (naive) Q, Lambda, RT	128253	133205	4726	3.85×10^{-14}	0.04481109	0.06750222
Watkins Q, Lambda	128327	133211	4407	9.85×10^{-11}	0.00356748	0.00737564
Watkins Q, LFA	128384	133215	2667	.627	-0.00088333	0.00167511

Table B.47: Experiment Group Three: Scalar ~ Ordinal-Normalised Wilcoxon, DD outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Scalar	Ordinal Norm	V	p-value	CI L	CI U
Actor/Critic	180217	180380	58	$< 2.2\times10^{-16}$	-0.217828	-0.171986
Actor/Critic, ET	133414	133460	238	3.79×10^{-15}	-0.13952039	-0.08889426
Actor/Critic, RT	133440	133462	312	2.80×10^{-14}	-0.2432353	-0.150642
Q-Learning	127288	133161	82	$< 2.2\times10^{-16}$	-0.04637332	-0.0334316
Double Q-Learning	127612	133163	317	3.20×10^{-14}	-0.0233847	-0.01501784
Expected SARSA	127633	133166	49	$< 2.2\times10^{-16}$	-0.03063813	-0.02152398
R Learning	127812	133087	0	$< 2.2\times10^{-16}$	-0.10597371	-0.09493283
SARSA	127850	133169	105	$< 2.2\times10^{-16}$	-0.04188668	-0.0295665
SARSA Lambda	127910	133170	4545	3.82×10^{-12}	0.1611196	0.247279
SARSA Lambda, RT	128039	133173	765	1.45×10^{-09}	-0.02067459	-0.00987055
Watkins (naive) Q, Lambda	128198	133177	4860	1.00×10^{-15}	0.2169836	0.2943958
Watkins (naive) Q, Lambda, RT	128253	133205	4357	3.03×10^{-10}	0.1323488	0.2258746
Watkins Q, Lambda	128327	133211	1048	3.84×10^{-07}	-0.03771816	-0.01192049
Watkins Q, LFA	128384	133215	3752.5	2.46×10^{-05}	0.00106592	0.00277179

Table B.48: Experiment Group Four: Ordinal ~ Ordinal-Normalised Wilcoxon, aggregate outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Ordinal	Ordinal Norm	V	p-value	CI L	CI U
Actor/Critic	180292	180380	48983	1.24×10^{-04}	0.02849704	0.07118751
Actor/Critic, ET	133454	133460	45950	.012	0.00145499	0.01468818
Actor/Critic, RT	133455	133462	43710	.119	-0.00159651	0.01412386
Q-Learning	129635	133161	48454	3.01×10^{-04}	0.00224914	0.00655378
Double Q-Learning	129642	180380	51223	1.53×10^{-06}	0.00674334	0.01403715
Expected SARSA	129656	133166	52508	8.21×10^{-08}	0.01134279	0.02478158
R Learning	129683	133087	49944	1.32×10^{-05}	0.00105084	0.00257417
SARSA	129710	133169	49862	2.45×10^{-05}	0.00388842	0.00896165
SARSA Lambda	129718	133170	36226	.094	-0.00363368	0.00023575
SARSA Lambda, RT	129723	133173	39935	.943	-0.00192346	0.00174204
Watkins (naive) Q, Lambda	129726	133177	42598	.28	-0.00053834	0.00168659
Watkins (naive) Q, Lambda, RT	129727	133205	38132	.395	-0.00200465	0.00081435
Watkins Q, Lambda	129757	133211	35790	.075	-0.00374519	0.00013889
Watkins Q, LFA	129758	133215	44057	.071	-0.000057521	0.00135607

Table B.49: Experiment Group Four: Ordinal ~ Ordinal-Normalised Wilcoxon, CC outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Ordinal	Ordinal Norm	V	p-value	CI L	CI U
Actor/Critic	180292	180380	5050	$< 2.2\times10^{-16}$	0.1347989	0.1701886
Actor/Critic, ET	133454	133460	4613	7.10×10^{-13}	0.06049727	0.10462295
Actor/Critic, RT	133455	133462	4745	2.32×10^{-14}	0.07566073	0.13212161
Q-Learning	129635	133161	5003	$< 2.2\times10^{-16}$	0.02706135	0.04007871
Double Q-Learning	129642	133163	4907	2.65×10^{-16}	0.0225849	0.03557682
Expected SARSA	129656	133166	4989	$< 2.2\times10^{-16}$	0.0475539	0.07658593
R Learning	129683	133087	4764.5	1.38×10^{-14}	0.00566535	0.01018887
SARSA	129710	133169	5025	$< 2.2\times10^{-16}$	0.02794735	0.04303068
SARSA Lambda	129718	133170	4797	5.71×10^{-15}	0.04621588	0.09096667
SARSA Lambda, RT	129723	133173	4669	1.71×10^{-13}	0.02817541	0.06789907
Watkins (naive) Q, Lambda	129726	133177	4636.5	3.92×10^{-13}	0.0102078	0.02265073
Watkins (naive) Q, Lambda, RT	129727	133205	4721	4.39×10^{-14}	0.04959186	0.09921312
Watkins Q, Lambda	129757	133211	4885	4.95×10^{-16}	0.06447191	0.11455207
Watkins Q, LFA	129758	133215	4040.5	1.90×10^{-07}	0.00266949	0.00565451

Table B.50: Experiment Group Four: Ordinal ~ Ordinal-Normalised Wilcoxon, CD outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Ordinal	Ordinal Norm	V	p-value	CI L	CI U
Actor/Critic	180292	180380	4799	5.41×10^{-15}	0.07736792	0.10373309
Actor/Critic, ET	133454	133460	3645	1.19×10^{-04}	0.0066149	0.03135787
Actor/Critic, RT	133455	133462	2595	.811	-0.01103327	0.0119124
Q-Learning	129635	133161	4261	2.41×10^{-09}	0.0049526	0.00879655
Double Q-Learning	129642	133163	4962.5	$< 2.2\times10^{-16}$	0.01690834	0.02402362
Expected SARSA	129656	133166	4909	2.50×10^{-16}	0.02717812	0.06037278
R Learning	129683	133087	4463	2.71×10^{-11}	0.00256963	0.00461086
SARSA	129710	133169	4678	1.35×10^{-13}	0.00799134	0.01178773
SARSA Lambda	129718	133170	2150	.198	-0.00307501	0.00046444
SARSA Lambda, RT	129723	133173	2506	.949	-0.0017468	0.00152633
Watkins (naive) Q, Lambda	129726	133177	3683.5	6.85×10^{-05}	0.00100599	0.00299844
Watkins (naive) Q, Lambda, RT	129727	133205	2181.5	.238	-0.00149914	0.00040972
Watkins Q, Lambda	129757	133211	1671	.003	-0.00325324	-0.00057229
Watkins Q, LFA	129758	133215	4163	3.87×10^{-09}	0.00230506	0.00454994

Table B.51: Experiment Group Four: Ordinal ~ Ordinal-Normalised Wilcoxon, DC outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Ordinal	Ordinal Norm	V	p-value	CI L	CI U
Actor/Critic	180292	180380	4800	5.26×10^{-15}	0.07529851	0.10169417
Actor/Critic, ET	133454	133460	3844	5.80×10^{-06}	0.007942	0.03353162
Actor/Critic, RT	133455	133462	2584	.841	-0.01100991	0.01112853
Q-Learning	129635	133161	4422	6.99×10^{-11}	0.00538657	0.00922316
Double Q-Learning	129642	133163	4933.5	$< 2.2\times10^{-16}$	0.01770251	0.0251179
Expected SARSA	129656	133166	4902	3.05×10^{-16}	0.02690267	0.05796907
R Learning	129683	133087	4213	1.33×10^{-09}	0.00220566	0.00398615
SARSA	129710	133169	4780	9.07×10^{-15}	0.00833984	0.01187145
SARSA Lambda	129718	133170	2218.5	.293	-0.0029206	0.00075655
SARSA Lambda, RT	129723	133173	2857	.254	-0.00057049	0.00355637
Watkins (naive) Q, Lambda	129726	133177	3614.5	1.81×10^{-04}	0.00098685	0.00306248
Watkins (naive) Q, Lambda, RT	129727	133205	2191.5	.252	-0.00193562	0.00047826
Watkins Q, Lambda	129757	133211	1620.5	.003	-0.0037004	-0.00059327
Watkins Q, LFA	129758	133215	179	7.35×10^{-16}	-0.01120571	-0.00662794

Table B.52: Experiment Group Four: Ordinal ~ Ordinal-Normalised Wilcoxon, DD outcomes. **ET:** Eligibility Traces **RT:** Replacing Traces **LFA:** Linear Function Approximation.

Algorithm	Exp ID		Wilcoxon			
	Ordinal	Ordinal Norm	V	p-value	CI L	CI U
Actor/Critic	180292	180380	39	$< 2.2\times10^{-16}$	-0.3877473	-0.2991819
Actor/Critic, ET	133454	133460	220	2.31×10^{-15}	-0.1711	-0.1087011
Actor/Critic, RT	133455	133462	1053	4.20×10^{-07}	-0.13285919	-0.05729592
Q-Learning	129635	133161	3	$< 2.2\times10^{-16}$	-0.05610783	-0.03900995
Double Q-Learning	129642	133163	5	$< 2.2\times10^{-16}$	-0.08647247	-0.05962742
Expected SARSA	129656	133166	24	$< 2.2\times10^{-16}$	-0.188571	-0.1149752
R Learning	129683	133087	303	2.20×10^{-14}	-0.01851065	-0.01058819
SARSA	129710	133169	5	$< 2.2\times10^{-16}$	-0.06677474	-0.04779989
SARSA Lambda	129718	133170	38	$< 2.2\times10^{-16}$	-0.07695046	-0.04105846
SARSA Lambda, RT	129723	133173	458	1.20×10^{-12}	-0.08471341	-0.04625496
Watkins (naive) Q, Lambda	129726	133177	98	$< 2.2\times10^{-16}$	-0.02679676	-0.01728623
Watkins (naive) Q, Lambda, RT	129727	133205	465	1.43×10^{-12}	-0.09949011	-0.05100133
Watkins Q, Lambda	129757	133211	219	2.24×10^{-15}	-0.10442598	-0.06005341
Watkins Q, LFA	129758	133215	3276	.01	0.00027446	0.00204639

B.3.4 Behavioural Profile Surface Maps

The Mutual Cooperation outcome for the first profile (*Actor/Critic* under a scalar representation, Figure B.2) is presented here to show the axes and labels clearly. In the remaining profiles this detail is not so easy to read but the important thing is the shape of the surfaces. Each set of four figures are the four representations for each algorithm.

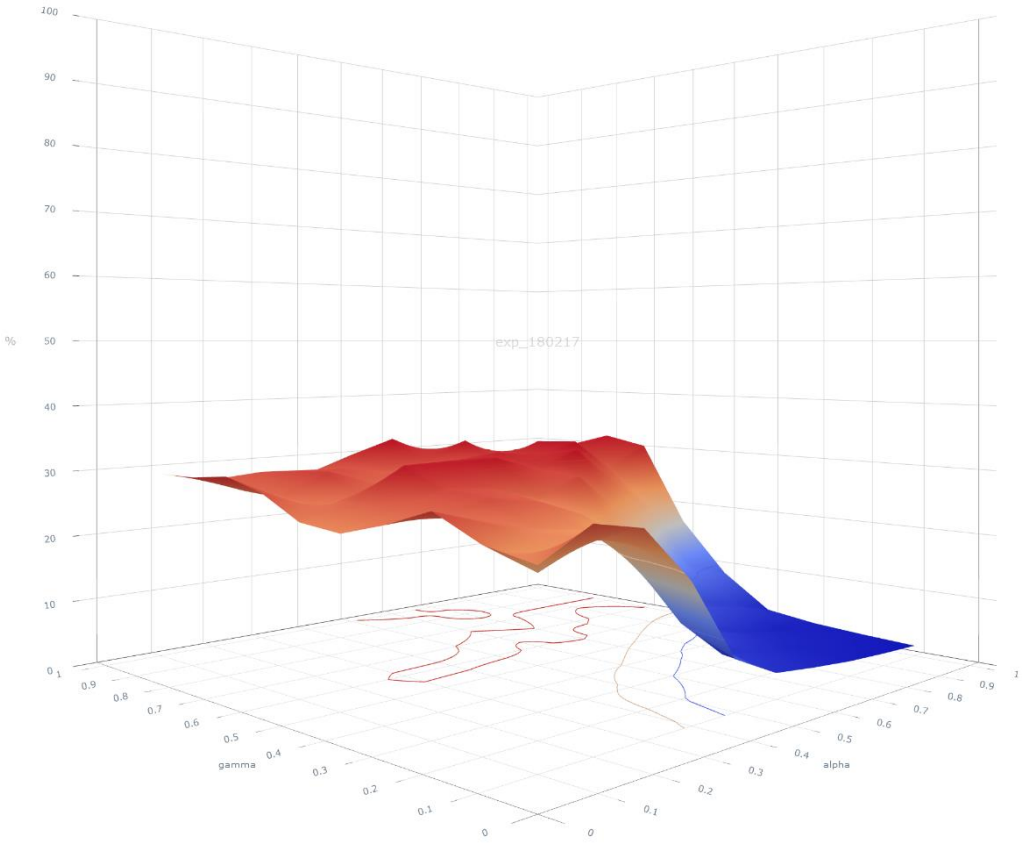


Figure B.2: Exp_ID: 180217; Mutual Cooperation; *Actor/Critic*; Scalar.

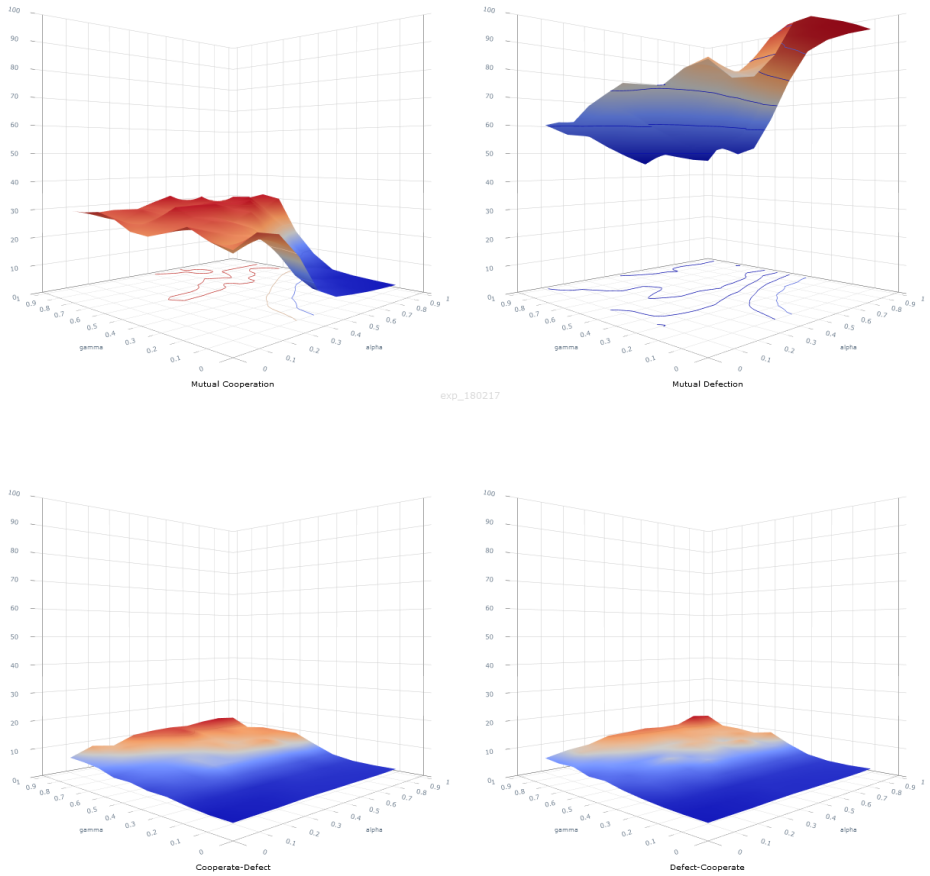


Figure B.3: Exp_ID: 180217; *Actor/Critic*; Scalar.

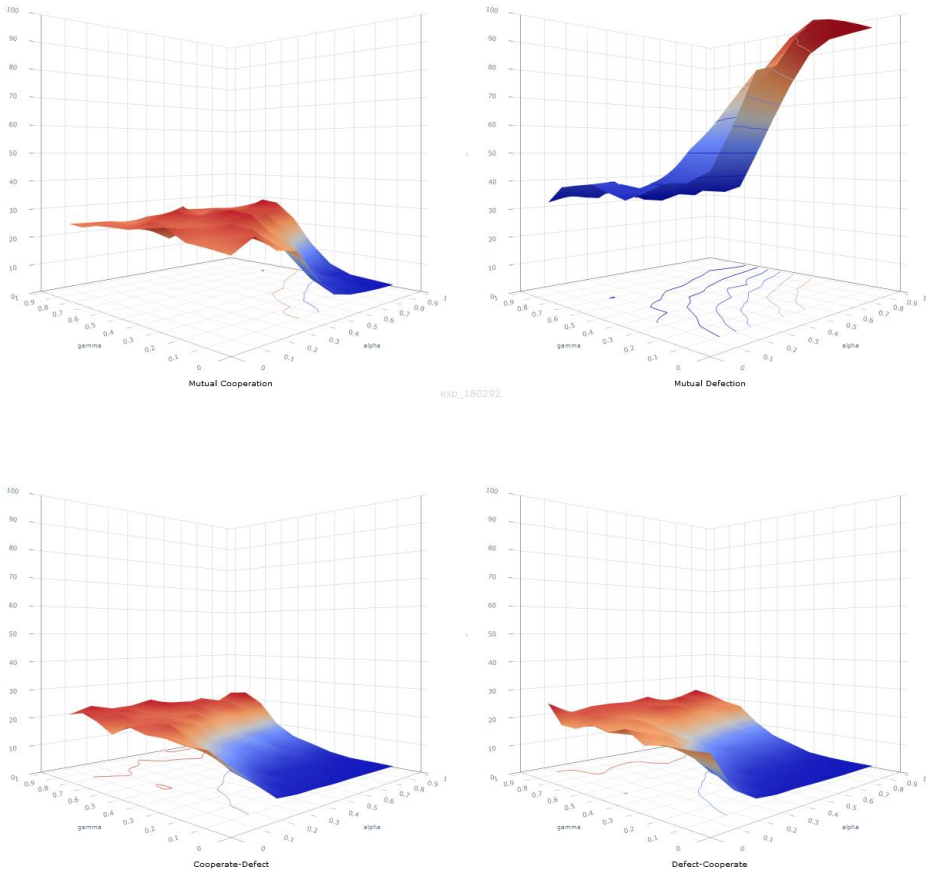


Figure B.4: Exp_ID: 180292; *Actor/Critic*; Ordinal.

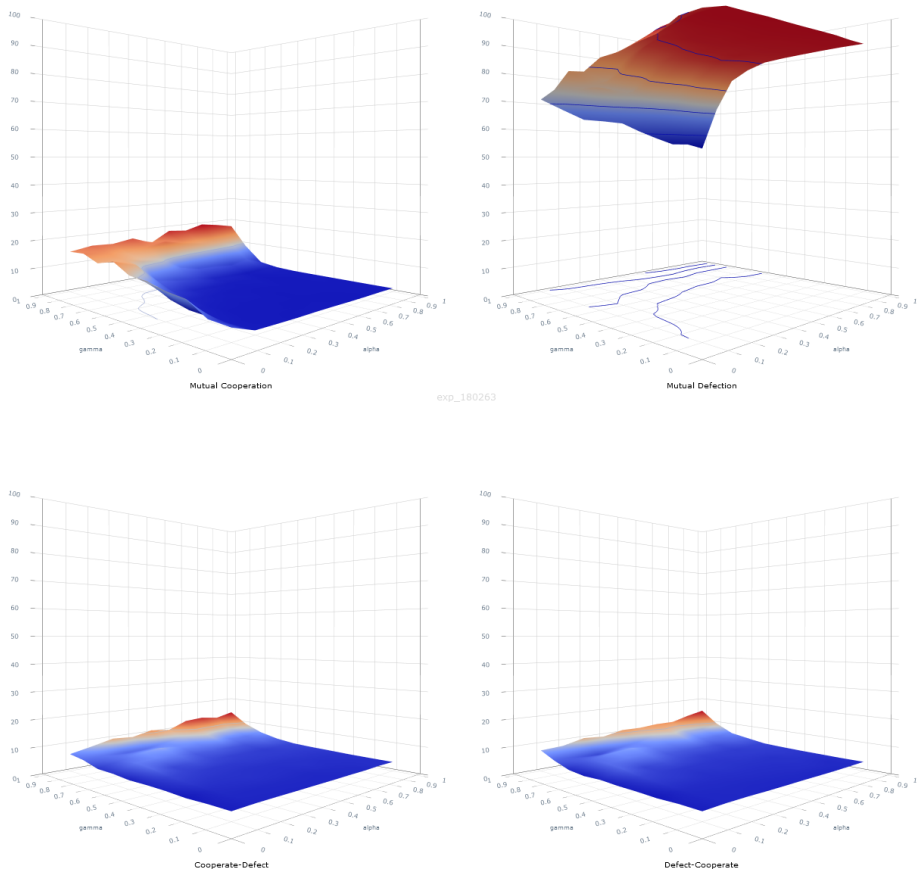


Figure B.5: Exp_ID: 180263; *Actor/Critic*; Normalised Scalar.

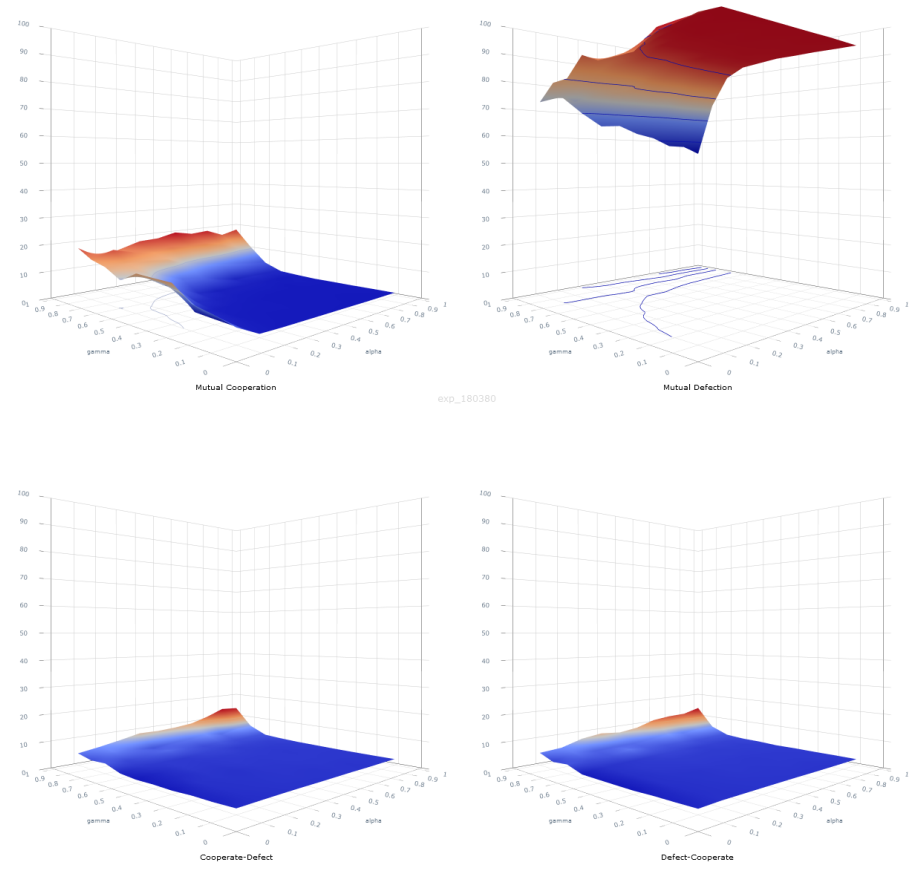


Figure B.6: Exp_ID: 180380; *Actor/Critic*; Normalised Ordinal.

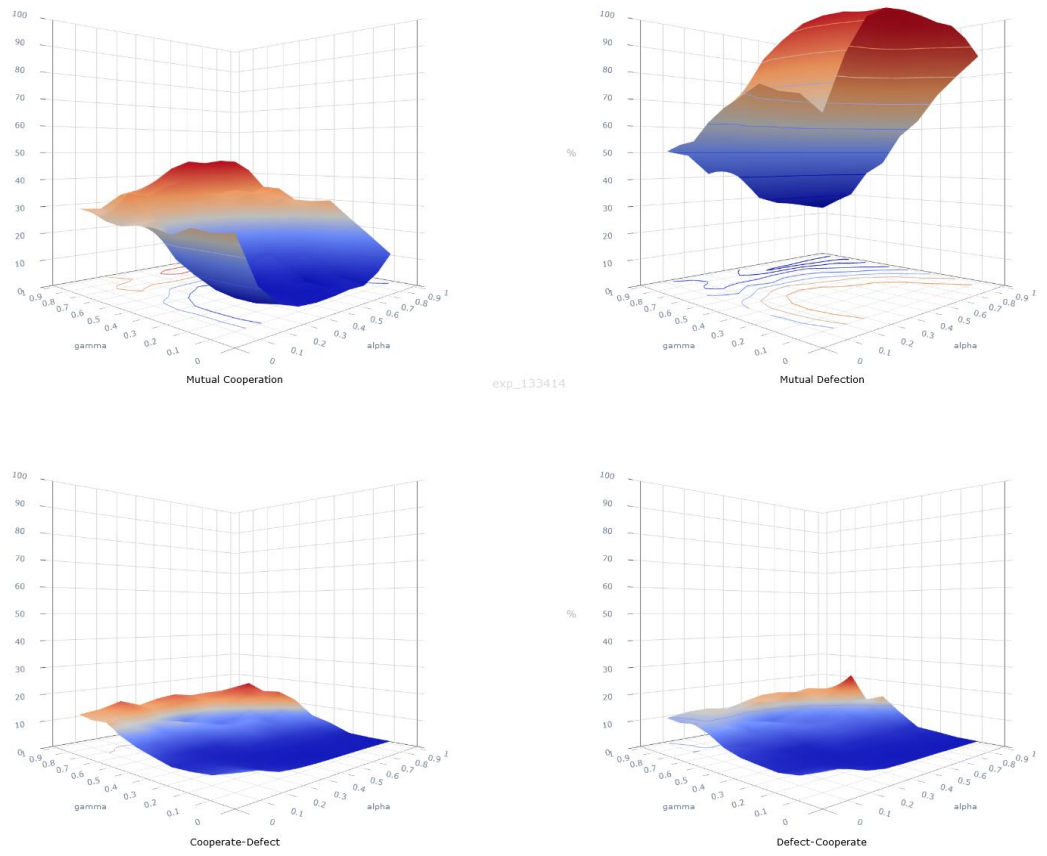


Figure B.7: Exp_ID: 133414; *Actor/Critic with Eligibility Traces*; Scalar.

