

Multiagent Training in N-Player General-Sum Games

Luke Marris

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

CoMPLEX
Centre for Mathematics and Physical Sciences in the Life Sciences and Experimental Biology
University College London

April 17, 2024

I, Luke Marris, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Recent successes in two-player, purely competitive (so-called zero-sum) games has made headlines around the world. Initially, perfect information games, with vast state spaces, such as Go and Chess, were conquered using a combination of search, deep neural network function approximation, and self-play reinforcement learning. Subsequently more challenging, imperfect information games such as StarCraft were tackled using recurrent deep neural networks, human-play priors, and policy-space response oracle reinforcement learning.

Although these are important advances, from a game theoretic perspective two-player zero-sum games are the simplest class, and many algorithms are known to converge to a solution concept called Nash Equilibrium in this setting. For two-player zero-sum games the Nash Equilibrium is unexploitable (there is no strategy that an opponent could play to that would reduce one's reward) and interchangeable (if there is more than one Nash equilibrium in a game, all equilibria are equally good, and it is not necessary for players to coordinate on the equilibria being played). But most importantly, Nash equilibrium coincides with the Minimax solution in two-player zero-sum, which is a single-objective optimization. Therefore when playing against rational opponents, the Nash equilibrium is the obvious, perhaps fundamental, objective to optimize for when training learning agents in two-player zero-sum games.

Progress on n-player general-sum games, however, has remained limited. It is both unclear what objective to optimize for and difficult to build algorithms that converge to any interesting objective. This is disappointing because the majority of the world's interactions are not purely competitive, have interesting mixed-motive dynamics, and have more than two players. This is an area of interest to economists, sociologists, policy makers, artificial intelligence researchers, and any other discipline that concerns multiple agents that have to compete or cooperate together to achieve their objectives.

Previous work has mainly involved applying single agent techniques, or two-player zero-sum techniques and hoping for the best. The overall aim of this work is to unlock progress beyond the narrow domains of two-player zero-sum, to the most general space of n-player, general-sum games using principled game theoretic techniques. In this end, this work utilizes more flexible mediated equilibrium solution concepts, correlated equilibrium and coarse correlated equilibrium, which are more suitable for general-sum games. This choice is for convenience and to enable the main goal of this work: building algorithms that scale in n-player general-sum settings.

This thesis a) builds an intuition for the space of games by studying transforms that do not alter their equilibrium, b) proposes a novel normal-form game embedding, c) develops tools for visualizing large extensive-form games, d) proposes an efficient equilibrium selection criteria, e) builds fast neural network solvers for finding equilibria in all normal-form games, f) proposes population based learning algorithms for training policies in n-player general-sum extensive-form games, g) proposes learning algorithms for training policies in Markov games, and h) develops novel ratings algorithms to evaluate strategies in n-player general-sum games.

Impact Statement

Interactions between intelligent, self-interested agents are ubiquitous. Such interactions are the primary complication underlying many real-world problems including climate negotiations, economic policy, defence treaties, pandemic coordination, market dynamics, and corporate competition. This thesis does not claim to directly solve any of the above problems. However, it does take a step towards developing learning algorithms that are capable of converging to solutions in games, the framework for describing interactions between self-interested agents. In particular, this thesis makes progress on the most difficult and general class of games, n-player general-sum, which have been notoriously difficult to solve.

Concretely, and most importantly, this work proposes learning algorithms for training agent policies towards mediated equilibria, in n-player general-sum normal-form, extensive-form, and Markov games. Additionally it makes several other contributions: it a) builds intuition for the space of games and their equilibrium solutions, b) proposes a novel game embedding, c) develops game theoretic visualization tools for games, d) proposes efficient equilibrium selection criteria, and e) proposes game theoretic strategy rating algorithms in n-player general-sum games.

This work bridges the fields of machine learning, reinforcement learning, and game theory. It is not the first to do so but does offer a unique perspective. Its algorithms are designed to be scaled with reinforcement learning and neural networks. This work is deliberate in avoiding the metric question: what is the perfect solution concept for n-player general-sum? Instead, it calls to action the need to build algorithms that scale to complex games with established solution concepts.

The fields of sociology, economics, game theory, and artificial intelligence directly benefit from the work in this thesis, however its indirect effects could be more widespread. Progress made in understanding strategic interactions between players could help improve cooperation amongst people or systems, increase their payoffs, and improve rules and regulation from which strategic interaction emerges.

Acknowledgements

Completing a PhD is said to be an individual undertaking however, in truth, a PhD is only possible with the help and support of many people. I would like to thank my primary supervisor Thore Graepel for his expert guidance, patience, and research perspectives. My secondary supervisor Jun Wang for his mentorship and perspective on the PhD process. Marc Lanctot for his excellent and tireless research mentorship. Karl Tuyls for his expert perspective on Game Theory. Georgios Piliouras for his bottomless knowledge on equilibrium concepts. My paper co-authors; Ian Gemp, Thore Graepel, Marc Lanctot, Siqui Lui, Paul Muller, Karl Tuyls, Andrea Tacchetti, Thomas Anthony, Jerome Connor, Shayegan Omidshafiei, and Georgios Piliouras for their helpful contributions. In particular, Ian Gemp and Siqui Lui who I had the privilege of working very closely with. Max Jaderberg, Wojtek Czarnecki, Julien Perolat, Ed Hughes, Guy Lever, Zhe Wang and Edgar Duéñez-Guzmán for helpful discussions. Georg Ostrovski, David Parkes, and John Agapiou for reviewing paper drafts. All anonymous reviewers for their thoughtful and constructive feedback. My managers Tiago Ramalho, Ian Dunning, David Budden, Ross Hemsley, and Georgios Piliouras for their sympathetic guidance during the PhD process. Tiago Ramalho and Geoff Parks for helping with and supporting my PhD application. The DeepMind PhD team, Nicole Hurley, Sarah Hodkinson, Claudia Pope, and Amber Nicklin-Clark for logistical support. The UCL PhD team and Lewis Griffin and Gemma Ludbrook for organizing the programme. DeepMind for funding the PhD, and establishing the programme in collaboration with UCL. John Shawe-Taylor and Jun Wang for examining my upgrade. Ran Spiegler and Milan Vojnovic for examining my viva. And, finally I would like to thank my wife, Harshnira Patani, for her love, support, proofreading, and patience throughout the process.

Paper Declaration

The following papers make up chapters in this thesis. The introduction of each chapter indicates which publications the content is drawn from. Copyright has been retained for all papers and can be freely included within this thesis.

Embedding Paper

L. Marris, I. Gemp, and G. Piliouras. Equilibrium-invariant embedding, metric space, and fundamental set of 2x2 normal-form games, 2023. URL <https://arxiv.org/abs/2304.09978>

Contributions: Marris derived the embeddings, wrote the proofs, wrote the paper, wrote the code, ran the experiments and produced the visualizations. Gemp provided useful discussions and feedback. Piliouras suggested the better-response-invariant transform, provided input for game visualizations, checked proofs, and provided useful direction.

JPSRO Paper

L. Marris, P. Muller, M. Lanctot, K. Tuyls, and T. Graepel. Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7480–7491. PMLR, 18–24 Jul 2021b. URL <http://proceedings.mlr.press/v139/marris21a.html>

Contributions: Marris wrote the paper, wrote the code, ran the experiments, and produced the figures. Muller helped with PSRO expertise, wrote the JPSRO convergence proofs, and suggested the (C)CE best-response distributions. Lanctot helped with PSRO discussion, checked proofs, wrote (C)CE best-response operator code, and provided writing feedback. Tuyls and Graepel provided valuable feedback and direction.

Neural Equilibrium Solver

L. Marris, I. Gemp, T. Anthony, A. Tacchetti, S. Liu, and K. Tuyls. Turbocharging solution concepts: Solving NEs, CEs and CCEs with neural equilibrium solvers. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 5586–5600. Curran Associates, Inc., 2022a. URL https://papers.nips.cc/paper_files/paper/2022/hash/24f420aa4c99642dbb9aae18b166bbbc-Abstract-Conference.html

Contributions: Marris contributed the idea, wrote the paper, wrote the code, and ran the experiments. Anthony provided counterexamples for relative entropy and welfare objectives. Gemp, Tacchetti and Liu reviewed code, the writing, and provided valuable feedback. Tuyls provided valuable feedback and direction.

Game-Theoretic Rating Paper

L. Marris, M. Lanctot, I. Gemp, S. Omidshafiei, S. McAleer, J. Connor, K. Tuyls, and T. Graepel. Game theoretic rating in n-player general-sum games with equilibria, 2022b. URL <https://arxiv.org/abs/2210.02205>

Contributions: Marris developed the idea, wrote the proofs, wrote the code, wrote the paper, and ran all

but one experiment. Lanctot trained the agents in agent rating experiments. Gemp helped with writing and provided helpful feedback. Omidshafiei provided useful feedback on game theoretic rating algorithms and α -Rank. McAleer helped check proofs. Connor suggested the connection of Nash Averaging with the derivative of the expected payoffs. Tuyls provided discussion and feedback. Graepel provided useful discussion, feedback and direction.

Contents

1	Introduction	18
1.1	Multiagent Problem	19
1.1.1	Inherently Multi-Objective	19
1.1.2	Non-Stationary Environment	20
1.1.3	An explosion in Complexity	21
1.1.4	Goals and a Non-Goal	23
1.2	The Multiagent Worldview	24
1.2.1	Gamification	24
1.2.2	Evolution and The Auto-Curriculum Property	25
1.2.3	Multiagent World	25
2	Literature Review	26
2.1	Reinforcement Learning	26
2.1.1	Fundamentals	28
2.1.2	Value-Based Model-Based Reinforcement Learning	32
2.1.3	Value-Based Model-Free Reinforcement Learning	38
2.1.4	Policy-Based Model-Free Reinforcement Learning	43
2.2	Game Theory and Multiagent Reinforcement Learning	45
2.2.1	Multiagent RL Problem	45
2.2.2	Normal-Form Games	46
2.2.3	Markov Games	64
2.2.4	Extensive-Form Games	66
3	Normal-Form Game Metric Spaces and Embeddings	70
3.1	Introduction	70
3.2	Equilibrium-Invariant Embedding	71
3.2.1	Reversible Deviation Gains	73
3.2.2	Sampling Equilibrium-Invariant Embedding	74
3.3	Better-Response Embedding	75
3.4	Symmetric Embedding	77
3.5	Discussion	78
3.6	Conclusion	79
3.A	Appendices	80
3.A.1	Deviation Gains as Linear Operators	80
3.A.2	Game Theoretic Rating	80
4	2×2 Metric Spaces and Embeddings	82

4.1	Introduction	82
4.2	Preliminaries	84
4.3	2×2 Equilibrium-Invariant Embedding	85
4.3.1	Deriving the 2×2 Equilibrium-Invariant Embedding	85
4.3.2	Invariant-Zero-Sum and Invariant-Common-Payoff Quadrants	86
4.3.3	2×2 Equilibrium-Invariant Distance Metric	89
4.4	2×2 Equilibrium-Symmetric Embedding	90
4.4.1	Deriving the 2×2 Equilibrium-Symmetric Embedding	90
4.4.2	2×2 Equilibrium-Symmetric Player-Agnostic Embedding	90
4.4.3	2×2 Equilibrium-Symmetric Distance Metric	91
4.5	2×2 Best-Response-Invariant Embedding	91
4.5.1	Deriving the Best-Response Embedding	91
4.5.2	Equivalence Classes, Graph Representation, and Measure	92
4.5.3	Naming the Best-Response-Invariant Embedding	93
4.5.4	Distance Metric	97
4.6	Discussion	97
4.7	Conclusion	99
5	Visualizing Large Games	103
5.1	Introduction	103
5.2	2×2 Game Visualization	103
5.3	Two-Player Game Visualization	106
5.3.1	Extensive-Form Games	107
5.3.2	AlphaStar League	108
5.4	N-Player Polymatrix Game Visualization	109
5.4.1	Three-Player Leduc Poker	110
5.4.2	Tiny Bridge 2vs2	110
5.5	Discussion	112
5.6	Conclusion	113
6	Equilibrium Selection of Correlated Equilibria and Coarse Correlated Equilibria	115
6.1	Introduction	115
6.1.1	Equilibrium Selection and Correlation Devices	116
6.1.2	Desired Properties	117
6.2	MG(C)CE and its Computation	117
6.2.1	Quadratic Program	118
6.2.2	Primal and Dual Forms	118
6.3	Properties of MG(C)CE	119
6.3.1	Uniqueness	119
6.3.2	Scalable Representation	119
6.3.3	Family of Solutions	120
6.3.4	Invariance	120
6.3.5	Computationally Tractable	120
6.4	Conclusion	121
6.A	Appendices	122
6.A.1	Generalized Entropy	122

6.A.2	Proofs of MG(C)CE Properties	122
6.A.3	MGCE Computation	124
7	Joint Policy-Space Response Oracles	127
7.1	Introduction	127
7.2	Preliminaries	128
7.2.1	Normal-Form and Extensive-Form Equilibria	128
7.2.2	Policy-Space Response Oracles (PSRO)	128
7.3	Joint PSRO	130
7.3.1	Meta-Game Estimation	130
7.3.2	Meta-Solvers	131
7.3.3	Best-Response Operators	132
7.3.4	Gap and Convergence	133
7.3.5	Evaluation	134
7.4	Experiments CEs and CCEs as Joint Meta-Solvers	134
7.5	Discussion	135
7.6	Conclusion	137
7.A	Appendices	138
7.A.1	JPSRO Hyper-parameters	138
7.A.2	Extended Experiments	138
7.A.3	Open Source Code	139
8	Neural Equilibrium Solvers	143
8.1	Introduction	143
8.2	Preliminaries	144
8.3	Maximum Welfare Minimum Relative Entropy (C)CEs	145
8.3.1	Dual of ϵ -MWMRE (C)CEs	145
8.4	Neural Network Training	146
8.4.1	Training Distribution and Input Preprocessing	147
8.4.2	Gradient Calculation	148
8.4.3	Equivariant Architectures	148
8.4.4	Parameterizations	151
8.5	Performance Experiments	151
8.6	Applications	154
8.7	Discussion	155
8.A	Appendices	156
8.A.1	Approximate Target Maximum Welfare Minimum Relative Entropy Equilibria	156
8.A.2	Equivariant Pooling Functions	158
8.A.3	Experiment Architecture and Hyper-Parameters	158
8.A.4	Relative Entropy and Welfare Objectives	159
9	Game Theoretic Rating	161
9.1	Introduction	161
9.2	Game-Theoretic Rating	162
9.2.1	Payoff Rating	162
9.2.2	Joint Strategy Distributions	163
9.2.3	Properties of Equilibria Ratings	164

9.3	Rating Algorithms	164
9.3.1	ϵ^{\min} -MECCE Payoff Rating	164
9.3.2	$\frac{\epsilon}{\epsilon_{\text{uni}}}$ -MECCE Payoff Rating	164
9.4	Implementation Considerations	165
9.4.1	Uncertainty in Payoffs	165
9.4.2	Repeated Strategy Problem	165
9.5	Experiments	166
9.5.1	Standard Normal Form Games	166
9.5.2	Constant-Sum Premier League Ratings	166
9.5.3	General-Sum Two-Player Premier League Ratings	168
9.5.4	Three-Player Premier League Ratings	168
9.5.5	Three-Player ATP Tennis Ratings	168
9.5.6	Multiagent Learning Dynamics	170
9.6	Discussion	173
9.7	Conclusion	173
9.A	Appendices	174
9.A.1	Algorithms	174
9.A.2	Full Support Conditions	174
9.A.3	Justification and Intuition	174
10	General Discussion and Conclusion	178
10.1	Introduction	178
10.2	Discussion	178
10.2.1	Equilibrium-Invariant Embeddings	178
10.2.2	2×2 games	179
10.2.3	Visualizing Large Games	179
10.2.4	Equilibrium Selection and Computation	179
10.2.5	JPSRO	179
10.2.6	Neural Equilibrium Solver	180
10.2.7	Game-Theoretic Rating	180
10.3	Conclusion	180