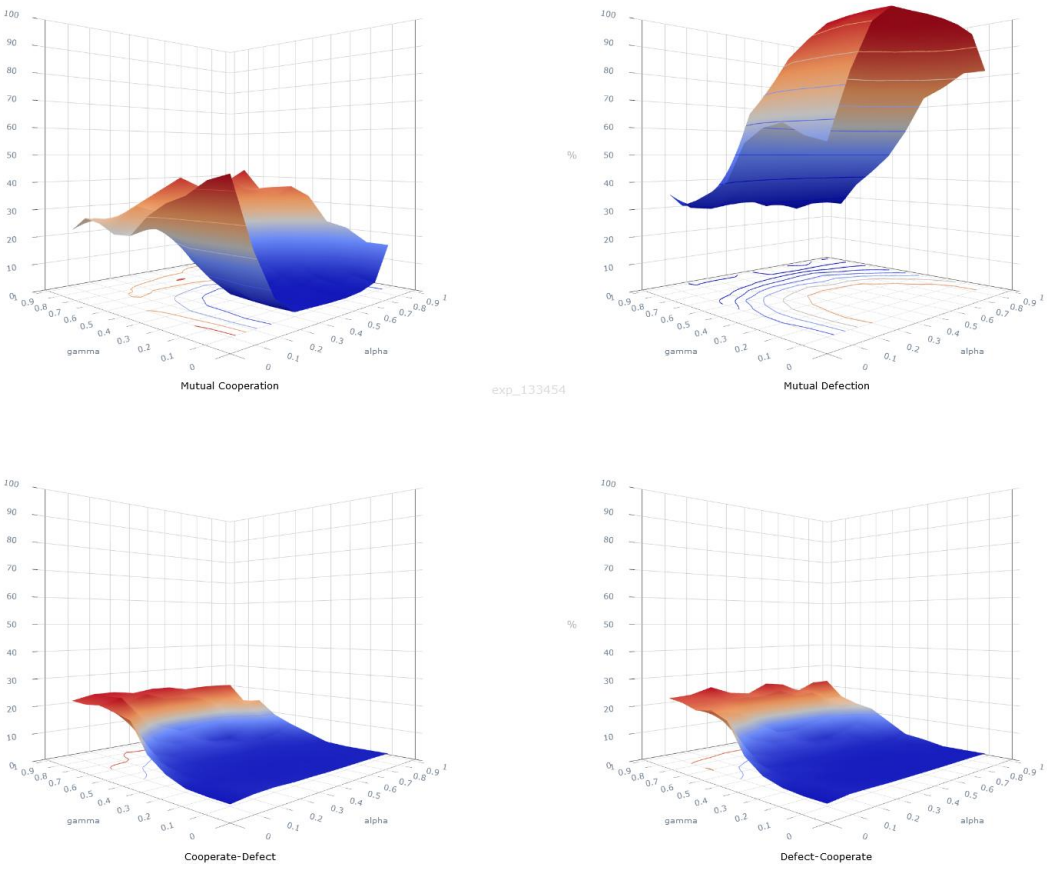


Figure B.8: Exp_ID: 133454; *Actor/Critic with Eligibility Traces*; Ordinal.

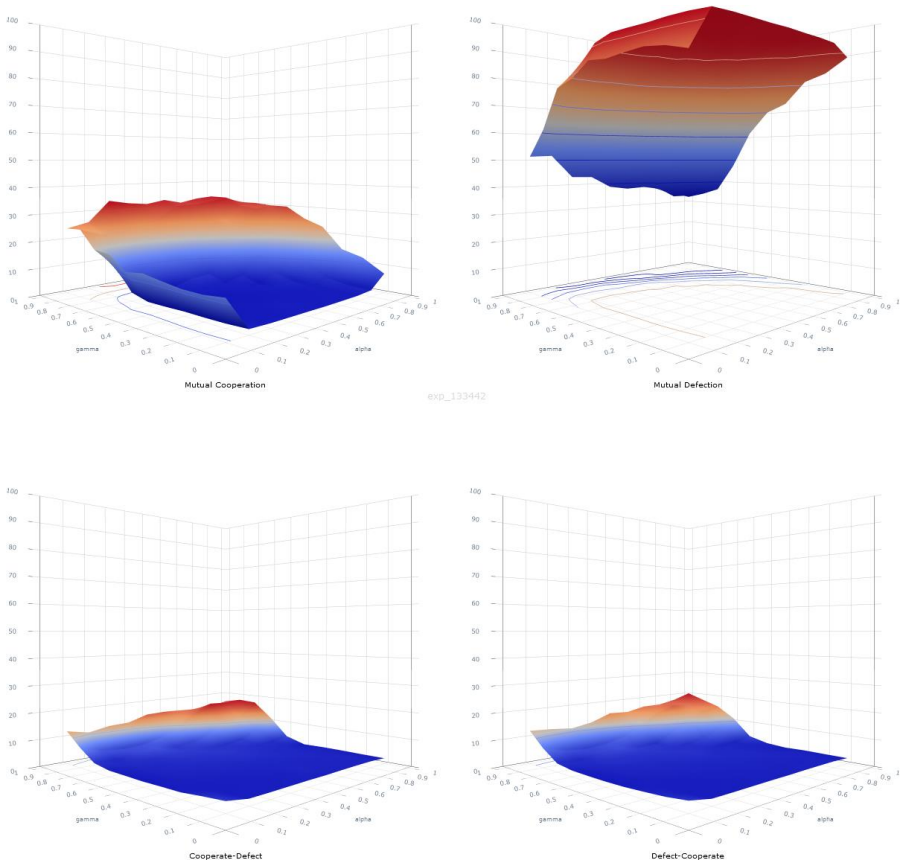


Figure B.9: Exp_ID: 133442; *Actor/Critic with Eligibility Traces*; Normalised Scalar.

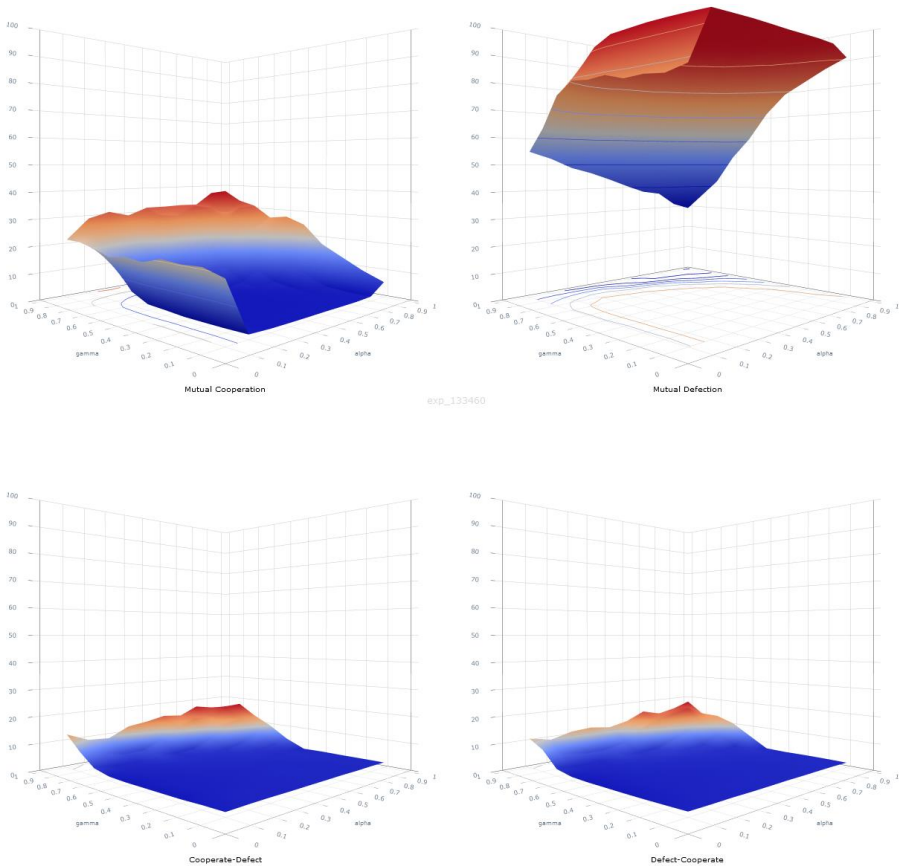


Figure B.10: Exp_ID: 133460; *Actor/Critic with Eligibility Traces*; Normalised Ordinal.

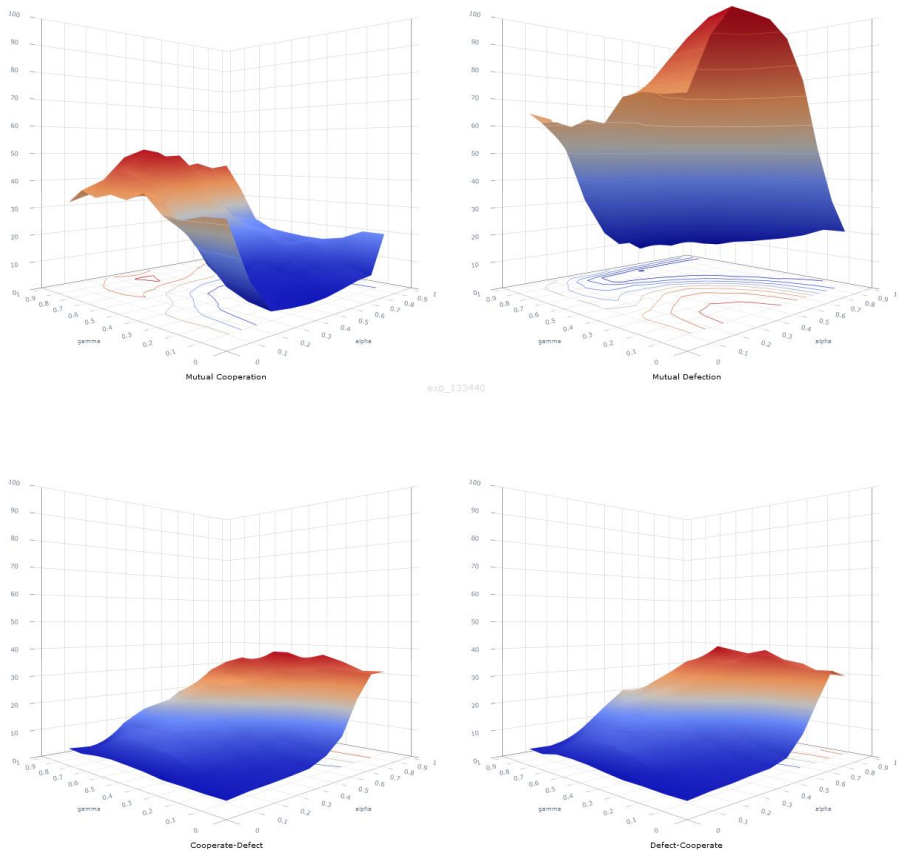


Figure B.11: Exp_ID: 133440; *Actor/Critic with Replacing Traces*; Scalar.

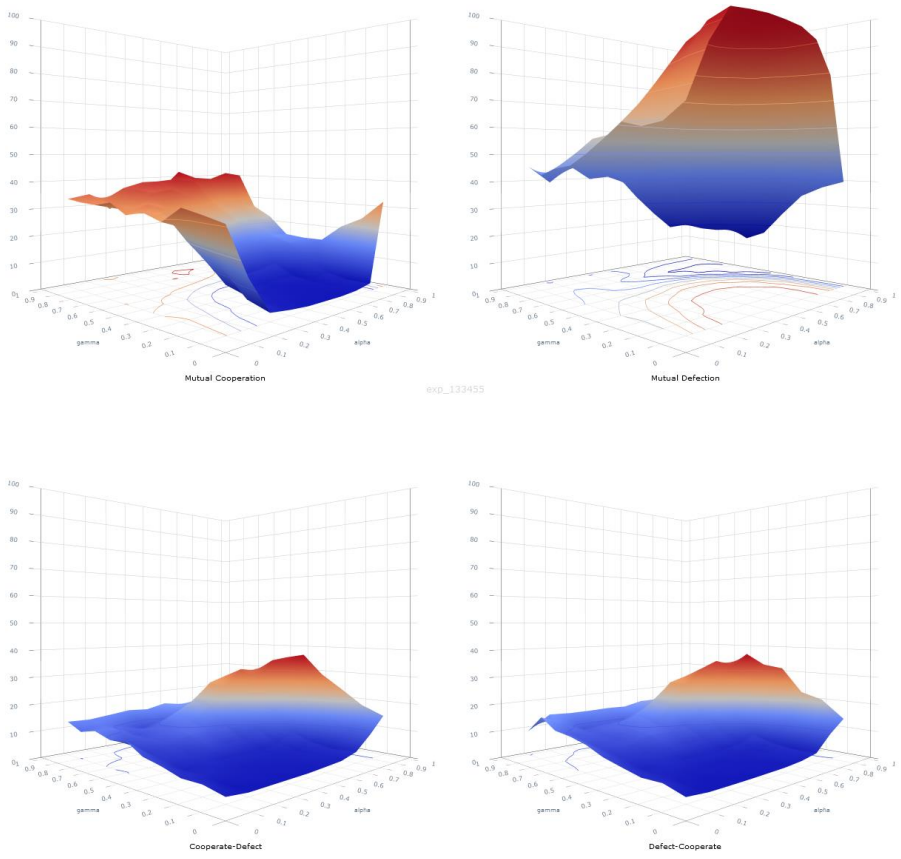


Figure B.12: Exp_ID: 133455; *Actor/Critic with Replacing Traces*; Ordinal.

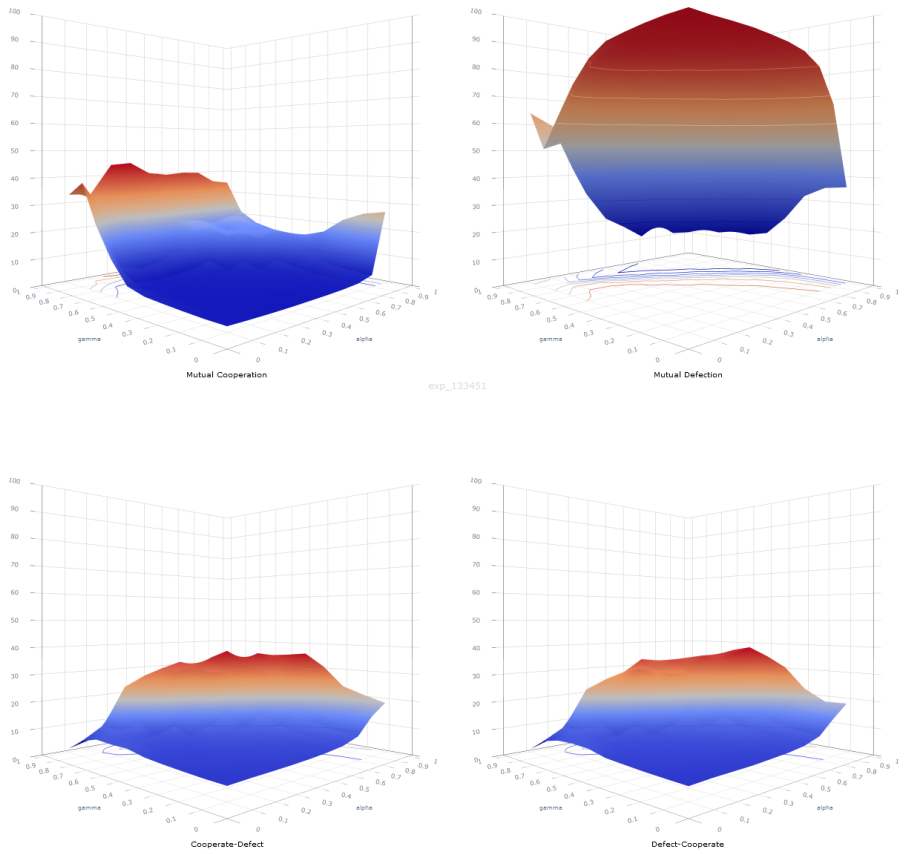


Figure B.13: Exp_ID: 133451; *Actor/Critic with Replacing Traces*; Normalised Scalar.

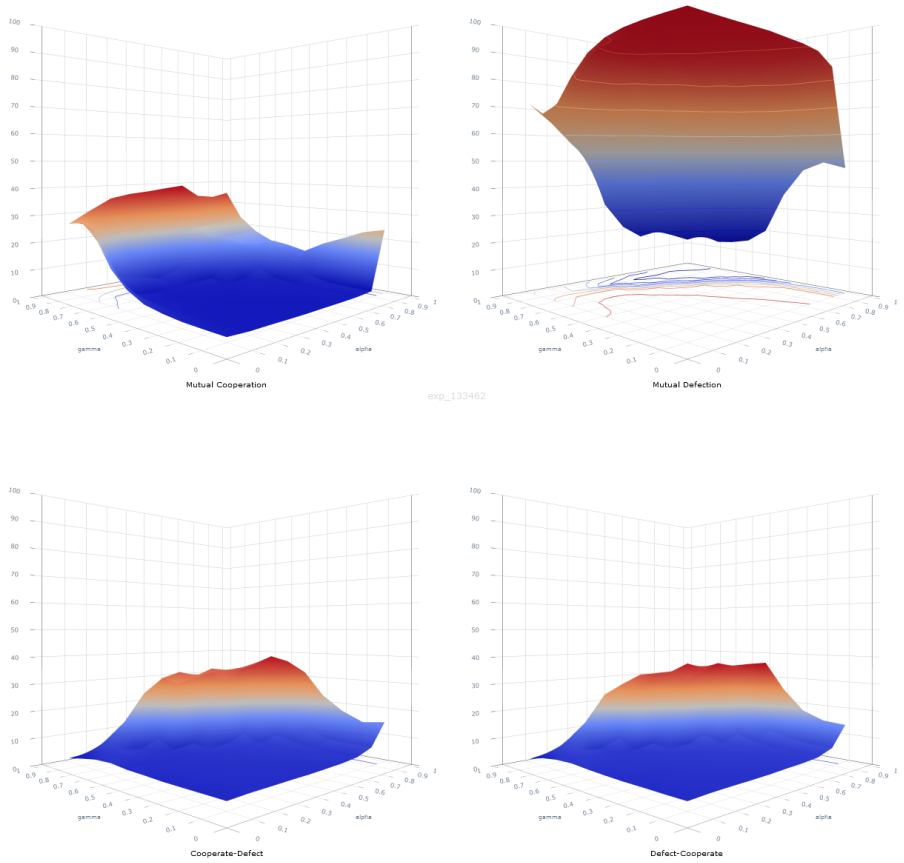


Figure B.14: Exp_ID: 133462; *Actor/Critic with Replacing Traces*; Normalised Ordinal.

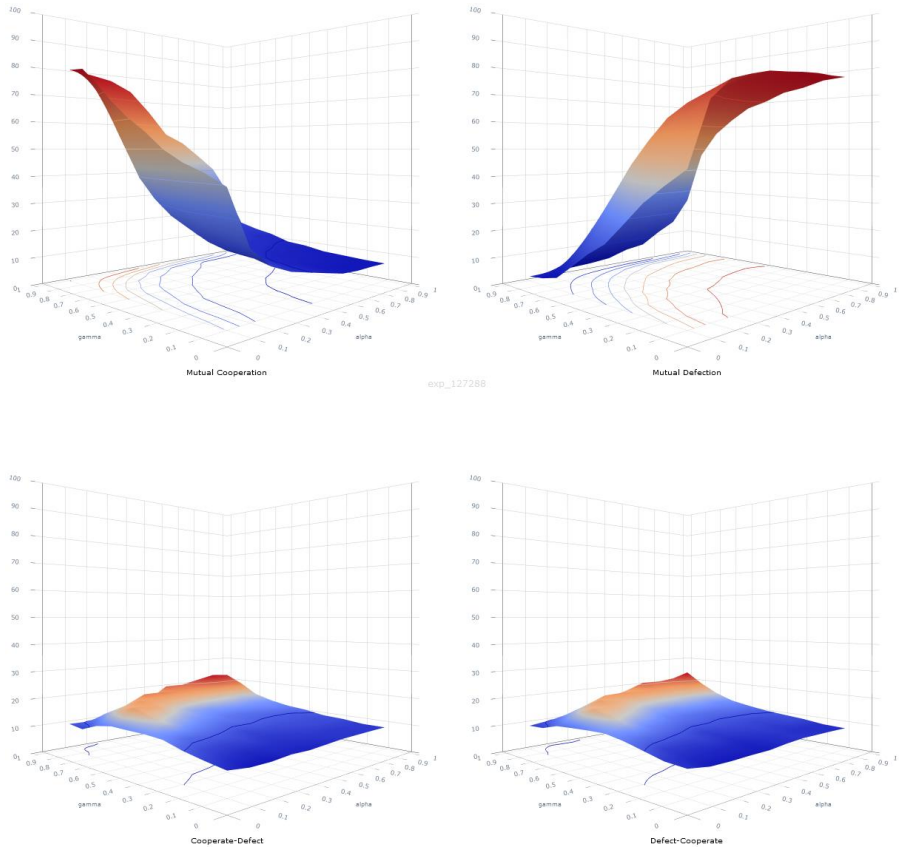


Figure B.15: Exp_ID: 127288; *Q-Learning*; Scalar.

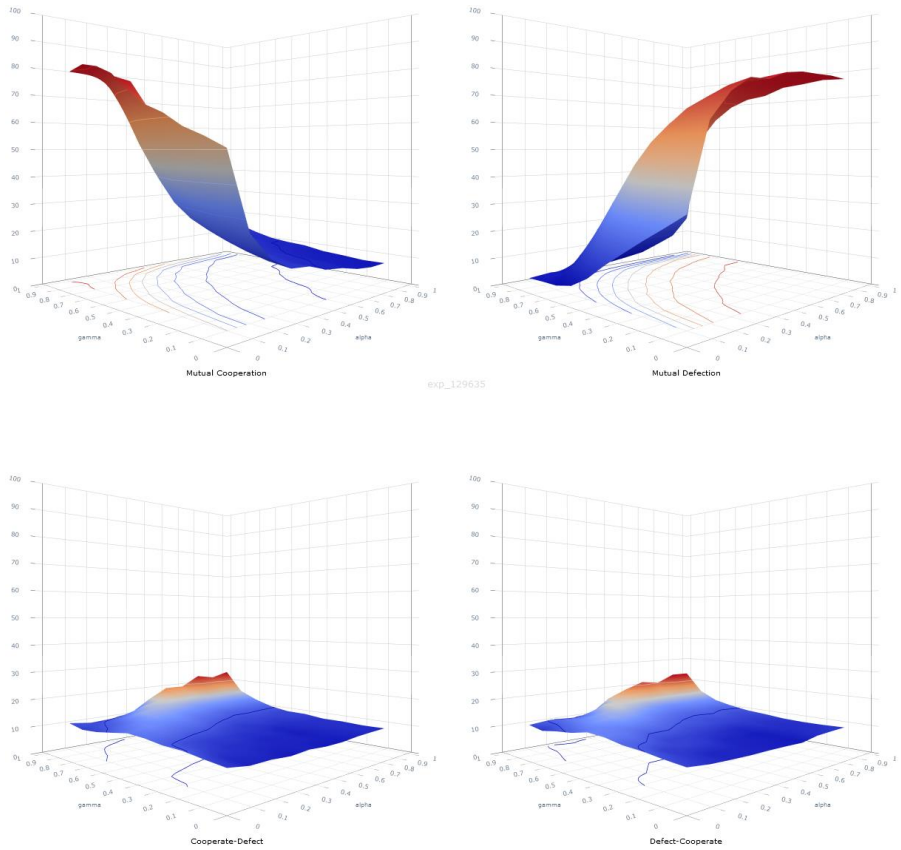


Figure B.16: Exp_ID: 129635; *Q-Learning*; Ordinal.

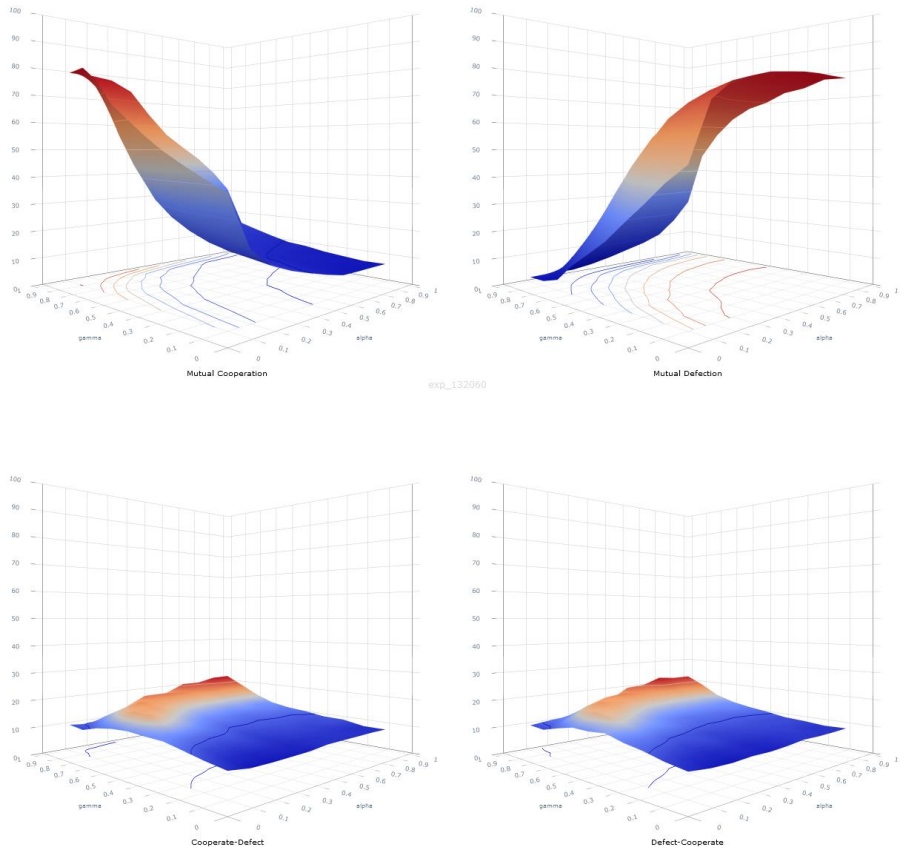


Figure B.17: Exp_ID: 132060; *Q-Learning*; Normalised Scalar.