List of Figures

1.1	Heider and Simmel Agency Experiment Stills	21
1.2	RPS learning dynamics	22
2.1	Agent-Environment Interaction	27
2.2	MARL formulations.	46
2.3	Probability Simplex and NE Manifold	61
2.4	(C)CE polytope of coordination game	62
2.5	Equilibria of canonical 2×2 games	63
3.1	Structure of deviation gain operators.	81
4.1	Common 2×2 normal-form games	83
4.2	Main 2×2 contributions	83
4.3	2×2 equilibrium-invariant embedding	87
4.4	Narrative payoff tables	94
4.5	Game Equilibria Polytopes	100
4.6	Equilibrium Support	101
4.7	15 Fundamental Games	102
5.1	Game shapes	104
5.2	Probability Simplex and NE Manifold	104
5.3	Visualization of 2×2 games	105
5.4	Two-Player Extensive-Form Game Visualization	106
5.5	AlphaStar Empirical Game Analysis	108
5.6	Local polymatrix approximation of three-player Leduc poker	111
5.7	Tiny bridge policy generation	112
5.8	Local polymatrix approximation of 2vs2 Tiny Bridge	113
5.9	Point cloud for two different 3×3 normal-form games.	114
7.1	Traffic Light Game	129
7.2	JPSRO(CCE) on various games	136
7.3	JPSRO(C)CE on Kuhn Poker	140
7.4	JPSRO(C)CE on Trade Comm	141
7.5	JPSRO(C)CE on Sheriff	142
8.1	Neural Equilibrium Solutions for Normal-Form Games	146
8.2	Network Architecture	151
8.3	Sweeps and Ablations	152

8.4	Sweep of Game Classes	154
8.5	Equivariant Pooling Functions	159
9.1	Symmetric Two-Player Constant-Sum Premier League Ratings	167
9.2	Symmetric Two-Player Constant-Sum Premier League Approximation Sweep	169
9.3	Symmetric Two-player General-Sum Premier League Ratings	170
9.4	Three-Player Premier League Ratings	171
9.5	Three-player ATP	171
9.6	Multiagent learning dynamics.	172

List of Tables

2.1	Temporal difference algorithms.	42
2.2	Value Function Approximation Taxonomy	42
2.3	Policy Gradient Algorithms.	45
2.4	Canonical Normal-Form Games	60
4.1	Number of variables in succinct games	84
4.2	Symmetries in 2×2 embeddings	90
4.3	Properties of the set of 2×2 best-response-invariant embeddings	93
4.4	Best-response-invariant embedding L_1 distance	98
6.1	Family of MG(C)CE solutions.	124
7.1	JPSRO Meta-Solvers	139
8.1	Network Parameterizations	151
8.2	Game Classes	153
8.3	Scaling Experiments	154
8.4	Generalization Experiments	155
8.5	Counterexample Game Payoffs	160
9.1	Dwayne, Pen, Sword, Rock, Paper, Scissors Game	165
9.2	Ratings for Standard Games	167
9.3	NES Parameterizations	174

List of Algorithms

2.1	Exhaustive Environment Tree Search	35
2.2	Double Oracle	64
3.1	Equilibrium-Invariant Embedding Sampling	74
3.2	Trivial Embedding Sampling	75
4.1	Equilibrium-Symmetric Embedding	90
5.1	Two-Player Global Visualization	107
5.2	N-Player Local Visualization	107
7.1	PSRO	130
7.2	JPSRO	130
9.1	Generalized Payoff Rating	164

Nomenclature

Sets

\mathbb{R}	Real space.
\mathbb{P}	Probability space.

Spaces

$s \sim S \in \mathcal{S}$	Sample state from random variable in state space.
$z \sim Z \in \mathcal{Z}$	Sample transient state from random variable in transient state space.
$b \sim B \in \mathcal{B}$	Sample absorbing state from random variable in absorbing state space.
$o \sim O \in \mathcal{O}$	Sample observation from random variable in observation space.
$a \sim A \in \mathcal{A}$	Sample action from random variable in action space.

Constants

$I[s, s]$	Identity matrix with main diagonal elements equal to one.
$e[s]$	One vector with all elements equal to one.
$E[s, s]$	One matrix with all elements equal to one.

Distributions

$d_0[s]$	Initial state distribution.
$d_k^\pi[s]$	State distribution after k steps under policy.
$d_\infty^\pi[s]$	Stationary state distribution under policy.

Markov Decision Process

$\pi[s, a]$	Policy distribution.
$\pi^*[s, a]$	Optimal policy distribution.
$T[s, a, s']$	Transition function of the environment.
$R[s, a, s']$	Expected reward function for a transition.
$\gamma[s, a, s']$	Expected discount function.

Markov Reward Process

$T^\pi[s, s']$	Transition function of the environment under a policy.
$R^\pi[s, s']$	Expected reward function for a transition under a policy.
$\gamma^\pi[s, s']$	Expected discount function under policy.

Absorbing Markov Process

$N_Z^\pi[z, z']$	Fundamental matrix for absorbing Markov chains.
$\lambda_Z^\pi(d_0)[z]$	Hitting counts for absorbing Markov chains.
$H_Z^\pi[z, z']$	Hitting probability for absorbing Markov chains.

Ergodic Markov Process

$N_L^\pi[s, s']$	Fundamental matrix for ergodic Markov chains.
$d_\infty^\pi[s]$	Stationary distribution for ergodic Markov chains.
$H_L^\pi[s, s']$	Hitting probability for ergodic Markov chains.

Discounted Markov Process

$N_{\gamma T}^\pi[s, s']$	Fundamental matrix for discounted Markov chains.
$\lambda_\gamma^\pi(d_0)[s]$	Discounted hitting counts for ergodic Markov chains.
$H_\gamma^\pi[s, s']$	Hitting probability for discounted Markov chains.

State Value Functions

$v^\pi[s]$	State value (expected return) function under policy.
$v^{\pi^*}[s]$	State value function under optimal policy.

Action Value Functions

$q^\pi[s, a]$	Action value (expected return) function under policy.
$q^{\pi^*}[s, a]$	Action value function under optimal policy.

Game Theory

$a_p \in \mathcal{A}_p$	Player p's action.
-------------------------	--------------------

$a_{-p} \in \mathcal{A}_{-p}$	Other players' actions.
$a \in \mathcal{A}$	Joint action
$G_p[a]$	Player p's payoff function.
$G[p, a]$	All players' payoffs function.
ϵ_p	Equilibrium approximation parameter.
$\sigma[a]$	Joint equilibrium distribution.
$\sigma_p[a_p]$	Marginal equilibrium distribution.
$\sigma[a_{-p} a_p]$	Conditional equilibrium distribution.

CEs and CCEs

$a \in \mathcal{A}$	Equilibrium actions.
$a'_p \in \mathcal{A}_p$	Player p's deviation action.
$a''_p \in \mathcal{A}_p$	Player p's recommended action.
$A_p^{\text{CE}}[a'_p, a''_p, a]$	CE deviation gain function.
$A_p^{\text{CCE}}[a'_p, a]$	CCE deviation gain function.
$\alpha_p^{\text{CE}}[a'_p, a''_p]$	CE deviation gain dual variables.
$\alpha_p^{\text{CCE}}[a'_p]$	CCE deviation gain dual variables.
$\beta[a]$	Nonnegative probability dual variables.
λ	Probability unit-sum dual variable.

Other

$\phi[a] = \phi[a_1, \dots, A_N]$	Target joint distribution.
-----------------------------------	----------------------------

$$a_{-p} = (a_1, \dots, a_{p-1}, a_{p+1}, \dots, a_N)$$

$$a = (a_1, \dots, a_N).$$

$$G_p[a] = G_p[a_1, \dots, a_N]$$

$$\sigma[a] = \sigma[a_1, \dots, a_N]$$

Chapter 1

Introduction

Impressive progress in single-agent environments (Atari ([Mnih et al., 2015](#))) has been driven by advances in deep reinforcement learning (DRL), where the generalization power of neural networks combined with the scalability of RL has enabled a surge in progress. Environments with more than a single player, called games, have had similar success, making headlines around the world (Backgammon ([Tesauro, 1995](#)), Go ([Silver et al., 2016](#)), Chess ([Campbell et al., 2002](#)), Shogi ([Silver et al., 2018](#)), and Stratego ([Perolat et al., 2022](#))). This success is also replicated in more complex model-free (Go, Chess, and Shogi ([Schrittwieser et al., 2019](#))) and partially observable (StarCraft ([Vinyals et al., 2019](#))) games. While this list appears diverse, the games are similar from a game theoretic perspective: they are all two-player zero-sum, where training algorithms are theoretically known to converge to approximately optimal solutions. Attempts at progress beyond two-player zero-sum (Capture the Flag ([Jaderberg et al., 2019](#)), Soccer ([Liu et al., 2021](#)), Hide and Seek ([Baker et al., 2019](#)), Poker ([Brown and Sandholm, 2019](#)), Dota ([Bernier et al., 2019](#)), Diplomacy ([Anthony et al., 2020a](#); [FAIR et al., 2022](#))), while impressive and in some cases super-human, have fallen short of being considered solutions to these games. The policies produced by these techniques can be exploitable, their training can be unstable, and convergence to a particular objective can be unsatisfactorily defined.

This disparity is caused by a theoretic and pragmatic step-change in complexity that arises when venturing beyond two-player zero-sum games ([von Neumann and Morgenstern, 1947](#)) or common-payoff games. Two-player zero-sum is a purely competitive context: one player's gain is the other's loss. Each player is only reacting to one other player's potential actions. There is no doubt that the relationship between the two players is adversarial, and therefore there is no need to coordinate with the other player. By comparison, in common-payoff games, all players receive the same payoff. In such a setting it is in all players' interests to coordinate as there is a single perfectly cooperative goal. The properties of these two game classes greatly simplify the training problem, and existing tools (reinforcement learning) and solutions (Nash equilibrium) can be readily leveraged to solve these games.

The term general-sum encompasses all possible game payoff structures, from purely competitive zero-sum, to purely cooperative common-payoff, to mixed-motive (the games in-between). The primary difficulty general-sum introduces is that a player may now want to coordinate while also retaining some competitive goals. For example, cars at a junction may want to coordinate to avoid colliding with each other, but individually also wish to minimize the time they spend idle waiting for other cars to pass. Having more than two players, sometimes called n-player, is also a complication. Now it is not obvious which players are working with, against, or are indifferent to other players. Even in a purely competitive setting, it may be advantageous

to collude with another player in order to avoid being exploited by a coalition yourself. Balancing these trade-offs is tricky and complex multiagent dynamics can arise in relatively simple games.

The difficulty in training multiagent systems arise from its decentralized multi-objective nature. It is not possible to simply consider the Pareto optimal set of solutions: each of the objectives (player payoffs) is self-incentivized, controls a subset of the parameters (actions), and has the prerogative to unilaterally modify its own actions if it is in its own interest. Therefore there is not just a notion of Pareto optimality, but also of stability. We wish to find solutions that satisfy both these conditions. It turns out that two-player zero-sum and common-payoff games have mechanisms to side step these issues by principally converting them from multi-objective optimization problems to single-objective optimization problems for which solutions are more readily available. The goal of this thesis is to develop algorithms that can converge to good, stable policies in n-player general-sum games.

1.1 Multiagent Problem

The polymath John von Neumann proved the Minimax theorem (von Neumann, 1928) which forms the basis of fundamental solutions to two-player zero-sum games that established the field of game theory (Casti, 1996). However he struggled to extend the theory much beyond this class (Bhattacharya, 2022). The theory of n-player general-sum games are still underdeveloped today. Disappointingly, the majority of contemporary research in game theory still focuses on two-player zero-sum. The difficulty is because of several reasons including multi-objectiveness, non-stationarity, and complexity explosion. In whole, we refer to these difficulties as the *multiagent problem*.

1.1.1 Inherently Multi-Objective

Finding solutions to games is a multi-objective problem with the additional properties that a) the objectives are competitive with one another, b) have agency to act in their own self interest, and c) have unilateral control over some portion of the parameters.

Training agents to perform well in such a setting raises a question of how to measure progress. If different players have different payoffs, which ones should we prioritise? Can the scales of the payoffs of each player even be compared? Should we aim for utilitarian outcomes, or fair outcomes, or just outcomes, however those could be defined? In single-objective RL, there is a single rational outcome: maximize expected reward or some risk profile over expected rewards. However in multiagent, in general there is no known one right way, which poses a problem. The lack of a unique target to work towards makes it difficult to design algorithms that optimize for a desired outcome.

Therefore the field of game theory instead chooses to focus on so-called *solution concepts*. The most popular class of these are equilibrium concepts. Instead of specifying desired outcomes, these concepts instead specify which outcomes are stable, meaning that rational agents would not have incentive to unilaterally deviate from such an outcome. Any particular multiagent problem may have infinitely many such stable points, and there is not necessarily a prescriptive way to choose amongst these points, so equilibrium concepts on their own do not solve the multiagent problem. The additional problem of choosing between solutions is known as the *equilibrium selection problem*.

However, there are some special cases when equilibrium solution concepts come close to defining a “true” objective. The mechanism by which this is achieved is through recasting the multi-objective problem as a single objective one through principled arguments. The easiest class of games where this is possible are *common-payoff* games, where the goals of all players are identical. In this case we could simply merge all the players into a single abstract player, which reduces the multi-objective problem to a single-agent

single-objective one. This new merged player would have an enlarged strategy-space which is the product of all the original players' strategies. Even if we wanted to avoid merging players the solutions are still relatively easy to find (for example a centralized maximum operator would find a Nash equilibrium) and, most importantly, there is a single obvious metric to optimize for.

Another, and the most famous, special case is when there are only two players and the objectives are purely competitive such that one player's gain is another's loss. In the literature, the second condition is often referred to as zero-sum¹, and the special case is therefore called *two-player zero-sum*. In this case, the Nash equilibrium and the Minimax solution coincide, and together define a stable solution that also minimizes the worst-case outcome for a player, whatever the opponent player does. For two-player zero-sum the Minimax solution is single-objective. Such a solution, while not necessarily unique, is tractable², unexploitable³, and interchangeable⁴. Therefore in two-player zero-sum, training agents towards a Nash equilibrium is a prescriptive goal. Because of this special case having a well-defined objective, huge progress has been made on extremely complex two-player, zero-sum games such as Go (Silver et al., 2016), Chess, StarCraft (Vinyals et al., 2019), and Stratego (Perolat et al., 2022). There exist artificial agents that are undeniably super-human in their performance.

These approaches for casting a multi-objective optimization problem as a single-objective optimization point to a fundamental question: is single-objective enough? This question is most explored in the reinforcement learning (RL) literature, where it is posed similarly: *is reward enough?* A drawback with RL is that it is single-objective, or at least imposes that multiple objectives must first be squashed onto a single scalar. The *reinforcement learning hypothesis*, coined by Littman, states that all goals can be characterized by maximizing over expected cumulative reward signal. Sutton is also a proponent of this hypothesis. This hypothesis is controversial. Rewards are easy to specify in a single-agent environment: they follow directly from the goal. However, in a multi-agent game, perhaps rewards would need additional shaping or mechanisms introduced to ensure convergence of greedy reward maximizing agents. Even if there existed scalar rewards one could give all agents that would ensure convergence to a solution (no such proof exists), calculating the reward function may be just as challenging as the original multi-agent problem. Without proof to the contrary, general multi-agent problems seem to be multi-objective in nature and this is a complication which must be overcome when designing algorithms in such settings.

1.1.2 Non-Stationary Environment

Conceptually, it is possible to ignore the other agents in a game such that they are subsumed into the rules of a single-agent environment: other players' actions are viewed no differently than the immutable physics of the world. Indeed this is a valid approach to take, so long as the environment remains stationary. This would mean that agents cannot adapt over time in response to your own, and others' changing behaviour. If we lived in such a world there would be no need to differentiate between players' actions and innate rules of the world (e.g. physics). Indeed the concepts of agency and theory of mind become nebulous.

This is, of course, an unrealistic model. Agents have their own agenda and will adapt behaviour over time, via learning or evolution, to ensure their goals are met. Therefore it is important to make a distinction between the entities in the world without agency and entities with agency. The distinction is so important that humans have evolved a cognitive bias for identifying agency in the world. This sense is so acute that stories can be visualized with simple shapes and people will instinctively construct a narrative that describes the behaviour of these shapes in the context of their individual goals as if they were alive (Figure 1.1) (Heider and Simmel, 1944). Humans are also known to see agency in complex natural phenomenon, such as

¹Summing over the two players' payoffs is equal to zero.

²Solvable in polynomial time.

³There is no strategy the opponent can play which will reduce your own payoff.

⁴If there are multiple equilibria, each is in equilibrium with any other, and all combinations result in the same payoff.

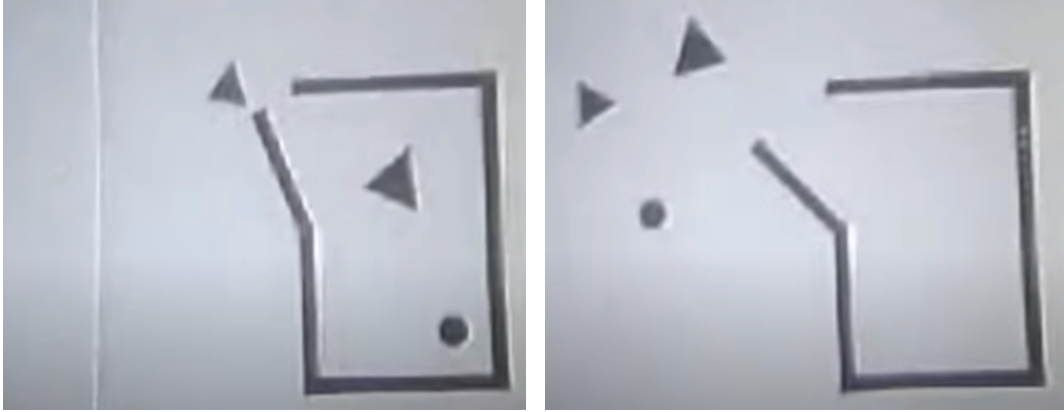


Figure 1.1: Stills from Heider and Simmel’s video ([Heider and Simmel, 1944](#)) demonstrating agency of simple shapes. When people are asked to describe what they see in the scene, they describe a story involving a conflict between the triangles and circles (entities with agency), in a world containing a room represented by the straight lines (entities without agency).

the stars, where there is none. This is evidence that there is an evolutionary advantage to overestimate the degree of agency in the world, rather than underestimate it. This shows that observing the world through a multiagent lens is, at least through human evolution, extremely important. People who have an impaired theory of mind struggle with many aspects of life. Perhaps the ability to recognise agency is to help learning: the ability to differentiate between rule-based changes in the world and agency-based changes in the world would be useful in separating the parts of the environment that are stationary and those that are non-stationary.

It is this complication that explains why training in a multiagent setting is complicated. When one player changes its behaviour, others necessarily respond by changing their own. It becomes advantageous to model what other agents may be thinking or know (theory of mind). It is not hard to see why building algorithms in such settings may be challenging. How do you ensure a population of learning agents converges to a sensible global solution? Do other agents’ minds and goals need to be modelled by individual agents?

As an example, consider the simple and popular two-player zero-sum children’s game, rock-paper-scissors. This game has cyclic dynamics: rock beats scissors, paper beats rock, scissors beats paper. The winning strategy is to play whatever counters the strategy of the other player. However if any player were to play deterministically they would be easily exploitable, so players should adopt a randomized strategy. Imagine a learning strategy where players would play some distribution over rock, paper, and scissors against others. They would increase the probability of playing strategies that win more on average and reduce the probability of strategies that lose on average. Naive learning dynamics (Figure 1.2) of such a non-stationary system does not result in convergence to the equilibrium of this game: playing rock, paper, and scissors with equal probability.

To make matters worse, many scalable RL methods that we hope to leverage assume a stationary environment, and either do not work or their convergence proofs break down when used in a non-stationary setting. Even if the RL methods did work, they would become moving targets for other agents.

1.1.3 An explosion in Complexity

The presence of players within an environment results in an explosion in complexity within games. The policies that other agents deploy influence the transition dynamics and hence the state visitation distribution of a game. Policies can be stochastic resulting in a continuum of possible modifications to the environment

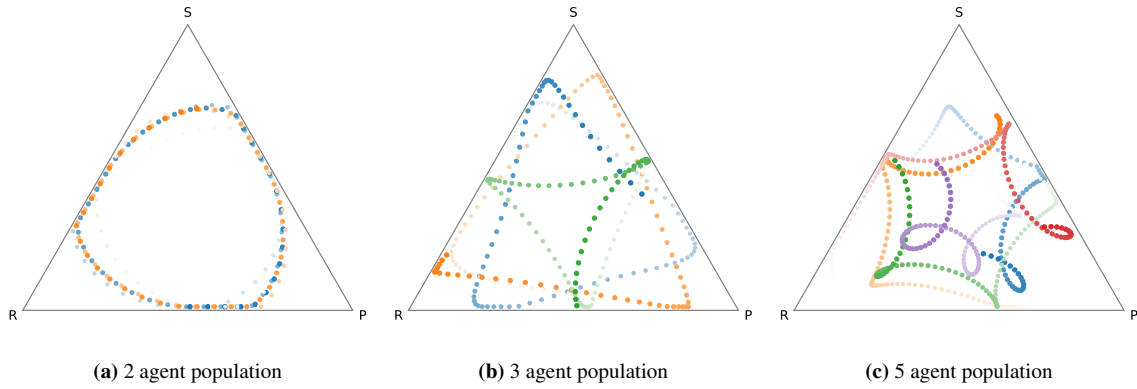


Figure 1.2: Non-stationary learning dynamics. Shows the evolution of the strategies of learning agents in rock-paper-scissors with different sized populations. Each agent’s mixed strategies over time are shown on a two-simplex. The points labelled “R”, “P” and “S” represent playing pure strategies and the space in between represents mixed strategies. The Nash equilibrium (the unexploitable solution to this two-player zero-sum game) is the point (not indicated) in the centre of the equilateral triangle: randomly playing all strategies with equal probability.)

which need to be modelled. For example, to emphasise the complexity that multiagency introduces, an analogous single agent environment would be like having parameterized rules which could be changed at any moment. Perhaps the laws of gravity could be inverted, or the melting point of water altered. A competent agent in such an environment would need to know how to adapt to such changes and clearly such a world would be very complex.

However, recent advances in machine learning provide optimism that increasingly complex environments can be tackled. The recent deluge of machine learning progress has been powered by three key ingredients; scalable and generalizable function approximation in the form of deep neural networks, staggering amounts of cheap compute power, and breakthroughs in scaling reinforcement learning.

Neural networks (NNs) were always promising because of their architectural flexibility, hierarchical structure, efficient gradient computation, and approximation power. The driving force behind training NNs is backpropagation (Kelley, 1960; Rumelhart et al., 1986), an efficient implementation of the chain rule. However, historically NNs with many layers, so called deep neural networks (DNNs), proved difficult to train because of issues with gradients diverging or vanishing as they progressively passed through more and more layers. The breakthroughs enabling DNNs can be attributed to many people. Hinton demonstrated that pre-training layers significantly helped. LeCun, popularized convolutional neural network (CNN) architectures which utilize weight sharing and are particularly efficient at image tasks. Hochreiter and Schmidhuber provided an architecture (LSTMs) (Hochreiter and Schmidhuber, 1997) that allowed training recurrent neural networks, which are deep over time. Regularization, like dropout (Srivastava et al., 2014), also helped with generalization issues. Additionally, benchmark datasets such as MNIST (Deng, 2012) and ImageNet (Deng et al., 2009) have enabled a research community to blossom around this field. Since then, there have been many more improvements including parameter initialization (Glorot and Bengio, 2010), nonlinearities (Agarap, 2018), adaptive optimizers (Kingma and Ba, 2014; Tieleman et al., 2012), batch normalization (Ioffe and Szegedy, 2015), gradient clipping (Mikolov, 2012), gradient scaling (Hessel et al., 2018), optimized hardware (GPUs, TPUs) (Google, 2023), and auto-differentiation libraries (Abadi et al., 2015; Bradbury et al., 2018; Paszke et al., 2019), which have made training DNNs comparatively easy.

Another key ingredient is reinforcement learning at scale. Early progress in planning (Deep Blue (Campbell et al., 2002)), and dynamic programming (Bellman, 1957a) showed that temporally extended tasks could be solved with the right algorithms. Value based methods deployed on larger games (TD-

Gammon (Tesauro, 1995)) were developed which improved performance further. Deep neural network function approximators were then used to learn values of games with large states (DQN (Mnih et al., 2015)). Search can be incorporated into the value targets to improve variance (AlphaGo (Silver et al., 2016)). Many policy gradient algorithms (Espeholt et al., 2018; Williams, 1992) have been developed which scale to even larger domains without knowledge of the dynamics of the environment.

Bigger networks, longer training times, bigger datasets, and more thorough hyper-parameter sweeping have consistently resulted in better performance. Therefore, it is clear to see why the growing abundance of cheaper, more powerful, and more specialised hardware translates to advances in machine learning. Moore’s Law (Moore, 1965; Schaller, 1997) is a well known phenomenon that had driven this progress, but more recently specialised hardware, such as Graphical Processing Units (GPUs), and custom Tensor Processing Units (TPUs) have driven compute power to even further heights. Progress in fast memory, such as RAM and SSDs have also allowed huge datasets to be streamed with ease. Indeed we are now approaching a level of computation comparable to the human brain: TPU pods can calculate 1 exaflop (goo), while the brain is estimated to be 100 teraflops (Moravec, 2000). Of course the brain is not a general computing unit, it possesses structural biases honed over billions of years of evolution to excel at the tasks of survival in the world we find ourselves in. Nevertheless, since we know the brain is a capable intelligence, this could be evidence that human-level performance of tasks may be within reach with the right algorithms and structural biases.

1.1.4 Goals and a Non-Goal

In summary, progress in the n-player general-sum multiagent settings has been slow. It is difficult because of three problems.

Metric Problem: Multiagent games are a multi-objective problem and it is not clear what the correct target objective, metric or solution concept is for n-player general-sum games.

Algorithm Problem: Multiagent games are non-stationary and even given an established solution concept, it is difficult to build algorithms that converge to a solution.

Complexity Problem: Multiagent games have more complexity, which is driven by players able to influence the game dynamics.

This thesis has an explicit non-goal: it will not search for an answer to the metric problem nor claim that the solution concepts studied within are the only metrics to measure progress on n-player general-sum games. It will attempt to explain the advantages and trade-offs of the solution concepts studied, however it will not dwell on these. While the metric problem is an important area of research it can also be a distraction. Instead this work focuses on the equally hard algorithm and complexity problems: given a reasonable solution concept, how does one build algorithms that converge to this solution in n-player general-sum games? This work exploits advances in machine learning to overcome the complexity problem.

Much of the work in this thesis is building a framework of tools necessary to tackle this problem. In order to scope the work, the thesis follows several themes. It focuses, without devotion, on mediated equilibrium solution concepts like *correlated equilibrium* (CE) (Aumann, 1974) and *coarse correlated equilibrium* (CCE) (Moulin and Vial, 1978) which are mathematically convenient (specifically they have a convex set of feasible solutions) and permit coordination between players (an important property in cooperative and mixed-motive settings). This thesis prefers, where possible, unique equilibrium selection for consistency and practicality. Assuming solution convexity, any strictly convex function could be used to select an equilibrium uniquely avoiding the equilibrium selection problem. Algorithms are built around game theoretic first principles and should have convergence guarantees. Finally, the primitive components of the algorithms should be scalable using deep learning or reinforcement learning. In summary this work develops techniques along these themes:

- Choose solution concepts for *convexity*.
- Choose equilibrium for *uniqueness*.
- Develop principled game theoretic algorithms for *convergence*.
- Use DL and RL primitives for *scale*.

A limitation of focusing on mediated equilibrium techniques is that, in non-zero-sum games, there must be a mediator⁵ at test time to enable implementation of the solution. In some ways, the presence of a mediator is advantageous: mediated equilibria permit coordination between players, are richer, and contain higher payoffs. However, the mediator’s presence in the game may not always be realistic for a number of reasons. Firstly, engineering a trustworthy entity to act as a mediator may be fraught. All players would have to have faith in the implementation and execution of such a central authority. Indeed, there are successful examples of mediators (traffic lights are widely trusted and observed, and the internet protocol is successful), however it obviously may not be pragmatic in all scenarios. Secondly, mediated solution concepts require communicating recommendations, be they individual actions (in extensive-form) or whole policies (in normal-form), to players. This communication channel may not be available in practice and therefore may limit the applicability of mediated solutions. Finally, the equilibrium solutions in n-player general-sum games are only useful if all players are aware of, and are willing to execute the equilibrium. Players *should* be incentivized to play an equilibrium, however there is no mechanism to force players to act in their own self-interest. This is in contrast to the Nash equilibrium in two-player zero-sum games which is unexploitable regardless of what an opponent chooses to do. Most of these drawbacks are fundamental to n-player general-sum games and would apply to other solution concepts too. Additionally, even if exact equilibria may not be easily implemented certain situations, having learning algorithms that converge to principled joint policies is still advantageous. For example, crudely marginalising an equilibrium such that it can be decentrally executed without a mediator may still result in better performance than non-game theoretic methods that train directly on decentralized solutions.

1.2 The Multiagent Worldview

Solving multiagent problems is difficult, but is it useful? The most obvious application of multiagent learning agents is to board games. Indeed most research, and research referenced in this thesis, has focused on such games. However this should not be mistaken for a narrow applicability. Board games are used as research benchmarks because they a) have engaging strategic dynamics, b) have rules honed over many years, c) are understood intuitively by people, d) have strong human baselines, e) have established theory, f) have code implementations, and g) are externally defined. In general, the learning algorithms developed in this work can be applied to many problems that have actors each striving for their goal. Climate negotiations, the stock market, national defence, and economic policy are all examples of difficult problems that can be modelled as games or multiagent problems. The multiagent worldview describes a world that is not just composed of many things with complex rule-based interactions: it is one where parts of the world have agency and it is advantageous to consider that agency.

1.2.1 Gamification

Many problems could be recast as multiagent, even when the underlying problem may not seem multiagent in nature. This framing is known as *gamification*. The most famous recent examples are GANs (Goodfellow et al., 2014) which constructs a two-player zero-sum game to generate high quality generated images that cannot be differentiated from a real image. In this approach, a generator player attempts to generate an image and a discriminator player attempts to classify whether an image is generated or real. Others have reframed fundamental mathematical concepts like the eigenvalue problem (Gemp et al., 2020) as a game.

⁵Also sometimes called a correlation device.

This field is underdeveloped but demonstrates that the tools of games are useful even for problems without agency.

1.2.2 Evolution and The Auto-Curriculum Property

We know that human intelligence arose through evolution, but it did not do so in isolation. Cooperation and competition between agents was an important component in driving intelligence of the human race. We also know that simple rules and interacting agents can lead to enormous complexity and open-ended outcomes (Leibo et al., 2019). It is therefore possible to see that from multiagent systems emerges an auto-curriculum property where, as interacting agents become more competent, the complexity of the system they are acting in increases, and therefore drives the necessity for further progress. This property is particularly desirable because it is clear that a hand-crafted set of ordered tasks (a curriculum) does not scale. This is true for many reasons including a) there are many possible tasks, b) it is not clear what subset of these tasks are useful for learning, and c) it is not clear in what order they should be presented.

Therefore, the multiagent auto-curriculum property (Graepel, 2020; Leibo et al., 2019) may be a key to designing a path to general intelligence. Evidence for this view is mounting: the game of Go was famously mastered using self-play (Silver et al., 2016) (and later Chess, and Shogi (Silver et al., 2018)), Capture the Flag with a population of agents (Jaderberg et al., 2019), and StarCraft using a league of agents (Vinyals et al., 2019). All these training regimes benefited from training against progressively stronger opponents in various forms (not too weak or too strong).

1.2.3 Multiagent World

The most basic argument for studying multiagent systems is that we live in a multiagent world, with multi-agent problems. Taking a broad view, many of the world's biggest problems are social dilemmas or moral hazards, whether it be agreeing to climate treaties, cooperating on free-trade, controlling nuclear arms proliferation, setting fiscal policy, or developing and sharing (or even taking) vaccines. These are complex multiagent problems with incentives that result in both competitive and cooperative behaviour. The study of multiagent systems need not only be about how to behave in such a system, but also includes the area of research about how to design multiagent systems that incentivize cooperative behaviour (mechanism design) that maximises some measure of social welfare. There have already been some examples of this, for example in auction design (Edelman et al., 2005). Even taking a more narrow view, a hypothesised artificially intelligent agent will necessarily have to interact with, and understand the incentives of its creators - humans - and of other agents it interacts with if it is to be of maximum utility. An intelligence that simply maximizes its own single-agent reward may be more limited than an agent that understands the goals of others.

Chapter 2

Literature Review

Understanding multiagent training beyond two-player zero-sum requires focusing on two areas of research; *Reinforcement Learning* (RL) and *Game Theory* (GT), and their intersection *Multiagent Reinforcement Learning* (MARL).