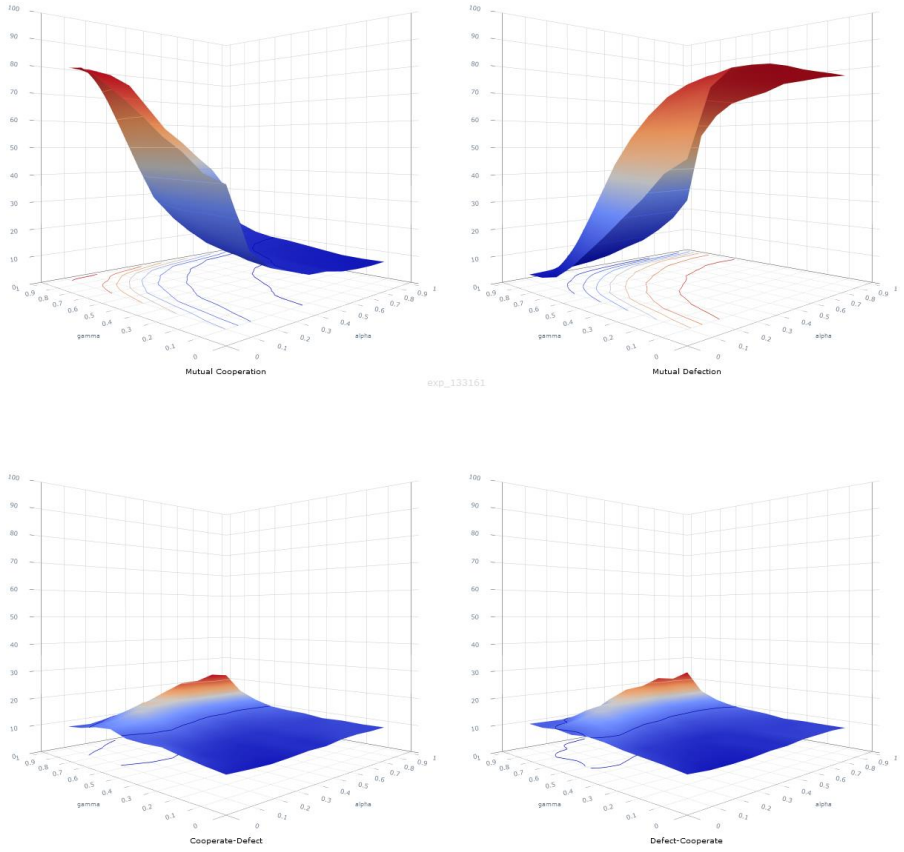


Figure B.18: Exp_ID: 133161; *Q-Learning*; Normalised Ordinal.

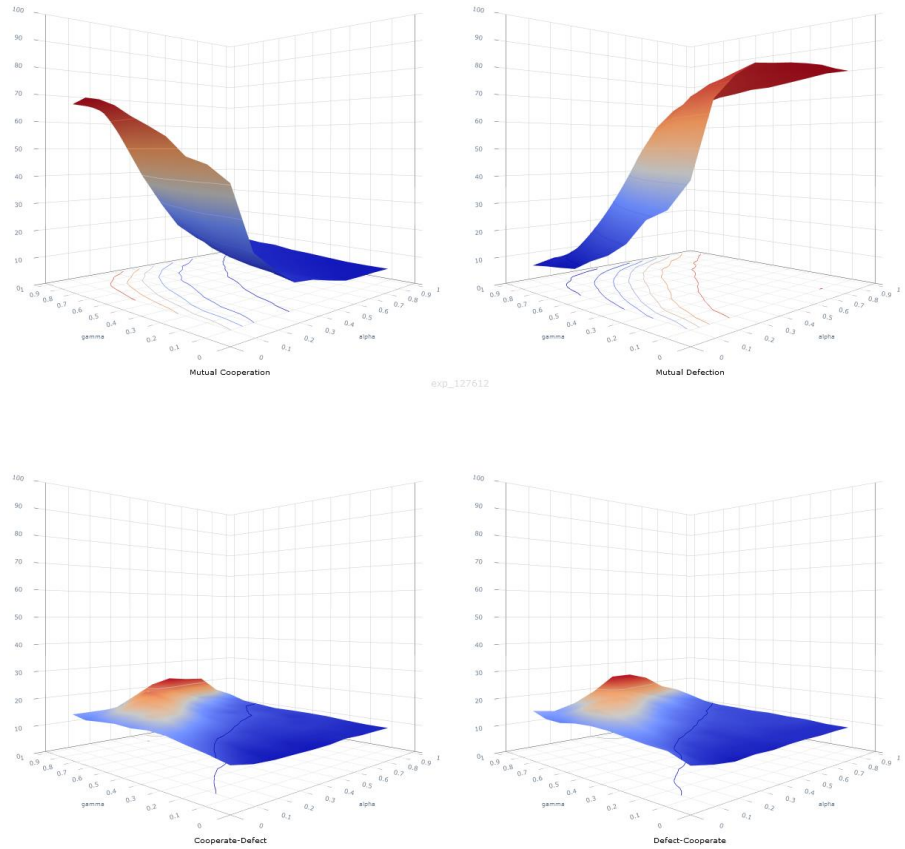


Figure B.19: Exp_ID: 127612; *Double Q-Learning*; Scalar.

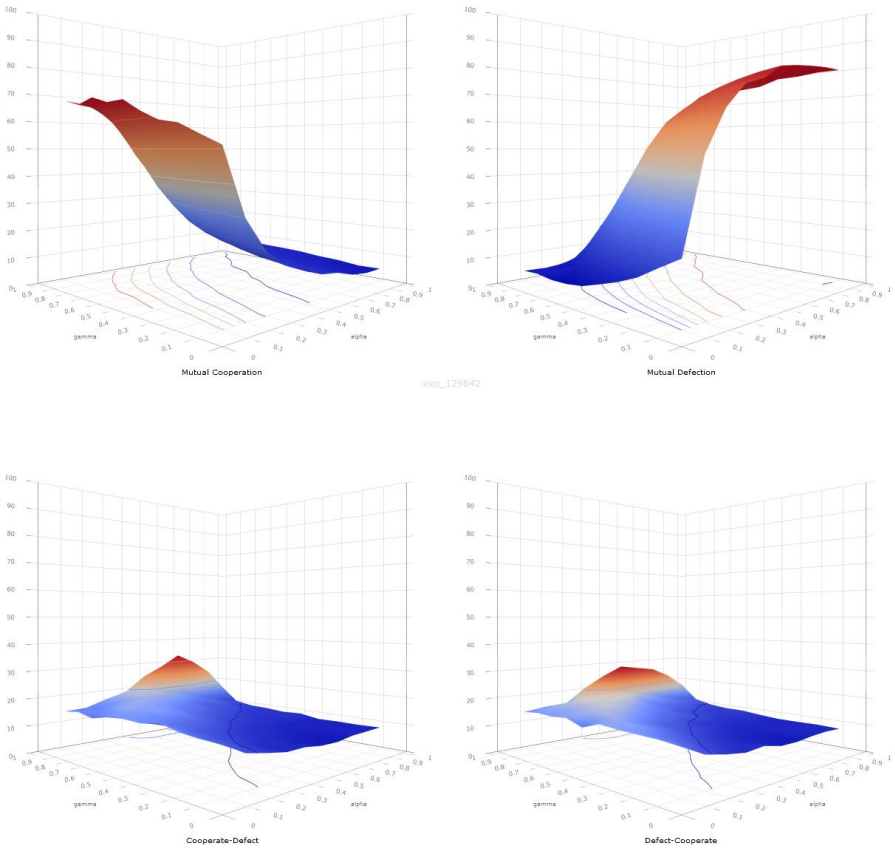


Figure B.20: Exp_ID: 129642; *Double Q-Learning*; Ordinal.

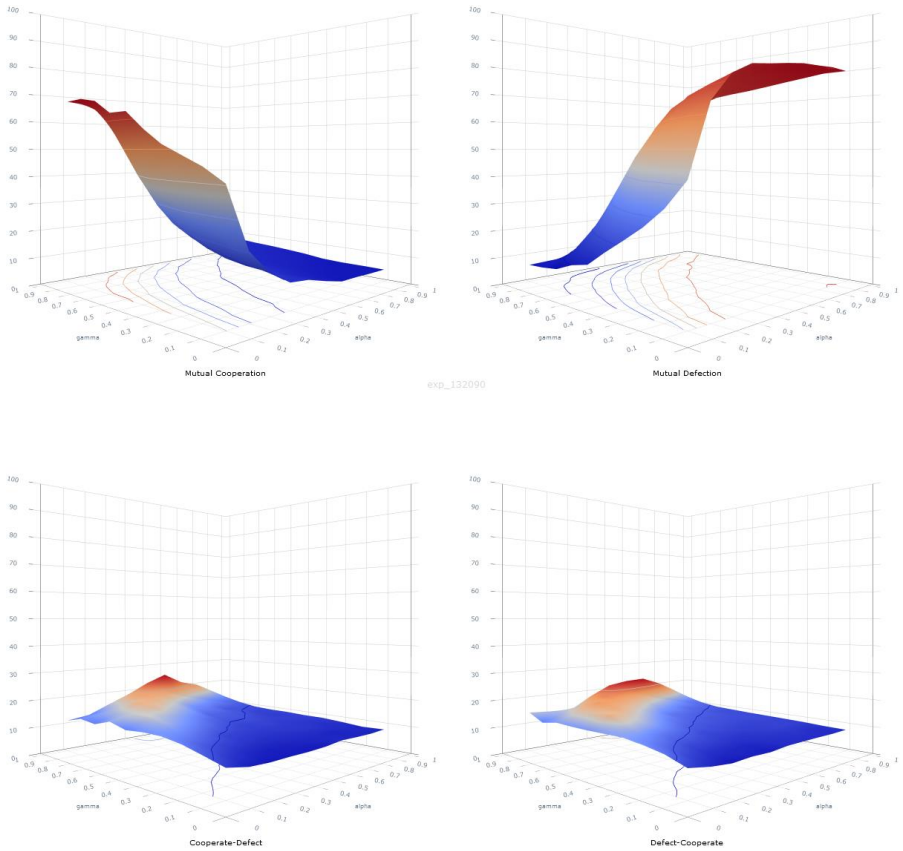


Figure B.21: Exp_ID: 132090; *Double Q-Learning*; Normalised Scalar.

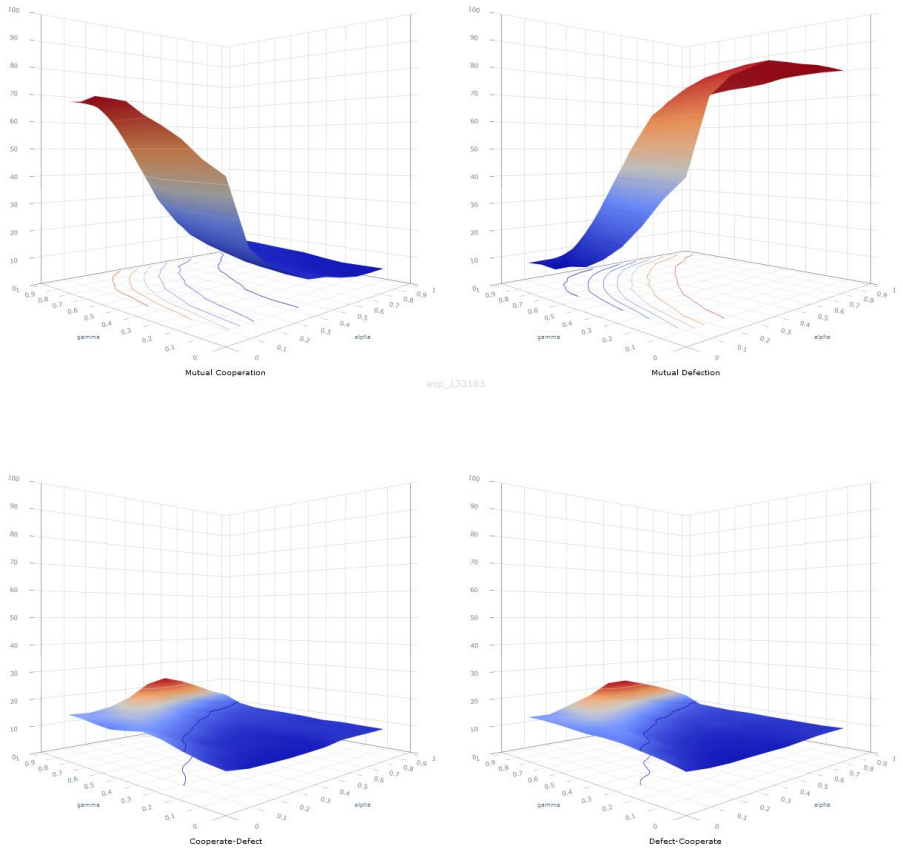


Figure B.22: Exp_ID: 133163; *Double Q-Learning*; Normalised Ordinal.

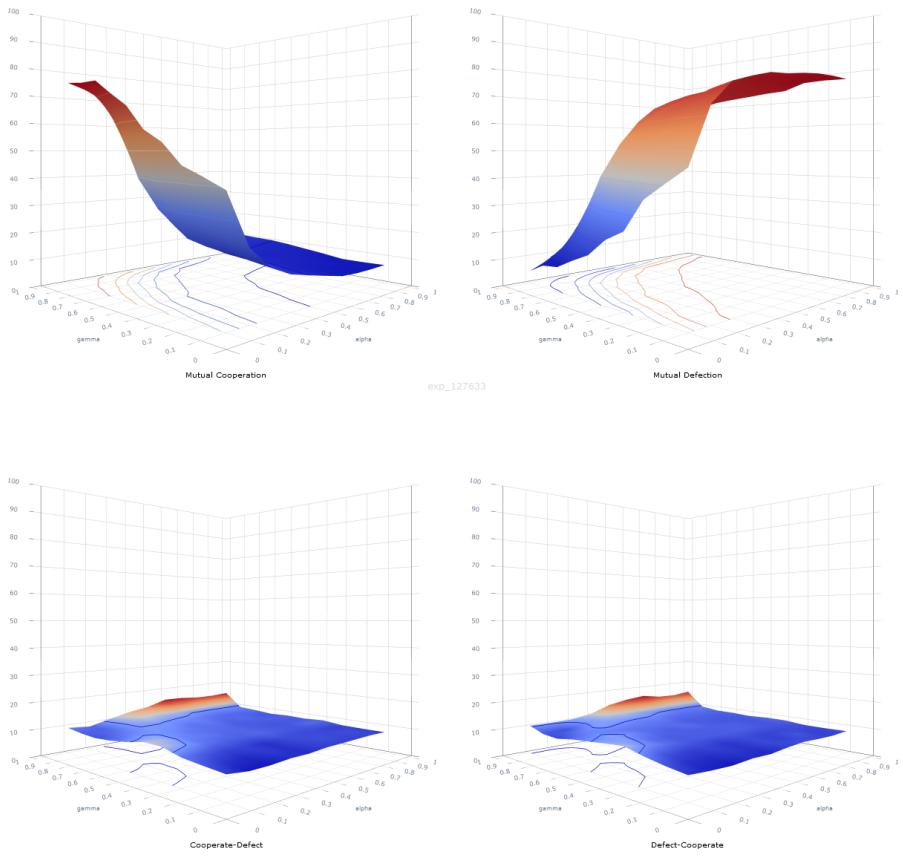


Figure B.23: Exp_ID: 127633; *Expected SARSA*; Scalar.

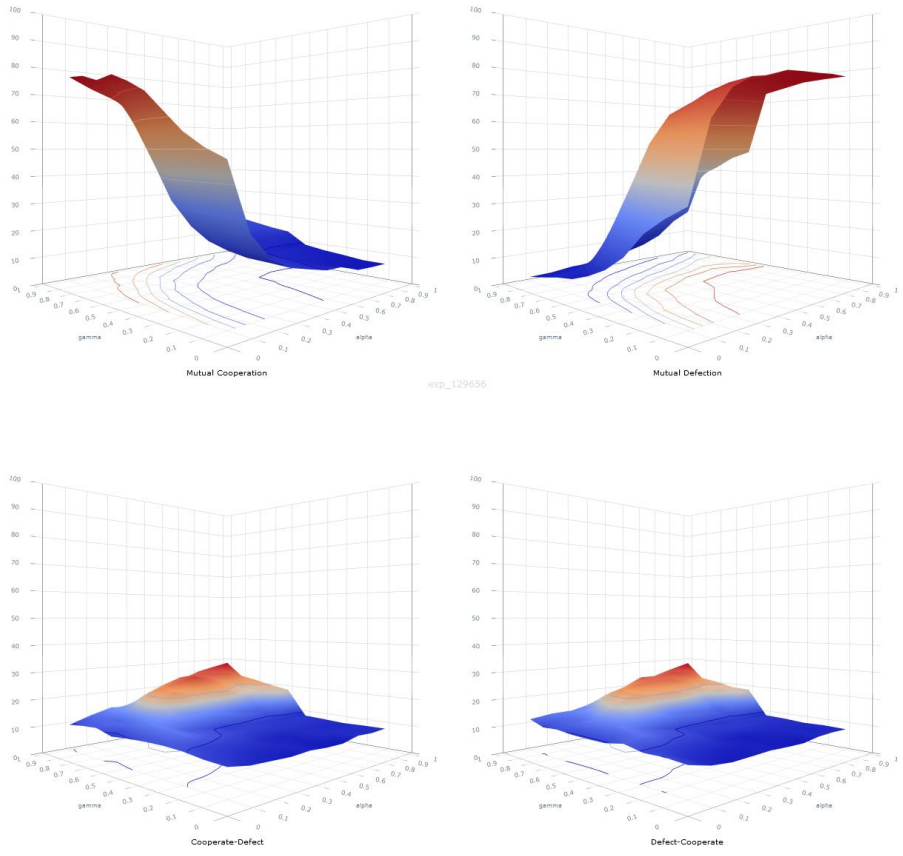


Figure B.24: Exp_ID: 129656; *Expected SARSA*; Ordinal.

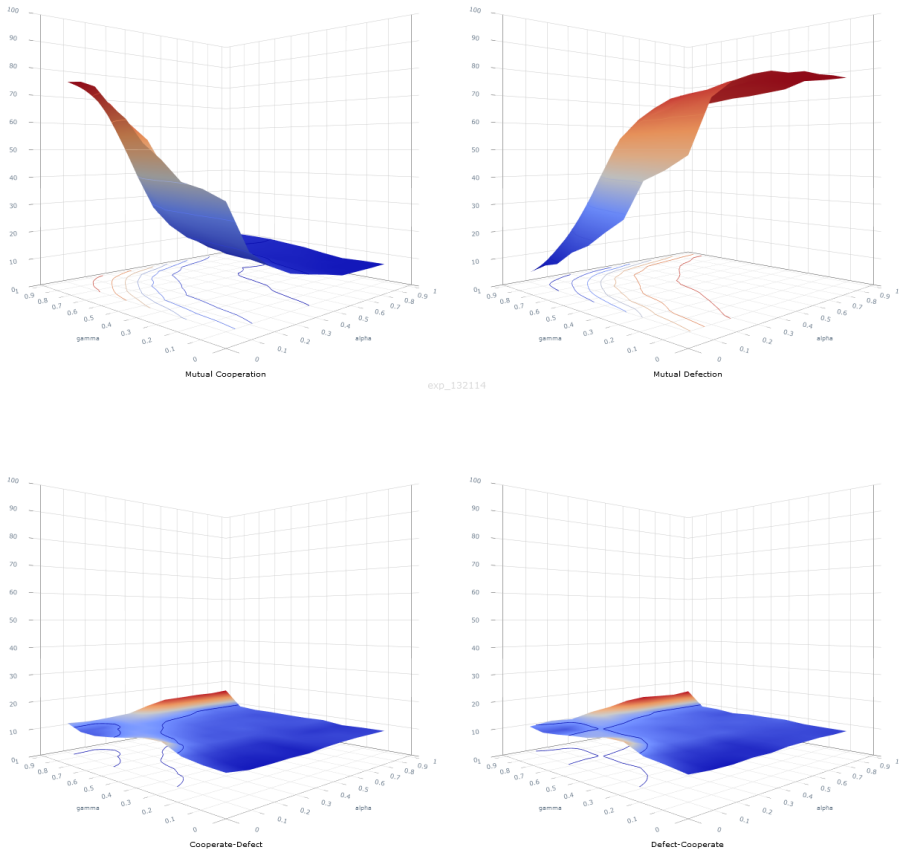


Figure B.25: Exp_ID: 132114 ; *Expected SARSA*; Normalised Scalar.

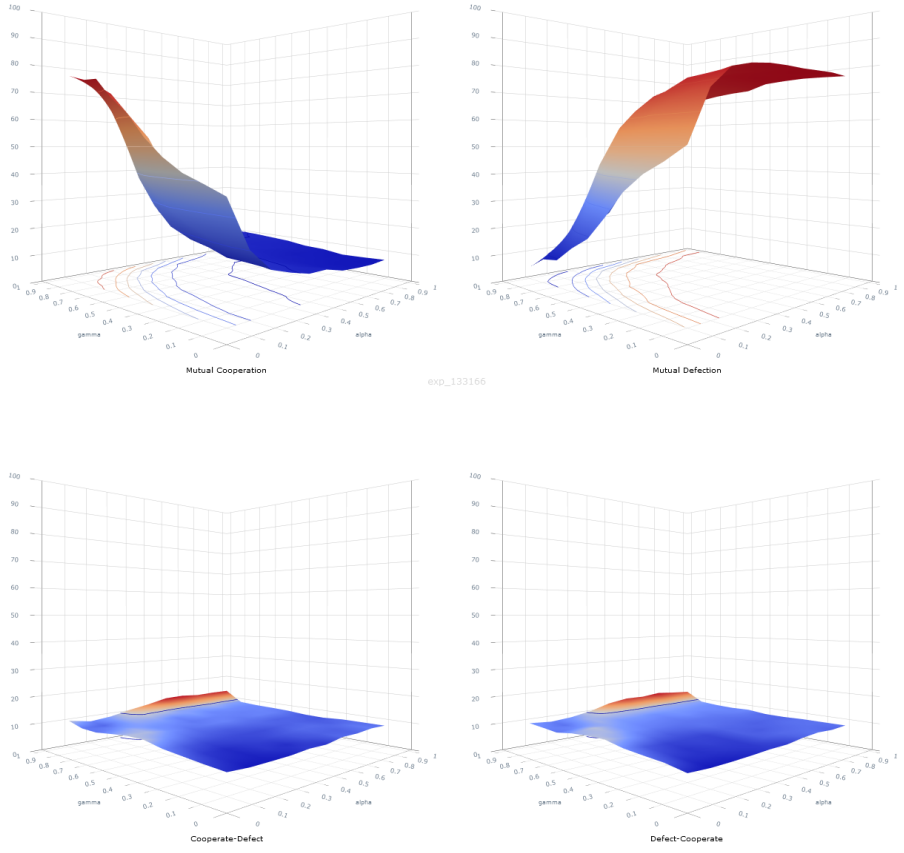


Figure B.26: Exp_ID: 133166; *Expected SARSA*; Normalised Ordinal.

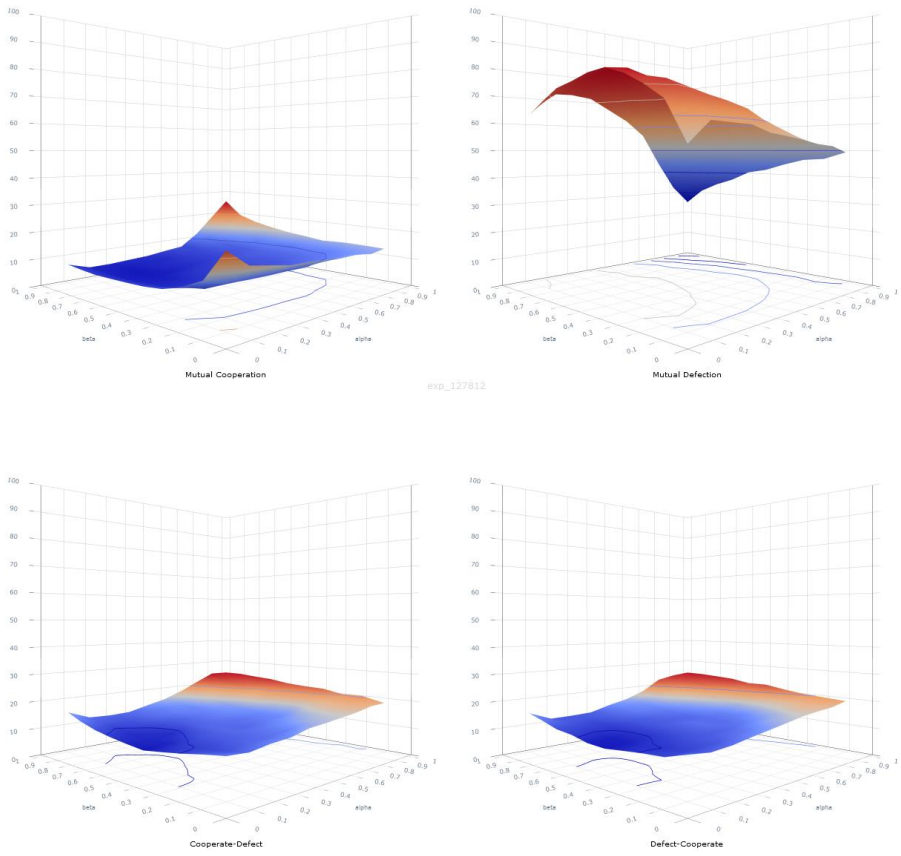


Figure B.27: Exp_ID: 127812; *R Learning*; Scalar.

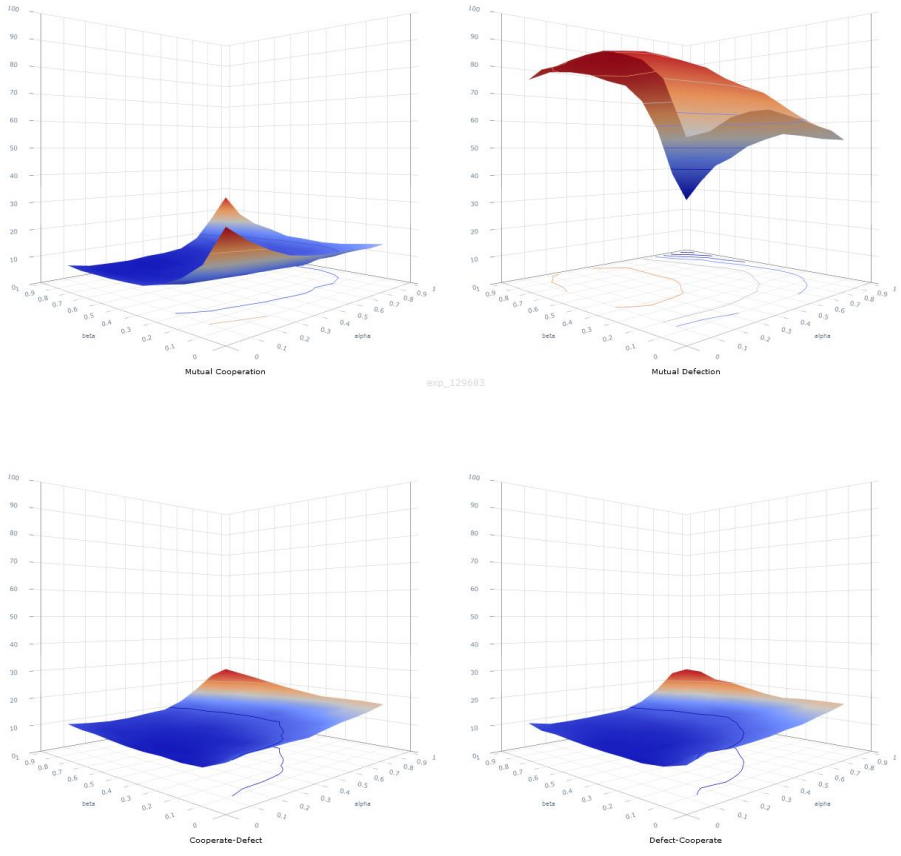


Figure B.28: Exp_ID: 129683; *R Learning*; Ordinal.