RL is focused on individual agent (single-objective) reward maximization, within a possibly partially observed environment, where the environment dynamics may or may not be known. In recent years, RL has exploded in popularity, primarily because of a breakthrough (DQN, (Mnih et al., 2015)) in scaling classic RL algorithms to use neural network function approximators and the commoditization of compute resources. Although there are still breakthroughs, the field is maturing, and increasingly more complex applications are being tackled: Atari (Mnih et al., 2015), Go (Silver et al., 2016), StarCraft (Vinyals et al., 2019), and Stratego (Perolat et al., 2022). Silver’s lecture course (Silver, 2015), Kaelbling’s survey (Kaelbling et al., 1996a,b), and Sutton & Barto’s book (Sutton and Barto, 2018) provide excellent introductions to the field of RL. The fundamentals, state of the art, and a taxonomy of RL are summarized in Section 2.1.

GT is focused on the interactions and behaviour of rational agents in competitive and cooperative scenarios. Although the field is mature, most theory is confined to two-player, zero-sum (purely competitive games where one player’s loss is another’s gain) games. Solutions to games are usually described via equilibria: where no player has incentive to deviate from a particular set of strategies, the most famous one being Nash equilibrium (Nash, 1951). GT differs from RL in the fact that there are no clear metrics to optimize for in general games. Theory beyond two-player zero-sum is incomplete but some progress has been made. Section 2.2 gives a background to GT, particularly focusing beyond two-player, zero-sum.

The intersection of RL and GT, however, is less developed. Loosely this area of research can be thought of as scaling multiagent game theoretic techniques using reinforcement learning and function approximation. There have been attempts at training RL agents in multiagent domains with more than two players: Capture the Flag (Jaderberg et al., 2019), and Soccer (Liu et al., 2021). There are a number of surveys on MARL (Hernandez-Leal et al., 2017, 2019; Zhang et al., 2021).

2.1 Reinforcement Learning

Reinforcement learning focuses on the problem of training an agent to maximize cumulative reward within an environment by observing a state of the environment and interacting via actions (Figure 2.1). An agent interacting with an environment observes and produces a sequence subscripted by the timestep; $s_0 \rightarrow a_0 \rightarrow$

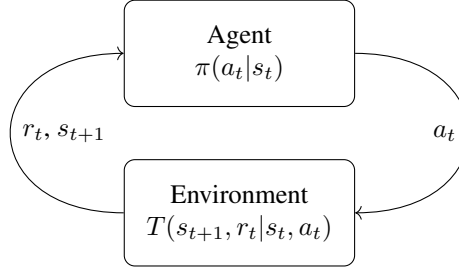


Figure 2.1: Agent-environment interaction in a Markov decision process. Where π is the *agent policy*, T is the *environment dynamics*, $a \sim A \in \mathcal{A}$ is the *sample agent action*, $s \sim S \in \mathcal{S}$ is the *sample environment state*, $\gamma \sim \Gamma \in [0, 1]$ is the *sample discount* and $r \sim R \in \mathcal{R}$ is the *sample reward*. This loop results in random trajectories $S_0 \rightarrow A_0 \rightarrow \Gamma_0 \rightarrow R_0 \rightarrow S_1 \rightarrow A_1 \rightarrow \Gamma_1 \rightarrow R_1 \rightarrow S_2 \rightarrow \dots \rightarrow S_T$ with sample trajectories $s_0 \rightarrow a_0 \rightarrow \gamma_0 \rightarrow r_0 \rightarrow s_1 \rightarrow a_1 \rightarrow \gamma_1 \rightarrow r_1 \rightarrow s_2 \rightarrow \dots \rightarrow s_T$.

$\gamma_0 \rightarrow r_0 \rightarrow s_1 \rightarrow a_1 \rightarrow \gamma_1 \rightarrow r_1 \rightarrow s_2 \rightarrow \dots$ sampled from random variables $S_0, A_0, \Gamma_0, R_0, S_1, A_1, \dots$. Environments and policies are often stochastic, therefore upper case is used to denote random variables, and lowercase to denote samples.

The goal of RL is to find the policy that maximizes the cumulative reward, the so called optimal policy. The cumulative reward from a time point t is a random variable called the *return*. Note that for long running environments, or environments that never terminate, the cumulative reward may be unbounded, therefore other definitions of return such as the *average return* or *discounted return* can be used.

$$\text{Return: } G_t = R_t + R_{t+1} + \dots + R_T = \sum_{k=0}^T R_{t+k} \quad (2.1a)$$

$$\text{Average Return: } G_t = \lim_{N \rightarrow \infty} \frac{1}{N} (R_t + R_{t+1} + \dots + R_N) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^N R_{t+k} \quad (2.1b)$$

$$\text{Discounted Return: } G_t^\gamma = R_t + \gamma R_{t+1} + \dots + \gamma^\infty R_\infty = \sum_{k=0}^{\infty} \gamma^k R_{t+k} \quad (2.1c)$$

Where $0 \leq \gamma < 1$ is called the *discount rate*. Values of γ close to 0 result in myopic agents and value close to 1 result in far-sighted agents. Using discounting is beneficial because it a) works for non-terminating environments, b) is more mathematically convenient than using average return, c) discounts rewards furthest into the future (which are also the most uncertain), resulting in a more stable target, d) it captures useful phenomenon in some applications (like finance where there is inflation or immediate reward can be reinvested for greater return), and e) there is biological evidence that animals prefer immediate reward over delayed reward (Mischel and Ebbsen, 1970; Vanderveldt et al., 2016). The return is a special case of discounted return with $\gamma = 1$. It is still valid to have the discount rate be defined independently for each timestep, or even be stochastic; $\gamma_t \sim \Gamma_t$. We define $\gamma[s, a, s'] = \mathbb{E}[\gamma_t | S_t = s, A_t = a]$ to be the expected discount at a transition and $\gamma^\pi[s] = \mathbb{E}_\pi[\gamma_t | S_t = s]$ to be the expected discount at state s , under policy π . Using this notation is it possible to model a termination of an environment at timestep t by using $\gamma_t = 0$. A natural environment terminal state and a state with zero discount are mathematically equivalent. For this reason, the discount rate is sometimes called referred to as the *probability to continue*.

The formulation of RL concerns the training of a single agent. However one may also naively use this in a multiagent setting by subsuming other agents into the environment. In a certain sense this is a valid approach because the distinction between innate “hard coded” entities within an environment and trained entities within environment is unimportant – the agent will maximize reward against whatever environment

dynamics it is faced with. It is not difficult to imagine, however, that things become less simple if the opponent agents are not fixed but change over time (for example perhaps the other agents are also training to maximize their own objectives). This means the environment becomes non-stationary. RL does not provide the tools to fully deal with this scenario, and so additional theory is required which is introduced in Section 2.2. Discussion on multiagent training will be paused until these later sections and this section will instead focus on single-agent training in a stationary environment.

2.1.1 Fundamentals

This section describes formalisations useful for training agents in the RL framework. Later sections introduce algorithms for performing training under different restrictions.

2.1.1.1 Markov Decision Processes

A Markov decision process (MDP) (Bellman, 1957b) is defined on a set of states, $s \in \mathcal{S}$, with actions available in each state $a \in \mathcal{A}$ (or more generally $a_s \in \mathcal{A}_s$). Transition probabilities are defined between these states which map state-action pairs to a next state $T : \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{S}' \mapsto \mathbb{P}$. If there are finite states and actions this transition can be tabulated in a tensor, $T[s, a, s']$, which is a categorical probability distribution over the next state; $\sum_{s'} T[s, a, s'] = 1 \quad \forall s, a$ and $T[s, a, s'] \geq 0 \quad \forall s, a, s'$. Furthermore, define a reward function $\tilde{R} : \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{S}' \otimes \mathcal{R} \mapsto \mathbb{R}$ between states and actions. In general this may be a stochastic function. If there are finite states and actions we can tabulate the expected reward for a transition; $R[s, a, s'] = \mathbb{E}_r [\tilde{R}(s, a, s', r)]$. The reward function is only defined where $T[s, a, s'] > 0$, however for simplicity zero can be used for undefined elements. Finally, we can define a discount $\Gamma : \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{S}' \mapsto \mathbb{P}$ which specifies the probability of continuing to the next state. Similarly this can be tabulated in $\Gamma[s, a, s']$. The discount serves two purposes; the firstly discounts rewards that are far into the future which it turn ensures infinite horizon domains are well defined. Secondly, it allows us to signal termination of an episode when $\gamma = 0$.

A key property of MDPs is that all states satisfy the Markov property (Gagniuc, 2017). A state is Markov if and only if the current state summarizes all important historical information: $P(S_{t+1}|S_t, A_t) = P^\pi(S_{t+1}|S_t, Q_t, S_{t-1}, A_{t-1} \dots)$. With this formulation, the current state completely determines the characteristics of the process, and therefore the historical states and action do not need to be considered, which greatly simplifies the problem setting. There is some redundancy in the formulation of terminating MDPs. For example, one could formulate for terminal states, z , $T[z, a, s'] = 1, z = s', = 0, z \neq s' \forall a$ and $R[z, a, s'] = 0 \forall a, s'$. Or one could build termination into the discount $\gamma[s] = 0, z = s, = \gamma, z \neq s$. There could be a single terminal node - or several terminal nodes.

2.1.1.2 Markov Reward Process and Markov Processes

A MDP can be converted to a simpler framework Markov reward process (MRP) (Howard, 1971) by applying a policy, $\pi[s, a]$, to result in a new transition function $T^\pi : \mathcal{S} \otimes \mathcal{S}' \mapsto \mathbb{P}$ tabulated as $T^\pi[s, s'] = \sum_a \pi[s, a] T[s, a, s']$ and expected reward function $R^\pi[s, s'] = \sum_a \pi[s, a] R[s, a, s']$. Furthermore, if the reward function is dropped, and we only consider the transitions, we are left with a Markov process (MP) (Kemeny et al., 1960). There is a rich body of theory built around MPs.

The sequence of states sampled under a Markov process is called a Markov chain. Given an initial state distribution $d_0[s]$, it is possible to calculate the state distribution k steps into the future under a policy π : $d_k^\pi[s'] = \sum_s d_0[s] T^\pi[s, s']^k$. It is often desirable to classify states according to whether it is possible to reach a state from another state. If a pair of states can be reached from one another, they are said to *communicate*. States can be divided into *equivalence classes*: if two states communicate they are in the same equivalence class. Equivalence classes can be partially ordered according to whether they feed or receive other equivalence class. For example, all terminal (also called *absorbing*) states are part of their own singleton equivalence class. A state, z , is absorbing only if $T^\pi[z, z] = 1$. The minimal elements of the

partial ordering of equivalence classes are called *absorbing sets*, all other classes are called *transient sets*. A state, z , is *transient* if there is a nonzero probability that it will never return to z . Otherwise a state is said to be *recurrent*. Every Markov chain must have at least one recurrent set (since every partial ordering has a minimal element). It is advantageous to represent the reordered transition matrix, $\hat{T}^\pi[s, s']$, in a canonical form that captures this partial ordering, where Z_i is the i th equivalence class and represents transitions within that class, and B_{ij} represents transitions between equivalence class i to j . If Z_i is recurrent, then $B_{i,j} = 0 \forall j$.

$$\hat{T}^\pi[s, s'] = \begin{bmatrix} Z_0 & B_{0,1} & \dots & B_{0,n-1} \\ 0 & Z_1 & \dots & B_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Z_{n-1} \end{bmatrix} \quad \hat{T}^\pi[s, s']^k = \begin{bmatrix} Z_0^k & B_{0,1} & \dots & B_{0,n-1} \\ 0 & Z_1^k & \dots & B_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Z_{n-1}^k \end{bmatrix} \quad (2.2)$$

If a Markov chain has only a single class (by construction a recurrent class) it is said to be *irreducible*. A Markov chain is *aperiodic* if there exists a k such that $T^\pi[s, s']^k > 0 \forall (s, s')$. Similarly, the period of the a state s has period p if p is the greatest common divisor of the number of transitions by which s can be reached, starting from s : $\gcd\{k > 0 : T[s, s']^k > 0\}$. In aperiodic Markov chains all states have period $p = 1$. A Markov chain is said to be *ergodic* if all states are both recurrent and aperiodic.

It is advantageous to study MPs separately according to the structure of their canonical form. Kemeny and Snell (Kemeny et al., 1960) provide a good overview of these different classes of MPs. Three important categories are *absorbing Markov chains*, *ergodic Markov chains*, and *irreducible Markov chains* which will be discussed further below. Note that there is some inconsistency in the literature about the names of some definitions, for example some redefine ergodic as regular, and irreducible as ergodic. Because this thesis studies MPs in the context of discounted MDPs, it will also consider some discounted versions of definitions that do not appear in MP literature.

Absorbing Markov chains

If a MP has absorbing states $b \in \mathcal{B}$ (terminal states that once entered cannot be left) and transient states $z \in \mathcal{Z}$. If $|\mathcal{B}| > 0$, and every transient state t can reach an absorbing state, it is called an *absorbing Markov chain*. All terminating environments induce absorbing Markov chains with any policy. Non-terminating environments can also induce an absorbing Markov chain under certain policies (for example if there is noop action that results in staying the same state). In this scenario, we can reorder the transition matrix into a canonical form \hat{T}^π , where Z is a matrix of shape $[z, z']$, B is a matrix of shape $[z, b']$, and I is the identity matrix of shape $[b, b']$.

$$\hat{T}^\pi[s, s'] = \begin{bmatrix} Z & B \\ 0 & I \end{bmatrix} \quad \hat{T}^\pi[s, s']^k = \begin{bmatrix} Z^k & \sum_k Z^k B \\ 0 & I \end{bmatrix} \quad \lim_{k \rightarrow \infty} \hat{T}^\pi[s, s']^k = \begin{bmatrix} 0 & NB \\ 0 & I \end{bmatrix} \quad (2.3)$$

Given this form, we define the *absorbing fundamental matrix*, $N[z, z'] = \sum_{k=0}^{\infty} Z^k = (I[z, z'] - Z[z, z'])^{-1}$ (using the Sherman-Morrison-Woodbury identity (Woodbury and of Statistics, 1950)). Quantities of the form $(I - A)^{-1}$ are sometimes called the *unity resolvent*. Note that $NZ = ZN = N - I$. The fundamental matrix describes the expected number of times the chain is in state z' given it started in state z . The variance of the fundamental matrix is given as $\hat{N}[z, z'] = N[z, z'](2\text{diag}(N)[z, z'] - I[z, z']) - N[z, z'] \odot N[z, z']$. We can further define an (undiscounted) frequency of being in a transient state, with initial state distribution $d_0[z]$, $\lambda^\pi(d_0)[z'] = \sum_z d_0[z]N[z, z']$. The expected number of steps of the chain before being absorbed, when starting in state z is given as $y[z] = \sum_{z'} N[z, z']1[z']$. The variance on the number of steps is $\sum_{z'} (2N[z, z'] - I[z, z'])y[z'] - y[z] \otimes y[z']$. The probability of being absorbed into

state b' from transient state t is known as the *absorbing probability* and is given by the upper right of the $\lim_{k \rightarrow \infty} \hat{T}^\pi[s, s']^k$ term, $\sum_{z'} N[z, z'] B[z', b']$. The probability of ever visiting state z' from state z is given by $H = (N - I) \text{diag}(N)^{-1}$.

Irreducible Ergodic Markov chains

A Markov process is ergodic if it is recurrent (each state is visited an infinite number of times) and aperiodic (each state is visited without any systematic period). By definition ergodic Markov chains are also irreducible. This means that any MPs with absorbing (terminal) states cannot be ergodic. This formulation is therefore best for analysing infinite horizon continuing environments.

An ergodic process has a limiting stationary distribution $d_\infty^\pi[s] = \sum_s d_0[s] (\lim_{k \rightarrow \infty} T^\pi[s, s']^k)$, where $d_0[s]$ is any valid probability distribution over initial states and $L = \lim_{k \rightarrow \infty} T^\pi[s, s']^k$ is called the *limiting distribution*. $d_\infty^\pi[s]$ is independent of initial conditions because all rows of L are the same for ergodic Markov chains and are equal to $d_\infty^\pi[s]$. All elements of $d_\infty^\pi[s]$ are positive.

$$L = \lim_{k \rightarrow \infty} T^\pi[s, s']^k = \begin{bmatrix} (d_\infty^\pi)^T \\ \vdots \\ (d_\infty^\pi)^T \end{bmatrix} \quad d_\infty^\pi = (d_\infty^\pi)^T \begin{bmatrix} (d_\infty^\pi)^T \\ \vdots \\ (d_\infty^\pi)^T \end{bmatrix} \quad \forall d_0^\pi \in \mathbb{P}^{|S|} \quad (2.4)$$

The stationary distribution can be calculated in closed form using the fact that $d_\infty^\pi[s] = T^\pi[s, s'] d_\infty^\pi[s]$. This is therefore the eigenvector of $T^\pi[s, s']$ corresponding with corresponding eigenvalue of 1. It can be computed by solving the linear program $\sum_s (I[s, s'] - T^\pi[s, s']) d_\infty^\pi[s] = 0$ with $\sum_s d_\infty^\pi[s] = 1$ and $d_\infty^\pi[s] \geq 0$. This can be converted to a linear system, where \tilde{T}^π is the transition matrix with the last column removed:

$$(d_\infty^\pi)^T [\tilde{I} - \tilde{T}^\pi \quad \bar{1}] = [\bar{0}^T \quad 1] \implies d_\infty^\pi = [\bar{0}^T \quad 1] [\tilde{I} - \tilde{T}^\pi \quad \bar{1}]^{-1} \quad (2.5)$$

Therefore, an ergodic Markov chain converges to an *equilibrium* with stationary distribution $d_\infty^\pi[s]$. Once in equilibrium, the chain has some interesting properties, for example it becomes reversible.

For ergodic Markov chains, we cannot define a fundamental matrix in the same way as for absorbing Markov chains because $\sum_k T^\pi[s, s']^k$ is unbounded. We therefore need to use alternative approaches. There are two approaches to this; firstly, leveraging the limiting distribution matrix, and secondly using discounting. It is also possible to define a modified matrix $T - L$ where L is the limiting distribution matrix defined above. Then define the *ergodic fundamental matrix*: $N_L = \sum_{k=0}^{\infty} (T - L)^k = I + \sum_{k=1}^{\infty} (T^k - L) = (I - T + L)^{-1}$. Note that $I - N = L - TN$. This fundamental matrix is not positive.

Discounted Markov chains

We have seen that although ergodic Markov chains have some interesting properties they can still pose some challenges because they are infinite horizon. There is an easy way of converting an any irreducible Markov chain into an absorbing one; using a discount, γ . Here the discount can be interpreted as “probability of continuing”, where $1 - \gamma$ is the probability of terminating at a particular state. This is a way of handling the infinite horizon case in RL, where states far into the future are down-weighted. We can put this in canonical form by introducing a single terminal state and discounted transition matrix.

$$\hat{T}^\pi[s, s'] = \begin{bmatrix} \gamma T & (1 - \gamma)e \\ 0 & 1 \end{bmatrix} \quad (2.6)$$

$$\hat{T}^\pi[s, s']^k = \begin{bmatrix} \gamma^k T^k & \sum_k \gamma^k T^k (1 - \gamma)e \\ 0 & 1 \end{bmatrix} \quad (2.7)$$

$$\lim_{k \rightarrow \infty} \hat{T}^\pi[s, s']^k = \begin{bmatrix} 0 & N_\gamma(1 - \gamma)e \\ 0 & 1 \end{bmatrix} \quad (2.8)$$

Leveraging discounting is it possible to define a *discounted fundamental matrix* $N_\gamma^\pi[s, s'] = \sum_{k=0}^{\infty} \gamma^k T^\pi[s, s']^k = (I[s, s'] - \gamma T^\pi[s, s'])^{-1}$. In this case N_γ^π is positive and all the other quantities can be defined like before (in the absorbing Markov chain case). Note that $\gamma T N_\gamma^\pi = I - N_\gamma^\pi$. The discounted cumulative frequency of being in a state, is defined for all MPs with $0 < \gamma < 1$: $\lambda(d_0)_\gamma^\pi[s'] = \sum_k \sum_s \gamma^k d_0[s] T^\pi[s, s']^k = \sum_s d_0[s] (I[s, s'] - \gamma T^\pi[s, s'])^{-1}$ and it is a function of an initial state distribution $d_0[s]$. The expected number of steps before terminating is $y[s] = \sum_{s'} N_\gamma^\pi[s, s'] 1[s'] = \sum_k \sum_{s'} \gamma^k T^\pi[s, s']^k 1[s'] = \sum_k \gamma^k = \frac{1}{1-\gamma}$. Therefore the *discounted state occupancy*, is defined $(1 - \gamma) \sum_s d_0[s] N_\gamma^\pi[s, s']$. The hitting probability of state s' from state s is given by $H_\gamma^\pi[s, s'] = \sum_{s''} (N_\gamma[s, s''] - I[s, s'']) \text{diag}(N_\gamma)[s'', s']^{-1}$.

$$\text{Fundamental:} \quad N_\gamma^\pi[s, s'] = \sum_{k=0}^{\infty} \gamma^k T^\pi[s, s']^k = (I[s, s'] - \gamma T^\pi[s, s'])^{-1} \quad (2.9a)$$

$$\text{Frequency:} \quad \lambda(d_0)_\gamma^\pi[s'] = \sum_k \sum_s \gamma^k d_0[s] T^\pi[s, s']^k = \sum_s d_0[s] N_\gamma^\pi[s, s'] \quad (2.9b)$$

$$\text{Termination Time:} \quad y[s] = \frac{1}{1 - \gamma} \quad (2.9c)$$

$$\text{Occupancy:} \quad d(d_0)_\gamma^\pi[s'] = (1 - \gamma) \sum_s d_0[s] N_\gamma^\pi[s, s'] \quad (2.9d)$$

$$\text{Hitting Prob:} \quad H_\gamma^\pi[s, s'] = \sum_{s''} (N_\gamma[s, s''] - I[s, s'']) \text{diag}(N_\gamma)[s'', s']^{-1} \quad (2.9e)$$

Irreducible Markov chains

Irreducible Markov chains include both periodic and aperiodic chains. Therefore ergodic chains are a subset of irreducible Markov chains. Some of the results of Ergodic chains can be generalised to irreducible chains. All irreducible Markov chains can be reversed $T[s', s] = \text{diag}(d)[s, s']^T T^\pi[s', s''] \text{diag}(d)[s'', s']$.

2.1.1.3 Environments

There are two main flavours of environments *terminating* (also called *episodic*) and *continuing*. Episodic environments have terminal states and are therefore *finite horizon*, and do not require discounting to be well defined. All terminating environments can be modelled using absorbing Markov chains. Continuing Markov chains can either be modelled with ergodic Markov chains, or discounted Markov chains.

Ergodic Continuing Environments

Define an ergodic continuing environment as an environment that induces a an ergodic Markov chain under some $\pi[s, a]$. It is sufficient to check if any positive policy $\pi[s, a] > 0 \forall (s, a)$ induces ergodic chain for this to be true. Continuing environments never terminate. This means they need to be simulated by either ergodic Markov chains or discounted Markov chains. Note that the process of converting an environment to a Markov chain (via a policy) can mean that it is no longer ergodic. This means that using discounting is most safe. A policy that is guaranteed to maintain ergodicity is a policy with all positive elements $\pi[s, a] > 0 \forall (s, a)$.

Periodic Continuing Environments

Define an periodic continuous environment as an environment that cannot induce an ergodic Markov chain under any $\pi[s, a]$. As a result, they can only be solved via discounted Markov chains.

Unbounded Terminating Environments

An example of a terminating environment is a Chess game. It is possible to win, lose, or draw the game to

terminate the episode, however it is also possible for the game to last infinite steps (one reason for a time limit on moves to be added in competitive Chess).

Bounded Terminating Environments

Termination is inevitable, but this could happen at several different timesteps depending on the path of chain taken.

Time Limited Terminating Environments

If there is a limit on the number of steps in an environment of each episode. This is a special case of a terminating environment. In this scenario to encode states in the MDP framework each state needs to specify the time (to ensure the Markov property). A corollary to this is that a state can never be revisited (because that would require going back in time). Absorbing Markov chains are best to model this type of environment.

2.1.1.4 Partially Observable Markov Decision Processes

So far only environments with perfect information have been considered: those where the player observes the exact state of the environment. For many environments, for example chess and Go this is a good model of the environment. However, for many problems, such as Poker and StarCraft, this is not a valid assumption to make. For many practical applications one does not observe the exact state of the world (*partial observation* in RL terminology or *imperfect information* in game theory terminology). Fortunately, we can model imperfect information through using partially observation Markov decision processes (POMDPs). Instead of agents directly receiving the state, $s \in \mathcal{S}$, of the game they now only receive observations, $o \in \mathcal{O}$. Unfortunately it is a much harder problem to solve. Although under some restrictions it is possible. Fortunately many of these difficulties can be side-stepped using model-free RL (Section 2.1.3.7).

History MDPs

We define a history of the observations, actions, and rewards seen so far. Rewards are included in the history because these could contain additional information on the underlying state.

$$h_t = (o_0, a_0, r_0, \dots, o_t) \sim H_t = (O_0, A_0, R_0, \dots, O_t) \quad (2.10)$$

By definition, the history satisfies the Markov property, therefore a policy can be made by conditioning on history, rather than the state. This is clearly less tractable than using Markovian states as the number of possible histories grows exponentially in the number of timesteps. This is sometimes referred to as the *curse of history* (Pineau et al., 2006).

$$|\mathcal{A}||\mathcal{R}| \sum_{t=0}^T |\mathcal{O}|^t = |\mathcal{A}||\mathcal{R}| \frac{|\mathcal{O}|^{T+1} - 1}{|\mathcal{O}| - 1} \quad (2.11)$$

For finite terminating environments, it is therefore possible to use model-based solvers on POMDPs after converting them to *history MDPs*. Non-terminating environments will result in infinite history MDPs and therefore cannot be solved via model-based methods. Model-free methods can be leveraged to solve infinite history MDPs with an additional trick to encode the infinite length observations (Section 2.1.3.7).

2.1.2 Value-Based Model-Based Reinforcement Learning

If the transition and reward probabilities of an environment are known (for any given state and action pair, the probability distribution over next states and rewards are known) the problem is known as model-based. It is worth emphasizing that it is not enough to have a simulation of the environment, we must be able to query for each state and action pair the probability distribution of future states and rewards.

2.1.2.1 Value Functions

The *state value function*, $v^\pi[s]$, is defined as the expected return, G_t , under policy π , starting from state s . The *action value function*, $q^\pi[s, a]$, is defined as the expected return, G_t , under policy π , starting from state s , after taking action a .

$$\text{State value function:} \quad v^\pi[s] = \mathbb{E}_\pi [G_t | S_t = s] \quad (2.12a)$$

$$\text{Action value function:} \quad q^\pi[s, a] = \mathbb{E}_\pi [G_t | S_t = s, A_t = a] \quad (2.12b)$$

One could also define these quantities in terms of hit frequencies, which are defined by the fundamental matrix. The value function is therefore defined in terms of which states are visited and how much reward those states give. This is sometimes called the dual formulation.

These quantities can be defined using hit frequencies, which are determined by the fundamental matrix. The value function thus quantifies the expected reward associated with visiting specific states. This is known as the dual formulation. To maintain consistency in magnitude across different discount factors (γ), value functions may be scaled by $(1 - \gamma)$, the reciprocal of the expected number of discounted steps. This scaling allows hit frequencies to represent occupancy probabilities.

$$\text{Dual State value function:} \quad v^\pi[s] = \sum_{s'} N_\gamma^\pi[s, s'] R^\pi[s, s'] \quad (2.13a)$$

$$\text{Dual Action value function:} \quad q^\pi[s, a] = \sum_{s'} \pi[s, a] N_\gamma^\pi[s, s'] R[s, a, s'] \quad (2.13b)$$

To identify the optimal policy, $\pi^*[s, a]$, we define two crucial quantities. The first, the action-value function, denoted $q^\pi[s, a]$, depends on the policy and quantifies the expected total return when action a is taken in state s , subsequently adhering to policy π . The second, the state-value function $v^\pi[s]$, represents the expected return when starting in state s and following policy π . Both functions are interrelated and can be defined in terms of each other, the transition dynamics of the environment, and the specific policy employed.

$$v^\pi[s] = \sum_a \pi[s, a] q^\pi[s, a] \quad (2.14a)$$

$$q^\pi[s, a] = \sum_{s'} T[s, a, s'] (R[s, a, s'] + \gamma v^\pi[s']) \quad (2.14b)$$

Action-value functions are associated with MDPs, while state-value functions pertain to Markov Reward Processes MRPs. Regarding notation, square brackets denote tabular lookup functions, analogous to indexing elements within tensors of corresponding dimensions. Subsequently, we will introduce round brackets to represent functions accepting real-valued inputs.

2.1.2.2 Bellman Operations

Recalling the definition of the return, G_t , we can express the value functions using recursive relationships. For instance, the value function is defined as follows:

$$\begin{aligned} v^\pi[s] &= \mathbb{E}_\pi [G_t | S_t = s] = \mathbb{E}_\pi [R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots | S_t = s] \\ &= \mathbb{E}_\pi [R_t + \gamma (R_{t+1} + \gamma R_{t+2} + \dots) | S_t = s] = \mathbb{E}_\pi [R_t + \gamma G_{t+1} | S_t = s] = \mathbb{E}_\pi [R_t + v^\pi[S_{t+1}] | S_t = s] \end{aligned}$$

Utilizing the Markov property and the MDP framework, we can define these quantities directly without

using expectations. These are sometimes referred to as the Bellman operators.

$$v^\pi[s] = \sum_a \pi[s, a] \sum_{s'} T[s, a, s'] (R[s, a, s'] + \gamma v^\pi[s']) \quad (2.15a)$$

$$q^\pi[s, a] = \sum_{s'} T[s, a, s'] \left(R[s, a, s'] + \gamma \sum_{a'} \pi[s', a'] q^\pi[s', a'] \right) \quad (2.15b)$$

Our objective is to identify an optimal policy, denoted as $\pi^*[s, a]$. This can be achieved by leveraging the optimal action-value function. While the optimal action-value function possesses a unique solution, the optimal policy itself may not be unique. For instance, when multiple actions share identical action values, one could arbitrarily select between them or distribute probability mass equally across all maximizing actions.

$$\pi^*[s, a] = \arg \max_a q^{\pi^*}[s, a] \quad (2.16)$$

Consequently, to determine an optimal policy, it suffices to find the optimal value functions, defined as $v^{\pi^*}[s] = \max_\pi v^\pi[s]$ or $q^{\pi^*}[s, a] = \max_\pi q^\pi[s, a]$. Notably, optimal Bellman operators can also be defined using an optimal policy.

$$v^{\pi^*}[s] = \max_a \sum_{s'} T[s, a, s'] \left(R[s, a, s'] + \gamma v^{\pi^*}[s'] \right) \quad (2.17a)$$

$$q^{\pi^*}[s, a] = \sum_{s'} T[s, a, s'] \left(R[s, a, s'] + \gamma \max_{a'} q^{\pi^*}[s', a'] \right) \quad (2.17b)$$

There exists an optimal policy, π^* , that is at least as good as all other policies $\pi^* \leq \pi \forall \pi$. All optimal policies achieve the optimal value function, $v^{\pi^*} = v^*$.

2.1.2.3 Exhaustive Tree Search

The most straightforward approach to determining the optimal value function within terminating environments involves an exhaustive tree search. This entails traversing the environment tree from each state to all its potential root states, a process achievable through a recursive algorithm (Algorithm 2.1)—sometimes referred to as a deep, full backup of the environment tree. Astute readers may recognize that initiating the calculation from the terminal states (leaf nodes) and progressing backward towards the initial states (root nodes) offers greater efficiency. This technique, known as *dynamic programming* or *backward induction*, forms the foundation for a broader class of iterative methods explored in the subsequent section. These methods, leveraging shallow backups of the environment tree, possess the distinct advantage of applicability even within non-terminating environments.

2.1.2.4 General Policy Iteration

The family of iterative algorithms for finding optimal value functions is called *general policy iteration* (GPI). All GPI algorithms revolve around two steps: *policy evaluation* and *policy improvement*. In a terminating environment, these techniques are also sometimes called *dynamic programming* or *backward induction* because it is only necessary to work backwards from the terminal states.

Policy Evaluation

Policy Evaluation is the process of determining the value function corresponding to a particular policy, π . This is achieved by solving the set of Bellman Equations which are a set of simultaneous equations. Therefore given a policy one can solve for the value function directly. Define $P^v[s, s'] = \sum_a \pi[s, a] \sum_{s'} T[s, a, s']$ and $b^v[s] = \sum_a \pi[s, a] \sum_{s'} T[s, a, s'] R[s, a, s']$ for the state value function and $P^q[s \otimes a, s' \otimes a'] = \sum_{s'} T[s \otimes a, s'] \sum_a \pi[s' \otimes a']$ and $b^q[s \otimes a] = \sum_{s'} T[s \otimes a, s'] R[s \otimes a, s']$ for the

Algorithm 2.1 Cached exhaustive depth-first environment tree search. Optimizations include caching optimal values found so far and only searching positive transition probability paths.