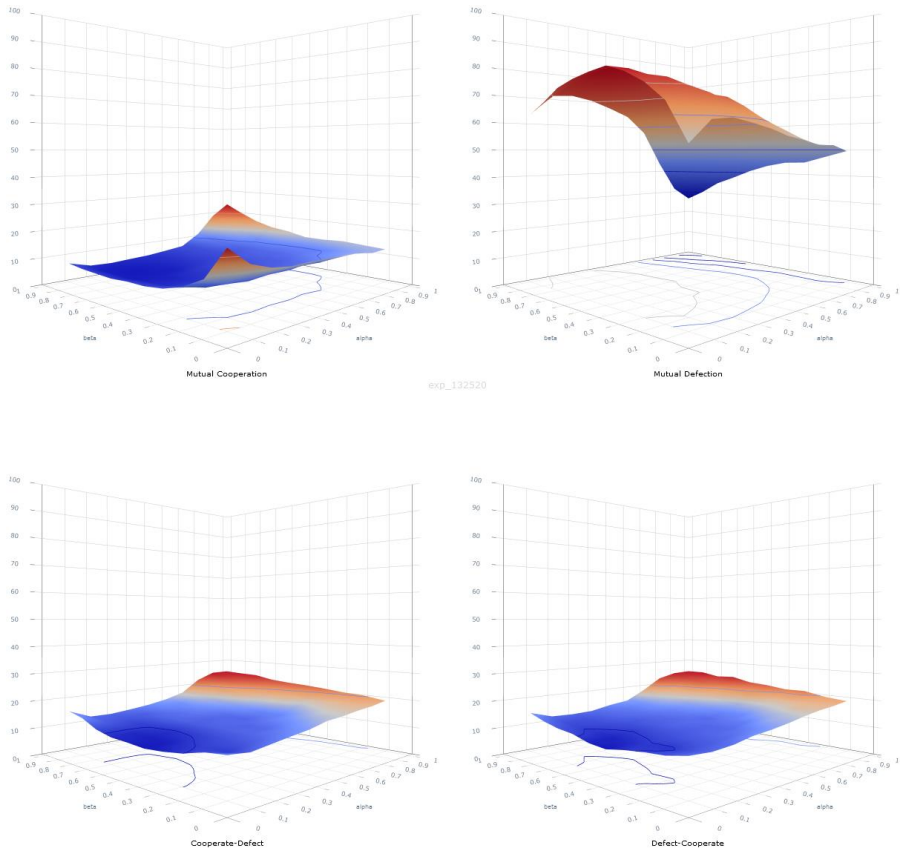


Figure B.29: Exp_ID: 132520; *R Learning*; Normalised Scalar.

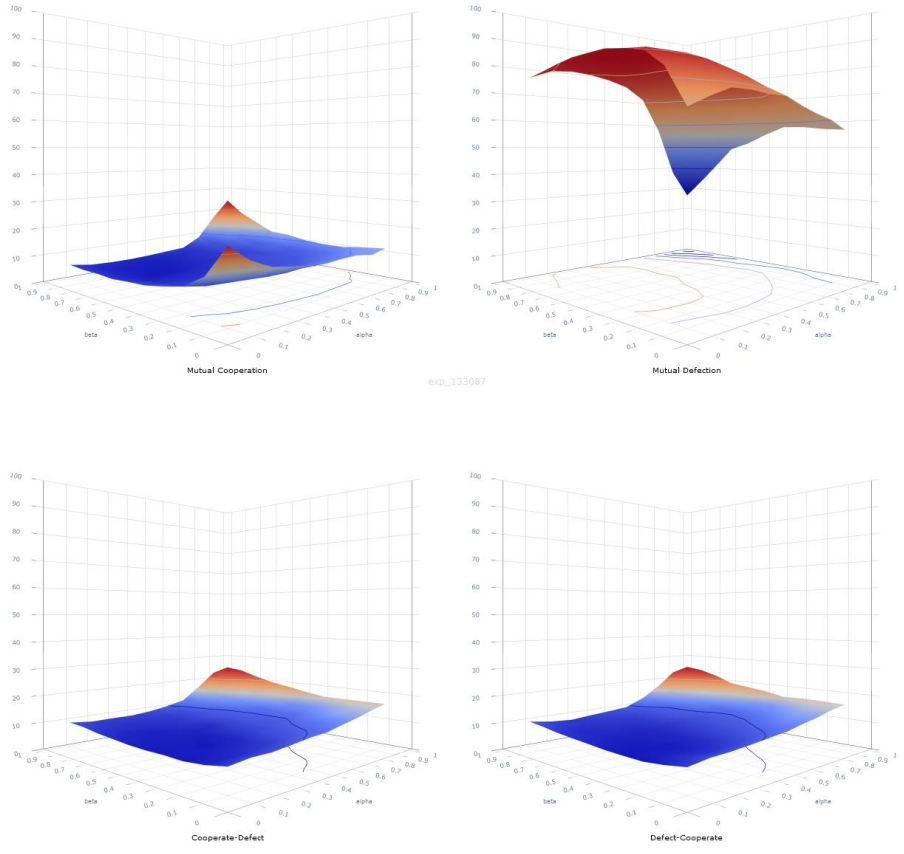


Figure B.30: Exp_ID: 133087; *R Learning*; Normalised Ordinal.

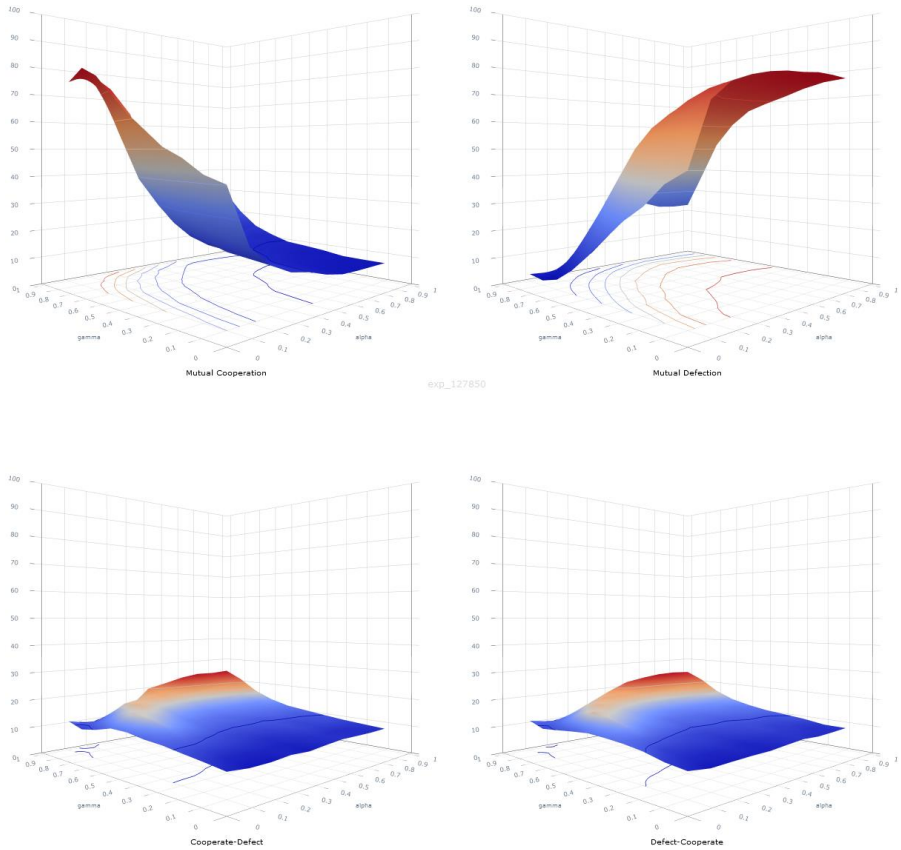


Figure B.31: Exp_ID: 127850; *SARSA*; Scalar.

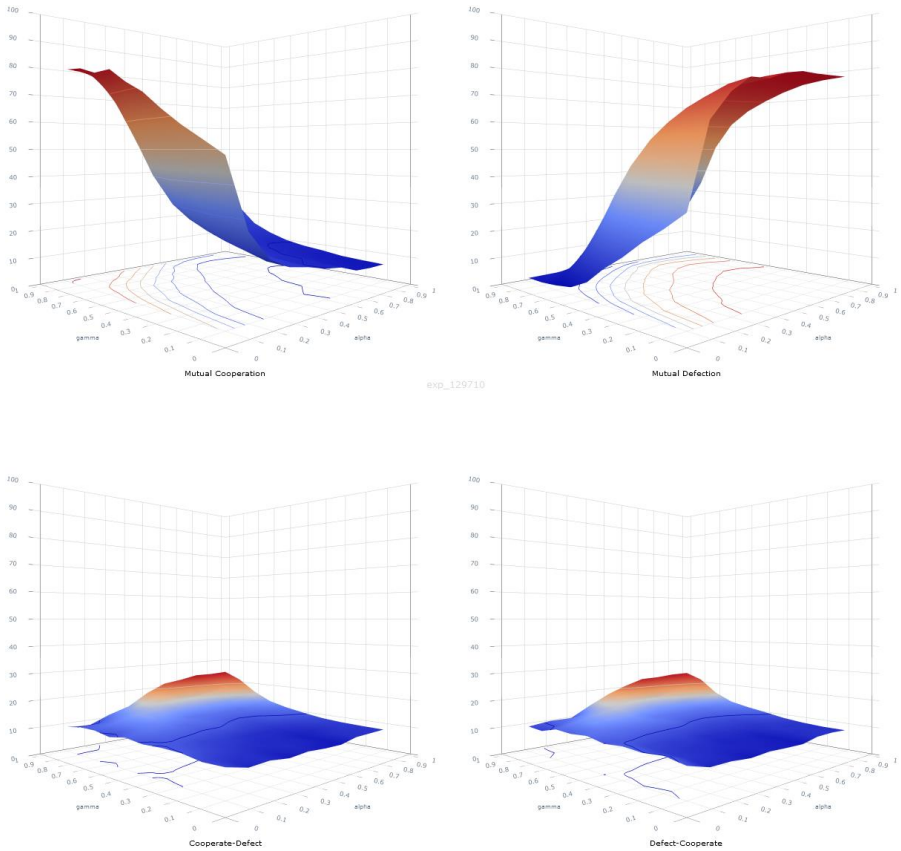


Figure B.32: Exp_ID: 129710; *SARSA*; Ordinal.

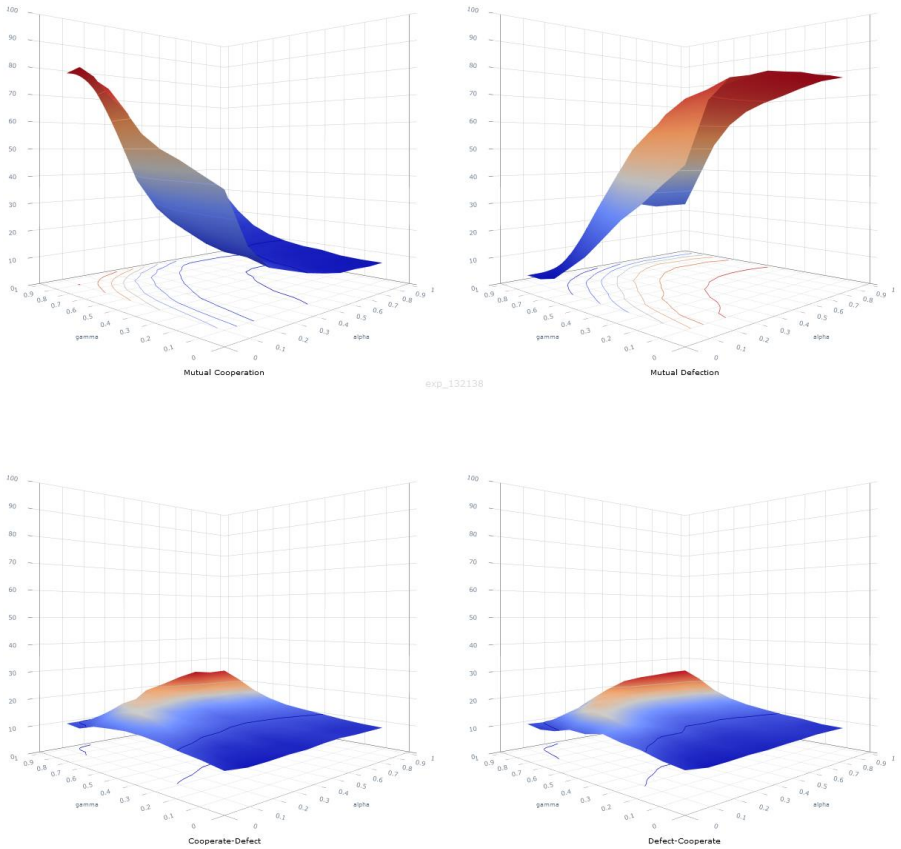


Figure B.33: Exp_ID: 132138; *SARSA*; Normalised Scalar.

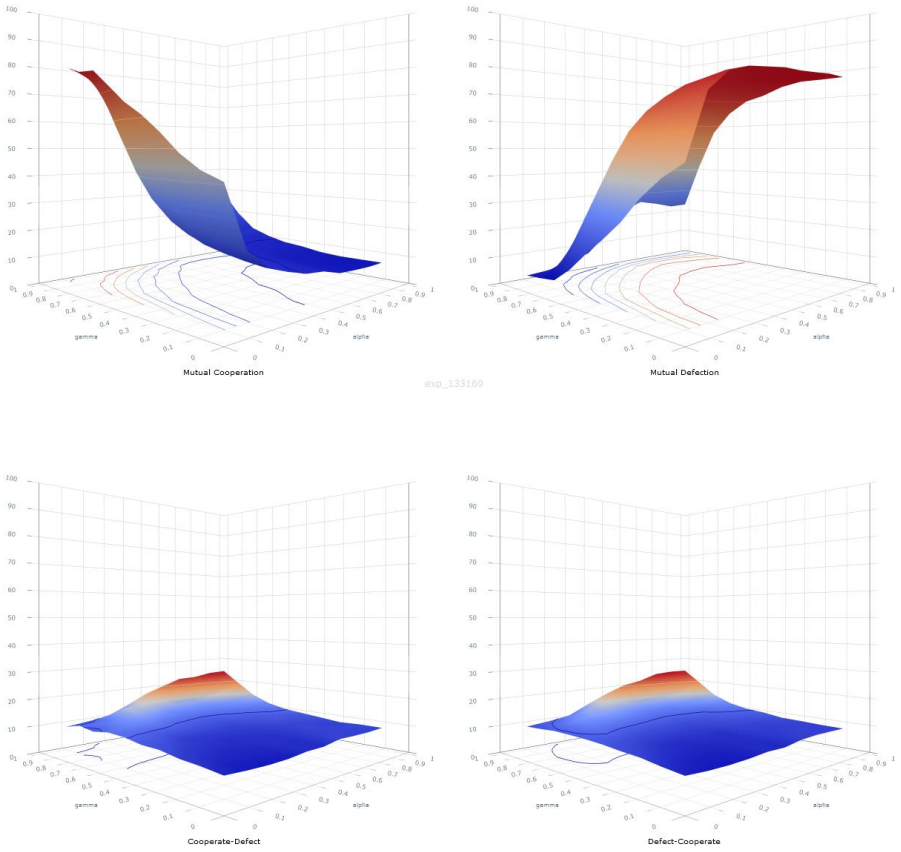


Figure B.34: Exp_ID: 133169; *SARSA*; Normalised Ordinal.

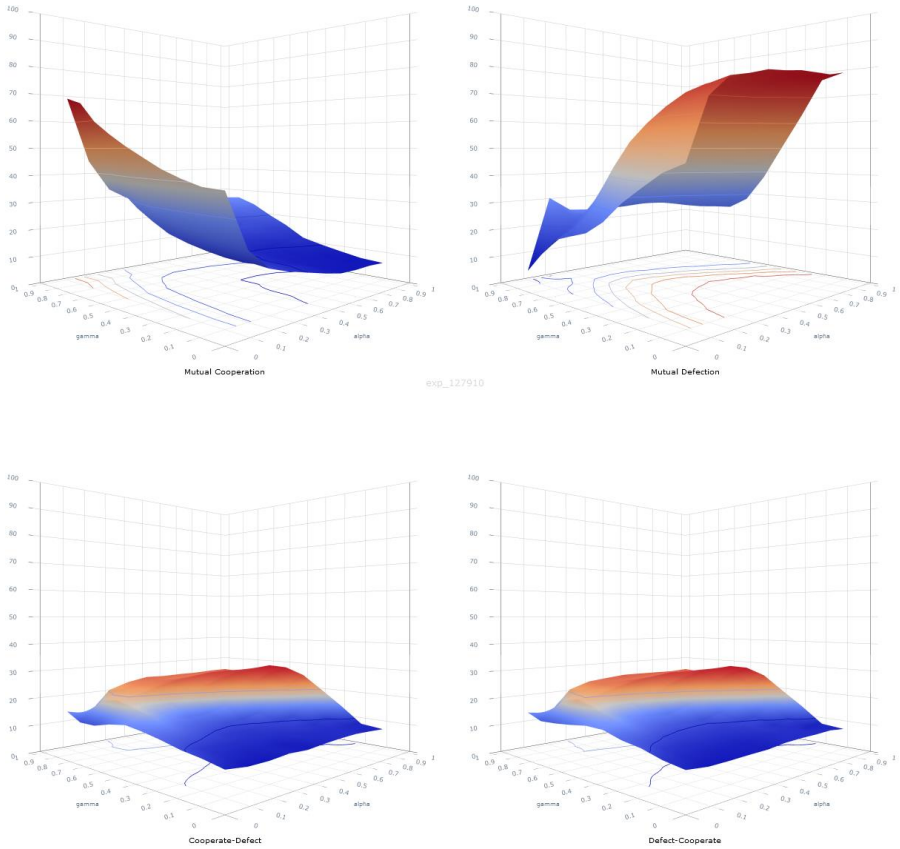


Figure B.35: Exp_ID: 127910; *SARSA Lambda*; Scalar.

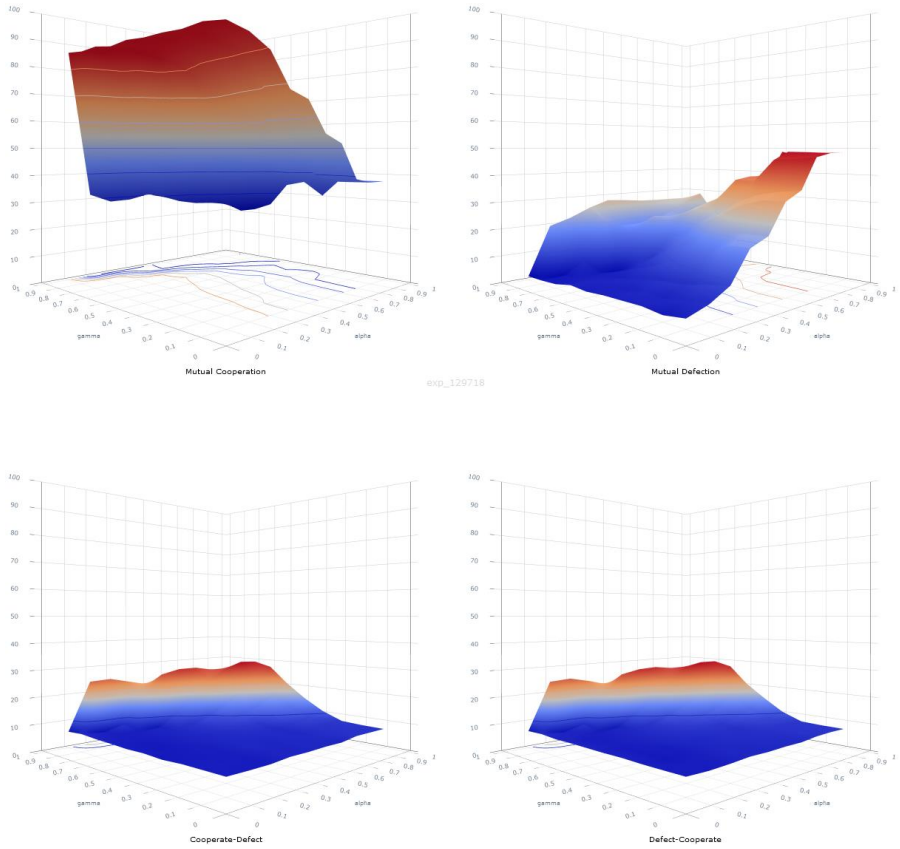


Figure B.36: Exp_ID: 129718; *SARSA Lambda*; Ordinal.

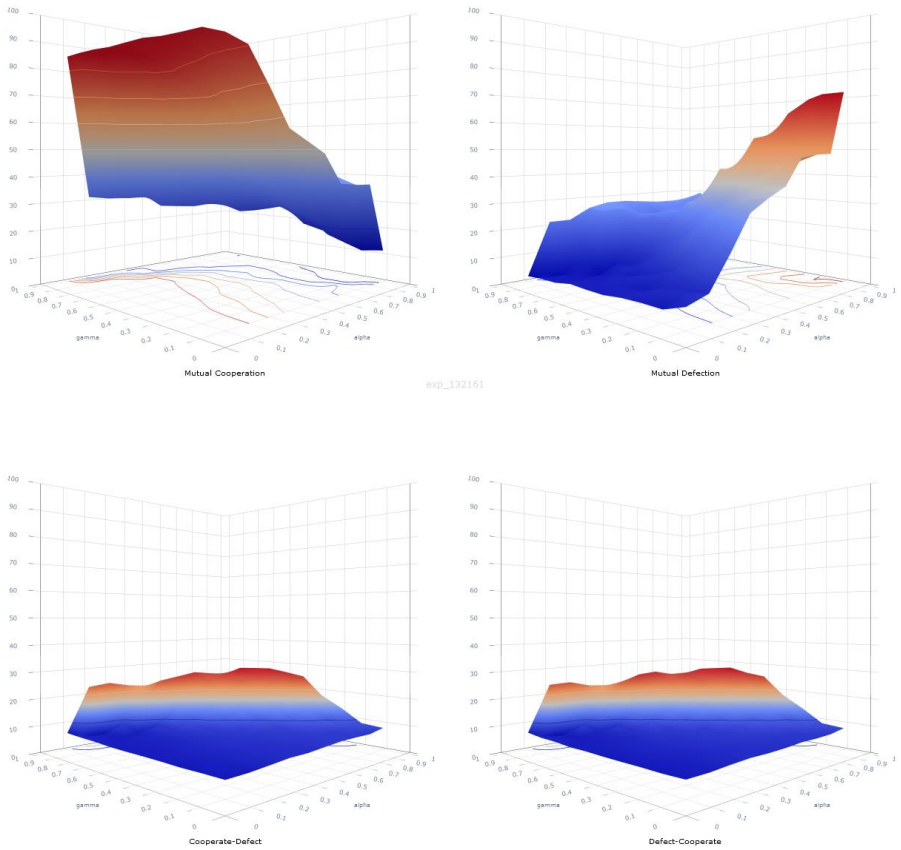


Figure B.37: Exp_ID: 132161; *SARSA Lambda*; Normalised Scalar.

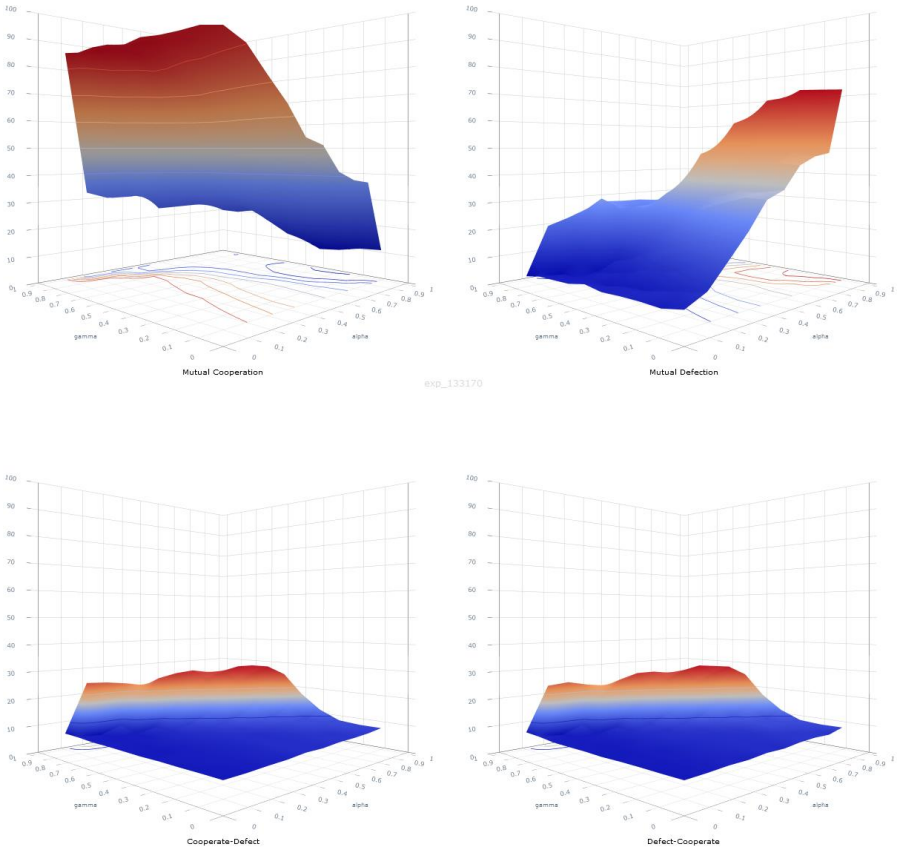


Figure B.38: Exp_ID: 133170; *SARSA Lambda*; Normalised Ordinal.