Require: A terminating undiscounted MDP with transitions T and expected reward R .

```

1: function OPTIMALSTATEVALUE( $v^*, s, T, R$ )
2:   if  $s \in v^*$  then                                     ▷ Check the cache.
3:     return  $v^*[s]$ 
4:   if  $s \in \mathcal{B}$  then                                       ▷ Equivalently:  $T[s, a, s] = 1 \ \forall a$ .
5:     return 0                                             ▷ Terminal states provide no further reward.
6:    $v^*[s] \leftarrow -\infty$ 
7:   for  $a \leftarrow \mathcal{A}$  do
8:      $v[s] \leftarrow 0$ 
9:     for  $s' \leftarrow \mathcal{S}'$  do
10:      if  $T[s, a, s'] > 0$  then
11:         $v[s] \leftarrow v[s] + T[s, a, s']R[s, a, s']$ 
12:         $v[s] \leftarrow v[s] + T[s, a, s'] \text{OPTIMALSTATEVALUE}(v^*, s', T, R)$ 
13:       $v^*[s] \leftarrow \max(v^*[s], v[s])$ 
14:   return  $v^*[s]$ 

```

action value function. The term $s \otimes a$ denotes the Cartesian product and can be interpreted as flattening the s and a dimensions into a single dimension of size $|\mathcal{S}||\mathcal{A}|$.

$$v^\pi[s] = (I[s, s'] - \gamma P^v[s, s'])^{-1} b^v[s] \quad (2.18)$$

$$q^\pi[s \otimes a] = (I[s \otimes a, s' \otimes a'] - \gamma P^q[s \otimes a, s' \otimes a'])^{-1} b^q[s \otimes a] \quad (2.19)$$

Notice that the $(\dots)^{-1}$ terms above correspond to the discounted fundamental matrix described in Section 2.1.1.2, and that the definitions of value looks similar to the mean number of steps before being absorbed scaled by the reward for being in a state. Note the fact that for transition functions $\sum_{s'} P^v[s, s'] = 1[s]$. Therefore when $0 < \gamma < 1$, the term $I[s, s'] - \gamma P^v[s, s']$ is diagonally dominant with spectral radius $\phi(I[s, s'] - \gamma P^v[s, s']) < 1$, and is full rank. The same argument can be made for the action value function. Therefore an inverse always exists for this problem. These properties mean that there are a number of iterative methods that also work. Define $A^v[s, s'] = I[s, s'] - \gamma P^v[s, s']$ and $A^q[s \otimes a, s' \otimes a'] = I[s \otimes a, s' \otimes a'] - \gamma P^q[s \otimes a, s' \otimes a']$. Let us now split the matrices into their upper triangular, diagonal, and lower triangular components: $A = U + D + L$. It is now possible to solve these equations using a number of iterative approaches for solving linear systems, $Ax = b$. These solutions follow the form of $x_{t+1} = M^{-1}((M - A)x_t + b)$.

$$\text{Richardson (Richardson, 1911): } M = \frac{1}{\omega}I \quad x_{t+1} = (I - \omega A)x_t + \omega b \quad (2.20a)$$

$$\text{Async Richardson: } M = \frac{1}{\omega}I + L \quad x_{t+1} = (I + \omega L)^{-1}(I + \omega L - \omega A)x_t + \omega b \quad (2.20b)$$

$$\text{Jacobi: } M = \frac{1}{\omega}D \quad x_{t+1} = D^{-1}((D - \omega A)x_t + \omega b) \quad (2.20c)$$

$$\text{Gauss-Seidel (Gauss, 1903): } M = \frac{1}{\omega}D + L \quad x_{t+1} = (D + \omega L)^{-1}((D + \omega L - \omega A)x_t + \omega b) \quad (2.20d)$$

Note that the Richardson method (Richardson, 1911) perfectly recovers the original Bellman equation and does not require performing matrix inversions. When used with a weight parameter it is sometimes

called modified Richardson Iteration. The optimal choice of ω is $\omega^* = \frac{2}{\lambda_{\max}(A) + \lambda_{\min}(A)}$. This is equivalent to simple Chebyshev iteration. The Jacobi method only requires a trivial diagonal inversion and converges faster. When used with a weighting parameter $\omega \neq 0$ it is called the damped Jacobi method. The Gauss-Seidel method requires inverting an lower triangular matrix and when used with a damping parameter is called Successive Over-Relaxation. An interpretation of how Richardson converges by observing that:

$$v = \gamma Av + b = \gamma A(\gamma Av + b) + b = \dots = \sum_{k=1}^{\infty} \gamma^k A^k b + b = (I - \gamma A)^{-1} b \quad (2.21)$$

In iterative methods, the updated value of each state depends on all possible predecessor states. Therefore iterative methods using the Bellman equation are usually known as *full-width* backups. Using *sample* backups is possible and this is what model-free approaches leverage (Section 2.1.3).

Policy Improvement

Policy Improvement is the process of improving the policy given an estimate of the action value function. For a policy to be improved it must be the case that it produces a payoff at least as good as the previous policy.

$$\text{Policy Improvement Condition: } \sum_a \pi_{t+1}[s, a] q^{\pi_t}[s, a] \geq \sum_a \pi_t[s, a] q^{\pi_t}[s, a] \quad \forall s \quad (2.22)$$

If the target policy is the optimal policy (sometimes also called greedy policy), this can be achieved using the maximizing policy, which trivially passes the above condition.

$$\pi[s, a] = \arg \max_a q^{\pi}[s, a] = \arg \max_a \sum_{s'} T[s, a, s'] (R[s, a, s'] + \gamma v^{\pi}[s']) \quad (2.23)$$

If more than one action is optimal we can pick between them arbitrarily. Common approaches are at random, the first action, or a uniform mixture.

$$\text{Rand Greedy: } \pi[s, a] = \begin{cases} 1, & \text{if } a = \text{rand}[\{a | q^{\pi}[s, a] = \max_{a'} q^{\pi}[s, a']\}] \\ 0, & \text{otherwise} \end{cases} \quad (2.24a)$$

$$\text{Maxent Greedy: } \pi[s, a] = \begin{cases} \frac{1}{|A^*|}, & \text{if } a \in A^* = \{a | q^{\pi}[s, a] = \max_{a'} q^{\pi}[s, a']\} \\ 0, & \text{otherwise} \end{cases} \quad (2.24b)$$

2.1.2.5 Policy Iteration

Policy iteration (Howard, 1960) involves a two-step process: a single policy evaluation step followed by a single policy improvement step. While the literature predominantly employs an unweighted ($\omega = 1$) Richardson update, Jacobi or Gauss-Seidel methods also remain viable alternatives. Although the state value function is typically favoured due to its lower memory requirements, the action value function can serve as a substitute. Finally, while the algorithm usually targets the optimal policy, other objectives can be pursued.

$$v_{t+1}^{\pi}[s] \leftarrow \sum_a \pi_t[s, a] \sum_{s'} T[s, a, s'] (R[s, a, s'] + \gamma v_t^{\pi}[s']) \quad (2.25a)$$

$$\pi_{t+1}[s, a] \leftarrow \arg \max_a \sum_{s'} T[s, a, s'] (R[s, a, s'] + \gamma v_{t+1}^{\pi}[s']) \quad (2.25b)$$

This algorithm does not require policy evaluation to converge before policy improvement. Notably,

it has no hyper-parameters to tune, such as a learning rate. The initial state value function, $v_0^*[s]$, and initial policy, $\pi_0[s, a]$, can be set arbitrarily, however $v_0^*[s] = \sum_a \pi_0[s, a] \sum_{s'} T[s, a, s'] R[s, a, s']$ and $\pi_0[s, a] = \frac{1}{|\mathcal{A}|}$ (or $v_0^*[s] = \max_a \sum_{s'} T[s, a, s'] R[s, a, s']$ and $\pi_0[s, a] = \arg \max_a \sum_{s'} T[s, a, s'] (R[s, a, s'] + \gamma v_0^*[s'])$) are sensible choices. Modified policy iteration (Nunen, 1976; Puterman and Shin, 1978) is a variant where the policy evaluation step is repeated several times rather than a single time. Async policy iteration (Williams and Baird, 1993) is equivalent to using an async Richardson update.

2.1.2.6 Value Iteration

The two steps of policy iteration can be combined using the optimal Bellman state value operator (Equation 2.17a) and the Richardson evaluation update. The resulting algorithm is known as value iteration (Bellman, 1957b; Shapley, 1953). It can be defined in terms of either state value functions or action-value functions.

$$v_{t+1}^*[s] \leftarrow \max_a \sum_{s'} T[s, a, s'] \left(R[s, a, s'] + \gamma v_t^*[s'] \right) \quad (2.26a)$$

$$q_{t+1}^*[s, a] \leftarrow \sum_{s'} T[s, a, s'] \left(R[s, a, s'] + \gamma \max_{a'} q_t^*[s', a'] \right) \quad (2.26b)$$

The iterative value iteration procedure allows for arbitrary initialization of $v_0^*[s']$. Common choices include $v_0^*[s'] = \max_a \sum_{s'} T[s, a, s'] R[s, a, s']$ or $v_0^*[s'] = 0$. Value iteration does not necessitate policy initialization or storage. Upon convergence, a policy can be derived using Random Greedy or Maximum Entropy Greedy (Section 2.1.2.4).

Analogous algorithms can be formulated with the optimal Bellman action-value operator (Equation 2.17b). However, the state value function is typically preferred due to its lower space complexity. Furthermore, any iterative policy evaluation technique from Section 2.1.2.4 is applicable.

Value iteration offers flexibility in state update order. Asynchronous updates (Equation 2.20b) result in an algorithm known as *Asynchronous value iteration* (Williams and Baird, 1993). In-place updates using a single value copy (v_{t+1} reusing v_t 's memory) are another common optimization. Asynchronous updates can be further extended by prioritizing states with the highest Bellman error (*prioritized sweeping* (Moore and Atkeson, 1993)). Alternatively, states visited by the agent's policy can be preferentially updated (*real-time dynamic programming* (Barto et al., 1995)).

To establish convergence, we demonstrate that the Bellman operator is a contraction. In other words, the distance between any two value estimates strictly decreases after applying the operator: $\|v_{k+1}^0 - v_{k+1}^1\|_\infty < \|v_k^0 - v_k^1\|_\infty$. Here, $\|\cdot\|_\infty$ represents the max norm, defined as $\|\cdot\|_\infty = \max_i |\cdot|$.

$$\begin{aligned} \|v_{k+1}^0 - v_{k+1}^1\|_\infty &= \left\| \max_{a^0} \left(\sum_{s'} T[s, a^0, s'] \gamma v_k^0 \right) - \max_{a^1} \left(\sum_{s'} T[s, a^1, s'] \gamma v_k^1 \right) \right\|_\infty \\ &\leq \left\| \max_a \sum_{s'} T[s, a, s'] (\gamma v_k^0 - \gamma v_k^1) \right\|_\infty = \max_s \left| \max_a \sum_{s'} T[s, a, s'] (\gamma v_k^0 - \gamma v_k^1) \right| \\ &\leq \max_{s,a} \left| \sum_{s'} T[s, a, s'] (\gamma v_k^0 - \gamma v_k^1) \right| \leq \max_{s,a} \gamma |v_k^0 - v_k^1| = \gamma \|v_k^0 - v_k^1\|_\infty \end{aligned}$$

Therefore for $\gamma < 1$, value iteration converges linearly with γ . A similar proof could be constructed with action values. Note that while the value function may take infinite time to fully converge, the greedy policy may be optimal much sooner. There are a number of stopping criterion including based on the Bellman residual (Williams et al., 1993), and the span semi-norm (Puterman, 2014).

This problem can be formulated as a linear program (LP) using the fact that $\max_i x_i$ is equivalent to

$\min y$ s.t. $y \geq x_i \forall i$. In operations research y is usually called a slack variable. LPs can be solved efficiently using a number of methods (Murty, 1983). $d_0[s] > 0$ is an initial distribution of states.

$$\min \sum_s d_0[s] v^{\pi^*}[s] \quad \text{s.t.} \quad v^{\pi^*}[s] \geq \sum_{s'} T[s, a, s'] \left(R[s, a, s'] + \gamma v^{\pi^*}[s'] \right) \quad \forall s \in \mathcal{S}, a \in \mathcal{A} \quad (2.27)$$

$$\min \sum_{s,a} d_0[s] q^{\pi^*}[s, a] \quad \text{s.t.} \quad q^{\pi^*}[s, a] \geq \sum_{s'} T[s, a, s'] \left(R[s, a, s'] + \gamma q^{\pi^*}[s', a'] \right) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}, a' \in \mathcal{A} \quad (2.28)$$

There is also a dual formulation of this problem.

$$\begin{aligned} \max_{\lambda} \quad & \sum_{s,a,s'} \lambda[s, a] T[s, a, s'] R[s, a, s'] \\ \text{s.t.} \quad & \sum_{a'} \lambda[s', a'] = d_0[s'] + \gamma \sum_{s,a} \lambda[s, a] T[s, a, s'] \quad \forall s' \end{aligned}$$

Examining the equality constraint for a particular actions, a , results in the $\lambda_a[s] = (I[s, s'] - \gamma T_a[s, s'])^{-1} d_0[s'] = \sum_{k=0}^{\infty} \sum_{s'} \gamma^k T_a[s, s']^k d_0[s']$. Therefore the dual variables can be interpreted as a cumulative discounted probability of being in a state.

$$\begin{aligned} \sum_{a'} \lambda[s', a'] &= d_0[s] + \gamma \sum_s \sum_a \lambda[s, a] T[s, a, s'] \\ &= d_0[s] + \gamma \sum_s \left(d_0[s] + \gamma \sum_s \sum_a \lambda[s, a] T[s, a, s'] \right) T[s, a, s'] \\ &= d_0[s] + \gamma \sum_s \left(d_0[s] + \gamma \sum_s \left(\mu_0[s] + \gamma \sum_s \sum_a \lambda[s, a] T[s, a, s'] \right) T[s, a, s'] \right) T[s, a, s'] \end{aligned}$$

2.1.3 Value-Based Model-Free Reinforcement Learning

Model-free reinforcement learning offers two key advantages over model-based RL. First, it is applicable when a model of the environment is unavailable. Second, it can handle environments with a large or even infinite number of states. In model-free RL, only samples from the environment are accessible. Despite this limitation, value-based methods can still be employed to determine the optimal policy. The key is to update estimates of action values using these samples, which are random variables with unknown distributions. Moreover, model-free approaches excel in partially observable environments by leveraging function approximation and recurrent architectures (see Section 2.1.3.7). This ability to address partial observability is a significant advantage.

2.1.3.1 Value-Based Model-Free Objective

In value-based model-free temporal difference learning, the objective is to estimate either the action-value or state-value function based on samples from the environment. A common approach is to minimize the squared error between the estimated and actual values.

$$J^{v^{\pi}} = \frac{1}{2} \sum_s (\hat{v}(s) - v(s))^2 \qquad J^{q^{\pi}} = \frac{1}{2} \sum_{s,a} (\hat{q}(s, a) - q(s, a))^2$$

$$\begin{aligned}\nabla_{v(s)} J^{v^\pi} &= (v(s) - \hat{v}(s)) & \nabla_{q(s,a)} J^{q^\pi} &= (q(s,a) - \hat{q}(s,a)) \\ v(s) &\leftarrow v(s) + \mu (\hat{v}(s) - v(s)) & q(s,a) &\leftarrow q(s,a) + \mu (\hat{q}(s,a) - q(s,a))\end{aligned}$$

It is unusual to know an exact targets $\hat{v}(s)$ and $\hat{q}(s,a)$, so instead they are defined in terms of sampled returns G , where $\hat{v}(s) = \mathbb{E}_{a_t \sim \pi(s_t), s_{t+1} \sim T(s_t, a_t)}[G]$ and $\hat{q}(s,a) = \mathbb{E}_{a_t = a, a_{t+1} \sim \pi(s_t), s_{t+1} \sim T(s_t, a_t)}[G]$. Returns can be estimated in a variety of ways, which can be classified across two dimensions. First, the horizon of sample rewards to build an estimate: an episode, a 1-step transition, or an n -step trajectory. Second, how the horizon is truncated: using a state-value, action-value function, or expected action-value.

$$\text{Episode} \left\{ \begin{array}{ll} \text{Sample R:} & G_{t:T} := \sum_{k=1}^{T-t} \gamma^{k-1} R_{t+k} \end{array} \right. \quad (2.29a)$$

$$\text{Transition} \left\{ \begin{array}{ll} \text{Sample V:} & G_{t:t+1}^v := R_{t+1} + \gamma v^\pi(S_{t+1}) \\ \text{Sample Q:} & G_{t:t+1}^q := R_{t+1} + \gamma q^\pi(S_{t+1}, A_{t+1}) \\ \text{Expected Q:} & G_{t:t+1}^{\pi q} := R_{t+1} + \gamma \sum_a \pi(S_{t+1}, a) q^\pi(S_{t+1}, a) \end{array} \right. \quad (2.29b)$$

$$\text{Trajectory} \left\{ \begin{array}{ll} \text{Sample V:} & G_{t:t+n}^v := \sum_{k=1}^n \gamma^{k-1} R_{t+k} + \gamma^n v^\pi(S_{t+n}) \\ \text{Sample Q:} & G_{t:t+n}^q := \sum_{k=1}^n \gamma^{k-1} R_{t+k} + \gamma^n q^\pi(S_{t+n}, A_{t+n}) \\ \text{Expected Q:} & G_{t:t+n}^{\pi q} := \sum_{k=1}^n \gamma^{k-1} R_{t+k} + \gamma^n \sum_a \pi(S_{t+n}, a) q^\pi(S_{t+n}, a) \end{array} \right. \quad (2.29c)$$

2.1.3.2 Bootstrapping and Bias-Variance Trade-Off

To estimate the value of a state $v(s)$ or an action $q(s,a)$, previously estimated values of these quantities are utilized. This process, known as *bootstrapping*, involves using prior estimates to derive new estimates. While the episode return provides an unbiased estimate of the value function, it suffers from high variance. In contrast, bootstrapped temporal difference targets are biased estimates due to their reliance on approximate value functions learned during training. However, they exhibit significantly lower variance. By using trajectories, it is possible to adjust the trade-off between bias and variance. Trajectories allow for a balance between the unbiased but high-variance episode return and the biased but low-variance temporal difference targets.