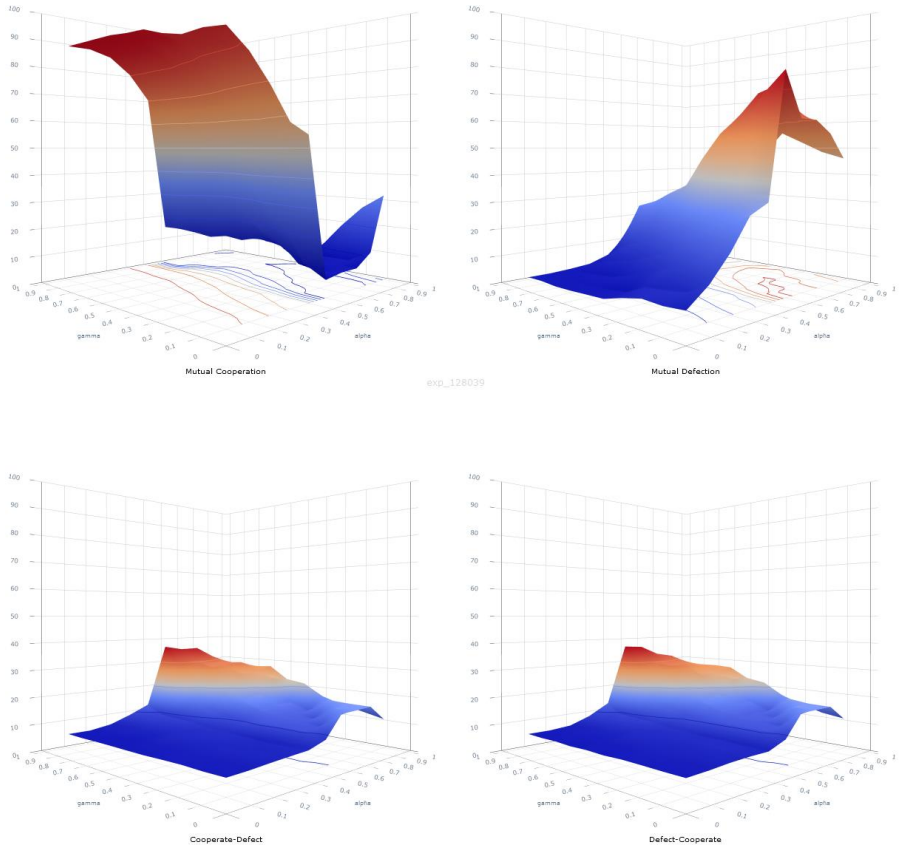


Figure B.39: Exp_ID: 128039; *SARSA Lambda, with Replacing Traces*; Scalar.

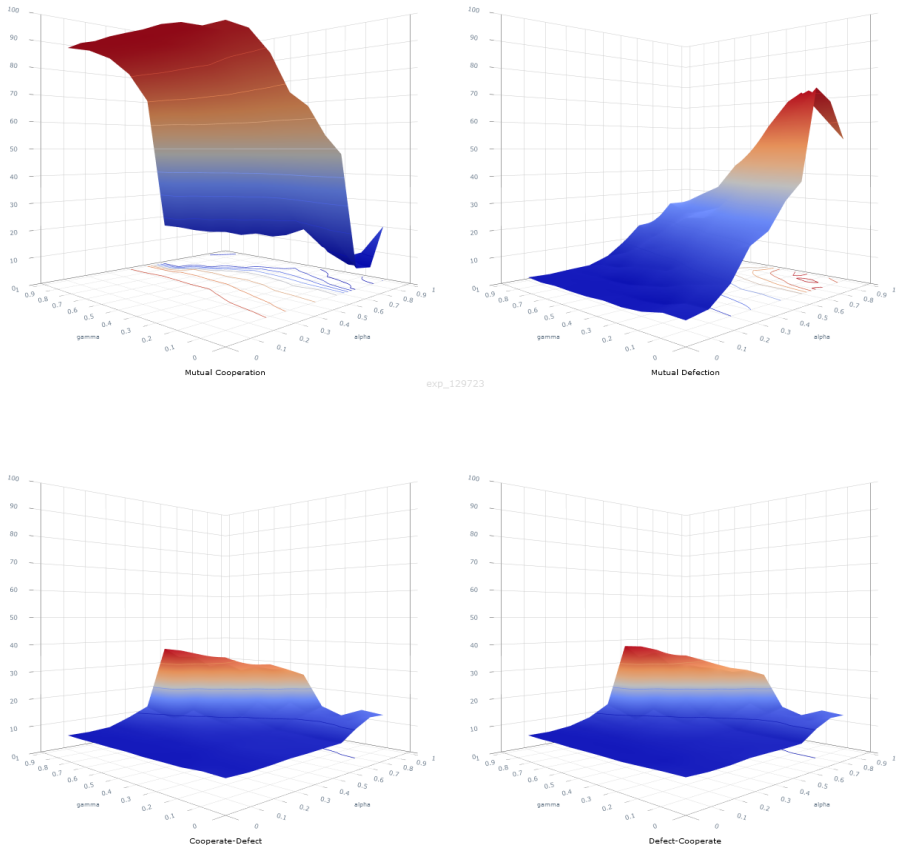


Figure B.40: Exp_ID: 129723; *SARSA Lambda, with Replacing Traces*; Ordinal.

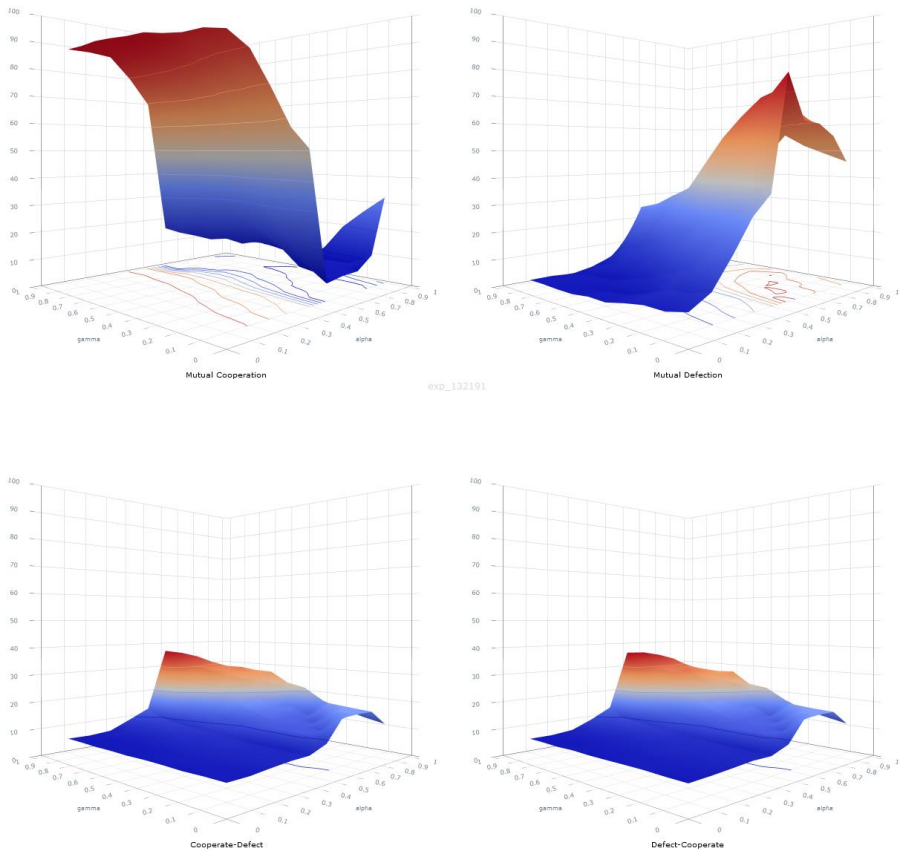


Figure B.41: Exp_ID: 132191; *SARSA Lambda, with Replacing Traces*; Normalised Scalar.

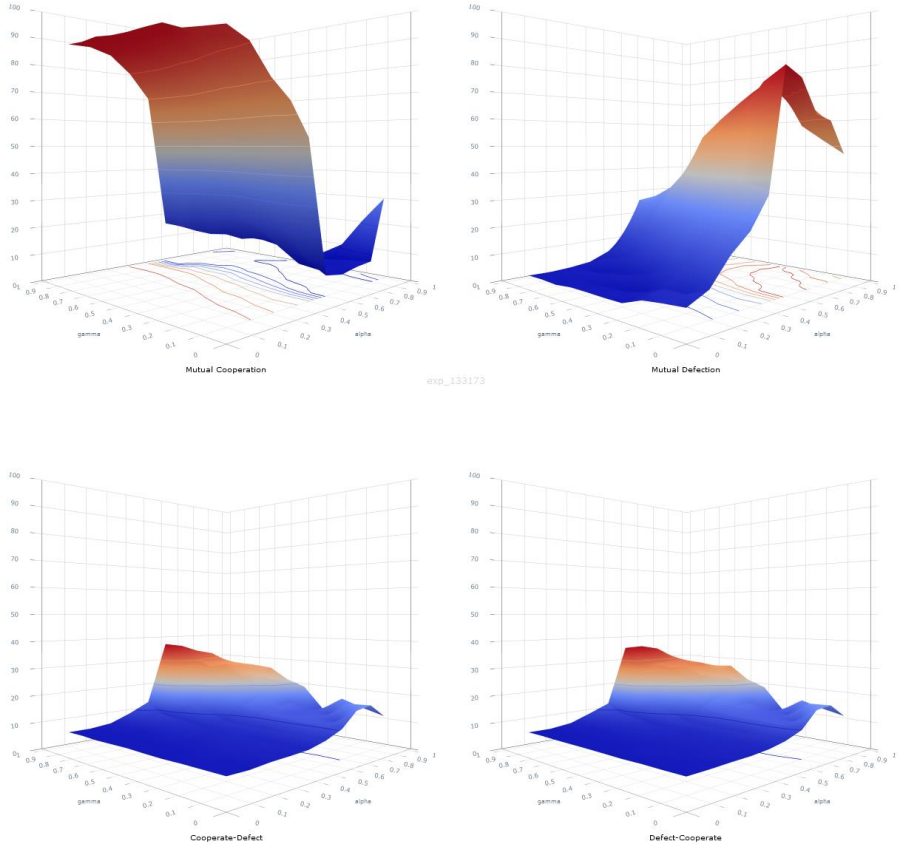


Figure B.42: Exp_ID: 133173; *SARSA Lambda, with Replacing Traces*; Normalised Ordinal.

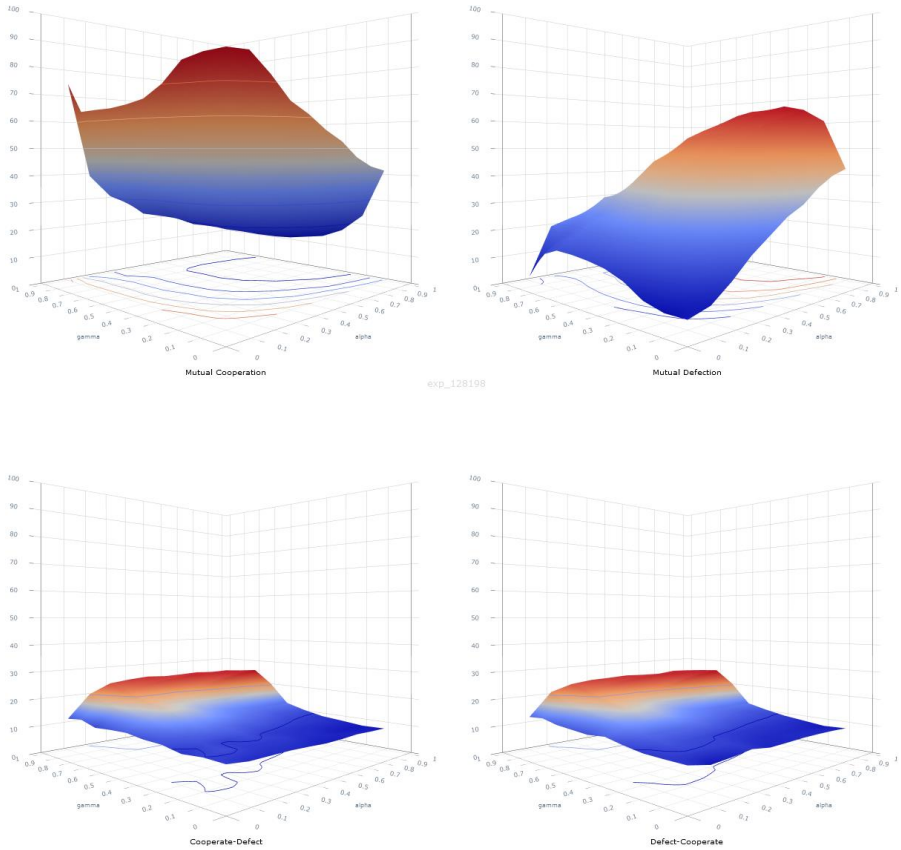


Figure B.43: Exp_ID: 128198; *Watkins (naïve) Q, Lambda*; Scalar.

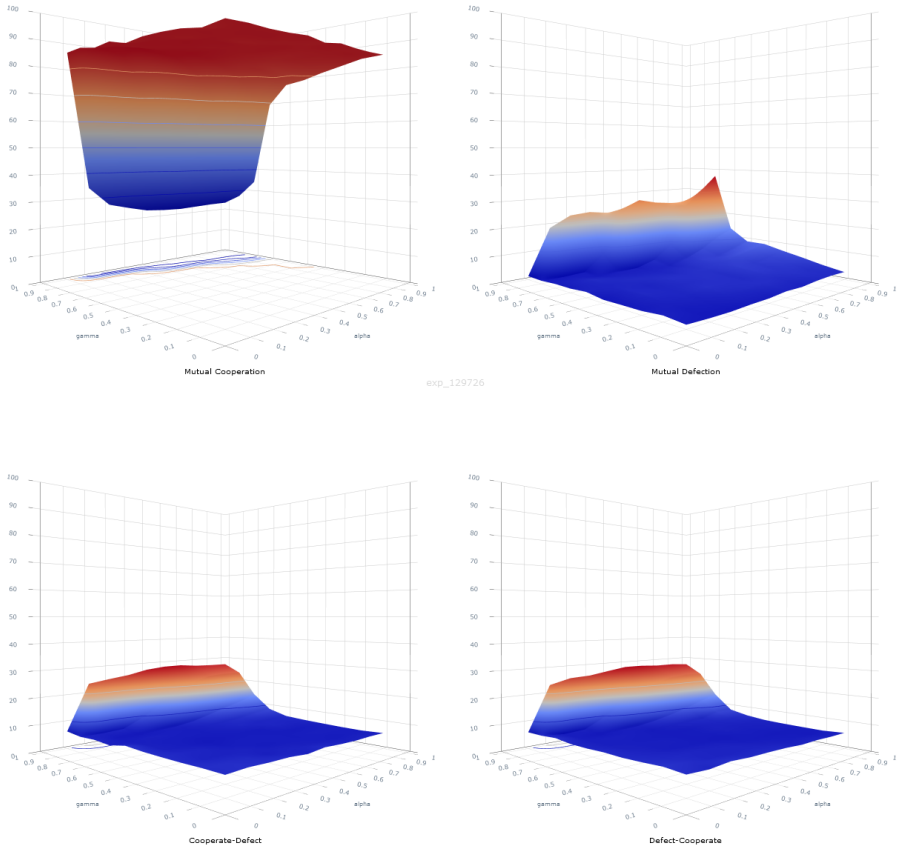


Figure B.44: Exp_ID: 129726; *Watkins (naïve) Q, Lambda*; Ordinal.

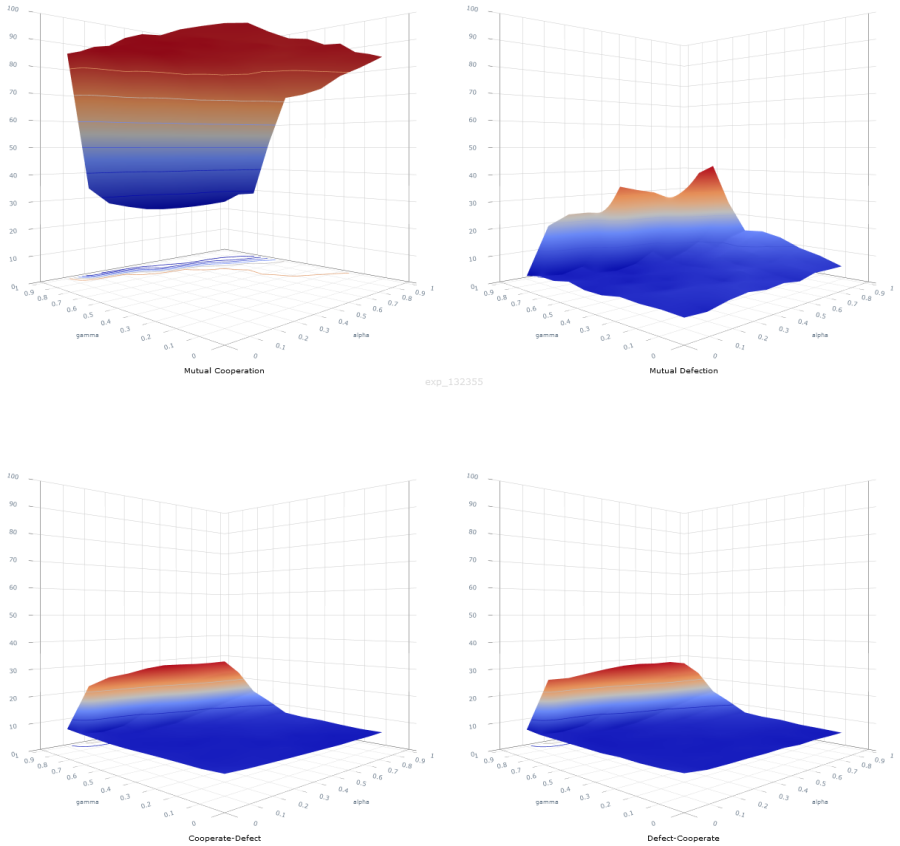


Figure B.45: Exp_ID: 132355; *Watkins (naïve) Q, Lambda*; Normalised Scalar.

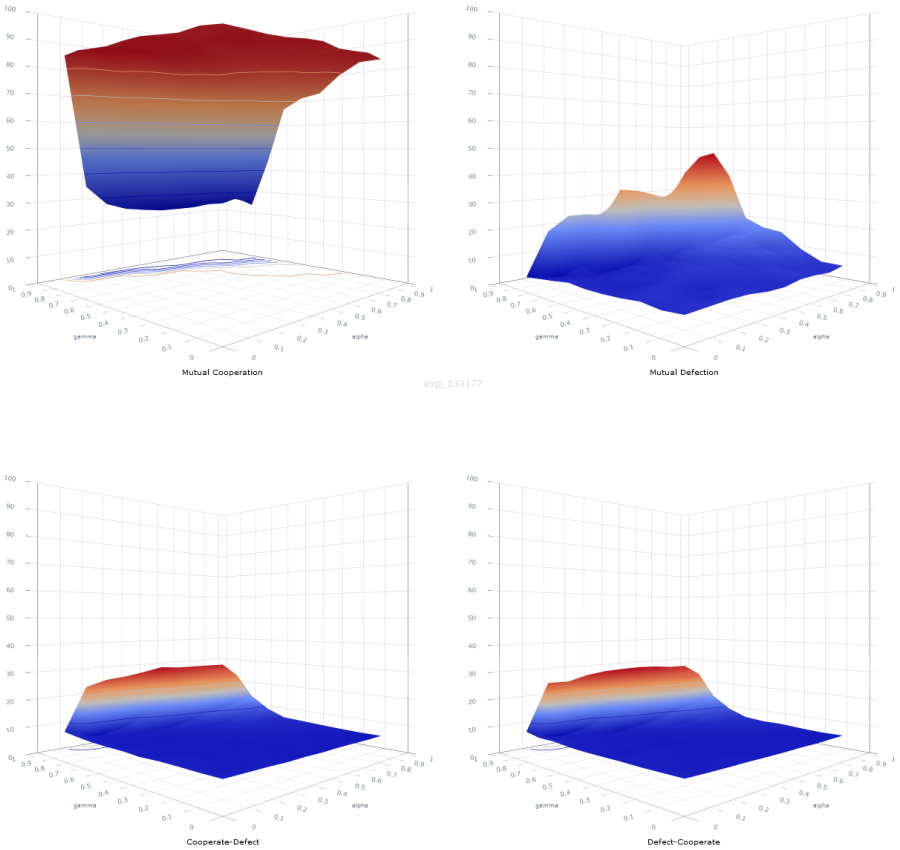


Figure B.46: Exp_ID: 133177; *Watkins (naïve) Q, Lambda*; Normalised Ordinal.

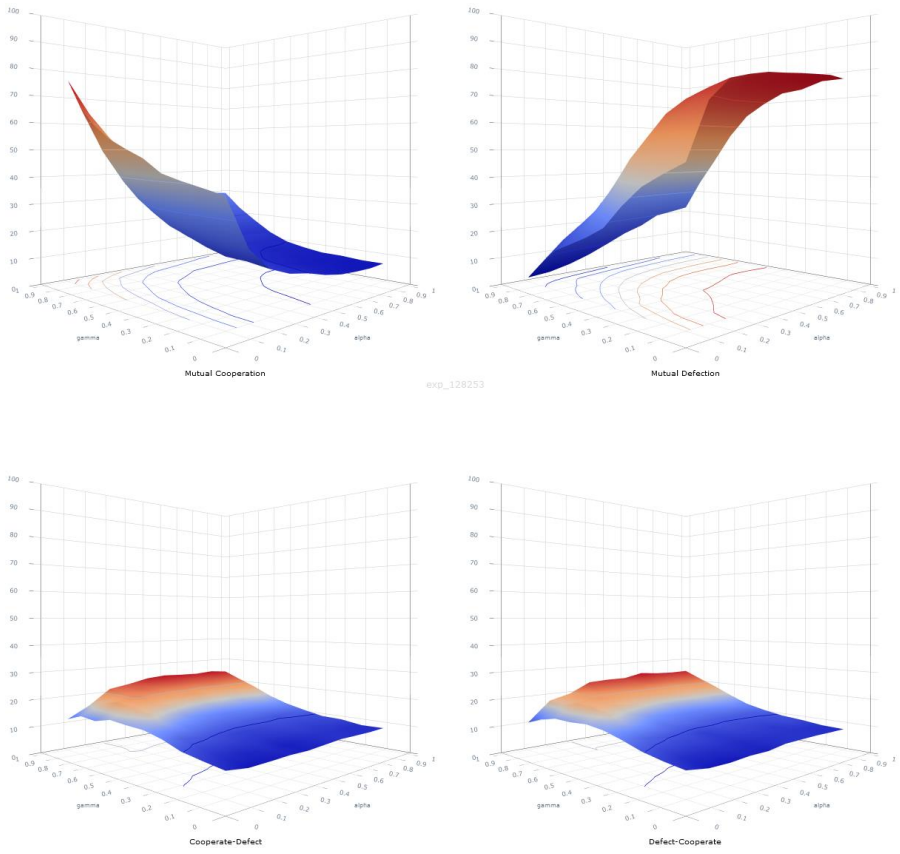


Figure B.47: Exp_ID: 128253; *Watkins (naïve) Q, Lambda, Replacing Traces*; Scalar.

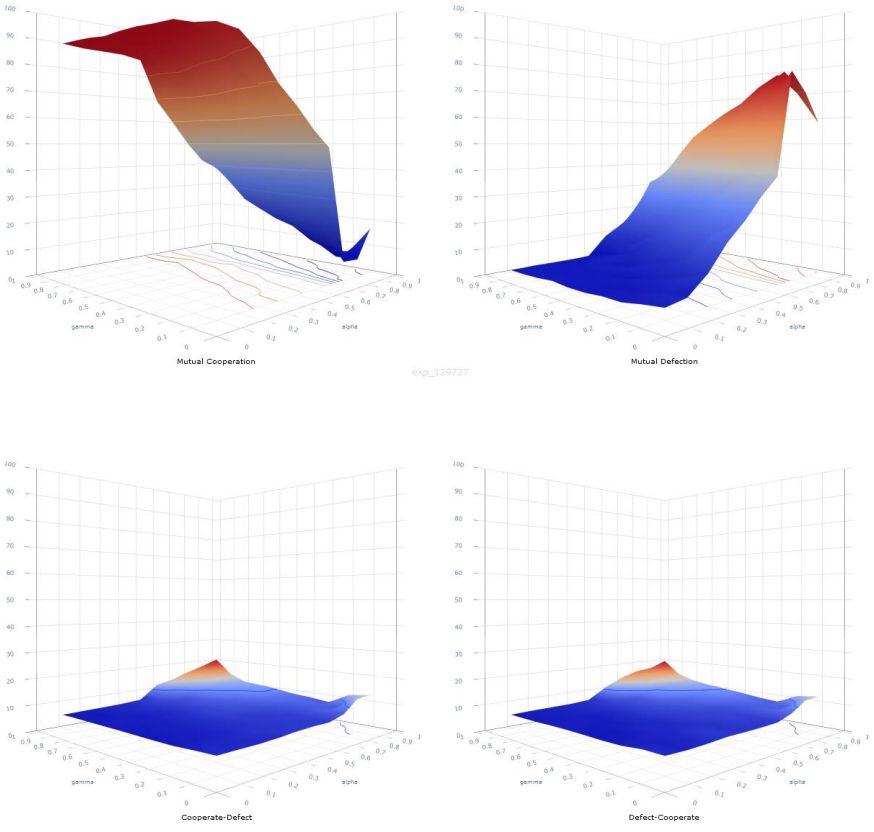


Figure B.48: Exp_ID: 129727; *Watkins (naïve) Q, Lambda, Replacing Traces*; Ordinal.

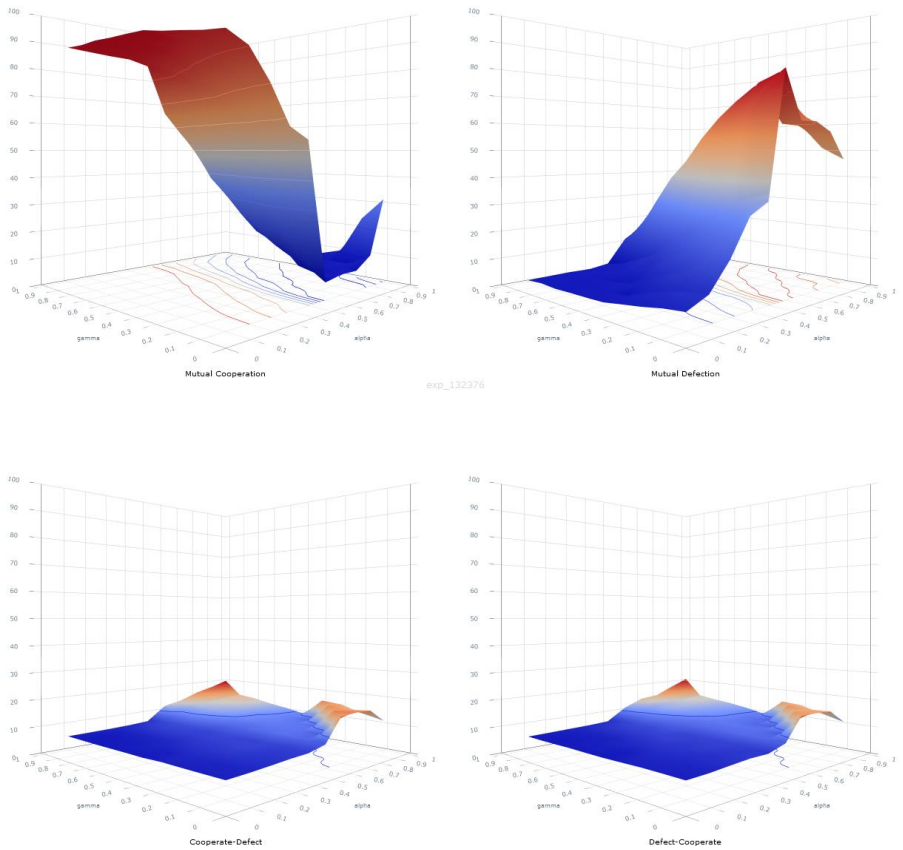


Figure B.49: Exp_ID: 132376; *Watkins (naïve) Q, Lambda, Replacing Traces*; Normalised Scalar.

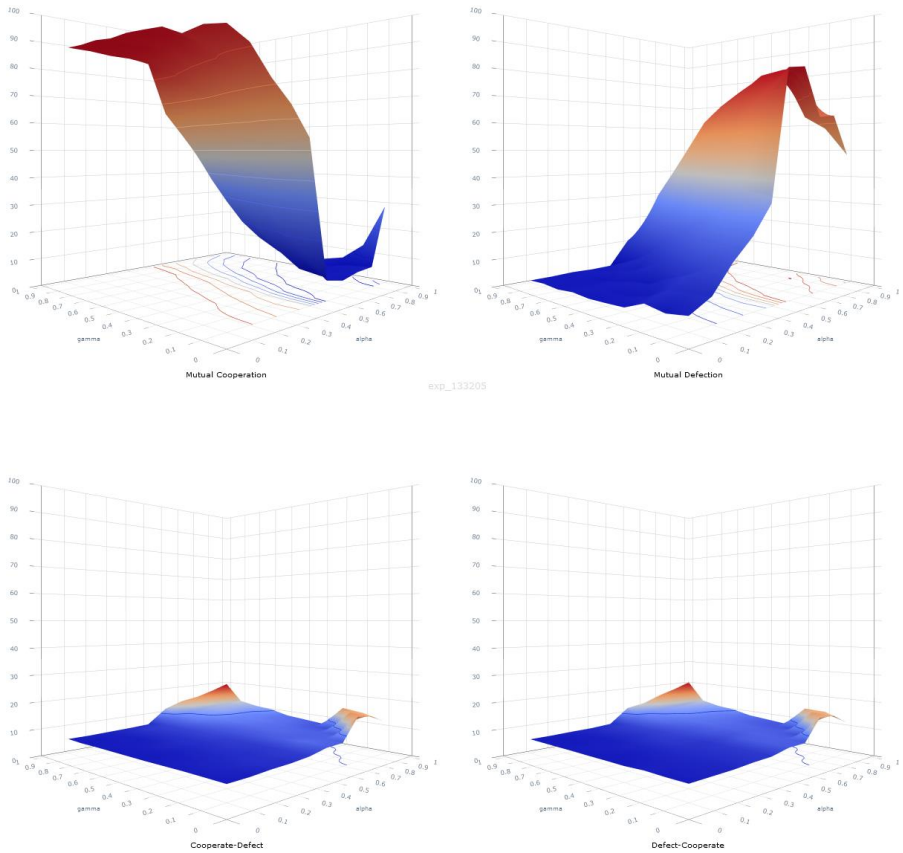


Figure B.50: Exp_ID: 133205; *Watkins (naïve) Q, Lambda, Replacing Traces*; Normalised Ordinal.

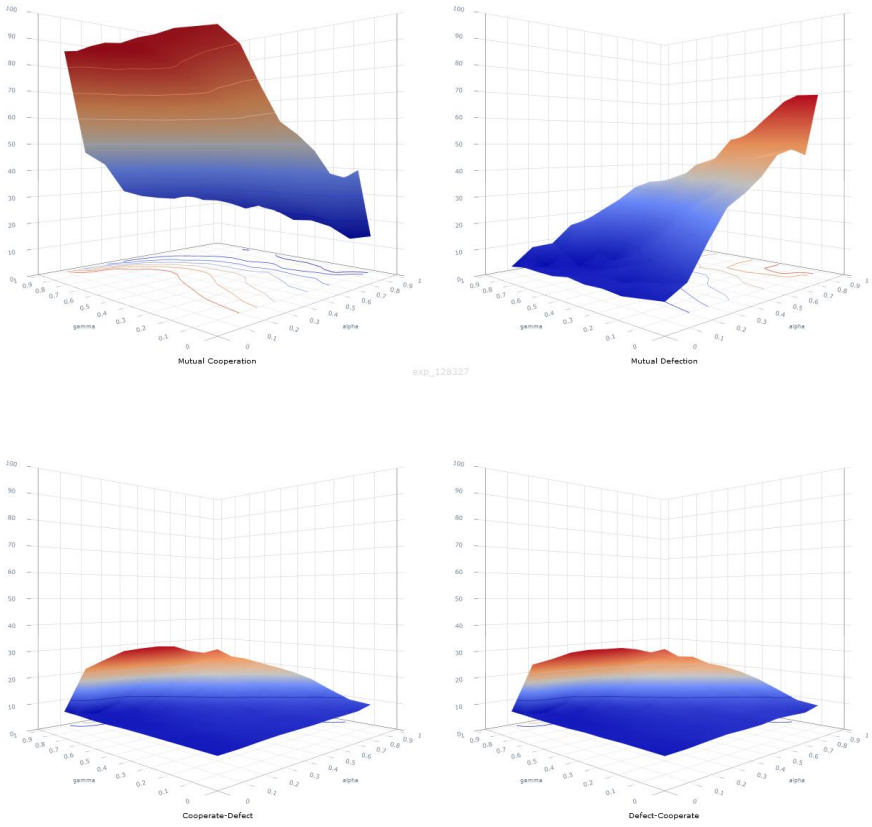


Figure B.51: Exp_ID: 128327; *Watkins Q*, *Lambda*; Scalar.

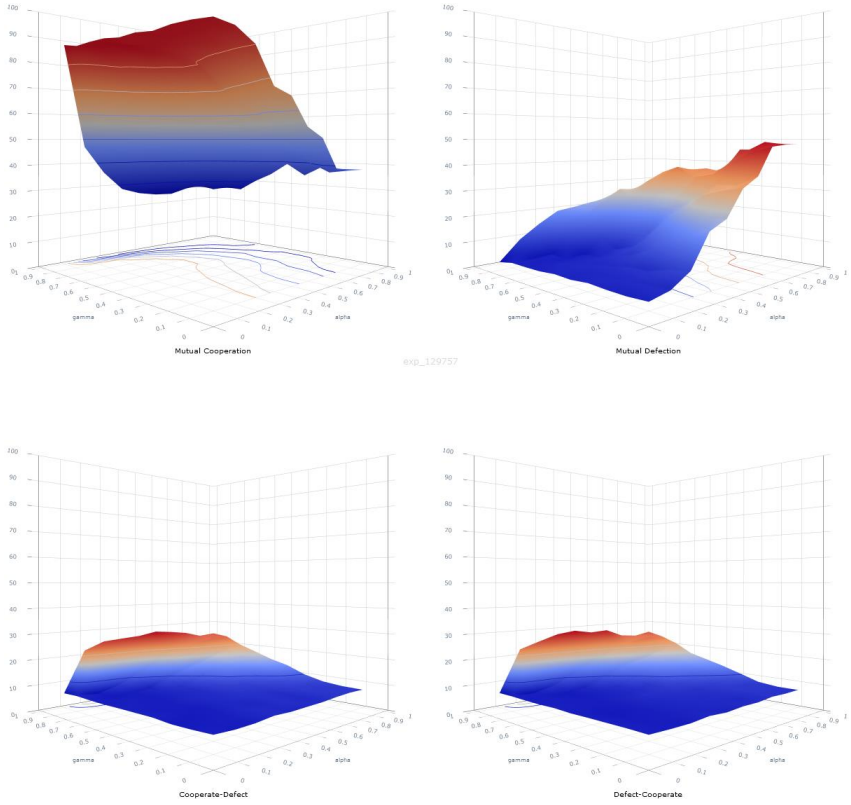


Figure B.52: Exp_ID: 129757; *Watkins Q*, *Lambda*; Ordinal.

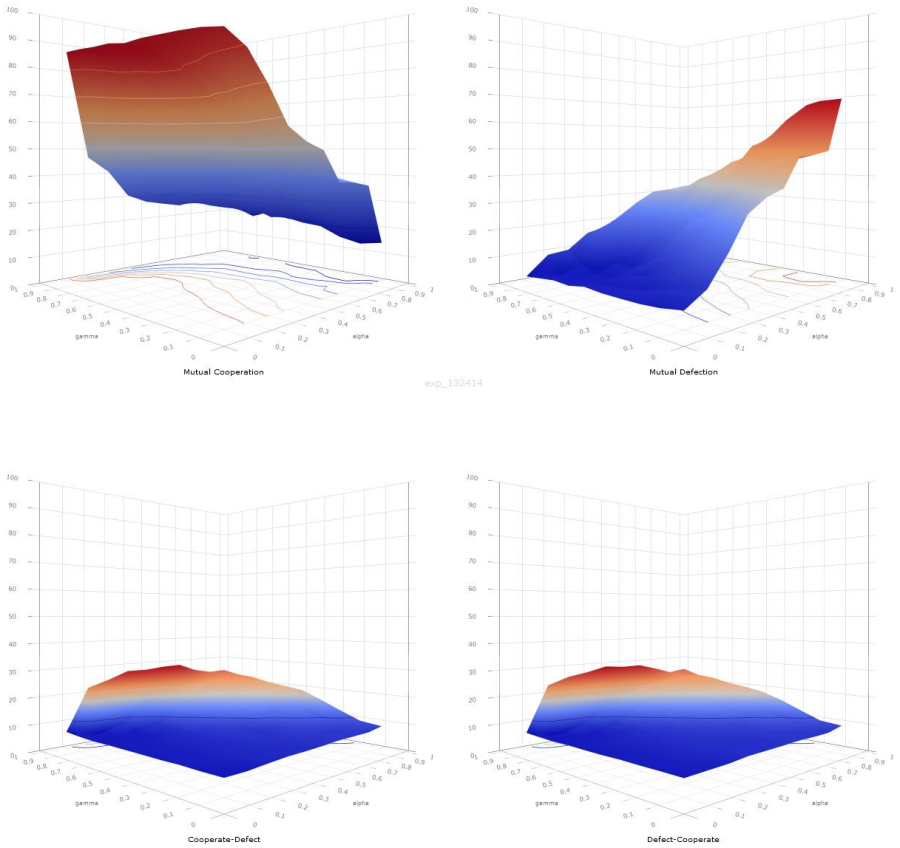


Figure B.53: Exp_ID: 132414; *Watkins Q, Lambda*; Normalised Scalar.

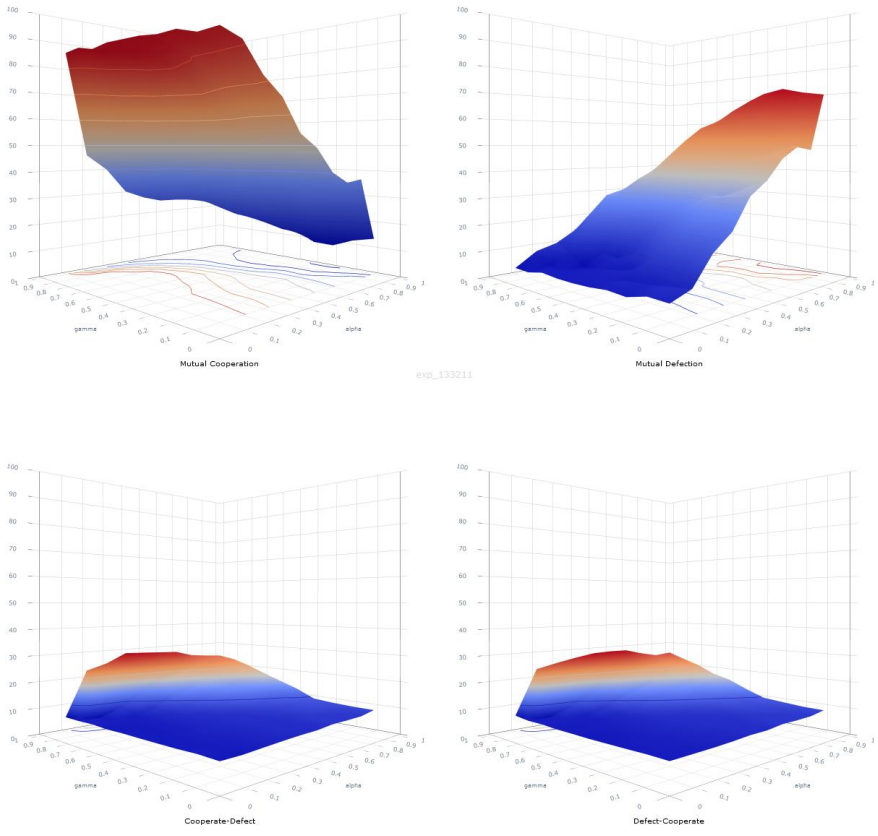


Figure B.54: Exp_ID: 133211; *Watkins Q, Lambda*; Normalised Ordinal.

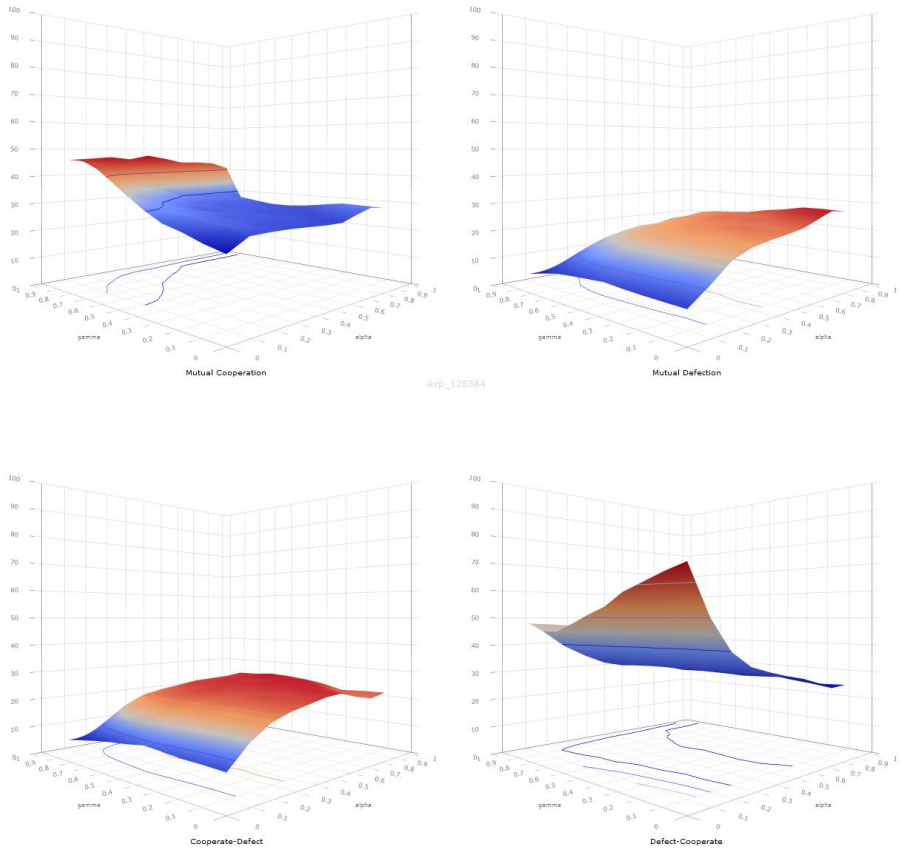


Figure B.55: Exp_ID: 128384; *Watkins Q, Linear Function Approximation*; Scalar.

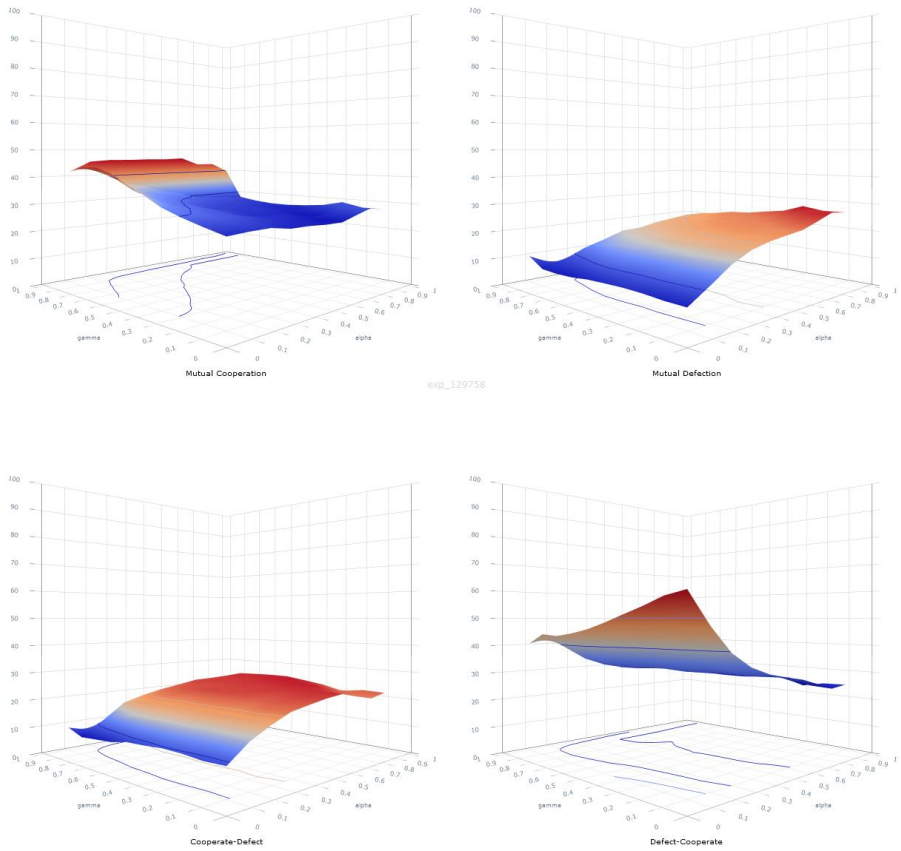


Figure B.56: Exp_ID: 129758; *Watkins Q, Linear Function Approximation*; Ordinal.

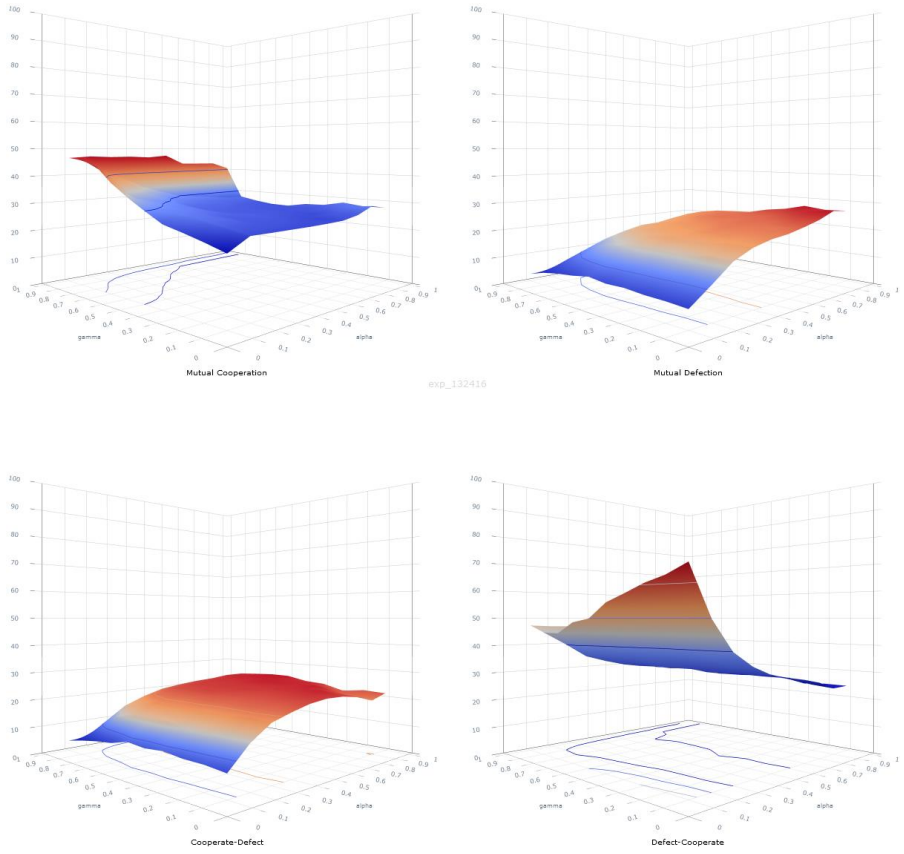


Figure B.57: Exp_ID: 132416; *Watkins Q, Linear Function Approximation*; Normalised Scalar.

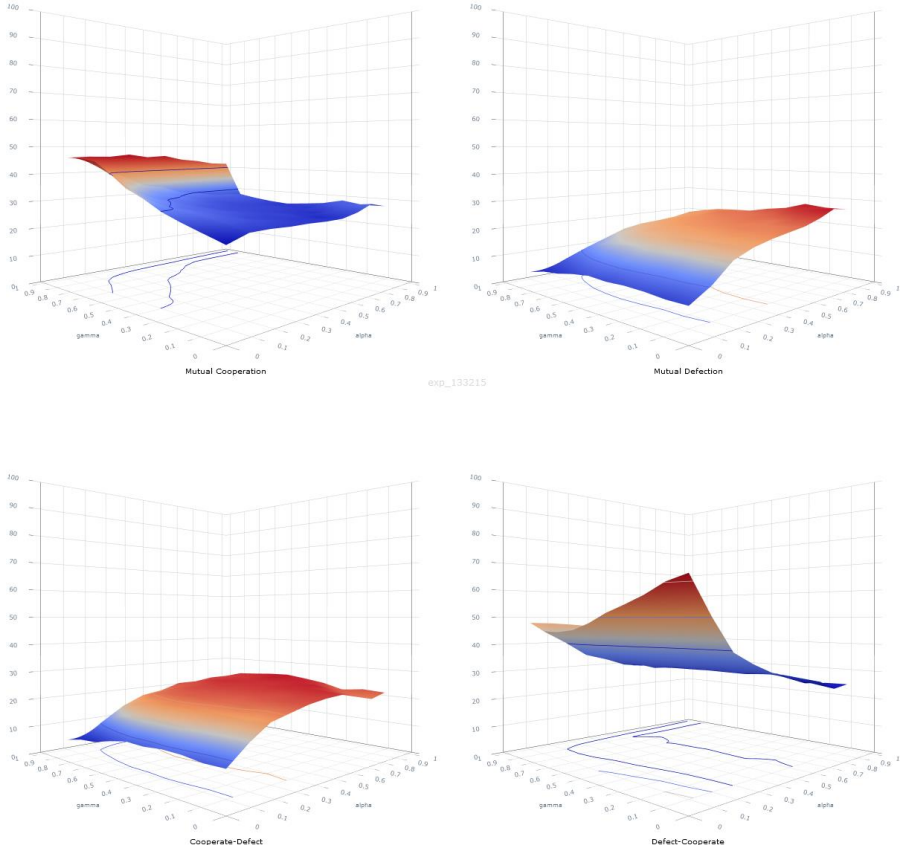


Figure B.58: Exp_ID: 133215; *Watkins Q, Linear Function Approximation*; Normalised Ordinal.

B.4 Experiment Series Three

B.4.1 Experiment IDs & Analysis Datasheets

Table B.53: Experiment Series Three Experiment IDs.

Experiment Series Three Experiment IDs					
Exp ID	**Target**	**Model**	**Job Count**	**Episodes**	**Timesteps**
477295	All *rRGS*	All *rRGS*	144	100	1000
14	g111	pd.canon.1	1	100	1000
15	g322	sh.canon.1	1	100	1000
16	g122	ch.canon.1	1	100	1000
18	g111	pd.canon.2	1	100	1000
19	g322	sh.canon.2	1	100	1000
20	g311	cn.canon.2	1	100	1000
21	g122	ch.canon.2	1	100	1000

Table B.54: Experiment Series Three Datasheets. Datasheets are packaged with a data release as per Appendix B.5.

Experiment Series Three Datasheets		
Item	**Filename**	**Description**
1	es3_eg2a.xlsx	Summary

Table B.55: Experiment Series Three Datasets. In change to previous experiment series, journal files are located in the experiment instances tarfile. Only PBS jobs (6-digit JOB_IDs) have pbs output files.

Experiment Series Three BasePath="/ES3/${Dir}/${Dir2}"					
Item	**Description**	**Data Type**	**Dir**	**Dir2**	**Size (MB)**
1	experiment data	Raw Episode/ Timestep	data		842
2	results data	Orderset, Summaries	results		0.9
3	pbs output files; .OU, .ER	Text	pbs_output	{exp_id}	10

B.4.2 Supplementary Data

Table B.56: Gamelock by Reward, All Game Models. Strategy algorithm is *Q-Learning*, with asymmetric parameters (see §6.1).

Each episode is 1000 timesteps. gL is gamelock count, i.e., the act of recognition. TTL is time-to-lock, i.e., how many timesteps until recognition occurs.

† Standard Deviation (σ) for population (all episodes, all game models) this row only, all others sample.
†† Standard Deviation (σ) over all episodes for the thirty-six game models in each layer.
††† Standard Deviation (σ) over all episodes for each individual game in each group.