2.1.3.3 Eligibility Traces

It is possible to use any weighted average of n -step returns from 1-step to ∞ -step returns. A common way of parameterizing this mixture is via the λ mixture:

$$\text{Continuing:} \quad {}^\lambda G_{t:\infty} := (1 - \lambda) \sum_{n=1}^{\infty} \lambda^{n-1} G_{t:t+n} \quad (2.30a)$$

$$\text{Terminating:} \quad {}^\lambda G_{t:T} := (1 - \lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} G_{t:t+n} + \lambda^{T-t-1} G_{t:T} \quad (2.30b)$$

Note that for terminating environments all returns $t+n > T$ are equal to $G_{t:T}$, therefore we can write the terminating λ -return as a finite sum. The weighting always has unit sum $(1 - \lambda) \sum_{n=1}^{\infty} \lambda^{n-1} = 1$. When $\lambda = 1$, the λ -return is equal to the episode return, and when $\lambda = 0$, the λ -return is equal to the transition return. Therefore eligibility traces allow one to mix between the two backup extremes. This interpretation of eligibility traces is sometimes known as the *forward view*.

Notice that computing the λ -return requires either running for an infinite number of timesteps for continuing environments or until termination for terminating environments. This can be unsatisfying to implement computationally. Fortunately there is an alternative but equivalent *backward view* that implements the same backups. It is implemented by maintaining an additional memory component for each state called

an *accumulating eligibility trace*.

$$\text{V-Eligibility Trace:} \quad e_t[s] = \begin{cases} \gamma \lambda e_{t-1}[s] & \text{if } s \neq s_t \\ \gamma \lambda e_{t-1}[s] + 1 & \text{if } s = s_t \end{cases} \quad (2.31)$$

$$\text{Q-Eligibility Trace:} \quad e_t[s, a] = \begin{cases} \gamma \lambda e_{t-1}[s, a] & \text{if } s \neq s_t \text{ and } a = a_t \\ \gamma \lambda e_{t-1}[s, a] + 1 & \text{otherwise} \end{cases} \quad (2.32)$$

It is then possible to modify the updates with an appropriate eligibility trace. Note that eligibility traces are inherently tabular which limits their application to large scale RL where states spaces are large or continuous.

$$\begin{aligned} v[s] &\leftarrow v[s] + \mu \delta_t e_t[s] & \forall s \in \mathcal{S} \\ q[s, a] &\leftarrow q[s, a] + \mu \delta_t e_t[s, a] & \forall s \in \mathcal{S}, a \in \mathcal{A} \end{aligned}$$

2.1.3.4 Off-Policy Learning

The concepts discussed thus far focus on *on-policy* learning, where the policy being learned is identical to the one being followed. However, when using action-values, it is also feasible to learn a target policy $\hat{\pi}$ that differs from the policy being followed π (*off-policy* learning). For off-policy trajectories, it is crucial to ensure that only the initial action ($k = 1$) is taken off-policy. If any subsequent actions are off-policy, the trajectory must be truncated immediately. However, if the return is truncated using the expected action-value under the target policy, it is permissible to utilize the final transition. To handle off-policy learning, it is necessary to define off-policy returns, which account for the difference between the target and followed policies.

$$\text{Episode} \left\{ \begin{array}{ll} \text{Sample R:} & G_{t:T}^{\hat{\pi}} := R_{t+1}^{\pi} + \sum_{k=2}^{T-t} \gamma^{k-1} R_{t+k}^{\hat{\pi}} \end{array} \right. \quad (2.33a)$$

$$\text{Transition} \left\{ \begin{array}{ll} \text{Sample Q:} & G_{t:t+1}^{q^{\hat{\pi}}} := R_{t+1}^{\pi} + \gamma q^{\hat{\pi}}(S_{t+1}, A_{t+1}^{\hat{\pi}}) \\ \text{Expected Q:} & G_{t:t+1}^{\hat{\pi} q^{\hat{\pi}}} := R_{t+1}^{\pi} + \gamma \sum_a \hat{\pi}(S_{t+1}, a) q^{\hat{\pi}}(S_{t+1}, a) \end{array} \right. \quad (2.33b)$$

$$\text{Trajectory} \left\{ \begin{array}{ll} \text{Sample Q:} & G_{t:t+n}^{q^{\hat{\pi}}} := R_{t+1}^{\pi} + \sum_{k=2}^n \gamma^{k-1} R_{t+k}^{\hat{\pi}} + \gamma^n q^{\hat{\pi}}(S_{t+n}, A_{t+n}^{\hat{\pi}}) \\ \text{Expected Q:} & G_{t:t+n}^{\hat{\pi} q^{\hat{\pi}}} := R_{t+1}^{\pi} + \sum_{k=2}^n \gamma^{k-1} R_{t+k}^{\hat{\pi}} + \gamma^n \sum_a \hat{\pi}(S_{t+n}, a) q^{\hat{\pi}}(S_{t+n}, a) \end{array} \right. \quad (2.33c)$$

These returns can converge to any target policy as long as the behaviour policy, $\pi(s, a) > 0 \forall s \in \mathcal{S}, a \in \mathcal{A}$. Commonly, the target policy is the optimal policy, π^* , and the behaviour policy is commonly ϵ -greedy, $\pi^{\epsilon*}$.

$$\begin{aligned} \pi^*(S_{t+n}, a) &= \begin{cases} 1 & \text{if } a = \arg \max_a q^{\hat{\pi}}(S_{t+n}, a) \\ 0 & \text{otherwise} \end{cases} \\ \pi^{\epsilon*}(S_{t+n}, a) &= \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}|} & \text{if } a = \arg \max_a q^{\hat{\pi}}(S_{t+n}, a) \\ \frac{\epsilon}{|\mathcal{A}|} & \text{otherwise} \end{cases} \end{aligned}$$

2.1.3.5 Algorithms

To approximate the value functions, the problem is formulated as an optimization task. The objective is to minimize a *temporal difference*, expressed as $L = \sum_s \sum_a \frac{1}{2} (\hat{q}(s, a) - q(s, a))^2$. Where $q(s, a)$ represents

the current estimate of the action-value function and $\hat{q}(s, a)$ denotes the temporal difference target, which is an updated estimate based on observed rewards and transitions.

The most basic on-policy algorithm is Monte Carlo which uses the sampled episodic return, $G_{t:T}$, as the target. SARSA (Rummery and Niranjan, 1994) uses a transition containing two sampled actions, and truncates the return estimate using the action-value of the final sampled action, $G_{t:t+1}^q$. n -step SARSA uses a trajectory of transitions. When combined with eligibility traces, it is known as SARSA(λ). It is possible to instead consider a transition with only a single sampled action and truncate on the expected action-value return under the policy, $G_{t:t+1}^{\pi q}$, resulting in the SARSE (van Seijen et al., 2009) algorithm. Again, there are n -step and eligibility trace versions of this algorithm. The policy used during training can be updated over time, and is often a function of the action-values themselves. If performing policy improvement, it is important that all actions have positive probability of being taken so that action values can be estimated accurately (or action-values could be initialized optimistically). A popular choice is to use ϵ -greedy policy, where ϵ is decayed to zero over time.

Off-policy algorithms are more complex. Consider the sampled episodic return, this will only be a valid target for a state if all subsequent actions are taken according to the target policy. Depending on how different the behaviour policy is from the target policy this could become vanishingly unlikely, so Off-Policy Monte Carlo is not a popular algorithm to employ. Similarly, Off-Policy SARSA requires the final action in the transition to be on-policy to learn from it, meaning many examples have to be dropped. Off-Policy SARSE, on the other hand does not require transitions to be dropped: the first action can be any and the value estimate is truncated using the target policy. When the target policy is equal to the greedy policy, Off-Policy SARSE is known as Q-Learning. There are two flavours of Q-Learning with eligibility traces, Watkin’s Q(λ) (Watkins, 1989) and Peng’s Q(λ) (Peng and Williams, 1996).

2.1.3.6 Scaling and Function Approximation

When the number of states is large or their representations exhibit structure that can benefit from generalization, function approximation becomes a valuable tool for estimating value functions.

There are a number of function approximators in the literature, including; linear features (Sutton and Barto, 2018), neural networks, decision trees, nearest neighbour. A crucial property for function approximators in reinforcement learning is differentiability. Deep neural networks are a popular choice due to this property.

Typically, these networks receive a state as input and output a vector of real values (either of cardinality $|\mathcal{A}|$ for action-value functions or a single value for state-value functions). The network’s weights, θ , parameterize these functions. This parameterization is advantageous as it allows for optimization over actions. However, it restricts the network to categorical action spaces. It is worth noting that networks capable of handling continuous actions can also be defined (Table 2.2).

2.1.3.7 Partial Observations and Recurrent Neural Networks

Model-free methods can be applied to partially observed environments by leveraging recurrent neural networks. It was shown in Section 2.1.1.4 that POMDPs can be converted to history MDPs if we use the history of observations, actions, and rewards. This was intractable because there are either too many histories to enumerate (terminating environments) or there are infinite histories (non-terminating environment). Function approximation, using recurrent architectures, provides a tractable way of encoding an infinite number of histories.

In general, this works by adding connections from the hidden activations of the previous time step to the hidden activations of the current time step. This means that the policy at a particular timestep can “see” the entire history of observations through these connections. Equivalently, when training the neural network, this allows gradients to flow through time back to all previous observations. This allows the

Algorithm	TD Target	Target Policy $\hat{\pi}$	Behaviour Policy π
Monte Carlo	$G_{t:T}$	π	π
SARSA (Rummery and Niranjan, 1994)	$G_{t:t+1}^q$	π	π
n -step SARSA	$G_{t:t+n}^q$	π	π
SARSA(λ)	$\lambda G_{t:\infty}^q$	π	π
SARSE (van Seijen et al., 2009)	$G_{t:t+1}^{\pi q}$	π	π
n -step SARSE	$G_{t:t+n}^{\pi q}$	π	π
SARSE(λ)	$\lambda G_{t:\infty}^{\pi q}$	π	π
Off-Policy Monte Carlo	$G_{t:T}^{\hat{\pi}}$	$\hat{\pi}$	π
Off-Policy SARSA	$G_{t:t+1}^{q^{\hat{\pi}}}$	$\hat{\pi}$	π
Off-Policy n -step SARSA	$G_{t:t+n}^{q^{\hat{\pi}}}$	$\hat{\pi}$	π
Off-Policy SARSA(λ)	$\lambda G_{t:\infty}^{q^{\hat{\pi}}}$	$\hat{\pi}$	π
Off-Policy SARSE	$G_{t:t+1}^{\hat{\pi} q}$	$\hat{\pi}$	π
Off-Policy n -step SARSE	$G_{t:t+n}^{\hat{\pi} q}$	$\hat{\pi}$	π
Off-Policy SARSE(λ)	$\lambda G_{t:\infty}^{\hat{\pi} q}$	$\hat{\pi}$	π
Q-Learning (Watkins, 1989)	$G_{t:t+1}^{\hat{\pi} q^{\hat{\pi}}}$	π^*	$\pi > 0$
n -step Q-Learning	$G_{t:t+n}^{\hat{\pi} q^{\hat{\pi}}}$	π^*	$\pi > 0$
Watkin's Q(λ) (Watkins, 1989)	$R_t + \gamma \max_a q^{\pi^*}(S_{t+1}, a)$	π^*	$\pi > 0$
Peng's Q(λ) (Peng and Williams, 1996)	$R_t + \gamma \max_a q^{\pi^*}(S_{t+1}, a)$	π, π^*	$\pi > 0$
Double Q-Learning (van Hasselt et al., 2015)	$R_t + \gamma q^{\pi^A}(S_{t+1}, \max_a q^{\pi^B}(S_{t+1}, a))$	π_A^*	$\pi_A > 0$
	$R_t + \gamma q^{\pi^B}(S_{t+1}, \max_a q^{\pi^A}(S_{t+1}, a))$	π_B^*	$\pi_B > 0$

Table 2.1: Temporal difference algorithms.

	Categorical Actions $a \in \mathcal{A}$		Continuous Actions $a \in \mathbb{R}^{ \mathcal{A} }$	
Categorical States $s \in \mathcal{S}$	$q^\pi[s, a]$	$v^\pi[s]$	$q_\theta^\pi[s, a]$	$v^\pi[s]$
Continuous States $s \in \mathbb{R}^O$	$q_\theta^\pi(s, a)$	$v^\pi(s)$	$q_\theta^\pi(s, a)$	$v^\pi(s)$

Table 2.2: Value function taxonomy with parameterization and their notation. Here O is the observation size. The square brackets indicates that the term is an output, while curved brackets indicate that the term is an input. Note that “continuous” is not a requirement it simply indicates that continuous space is supported. The most common architectures are either *tabular*, $q^\pi[s, a]$, or with state as an input, $q_\theta^\pi(s, a)$.

network to learn an efficient encoding of the history, h_t , within the hidden activations of the recurrent neural network (presumably only keeping information necessary to maximize return). Remember that the history contains not just the observations, but also the actions and rewards. Therefore the reward and action sampled by the policy should be made available to the recurrent part of the network too.

The intuition at play here is that recurrent networks, and adding additional information such as previous sampled action and observed reward, make the (non-Markovian) partially-observed environment appear more Markovian. There are other ways of achieving the same goal, for example in the Atari environment some frames are repeated several timesteps in a row. This makes a frame partially observable because there is no way of knowing from the frame alone whether this is the first or second time the agent has observed this frame. Rather than use a recurrent architecture, the DQN algorithm stacked the previous four frames (Mnih et al., 2015) resulting in Markov state observations.

Recurrent architectures are more difficult to train than traditional feedforward architectures because passing gradients back through many timesteps can result in gradients either getting watered down to zero,

or expanding due to a positive feedback loop. This phenomenon is known as the *vanishing and exploding gradients problem* (Hochreiter, 1991; Hochreiter et al., 2001).

This problem can be mitigated by a) using certain activation functions, b) careful initialization of the network parameters, c) clipping gradients before updating network parameters, and most importantly d) using special recurrent architectures. Of the architectures used for recurrent networks the most famous is the *long short-term memory* (LSTM) architecture (Hochreiter and Schmidhuber, 1997), which works by ensuring that gradients are summed between timesteps (rather than multiplied). There have been a number of refinements of LSTMs, including peepholes (Gers and Schmidhuber, 2000) and dropout (Zaremba et al., 2014). Other recurrent architectures of interest include gated recurrent units (GRUs) (Cho et al., 2014).

2.1.3.8 Training in Practice

There are several tricks that can be used to successfully train agents using function approximation in practice. Firstly, when training many parameters, careful renormalization of gradients and adaptive optimizers (Kingma and Ba, 2014; Tieleman et al., 2012) should be utilized. Secondly, target and behaviour networks should be used to ensure stability. Thirdly, replay buffers (Mnih et al., 2015; Schaul et al., 2016) can be utilized for efficiency. Fourthly, auxiliary tasks can be used accelerate learning representations at the beginning of training (Jaderberg et al., 2016). Finally, hyper-parameter optimization is useful for searching over the large resulting model space.