Gamelock by Reward, Observer Summary
Q-Learning, RGS, 1k

Model	Episodes	gL Count	gL Rate	Mean TTL	TTL σ	Min TTL	Max TTL
All †	14400	14219	.987	103.62	149.15	4	999
Layer 1 ††	3600	3591	.998	70.75	91.42	4	989
Layer 2	3600	3545	.985	133.22	181.7	4	999
Layer 3	3600	3494	.971	139.05	182.2	4	998
Layer 4	3600	3589	.997	71.47	101	4	998
g111 †††	100	100	1	48.93	33.44	4	161
g112	100	100	1	58.68	47.48	6	317
g113	100	100	1	53.29	39.2	7	218
g114	100	100	1	60.71	43.8	4	246
g115	100	100	1	54.01	40.17	8	222
g116	100	100	1	53.21	30.53	4	136
g161	100	100	1	91.47	147.75	4	870
g162	100	100	1	80.77	122.62	6	989
g163	100	98	.98	91.49	106.08	7	632
g164	100	99	.99	91.31	130.53	5	889
g165	100	99	.99	88.09	120.21	9	751
g166	100	100	1	80.86	103.57	8	617
g151	100	100	1	60.88	75.95	8	666
g152	100	99	.99	79.08	130.42	4	881
g153	100	100	1	65.83	71.22	4	404
g154	100	100	1	96.1	123.53	5	724
g155	100	100	1	98.19	113.56	5	641
g156	100	100	1	80.05	111.43	5	841
g141	100	99	.99	102.2	161.2	6	912
g142	100	100	1	80.3	109.15	4	815
g143	100	99	.99	87.68	139.61	8	890
g144	100	99	.99	78.06	113.35	5	791
g145	100	99	.99	93.18	128.65	4	708
g146	100	100	1	76.81	101.21	5	834
g131	100	100	1	59.62	42.43	8	245
g132	100	100	1	52.13	31.77	4	153
g133	100	100	1	61.53	36.7	9	202
g134	100	100	1	67.27	99.07	5	898
g135	100	100	1	61.18	42.73	7	225
g136	100	100	1	57.8	45.2	4	201
g121	100	100	1	52.91	34.64	9	206
g122	100	100	1	58.85	43.68	5	251
g123	100	100	1	53.19	38.91	4	198
g124	100	100	1	64.13	56.91	6	366
g125	100	100	1	53.73	42.11	6	246
g126	100	100	1	53.31	42.45	4	198

Gamelock by Reward, Observer Summary Q-Learning, RGS, 1k							
Model	Episodes	gL Count	gL Rate	Mean TTL	TTL σ	Min TTL	Max TTL
g211	100	100	1	76.91	56.39	6	298
g212	100	100	1	71.28	53.61	7	305
g213	100	100	1	68.28	53.2	4	266
g214	100	100	1	80.7	58.14	12	288
g215	100	100	1	60.25	42.04	9	190
g216	100	99	.99	86.49	82.87	6	679
g261	100	93	.93	210.83	251.1	13	942
g262	100	93	.93	181.13	211.9	5	807
g263	100	98	.98	201.27	261.3	9	922
g264	100	98	.98	187.88	222	5	864
g265	100	100	1	201.68	219.3	11	883
g266	100	96	.96	226.99	259	9	951
g251	100	94	.94	173.57	199.27	5	869
g252	100	94	.94	177.28	234.87	5	973
g253	100	97	.97	181.31	226.59	7	999
g254	100	97	.97	164.54	202.37	6	805
g255	100	100	1	190.14	230.77	5	817
g256	100	100	1	225.77	262.91	6	922
g241	100	98	.98	234.07	269.49	7	997
g242	100	95	.95	209.83	255.23	4	967
g243	100	95	.95	217.27	243.25	7	967
g244	100	96	.96	159.36	192.32	4	855
g245	100	100	1	194.06	207.95	5	900
g246	100	100	1	208.22	248.88	9	959
g231	100	96	.96	63.09	49.44	4	227
g232	100	100	1	80.62	63.96	6	280
g233	100	100	1	70.31	57.08	6	311
g234	100	100	1	59.44	45.38	5	258
g235	100	100	1	55.29	40.82	4	211
g236	100	100	1	68.19	52.38	4	264
g221	100	100	1	70.19	57.86	4	287
g222	100	100	1	75.01	58.67	9	276
g223	100	100	1	66.07	45.38	10	268
g224	100	100	1	56.96	49.46	5	291
g225	100	100	1	69.05	48.35	7	197
g226	100	100	1	72.54	58.62	8	299
g311	100	100	1	68.73	54.13	5	259
g312	100	100	1	80.57	62.25	6	291
g313	100	100	1	75.76	63.9	5	290
g314	100	100	1	74.55	53.84	5	276
g315	100	100	1	70.99	57.57	6	347
g316	100	100	1	76.96	59.27	6	329
g361	100	93	.93	198.7	207	4	832
g362	100	90	.9	184.61	206.6	8	949
g363	100	93	.93	185.72	236.7	4	966
g364	100	92	.92	161.24	204.1	4	870
g365	100	99	.99	196.78	238.6	6	977
g366	100	99	.99	215.06	246.7	4	921
g351	100	97	.97	250.93	250	4	972
g352	100	92	.92	223.51	246.43	4	978
g353	100	94	.94	229.99	249.38	9	902
g354	100	91	.91	171.14	200.59	7	845
g355	100	94	.94	196.55	230.31	4	998
g356	100	98	.98	246.73	255.62	6	989
g341	100	94	.94	199.47	215.97	7	815
g342	100	95	.95	212.7	225.3	4	903
g343	100	88	.88	149.42	211.01	7	869
g344	100	98	.98	191.74	208.42	5	907
g345	100	93	.93	207.34	278.43	6	928

Gamelock by Reward, Observer Summary Q-Learning, RGS, 1k							
Model	Episodes	gL Count	gL Rate	Mean TTL	TTL σ	Min TTL	Max TTL
g346	100	94	.94	236.5	263.76	7	941
g331	100	100	1	81.78	56.23	5	255
g332	100	100	1	70.54	53.87	6	271
g333	100	100	1	89.57	79.73	9	584
g334	100	100	1	79.53	68.39	5	477
g335	100	100	1	70.86	53.71	5	264
g336	100	100	1	68.32	58.72	4	435
g321	100	100	1	80.14	56.84	7	304
g322	100	100	1	65.42	56.99	4	285
g323	100	100	1	72.92	57.13	5	397
g324	100	100	1	68.09	49.03	7	243
g325	100	100	1	69.62	53.12	4	260
g326	100	100	1	83.25	76.24	4	401
g411	100	100	1	59.05	43.78	6	274
g412	100	100	1	51.78	32.79	6	194
g413	100	100	1	50.22	35.28	7	191
g414	100	100	1	49.98	44.2	4	391
g415	100	100	1	54.84	45.93	6	251
g416	100	100	1	54.11	46.52	4	314
g461	100	99	.99	100.17	153.99	5	952
g462	100	100	1	124.47	199.64	7	935
g463	100	100	1	112.66	174.16	8	915
g464	100	98	.98	76.38	126.1	6	998
g465	100	99	.99	72.24	124.24	7	952
g466	100	99	.99	75.34	92.82	7	567
g451	100	100	1	74.19	122.7	4	931
g452	100	100	1	85.09	122.03	5	946
g453	100	100	1	80.65	122.78	6	874
g454	100	100	1	97.05	127.22	5	742
g455	100	97	.97	100.98	161.24	6	869
g456	100	99	.99	74.52	120.81	6	949
g441	100	100	1	82.11	138.5	8	963
g442	100	98	.98	100.95	137.62	7	697
g443	100	100	1	70.49	91.03	7	619
g444	100	100	1	91.68	127.77	6	714
g445	100	100	1	91.15	131.96	6	717
g446	100	100	1	83.51	110.94	5	572
g431	100	100	1	50.61	36.51	5	190
g432	100	100	1	48.8	31.93	5	173
g433	100	100	1	63.91	52.2	5	360
g434	100	100	1	61.87	44.61	7	202
g435	100	100	1	53.54	31.74	4	178
g436	100	100	1	53.6	36.75	4	222
g421	100	100	1	56.31	34.31	4	200
g422	100	100	1	57.59	35.73	7	188
g423	100	100	1	57.14	39.2	6	212
g424	100	100	1	48.63	35.98	8	219
g425	100	100	1	51.52	34.17	5	156
g426	100	100	1	55.62	35.77	6	185

Table B.57: Interlock by Reward, Agent Zero, All Game Models. Strategy algorithm is *Q-Learning*, with asymmetric parameters (see §6.1).

Each episode is 1000 timesteps. ***gL*** is gamelock count, i.e., the act of recognition. ***TTL*** is time-to-lock, i.e., how many timesteps until recognition occurs.

† Standard Deviation (σ) for population (all episodes, all game models) this row only, all others sample.
†† Standard Deviation (σ) over all episodes for the thirty-six game models in each layer.
††† Standard Deviation (σ) over all episodes for each individual game in each group.

Interlock by Reward, Agent Zero Q-Learning, RGS, 1k							
Model	**Episodes**	***gL* Count**	***gL* Rate**	**Mean TTL**	**TTL σ**	**Min TTL**	**Max TTL**
All †	14400	14221	.988	103.76	149.15	4	1000
Layer 1 ††	3600	3591	.998	70.75	91.42	4	989
Layer 2	3600	3545	.985	133.22	181.74	4	999
Layer 3	3600	3494	.971	139.6	183.26	4	1000
Layer 4	3600	3589	.997	71.47	100.96	4	998
g111 †††	100	100	1	48.93	33.44	4	161
g112	100	100	1	58.68	47.48	6	317
g113	100	100	1	53.29	39.2	7	218
g114	100	100	1	60.71	43.8	4	246
g115	100	100	1	54.01	40.17	8	222
g116	100	100	1	53.21	30.53	4	136
g161	100	100	1	91.47	147.8	4	870
g162	100	100	1	80.77	122.6	6	989
g163	100	98	.98	91.49	106.1	7	632
g164	100	99	.99	91.31	130.5	5	889
g165	100	99	.99	88.09	120.2	9	751
g166	100	100	1	80.86	103.6	8	617
g151	100	100	1	60.88	75.95	8	666
g152	100	99	.99	79.08	130.4	4	881
g153	100	100	1	65.83	71.22	4	404
g154	100	100	1	96.1	123.5	5	724
g155	100	100	1	98.19	113.6	5	641
g156	100	100	1	80.05	111.4	5	841
g141	100	99	.99	102.2	161.2	6	912
g142	100	100	1	80.3	109.1	4	815
g143	100	99	.99	87.68	139.6	8	890
g144	100	99	.99	78.06	113.3	5	791
g145	100	99	.99	93.18	128.7	4	708
g146	100	100	1	76.81	101.2	5	834
g131	100	100	1	59.62	42.43	8	245
g132	100	100	1	52.13	31.77	4	153
g133	100	100	1	61.53	36.7	9	202
g134	100	100	1	67.27	99.07	5	898
g135	100	100	1	61.18	42.73	7	225
g136	100	100	1	57.8	45.24	4	201
g121	100	100	1	52.91	34.63	9	206
g122	100	100	1	58.85	43.68	5	251
g123	100	100	1	53.19	38.91	4	198
g124	100	100	1	64.13	56.91	6	366
g125	100	100	1	53.73	42.11	6	246
g126	100	100	1	53.31	42.45	4	198
g211	100	100	1	76.91	56.38	6	298
g212	100	100	1	71.28	53.61	7	305
g213	100	100	1	68.28	53.2	4	266
g214	100	100	1	80.7	58.14	12	288
g215	100	100	1	60.25	42.04	9	190
g216	100	99	.99	86.49	82.86	6	679

Interlock by Reward, Agent Zero Q-Learning, RGS, 1k							
Model	Episodes	gL Count	gL Rate	Mean TTL	TTL σ	Min TTL	Max TTL
g261	100	93	.93	210.83	251.1	13	942
g262	100	93	.93	181.13	211.9	5	807
g263	100	98	.98	201.27	261.3	9	922
g264	100	98	.98	187.88	222	5	864
g265	100	100	1	201.68	219.3	11	883
g266	100	96	.96	226.99	259	9	951
g251	100	94	.94	173.57	199.3	5	869
g252	100	97	.97	177.28	234.9	5	973
g253	100	97	.97	181.31	226.6	7	999
g254	100	100	1	164.54	202.4	6	805
g255	100	100	1	190.14	230.8	5	817
g256	100	98	.98	225.77	262.9	6	922
g241	100	95	.95	234.07	269.5	7	997
g242	100	95	.95	209.83	255.2	4	967
g243	100	96	.96	217.27	243.3	7	967
g244	100	100	1	159.36	192.3	4	855
g245	100	100	1	194.06	207.9	5	900
g246	100	96	.96	208.22	248.9	9	959
g231	100	100	1	63.09	49.44	4	227
g232	100	100	1	80.62	63.96	6	280
g233	100	100	1	70.31	57.08	6	311
g234	100	100	1	59.44	45.38	5	258
g235	100	100	1	55.29	40.82	4	211
g236	100	100	1	68.19	52.38	4	264
g221	100	100	1	70.19	57.86	4	287
g222	100	100	1	75.01	58.67	9	276
g223	100	100	1	66.07	45.38	10	268
g224	100	100	1	56.96	49.46	5	291
g225	100	100	1	69.05	48.34	7	197
g226	100	100	1	72.54	58.62	8	299
g311	100	100	1	68.73	54.13	5	259
g312	100	100	1	80.57	62.25	6	291
g313	100	100	1	75.76	63.9	5	290
g314	100	100	1	74.55	53.84	5	276
g315	100	100	1	70.99	57.57	6	347
g316	100	100	1	76.96	59.27	6	329
g361	100	93	.93	198.7	207	4	832
g362	100	90	.9	184.61	206.6	8	949
g363	100	93	.93	195.72	249.6	4	966
g364	100	92	.92	161.24	204.1	4	870
g365	100	99	.99	196.78	238.6	6	977
g366	100	99	.99	215.06	246.7	4	921
g351	100	97	.97	250.93	250	4	972
g352	100	92	.92	223.51	246.4	4	978
g353	100	94	.94	229.99	249.4	9	902
g354	100	91	.91	171.14	200.6	7	845
g355	100	94	.94	196.55	230.3	4	998
g356	100	98	.98	246.73	255.6	6	989
g341	100	94	.94	199.47	216	7	815
g342	100	95	.95	212.7	225.3	4	903
g343	100	89	.89	159.42	226.9	7	1000
g344	100	98	.98	191.74	208.4	5	907
g345	100	93	.93	207.34	278.4	6	928
g346	100	94	.94	236.5	263.8	7	941
g331	100	100	1	81.78	56.23	5	255
g332	100	100	1	70.54	53.87	6	271
g333	100	100	1	89.57	79.73	9	584
g334	100	100	1	79.53	68.39	5	477
g335	100	100	1	70.86	53.71	5	264

Interlock by Reward, Agent Zero Q-Learning, RGS, 1k							
Model	Episodes	gL Count	gL Rate	Mean TTL	TTL σ	Min TTL	Max TTL
g336	100	100	1	68.32	58.72	4	435
g321	100	100	1	80.14	56.84	7	304
g322	100	100	1	65.42	56.99	4	285
g323	100	100	1	72.92	57.13	5	397
g324	100	100	1	68.09	49.03	7	243
g325	100	100	1	69.62	53.12	4	260
g326	100	100	1	83.25	76.24	4	401
g411	100	100	1	59.05	43.78	6	274
g412	100	100	1	51.78	32.79	6	194
g413	100	100	1	50.22	35.28	7	191
g414	100	100	1	49.98	44.2	4	391
g415	100	100	1	54.84	45.93	6	251
g416	100	100	1	54.11	46.52	4	314
g461	100	99	.99	100.17	154	5	952
g462	100	100	1	124.47	199.6	7	935
g463	100	100	1	112.66	174.2	8	915
g464	100	98	.98	76.38	126.1	6	998
g465	100	99	.99	72.24	124.2	7	952
g466	100	99	.99	75.34	92.82	7	567
g451	100	100	1	74.19	122.7	4	931
g452	100	100	1	85.09	122	5	946
g453	100	100	1	80.65	122.8	6	874
g454	100	100	1	97.05	127.2	5	742
g455	100	97	.97	100.98	161.2	6	869
g456	100	99	.99	74.52	120.8	6	949
g441	100	100	1	82.11	138.5	8	963
g442	100	98	.98	100.95	137.6	7	697
g443	100	100	1	70.49	91.03	7	619
g444	100	100	1	91.68	127.8	6	714
g445	100	100	1	91.15	132	6	717
g446	100	100	1	83.51	110.9	5	572
g431	100	100	1	50.61	36.51	5	190
g432	100	100	1	48.8	31.93	5	173
g433	100	100	1	63.91	52.2	5	360
g434	100	100	1	61.87	44.61	7	202
g435	100	100	1	53.54	31.74	4	178
g436	100	100	1	53.6	36.75	4	222
g421	100	100	1	56.31	34.31	4	200
g422	100	100	1	57.59	35.73	7	188
g423	100	100	1	57.14	39.2	6	212
g424	100	100	1	48.63	35.98	8	219
g425	100	100	1	51.52	34.17	5	156
g426	100	100	1	55.62	35.77	6	185

Table B.58: Interlock by Reward, Agent One, All Game Models. Strategy algorithm is *Q-Learning*, with asymmetric parameters (see §6.1).

Each episode is 1000 timesteps. ***gL*** is gamelock count, i.e., the act of recognition. ***TTL*** is time-to-lock, i.e., how many timesteps until recognition occurs.

† Standard Deviation (***σ***) for population (all episodes, all game models) this row only, all others sample.
†† Standard Deviation (***σ***) over all episodes for the thirty-six game models in each layer.
††† Standard Deviation (***σ***) over all episodes for each individual game in each group.

Interlock by Reward, Agent One Q-Learning, RGS, 1k							
Model	**Episodes**	***gL* Count**	***gL* Rate**	**Mean TTL**	**TTL *σ***	**Min TTL**	**Max TTL**
All †	14400	14221	.987	102.87	149.77	4	1000
Layer 1 ††	3600	3591	.998	70.02	91.26	4	989
Layer 2	3600	3545	.985	132.35	182.36	4	999
Layer 3	3600	3496	.971	138.21	183.03	4	1000
Layer 4	3600	3589	.997	70.9	101.29	4	998
g111 †††	100	100	1	48.23	33.55	4	161
g112	100	100	1	58.28	47.68	6	317
g113	100	100	1	52.18	38.96	7	218
g114	100	100	1	60.67	43.65	4	246
g115	100	100	1	53.5	40.37	8	222
g116	100	100	1	52.09	30.1	4	136
g161	100	100	1	91.02	148.43	4	870
g162	100	100	1	80.67	123.03	6	989
g163	100	98	.98	90.82	106.59	7	632
g164	100	99	.99	90.64	131.17	5	889
g165	100	99	.99	87.95	120.58	9	751
g166	100	100	1	80.52	103.99	8	617
g151	100	100	1	54.22	45.31	8	666
g152	100	99	.99	78.82	130.97	4	881
g153	100	100	1	65.15	71.58	4	404
g154	100	100	1	95.5	124.1	5	724
g155	100	100	1	97.77	114	5	641
g156	100	100	1	79.53	111.96	5	841
g141	100	99	.99	100.9	161.99	6	912
g142	100	100	1	79.9	109.63	4	815
g143	100	99	.99	86.75	140.32	8	890
g144	100	99	.99	77.99	113.69	5	791
g145	100	99	.99	93.08	129.03	4	708
g146	100	100	1	76.22	101.71	5	834
g131	100	100	1	58.52	42.34	8	245
g132	100	100	1	51.51	31.92	4	153
g133	100	100	1	61.36	36.61	9	202
g134	100	100	1	67.1	99.44	5	898
g135	100	100	1	60.82	42.87	7	225
g136	100	100	1	57.46	45.41	4	201
g121	100	100	1	52.14	34.73	9	206
g122	100	100	1	58.11	43.88	5	251
g123	100	100	1	52.33	38.97	4	198
g124	100	100	1	62.94	56.92	6	366
g125	100	100	1	53.18	42.33	6	246
g126	100	100	1	53	42.61	4	198
g211	100	100	1	76.85	56.21	6	298
g212	100	100	1	71.06	53.65	7	305
g213	100	100	1	68.02	53.3	4	266
g214	100	100	1	80.17	58.37	12	288
g215	100	100	1	60.06	42.04	9	190
g216	100	99	.99	84.71	82.77	6	679

Interlock by Reward, Agent One Q-Learning, RGS, 1k							
Model	Episodes	gL Count	gL Rate	Mean TTL	TTL σ	Min TTL	Max TTL
g261	100	93	.93	210.59	251.62	13	942
g262	100	93	.93	181.06	212.29	5	807
g263	100	98	.98	200.86	262.15	9	922
g264	100	98	.98	186.78	223.03	5	864
g265	100	100	1	201.25	219.82	11	883
g266	100	96	.96	226.06	259.92	9	951
g251	100	94	.94	173.57	199.51	5	869
g252	100	97	.97	176.01	236.01	5	973
g253	100	97	.97	178.56	227.55	7	999
g254	100	100	1	162.98	203.39	6	805
g255	100	100	1	189.46	231.61	5	817
g256	100	98	.98	221.52	263.47	6	922
g241	100	95	.95	231.81	270.86	7	997
g242	100	95	.95	209.21	256.09	4	967
g243	100	96	.96	216.57	244.03	7	967
g244	100	100	1	158.79	193.02	4	855
g245	100	100	1	191.18	208.79	5	900
g246	100	96	.96	207.95	249.47	9	959
g231	100	100	1	62.98	49.41	4	227
g232	100	100	1	80.35	64.06	6	280
g233	100	100	1	69.12	57.16	6	311
g234	100	100	1	58.75	45.60	5	258
g235	100	100	1	54.99	40.94	4	211
g236	100	100	1	68	52.41	4	264
g221	100	100	1	69.05	57.99	4	287
g222	100	100	1	74	58.91	9	276
g223	100	100	1	64.94	45.36	10	268
g224	100	100	1	56.85	49.49	5	291
g225	100	100	1	68.18	48.56	7	197
g226	100	100	1	72.37	58.65	8	299
g311	100	100	1	68.59	54.12	5	259
g312	100	100	1	80.06	62.49	6	291
g313	100	100	1	75.46	64.06	5	290
g314	100	100	1	74.17	53.99	5	276
g315	100	100	1	70.72	57.69	6	347
g316	100	100	1	76.8	59.25	6	329
g361	100	93	.93	191.01	199.84	4	832
g362	100	90	.9	184.1	207.24	8	949
g363	100	94	.94	194.13	250.84	4	1000
g364	100	92	.92	158.5	204.83	4	870
g365	100	99	.99	193.1	239.22	6	977
g366	100	99	.99	214.41	247.45	4	921
g351	100	97	.97	245.59	249.63	4	972
g352	100	92	.92	220.81	247.64	4	978
g353	100	94	.94	223.39	246.81	9	902
g354	100	91	.91	170.99	200.99	7	845
g355	100	94	.94	196.55	230.62	4	998
g356	100	98	.98	246.33	256.06	6	989
g341	100	94	.94	197.49	217.07	7	815
g342	100	95	.95	211.22	226.35	4	903
g343	100	89	.89	158.98	227.8	7	1000
g344	100	98	.98	189.89	209.48	5	907
g345	100	93	.93	207.15	279.19	6	928
g346	100	94	.94	234.51	265.08	7	941
g331	100	100	1	81.15	56.49	5	255
g332	100	100	1	70.02	54.11	6	271
g333	100	100	1	88.67	80.13	9	584
g334	100	100	1	78.53	68.70	5	477
g335	100	100	1	70.72	53.68	5	264

Interlock by Reward, Agent One Q-Learning, RGS, 1k							
Model	**Episodes**	**gL Count**	**gL Rate**	**Mean TTL**	**TTL σ**	**Min TTL**	**Max TTL**
g336	100	100	1	68.25	58.69	4	435
g321	100	100	1	79.06	57.06	7	304
g322	100	100	1	64.93	57.25	4	285
g323	100	100	1	71.93	57.36	5	397
g324	100	100	1	67.3	49.26	7	243
g325	100	100	1	68.83	53.38	4	260
g326	100	100	1	82.29	76.62	4	401
g411	100	100	1	58.81	43.85	6	274
g412	100	100	1	51.06	32.89	6	194
g413	100	100	1	49.28	35.18	7	191
g414	100	100	1	49.51	44.43	4	391
g415	100	100	1	54.72	45.96	6	251
g416	100	100	1	53.94	46.60	4	314
g461	100	99	.99	99.85	154.62	5	952
g462	100	100	1	124.36	200.32	7	935
g463	100	100	1	112.46	174.8	8	915
g464	100	98	.98	76.13	126.63	6	998
g465	100	99	.99	71.79	124.84	7	952
g466	100	99	.99	75.1	93.15	7	567
g451	100	100	1	73.79	123.27	4	931
g452	100	100	1	84.91	122.46	5	946
g453	100	100	1	80.41	123.27	6	874
g454	100	100	1	96.63	127.74	5	742
g455	100	97	.97	98.62	161.48	6	869
g456	100	99	.99	73.49	121.39	6	949
g441	100	100	1	81.7	139.15	8	963
g442	100	98	.98	100.66	138.13	7	697
g443	100	100	1	69.8	91.49	7	619
g444	100	100	1	88.72	126.74	6	714
g445	100	100	1	90.01	132.61	6	717
g446	100	100	1	83.21	111.37	5	572
g431	100	100	1	50.18	36.68	5	190
g432	100	100	1	48.41	32.07	5	173
g433	100	100	1	63.49	52.42	5	360
g434	100	100	1	60.82	44.62	7	202
g435	100	100	1	52.62	31.66	4	178
g436	100	100	1	53.25	36.89	4	222
g421	100	100	1	56.12	34.27	4	200
g422	100	100	1	57.25	35.84	7	188
g423	100	100	1	56.19	39.21	6	212
g424	100	100	1	48.52	35.97	8	219
g425	100	100	1	51.12	34.32	5	156
g426	100	100	1	55.3	35.87	6	185

Table B.59: Gamelock by Preference, All Game Models. Strategy algorithm is *Q-Learning*, with asymmetric parameters (see §6.1).

Each episode is 1000 timesteps. ***gL*** is gamelock count, i.e., the act of recognition. ***TTL*** is time-to-lock, i.e., how many timesteps until recognition occurs.

† Standard Deviation (σ) for population (all episodes, all game models) this row only, all others sample.
†† Standard Deviation (σ) over all episodes for the thirty-six game models in each layer.
††† Standard Deviation (σ) over all episodes for each individual game in each group.

Gamelock by Preference, Observer Summary
Q-Learning, RGS, 1k

Model	Episodes	*gL* Count	*gL* Rate	Mean TTL	TTL σ	Min TTL	Max TTL
All †	14400	14136	.982	140.32	170.25	5	1000
Layer 1 ††	3600	3577	.994	109.8	128.11	5	993
Layer 2	3600	3527	.98	171.01	197.03	6	999
Layer 3	3600	3460	.961	175.32	199.13	6	1000
Layer 4	3600	3572	.992	105.16	129.14	7	998
g111 †††	100	100	1	70.8	46.66	10	276
g112	100	100	1	81.45	52.21	13	339
g113	100	100	1	78.31	51.33	11	286
g114	100	100	1	81.07	50.94	9	267
g115	100	100	1	75.14	46.24	14	242
g116	100	100	1	77.08	40.54	13	202
g161	100	100	1	147.85	192.84	10	913
g162	100	99	0.99	135.09	168	13	993
g163	100	95	0.95	145.62	158.68	12	852
g164	100	99	0.99	158.9	181.56	5	969
g165	100	99	0.99	139.22	165.57	11	966
g166	100	100	1	124.45	135.75	17	633
g151	100	98	0.98	127.6	179.82	17	957
g152	100	97	0.97	151.32	183.26	11	893
g153	100	97	0.97	126.31	141.52	9	667
g154	100	99	0.99	118.62	135.49	9	757
g155	100	99	0.99	141.9	160.84	17	931
g156	100	100	1	140	180.54	11	922
g141	100	99	0.99	141.48	180.54	8	966
g142	100	100	1	142.44	172.26	12	917
g143	100	99	0.99	143.77	168.95	24	910
g144	100	99	0.99	139.73	177.83	9	991
g145	100	98	0.98	136.71	157.32	13	799
g146	100	100	1	128.16	153.55	7	839
g131	100	100	1	85.28	58.12	8	316
g132	100	100	1	79.38	45.73	15	284
g133	100	100	1	78.35	41.91	17	218
g134	100	100	1	91.02	106.51	11	949
g135	100	100	1	89.13	58.81	14	327
g136	100	100	1	85.82	55.14	10	238
g121	100	100	1	79.06	46.33	15	239
g122	100	100	1	88.34	88.48	11	833
g123	100	100	1	77.74	51.56	12	314
g124	100	100	1	93.33	65.38	9	392
g125	100	100	1	82.01	60.6	11	345
g126	100	100	1	70.28	46.18	9	219
g211	100	100	1	92.9	54.79	9	298
g212	100	100	1	83.43	51.78	10	305
g213	100	100	1	86.89	58.15	7	326
g214	100	100	1	95.97	64.97	12	305
g215	100	100	1	76.6	42.9	9	190
g216	100	99	0.99	93.78	81.4	8	679

Gamelock by Preference, Observer Summary Q-Learning, RGS, 1k							
Model	Episodes	gL Count	gL Rate	Mean TTL	TTL σ	Min TTL	Max TTL
g261	100	91	0.91	249.94	246.72	23	942
g262	100	93	0.93	264.42	242.61	14	994
g263	100	98	0.98	278.72	264.51	19	922
g264	100	98	0.98	251.34	233.81	18	864
g265	100	100	1	266.08	232.83	18	883
g266	100	95	0.95	264.46	251.55	16	951
g251	100	91	0.91	230.79	220.46	10	869
g252	100	95	0.95	259.16	265.13	23	973
g253	100	95	0.95	233.29	237.47	8	999
g254	100	100	1	218.14	206.03	14	805
g255	100	100	1	273.54	249.57	12	884
g256	100	97	0.97	285.38	260.69	12	922
g241	100	94	0.94	287.32	267.42	17	997
g242	100	93	0.93	259.94	252	9	967
g243	100	93	0.93	267.29	257.93	13	967
g244	100	100	1	221.28	210.44	12	855
g245	100	100	1	238.93	210.62	22	900
g246	100	95	0.95	282.01	270.77	9	960
g231	100	100	1	77.39	50.00	17	228
g232	100	100	1	99.75	64.24	10	280
g233	100	100	1	89.05	61.13	10	311
g234	100	100	1	75.7	53.17	7	351
g235	100	100	1	69.54	46.14	7	288
g236	100	100	1	85.85	51.00	10	264
g221	100	100	1	89.95	59.56	6	287
g222	100	100	1	85.85	56.87	10	276
g223	100	100	1	81.07	49.3	12	289
g224	100	100	1	71.06	53.56	9	291
g225	100	100	1	81.16	51.01	8	285
g226	100	100	1	88.42	58.13	11	299
g311	100	100	1	81.54	51.14	7	259
g312	100	100	1	93.71	61.85	6	291
g313	100	100	1	91.47	64.17	11	290
g314	100	100	1	82.97	52.93	7	276
g315	100	100	1	80.68	57.28	15	347
g316	100	100	1	93.34	64.77	12	329
g361	100	90	0.9	251.01	236.49	19	925
g362	100	87	0.87	241.43	227.19	8	975
g363	100	92	0.92	266.45	262.55	14	1000
g364	100	90	0.9	216.15	213.55	11	870
g365	100	94	0.94	264.3	257.51	11	977
g366	100	93	0.93	283.54	263.92	13	921
g351	100	96	0.96	288.8	247.89	11	972
g352	100	92	0.92	289.12	249.66	15	978
g353	100	92	0.92	273.39	250.38	17	902
g354	100	89	0.89	231.6	229.11	7	989
g355	100	92	0.92	259.12	234.02	7	998
g356	100	97	0.97	293.22	259.26	14	989
g341	100	92	0.92	248.94	225.82	23	879
g342	100	94	0.94	260.31	233.03	9	903
g343	100	89	0.89	247.18	247	11	1000
g344	100	98	0.98	244.02	217.79	24	907
g345	100	91	0.91	267.3	287.93	6	928
g346	100	92	0.92	296.05	274.83	12	979
g331	100	100	1	93.04	53.87	7	255
g332	100	100	1	88.65	52.39	16	271
g333	100	100	1	98.88	77.37	12	584
g334	100	100	1	92.88	68.2	11	477
g335	100	100	1	81.41	50.97	15	264

Gamelock by Preference, Observer Summary Q-Learning, RGS, 1k							
Model	Episodes	gL Count	gL Rate	Mean TTL	TTL σ	Min TTL	Max TTL
g336	100	100	1	82.84	58.44	12	435
g321	100	100	1	100.38	61.03	7	304
g322	100	100	1	85.47	67.7	8	349
g323	100	100	1	87.44	57.09	10	397
g324	100	100	1	74.27	47.65	7	243
g325	100	100	1	81.99	52.63	8	260
g326	100	100	1	98.49	76.16	7	401
g411	100	100	1	74.3	47.72	12	274
g412	100	100	1	70.53	36.63	11	194
g413	100	100	1	69.9	41.66	7	212
g414	100	100	1	64.03	47.06	9	391
g415	100	100	1	70.15	46.70	11	251
g416	100	100	1	68.41	45.94	8	314
g461	100	97	0.97	145.86	187.77	13	952
g462	100	99	0.99	156.6	207.34	13	935
g463	100	99	0.99	156.38	186.87	12	915
g464	100	98	0.98	118.22	145.35	12	998
g465	100	98	0.98	137.3	168.17	16	952
g466	100	98	0.98	127.32	147.33	19	966
g451	100	100	1	126.25	173.6	8	983
g452	100	100	1	123.68	143.42	10	946
g453	100	100	1	129.5	168.43	11	905
g454	100	98	0.98	147.55	160.61	10	742
g455	100	95	0.95	151.72	212.91	11	960
g456	100	99	0.99	112.19	149.69	8	949
g441	100	100	1	121.79	162.12	10	963
g442	100	97	0.97	175.05	205.73	14	907
g443	100	100	1	139.27	155.7	16	720
g444	100	97	0.97	132.5	154.29	10	824
g445	100	99	0.99	159.42	171.51	8	717
g446	100	98	0.98	139.02	147.54	9	808
g431	100	100	1	65.03	41.67	15	208
g432	100	100	1	66.91	45.06	11	315
g433	100	100	1	77.92	54.43	8	360
g434	100	100	1	79.56	51.18	11	254
g435	100	100	1	70.17	38.22	20	188
g436	100	100	1	74.23	45.66	13	265
g421	100	100	1	72.29	41.63	11	200
g422	100	100	1	72.67	38.81	8	200
g423	100	100	1	72.13	43.48	18	251
g424	100	100	1	72.1	44.42	15	219
g425	100	100	1	70.61	40.85	7	210
g426	100	100	1	75.32	41.04	8	236

Table B.60: Interlock by Preference, Agent Zero, All Game Models. Strategy algorithm is *Q-Learning*, with asymmetric parameters (see §6.1).

Each episode is 1000 timesteps. ***gL*** is gamelock count, i.e., the act of recognition. ***TTL*** is time-to-lock, i.e., how many timesteps until recognition occurs.

† Standard Deviation (σ) for population (all episodes, all game models) this row only, all others sample.
†† Standard Deviation (σ) over all episodes for the thirty-six game models in each layer.
††† Standard Deviation (σ) over all episodes for each individual game in each group.

Interlock by Preference, Agent Zero Q-Learning, RGS, 1k							
Model	**Episodes**	***iL* Count**	***iL* Rate**	**Mean TTL**	**TTL σ**	**Min TTL**	**Max TTL**
All †	14400	14139	.982	136.56	168.9	4	1000
Layer 1 ††	3600	3577	.994	101.83	125.28	4	991
Layer 2	3600	3530	.981	164.17	193.91	6	999
Layer 3	3600	3460	.961	175.32	199.13	6	1000
Layer 4	3600	3572	.992	104.91	129.27	5	998
g111 †††	100	100	1	63.67	43.16	8	276
g112	100	100	1	74.4	48.22	13	317
g113	100	100	1	74.33	50.87	11	286
g114	100	100	1	74.09	49.80	4	246
g115	100	100	1	66.44	42.24	8	222
g116	100	100	1	67.88	35.23	13	202
g161	100	100	1	141.71	190.87	7	870
g162	100	99	.99	130.35	166.49	6	989
g163	100	95	.95	139.92	154.05	12	852
g164	100	99	.99	154.86	179.94	5	969
g165	100	99	.99	131.05	163.14	9	966
g166	100	100	1	118.26	131.02	10	617
g151	100	98	.98	119.93	180.14	10	957
g152	100	97	.97	148.53	181.75	9	881
g153	100	97	.97	120.09	140.85	7	667
g154	100	99	.99	111.88	127.78	7	724
g155	100	99	.99	134.72	154.06	14	931
g156	100	100	1	133.41	180.32	5	922
g141	100	99	.99	135.58	175.15	8	912
g142	100	100	1	136.99	170.05	12	917
g143	100	99	.99	137.72	168.03	24	890
g144	100	99	.99	133.07	176.34	5	991
g145	100	98	.98	128.15	147.2	7	708
g146	100	100	1	120.83	153.91	5	834
g131	100	100	1	75.62	55.54	8	316
g132	100	100	1	72.34	45.15	15	284
g133	100	100	1	69.57	38.02	10	202
g134	100	100	1	80.7	102.56	10	949
g135	100	100	1	78.46	48.93	13	270
g136	100	100	1	74.98	51.24	7	233
g121	100	100	1	69.9	45.81	15	239
g122	100	100	1	72.25	44.76	10	251
g123	100	100	1	67	41.61	9	198
g124	100	100	1	78.48	62.64	9	392
g125	100	100	1	68.76	50.45	6	254
g126	100	100	1	59.98	42.54	4	201
g211	100	100	1	88.47	55.32	6	298
g212	100	100	1	82.2	51.98	10	305
g213	100	100	1	81.97	52.54	7	266
g214	100	100	1	90.88	63.93	12	305
g215	100	100	1	72.38	41.52	9	190
g216	100	99	.99	93.78	81.4	8	679

Interlock by Preference, Agent Zero Q-Learning, RGS, 1k							
Model	Episodes	*iL* Count	*iL* Rate	Mean TTL	TTL σ	Min TTL	Max TTL
g261	100	91	.91	236.12	244.48	17	942
g262	100	93	.93	250.94	234.17	12	929
g263	100	98	.98	265.45	260.02	9	922
g264	100	98	.98	243.08	236.39	18	864
g265	100	100	1	263.2	233.94	18	883
g266	100	95	.95	263.15	252.27	16	951
g251	100	92	.92	216.74	219.49	10	869
g252	100	95	.95	250.67	262.25	17	973
g253	100	95	.95	229.47	238.22	8	999
g254	100	100	1	208.98	209.06	14	805
g255	100	100	1	263.24	246.29	12	884
g256	100	97	.97	275.19	258.87	11	922
g241	100	94	.94	264.33	263.9	17	997
g242	100	93	.93	248.57	251.86	9	967
g243	100	94	.94	255.48	249.95	13	967
g244	100	100	1	208.12	207.12	11	855
g245	100	100	1	227.46	208.85	17	900
g246	100	96	.96	266.23	268.54	9	960
g231	100	100	1	72.91	48.98	14	228
g232	100	100	1	96.59	62.73	10	280
g233	100	100	1	87.11	62.21	10	311
g234	100	100	1	72.38	53.04	7	351
g235	100	100	1	68.56	46.32	7	288
g236	100	100	1	84.29	50.69	9	264
g221	100	100	1	88.7	60.28	6	287
g222	100	100	1	84.89	57.26	10	276
g223	100	100	1	77.76	45.12	12	268
g224	100	100	1	67.51	51.32	8	291
g225	100	100	1	78.13	49.97	8	285
g226	100	100	1	85.21	59.01	11	299
g311	100	100	1	81.54	51.14	7	259
g312	100	100	1	93.71	61.85	6	291
g313	100	100	1	91.47	64.17	11	290
g314	100	100	1	82.97	52.93	7	276
g315	100	100	1	80.68	57.28	15	347
g316	100	100	1	93.34	64.77	12	329
g361	100	90	.9	251.01	236.49	19	925
g362	100	87	.87	241.43	227.19	8	975
g363	100	92	.92	266.45	262.55	14	1000
g364	100	90	.9	216.15	213.55	11	870
g365	100	94	.94	264.3	257.51	11	977
g366	100	93	.93	283.54	263.92	13	921
g351	100	96	.96	288.8	247.89	11	972
g352	100	92	.92	289.12	249.66	15	978
g353	100	92	.92	273.39	250.38	17	902
g354	100	89	.89	231.6	229.11	7	989
g355	100	92	.92	259.12	234.02	7	998
g356	100	97	.97	293.22	259.26	14	989
g341	100	92	.92	248.94	225.82	23	879
g342	100	94	.94	260.31	233.03	9	903
g343	100	89	.89	247.18	247	11	1000
g344	100	98	.98	244.02	217.79	24	907
g345	100	91	.91	267.3	287.93	6	928
g346	100	92	.92	296.05	274.83	12	979
g331	100	100	1	93.04	53.87	7	255
g332	100	100	1	88.65	52.39	16	271
g333	100	100	1	98.88	77.37	12	584
g334	100	100	1	92.88	68.2	11	477
g335	100	100	1	81.41	50.97	15	264

Interlock by Preference, Agent Zero Q-Learning, RGS, 1k							
Model	**Episodes**	***iL* Count**	***iL* Rate**	**Mean TTL**	**TTL σ**	**Min TTL**	**Max TTL**
g336	100	100	1	82.84	58.44	12	435
g321	100	100	1	100.38	61.03	7	304
g322	100	100	1	85.47	67.69	8	349
g323	100	100	1	87.44	57.09	10	397
g324	100	100	1	74.27	47.65	7	243
g325	100	100	1	81.99	52.63	8	260
g326	100	100	1	98.49	76.16	7	401
g411	100	100	1	74.07	47.99	7	274
g412	100	100	1	69.69	36.86	11	194
g413	100	100	1	69.62	41.97	7	212
g414	100	100	1	63.98	47.12	5	391
g415	100	100	1	69.73	47.14	9	251
g416	100	100	1	67.7	46.10	8	314
g461	100	97	.97	145.86	187.77	13	952
g462	100	99	.99	156.58	207.34	13	935
g463	100	99	.99	156.36	186.88	12	915
g464	100	98	.98	117.96	145.43	10	998
g465	100	98	.98	137.08	168.32	15	952
g466	100	98	.98	126.77	147.67	11	966
g451	100	100	1	126.23	173.61	8	983
g452	100	100	1	123.4	143.59	7	946
g453	100	100	1	129.5	168.43	11	905
g454	100	98	.98	147.55	160.61	10	742
g455	100	95	.95	151.72	212.91	11	960
g456	100	99	.99	112.14	149.73	6	949
g441	100	100	1	121.5	162.27	10	963
g442	100	97	.97	175.05	205.73	14	907
g443	100	100	1	138.75	156.05	7	720
g444	100	97	.97	131.99	154.64	10	824
g445	100	99	.99	159.42	171.51	8	717
g446	100	98	.98	139.02	147.54	9	808
g431	100	100	1	64.8	41.89	14	208
g432	100	100	1	66.2	45.35	11	315
g433	100	100	1	77.69	54.63	8	360
g434	100	100	1	78.63	52.01	8	254
g435	100	100	1	69.12	38.37	20	188
g436	100	100	1	74.23	45.66	13	265
g421	100	100	1	72.05	41.83	11	200
g422	100	100	1	72.49	39.06	8	200
g423	100	100	1	72.12	43.48	18	251
g424	100	100	1	71.75	44.76	11	219
g425	100	100	1	70.61	40.85	7	210
g426	100	100	1	75.27	41.10	8	236

Table B.61: Interlock by Preference, Agent One, All Game Models. Strategy algorithm is *Q-Learning*, with asymmetric parameters (see §6.1).

Each episode is 1000 timesteps. ***gL*** is gamelock count, i.e., the act of recognition. ***TTL*** is time-to-lock, i.e., how many timesteps until recognition occurs.

† Standard Deviation (σ) for population (all episodes, all game models) this row only, all others sample.
†† Standard Deviation (σ) over all episodes for the thirty-six game models in each layer.
††† Standard Deviation (σ) over all episodes for each individual game in each group.

Interlock by Preference, Agent One Q-Learning, RGS, 1k							
Model	**Episodes**	***iL* Count**	***iL* Rate**	**Mean TTL**	**TTL σ**	**Min TTL**	**Max TTL**
All †	14400	14162	.983	128.44	163.91	4	1000
Layer 1 ††	3600	3591	.998	83.28	96.74	4	993
Layer 2	3600	3527	.98	170.68	197.24	5	999
Layer 3	3600	3460	.961	175.32	199.13	6	1000
Layer 4	3600	3584	.996	84.47	107.79	6	998
g111 †††	100	100	1	60.66	39.92	10	251
g112	100	100	1	71.87	51.08	9	339
g113	100	100	1	62.62	41.86	9	234
g114	100	100	1	70.27	45.96	9	267
g115	100	100	1	67.85	46.09	9	242
g116	100	100	1	67.41	37.46	13	187
g161	100	100	1	101.03	151.26	9	913
g162	100	100	1	87.94	126.01	10	993
g163	100	98	.98	102.11	114.1	9	717
g164	100	99	.99	100.03	133.72	5	896
g165	100	99	.99	103.05	124.49	11	770
g166	100	100	1	90.43	110.43	10	633
g151	100	100	1	73.86	79.97	8	698
g152	100	99	.99	89.41	133.35	10	893
g153	100	100	1	81.62	75.18	9	431
g154	100	100	1	107.25	130.71	9	757
g155	100	100	1	108.81	123.67	16	674
g156	100	100	1	92.25	113.65	7	844
g141	100	99	.99	111.51	167.24	8	966
g142	100	100	1	90.14	113.65	10	842
g143	100	99	.99	98.42	141.22	13	910
g144	100	99	.99	87.96	118.1	9	833
g145	100	99	.99	105.77	141.33	11	799
g146	100	100	1	87.6	102.63	7	839
g131	100	100	1	72.18	48.45	8	298
g132	100	100	1	66.21	36.79	11	184
g133	100	100	1	72.87	41.32	9	218
g134	100	100	1	82.04	102.79	9	898
g135	100	100	1	77.35	54.51	13	327
g136	100	100	1	74.68	51.91	4	238
g121	100	100	1	66.71	37.82	12	206
g122	100	100	1	79.32	89.31	9	833
g123	100	100	1	68.18	50.93	10	314
g124	100	100	1	82.96	63.02	9	366
g125	100	100	1	70.37	56.11	11	345
g126	100	100	1	65.36	46.82	7	219
g211	100	100	1	92.9	54.79	9	298
g212	100	100	1	83.05	52.11	10	305
g213	100	100	1	86.48	58.5	7	326
g214	100	100	1	95.04	65.5	12	305
g215	100	100	1	75.83	43.49	9	190
g216	100	99	.99	93.77	81.41	8	679

Interlock by Preference, Agent One Q-Learning, RGS, 1k							
Model	Episodes	*iL* Count	*iL* Rate	Mean TTL	TTL σ	Min TTL	Max TTL
g261	100	91	.91	249.94	246.72	23	942
g262	100	93	.93	264.22	242.81	12	994
g263	100	98	.98	278.3	264.88	19	922
g264	100	98	.98	250.95	234.2	6	864
g265	100	100	1	265.28	233.65	14	883
g266	100	95	.95	264.18	251.82	10	951
g251	100	91	.91	230.43	220.76	9	869
g252	100	95	.95	259.09	265.19	23	973
g253	100	95	.95	232.85	237.85	8	999
g254	100	100	1	217.74	206.38	14	805
g255	100	100	1	273.52	249.59	12	884
g256	100	97	.97	285.38	260.69	12	922
g241	100	94	.94	287.32	267.42	17	997
g242	100	93	.93	259.59	252.28	9	967
g243	100	93	.93	267.04	258.17	13	967
g244	100	100	1	221.28	210.44	12	855
g245	100	100	1	238.85	210.71	14	900
g246	100	95	.95	281.65	271.07	9	960
g231	100	100	1	77.39	50	17	228
g232	100	100	1	99.45	64.61	10	280
g233	100	100	1	88.94	61.22	10	311
g234	100	100	1	75.27	53.49	7	351
g235	100	100	1	68.28	46.68	5	288
g236	100	100	1	85.7	51.14	10	264
g221	100	100	1	88.73	60.45	6	287
g222	100	100	1	85.73	57.02	9	276
g223	100	100	1	80.32	49.58	10	289
g224	100	100	1	70.94	53.7	6	291
g225	100	100	1	80.59	51.54	8	285
g226	100	100	1	88.42	58.13	11	299
g311	100	100	1	81.54	51.14	7	259
g312	100	100	1	93.71	61.85	6	291
g313	100	100	1	91.47	64.17	11	290
g314	100	100	1	82.97	52.93	7	276
g315	100	100	1	80.68	57.28	15	347
g316	100	100	1	93.34	64.77	12	329
g361	100	90	.9	251.01	236.49	19	925
g362	100	87	.87	241.43	227.19	8	975
g363	100	92	.92	266.45	262.55	14	1000
g364	100	90	.9	216.15	213.56	11	870
g365	100	94	.94	264.3	257.51	11	977
g366	100	93	.93	283.54	263.92	13	921
g351	100	96	.96	288.8	247.89	11	972
g352	100	92	.92	289.12	249.66	15	978
g353	100	92	.92	273.39	250.38	17	902
g354	100	89	.89	231.6	229.11	7	989
g355	100	92	.92	259.12	234.02	7	998
g356	100	97	.97	293.22	259.26	14	989
g341	100	92	.92	248.94	225.82	23	879
g342	100	94	.94	260.31	233.03	9	903
g343	100	89	.89	247.18	247	11	1000
g344	100	98	.98	244.02	217.79	24	907
g345	100	91	.91	267.3	287.93	6	928
g346	100	92	.92	296.05	274.83	12	979
g331	100	100	1	93.04	53.87	7	255
g332	100	100	1	88.65	52.39	16	271
g333	100	100	1	98.88	77.37	12	584
g334	100	100	1	92.88	68.2	11	477
g335	100	100	1	81.41	50.97	15	264

Interlock by Preference, Agent One Q-Learning, RGS, 1k							
Model	**Episodes**	***iL* Count**	***iL* Rate**	**Mean TTL**	**TTL σ**	**Min TTL**	**Max TTL**
g336	100	100	1	82.84	58.44	12	435
g321	100	100	1	100.38	61.03	7	304
g322	100	100	1	85.47	67.69	8	349
g323	100	100	1	87.44	57.09	10	397
g324	100	100	1	74.27	47.65	7	243
g325	100	100	1	81.99	52.63	8	260
g326	100	100	1	98.49	76.16	7	401
g411	100	100	1	68.76	46.71	12	274
g412	100	100	1	63.34	33.59	9	194
g413	100	100	1	60.66	33.79	7	191
g414	100	100	1	57.52	45.15	9	391
g415	100	100	1	63.43	45.05	11	251
g416	100	100	1	63.04	44.72	6	314
g461	100	98	.98	126.29	173.7	7	952
g462	100	99	.99	136.64	202.8	9	935
g463	100	99	.99	126.69	179.43	12	915
g464	100	98	.98	89.17	129.79	11	998
g465	100	99	.99	100.8	150.76	16	952
g466	100	99	.99	87.69	95.25	18	567
g451	100	100	1	87.1	131.74	8	931
g452	100	100	1	96.9	120.93	9	946
g453	100	100	1	105.67	143.42	11	874
g454	100	100	1	113.6	132.35	10	742
g455	100	97	.97	107.04	159.38	11	869
g456	100	99	.99	87.92	119.96	8	949
g441	100	100	1	94.82	141.69	10	963
g442	100	98	.98	133.12	172.07	8	907
g443	100	100	1	90.92	104.14	16	619
g444	100	99	.99	97.65	125.64	10	714
g445	100	100	1	108.51	141	8	717
g446	100	99	.99	104.12	122.92	9	610
g431	100	100	1	58.82	36.46	8	190
g432	100	100	1	55.73	30.9	11	173
g433	100	100	1	70.64	51.27	8	360
g434	100	100	1	72.87	47.93	11	254
g435	100	100	1	61.52	34.44	12	188
g436	100	100	1	66.63	41.75	13	265
g421	100	100	1	64.32	36.9	11	200
g422	100	100	1	66.39	36.22	8	188
g423	100	100	1	64.4	38.96	18	212
g424	100	100	1	61.03	37.2	9	219
g425	100	100	1	59.72	31.89	7	156
g426	100	100	1	67.4	36.15	6	185

B.5 Code & Data Availability

Table B.62: Thesis Code Availability.

Title	Tag	DOI
Code -- Experiment Series One -- Multi-Model Tournament	v0.1-alpha	10.5281/zenodo.8188104
Code -- Experiment Series Two -- Representational Equivalence	v0.1-alpha	10.5281/zenodo.8188122
Code -- Experiment Series Three -- Game Model Recognition	v0.1-alpha	10.5281/zenodo.8188155

Table B.63: Thesis Data Availability.

Title	Size (Mb)	DOI
Data -- Experiment Series One -- Multi-Model Tournament	777.7	10.5281/zenodo.8170123
Data -- Experiment Series Two -- Representational Equivalence	9520	10.5281/zenodo.8171272
Data -- Experiment Series Three -- Game Model Recognition	381.5	10.5281/zenodo.8173539