2.1.4 Policy-Based Model-Free Reinforcement Learning

Explicitly learning and storing action-values for every state and action can be computationally demanding. As an alternative, *policy gradient* methods directly optimize the policy without explicitly computing action-values. These on-policy algorithms aim to directly learn the optimal policy, resulting in a significant reduction in computational overhead (Sutton et al., 1999).

2.1.4.1 Policy Gradient Theorem

In the infinite horizon case, the objective function we wish to maximize is the value V^{π_θ} , under the stationary distribution, $d_\infty^{\pi_\theta}(s)$, under the policy π_θ . For the episodic case, without loss of generality, we can consider only the value of the initial states under initial distribution $d_0(s)$. This simplifies analysis because it is necessary to differentiate through $d_\infty^{\pi_\theta}(s)$, a term involving the policy.

$$\text{Continuing:} \quad J^{\pi_\theta} = \sum_{s \in S} d_\infty^{\pi_\theta}(s) V^{\pi_\theta}(s) = \sum_{s \in S} d_\infty^{\pi_\theta}(s) \sum_{a \in A} \pi_\theta(a|s) Q^{\pi_\theta}(s, a) \quad (2.34a)$$

$$\text{Episodic:} \quad J^0 = \sum_{s \in S} d_0(s) V^{\pi_\theta}(s) = \sum_{s \in S} d_0(s) \sum_{a \in A} \pi_\theta(a|s) Q^{\pi_\theta}(s, a) \quad (2.34b)$$

The policy gradient theorem (Sutton and Barto, 2018; Sutton et al., 1999) states that the gradient of this loss function can be computed without knowing the dynamics of the environment¹.

$$\nabla_\theta J(\theta) = \nabla_\theta \sum_{s \in S} d^{\pi_\theta}(s) \sum_{a \in A} Q^{\pi_\theta}(s, a) \pi_\theta(a|s) \quad (2.35a)$$

$$\propto \sum_{s \in S} d^{\pi_\theta}(s) \sum_{a \in A} Q^{\pi_\theta}(s, a) \nabla_\theta \pi_\theta(a|s) \quad (2.35b)$$

$$= \sum_{s \in S} d^{\pi_\theta}(s) \sum_{a \in A} \pi_\theta(a|s) Q^{\pi_\theta}(s, a) \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)} \quad (2.35c)$$

¹Note the calculus identity $\frac{\nabla_x f(x)}{f(x)} = \nabla_x \ln f(x)$.

$$= \mathbb{E}_{s,a \sim \tau^\pi} \left[Q^{\pi_\theta}(s, a) \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)} \right] \quad (2.35d)$$

$$= \mathbb{E}_{s,a \sim \tau^\pi} [Q^{\pi_\theta}(s, a) \nabla_\theta \ln \pi_\theta(a|s)] \quad (2.35e)$$

This reformulation is useful because it enables updates on θ without calculating the derivative of the state distribution, nor calculating the stationary state distribution.

2.1.4.2 Baseline

It is common to modify this update to reduce the variance of the estimator by subtracting a *baseline*, $b(s)$ (Williams, 1992).

$$\nabla_\theta J(\theta) \propto \sum_{s \in S} d^{\pi_\theta}(s) \sum_{a \in A} (Q^{\pi_\theta}(s, a) - b(s)) \nabla_\theta \pi_\theta(a|s) \quad (2.36)$$

Often it is common to use the value function as a baseline. This results in an update using the so-called *advantage function*, $A(s, a) = Q(s, a) - v(s)$. However, the baseline can be any function that does not depend on the action a . This is valid because the derivative of the baseline term is zero.

$$\sum_{a \in A} b(s) \nabla_\theta \pi_\theta(a|s) = b(s) \nabla_\theta \sum_{a \in A} \pi_\theta(a|s) = b(s) \nabla_\theta 1 = 0 \quad (2.37)$$

2.1.4.3 Function Approximation

It is common to use function approximation to represent both the policy and any value functions that are required for policy updates. Sutton's *compatible function approximation theorem* (Sutton et al., 1999) states that if the following conditions are satisfied, the policy gradient is exact. These conditions imply that the policy has to be stochastic.

1. The function approximator is compatible to the policy $\nabla_w q_w(s, a) = \nabla_\theta \ln \pi_\theta(a|s)$.
2. The value function minimizes the mean squared error $\epsilon = \mathbb{E}_{\pi_\theta} [(q_\pi(s, a) - q_w(s, a))^2]$.

2.1.4.4 Off Policy Correction

When using a replay buffer or a distributed setting, the training data is slightly off-policy. It is still possible to use on-policy learning algorithms, but the policy drift must be accounted for, usually by importance sampling or trust region techniques. Re-Trace (Munos et al., 2016) requires learning action value functions but does not exactly reduce to the on-policy update resulting in a bias. V-Trace (Espeholt et al., 2018) uses state value function and reduces to bias-free on-policy updates with on-policy experience.

2.1.4.5 Policy Gradient Algorithms

Some policy gradient algorithms are summarized in Table 2.3. Using policy gradient with the sampled episode return is called REINFORCE (Williams, 1992) (or *Monte Carlo policy gradient*). Using policy gradient with a value function baseline, and any other estimate of the action-value function is known as actor-critic (Konda and Tsitsiklis, 1999). When used with a value function base, it is known as advantage actor critic (A2C) (Mnih et al., 2016). This algorithm can be deployed in a parallel, with asynchronous update, in which case it is known as asynchronous advantage actor-critic (A3C) (Mnih et al., 2016). Parallel execution introduces non-determinism and therefore the policy updates may be slightly off-policy. Furthermore, in order to maximize learner utilization it is possible to use a small replay-buffer. In order to correct for this, off-policy correction was added to some algorithms. Using an action-value based correction results in ACER (Wang et al., 2016). Using a state-value based correction results in IMPALA (Espeholt et al., 2018). Most algorithms also have an entropy maximization in the policy to encourage exploration (Mnih et al., 2016; Williams and Peng, 1991). It is advantageous for policy and value networks to not share pa-

rameters to reduce correlation. Other policy gradient algorithms include TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017).

Algorithm		Action-Value	Baseline	Correction
REINFORCE	(Williams, 1992)	$G_{t:T}$	0	
Advantage REINFORCE	(Williams, 1992)	$G_{t:T}$	$v(s)$	
Actor-Critic	(Konda and Tsitsiklis, 1999)	$Q(s, a)$	0	
Advantage Actor-Critic (A2C)	(Mnih et al., 2016)	$Q(s, a)$	$v(s)$	
ACER	(Wang et al., 2016)	$Q(s, a)$	$v(s)$	Retrace
IMPALA	(Espeholt et al., 2018)	$Q(s, a)$	$v(s)$	V-Trace

Table 2.3: Policy Gradient Algorithms.

2.2 Game Theory and Multiagent Reinforcement Learning

Game theory is the study of the behaviour of rational payoff maximizing agents, in the presence of other agents. There are many ways for predicting how agents behave; the different methods of prediction are referred to as solution concepts. The world that agents operate in is often called a game. Games can be formulated in several ways, and the formulation dictates suitable solution concepts. These formulations often concern how many temporal steps there are in the game (normal-form game when there is only one step) and whether agents take actions simultaneously (Markov games) or sequentially (extensive-form games).

Multiagent reinforcement learning (MARL) is the broad field of applying machine learning techniques to multiagent problems. It often utilizes reinforcement learning (RL), function approximation, and sometimes supervised learning. It is most applicable to many-player large games with intractable numbers of states. Therefore MARL should be thought of as not just multiagent with RL, but multiagent in combination with many ML techniques which are used to scale to large problems. Game theory (GT) is an essential component, especially when considering environments with more than two players or general-sum payoffs. An excellent discussion on the overlap of RL and GT is given by Shoham et al. (2007).

2.2.1 Multiagent RL Problem

Different formulations of the MARL problem can be found in Figure 2.2. The most simple formulation is where every agent is independent and learns at once (Figure 2.2a). While the most general, it can be difficult to reason about the dynamics of learning agents in such a formulation. Moreover, there are many possible deterministic opponent policies, infinite stochastic opponent policies, and therefore there are an infinite number of induced environments. One simplification is to subsume all but one agent into the environment (Figure 2.2b), and freeze the other agents' adaptation. In this formulation, a single agent trains against a fixed set of opponents resulting in a stationary single-agent learning environment, where the tools for RL are well developed. A final formulation is to combine all the agents into a single centralized agent (2.2c). In this formulation, joint actions are trained directly, to change the learning problem into a single-agent problem. In this formulation, there is often a mechanism to allow agents to act independently at test time, so that multi-agency is maintained. Many of the algorithms discussed in this section assume one of these formulations during training.

2.2.1.1 Goals of Game Theory

Game theory is most developed in a subset of games: those with two players and a restriction on the payoffs, $G_1(a_1, a_2) = -G_2(a_1, a_2)$, known as zero-sum, which corresponds to purely competitive games. In this setting the objective is clear: maximize your payoff in the presence of another player who is also maximizing their payoff. Because one player's gain is the other player's loss there is a certain symmetry

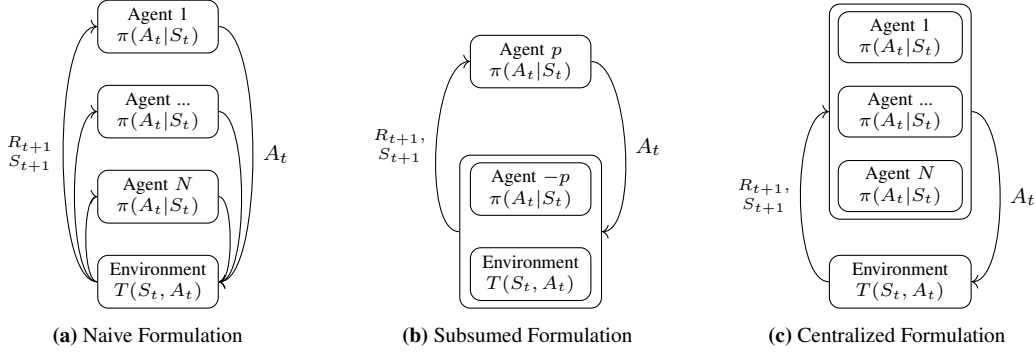


Figure 2.2: MARL formulations.

about this class of games. It turns out that there is a single metric, distance to a Nash equilibrium, that players should minimize. Being close to an NE means that a player is unexploitable: there is nothing an opponent can do to reduce your payoff. Particularly in n -player, general-sum games, it is difficult to define a single criterion to find solutions to games. There are numerous reasons why this is the case:

1. There may be opportunities to collaborate.
2. Choice of which players to collaborate with.
3. To what extent should a player collaborate or exploit.
4. How does one enforce or incentivize collaboration.
5. How to coordinate on behaviour.

One approach is to instead consider joints that are in equilibrium: distributions such that no player has incentive to unilaterally deviate from a recommendation. NE is one such equilibrium, but this thesis also focuses on other concepts, namely correlated equilibrium and coarse correlated equilibrium.

2.2.2 Normal-Form Games

It is useful to model multiagent environments as games. There are a number of possible formalisms including normal-form, extensive-form, and stochastic games. A particular type of single-shot, simultaneous move game is called a *normal-form* game (von Neumann, 1928). Also called strategic-form, a normal-form game consists of N players, $p \in [1, N]$, a set of strategies available to each player, $a_p \in \{a_p^A, a_p^B, \dots\} = \mathcal{A}_p$, and a payoff for each player under a particular joint strategy, $G_p(a)$, where $a = (a_1, \dots, a_N) \in \mathcal{A}$ and $\mathcal{A} = \otimes_p \mathcal{A}_p$. The distribution of play is described by a joint $\sigma(a)$, which follows the usual probability constraints $0 \leq \sigma(a) \leq 1$ and $\sum_{a \in \mathcal{A}} \sigma(a) = 1$. The goal of each player is to maximize their expected payoff under the joint, $v_p = \sum_{a \in \mathcal{A}} \sigma(a) G_p(a)$. Players could play independently by selecting a strategy according to their marginal $\sigma(a_p)$ over joint strategies, such that the joint factorizes $\sigma(a) = \otimes_p \sigma(a_p)$. However this is limiting because it does not allow players to coordinate. A mediator called a *correlation device* could be employed to allow players to execute arbitrary joints $\sigma(a)$ that do not necessarily factorize into their marginals. Such a mediator would sample from a publicly known joint $\sigma(a)$ and secretly communicate to each player their recommended strategy. Solution concepts for normal form-games focus on finding suitable joints, $\sigma(a)$, which have desirable properties such as stability and dominance.

If a player opts to play a single strategy deterministically, we say the player is employing a *pure strategy*, a_p . A grouping over all players' pure strategies is called a *pure strategy profile*, $a = (a_1, \dots, a_N)$. Sometimes a pure strategy profile is also called a *joint strategy*. It is often advantageous to denote a joint strategy as a tuple consisting of player p 's strategy along with the rest of the players' strategies: (a_p, a_{-p}) , where $a_{-p} = (a_1, \dots, a_{p-1}, a_{p+1}, \dots, a_N)$. Henceforth the notation $-p$ means the set of all players excluding player p .

If a player opts to play randomly across multiple strategies, we say the player is employing a *mixed strategy*, and is defined as a distribution over strategies $\sigma(a_p)$. Any strategy that has positive probability of being played is said to be in the *support*. A grouping over all players' mixed strategies is called a *mixed strategy profile*, $\otimes_p \sigma(a_p)$. A mixed strategy profile is a *joint*, however a joint is more general: the distribution does not need to factorize into its marginals. For example, consider some two-player normal form games, also called *bimatrix games* (Figure 2.4). Bimatrix games are often represented in a table, where the strategies available to player 1 are enumerated on the row headers and the strategies available to player 2 are listed on the column headers. Two numbers indicating the payoffs for player 1 and player 2 are found at the intersection of these strategies.

A common way of classifying games is through how each player's payoff relates to the other players' payoffs. A *zero-sum* game is one where $\sum_p G_p(a) = 0 \forall a$. A *constant-sum* game is similar: $\sum_p G_p(a) = c \forall a$. These properties are most useful when there are also only two players. Two-player, zero-sum games are sometimes also called *purely-competitive* games. In *common-payoff* games all players have the same payoff $G_p(a) = G_q(a) \forall p, q$, and are also sometimes known as *pure-coordination* games or *team* games. In general, we say a game is *general-sum* if there are no restrictions on the payoffs between players. In *symmetric* games, $G_p(a_p, a_{-p}) = G_q(a_q, a_{-q}) \forall p, q \in [1, N]$.

In the multiagent scenario there are multiple agents that are each interested in maximizing their payoffs. Therefore this is now a multi-objective optimization setting. The idea of a player having an optimal strategy is no longer well-defined, such a strategy now depends on the strategies of the other players. There are two types of approaches of dealing with optimality in this setting. Firstly we may reduce the space of possible strategies by excluding those which we know are strictly worse than others according to some definition. Secondly, we can select amongst the remaining strategies according to some, ideally uniquely, rule which necessarily converts the multi-objective problem into a single-objective problem.

2.2.2.1 Normal-Form Operators

There are two operators that are used for definitions of solution concepts and as primitives in multiagent algorithms.

Subsume Players

A game, $G(a)$, can be simplified, $\bar{G}(a_{-q})$, by fixing another player's, q , mixed strategy, $\sigma_q(a_q)$. The resulting game is an instance of the original game but with one less player.

$$\bar{G}_p(a_{-q}) = G_p(\sigma_q(a_q), a_{-q}) = \sum_{a_q} \sigma(a_q) G_p(a_q, a_{-q}) \quad \forall p \neq q \in [1, N] \quad (2.38)$$

Of course, it is also possible to “mix” over pure strategies, in which case this operator is more akin to indexing. If $\sigma(a_p)$ is sparse this calculation can be relatively cheap. This operator is linear $f : \mathbb{R}^{|\mathcal{A}_1| \times \dots \times |\mathcal{A}_N|} \rightarrow \mathbb{R}^{|\mathcal{A}_1| \times \dots \times |\mathcal{A}_{p-1}| \times |\mathcal{A}_{p+1}| \times \dots \times |\mathcal{A}_N|}$. It is possible to do this for more than one player at a time. It is common to subsume all other players in a game around fixed mixed joint strategies, $\sigma(a_{-p})$. It is not a requirement for the joint strategy to factorize, $\otimes_{-p} \sigma(a_p)$.

$$\bar{G}_p(a_p) = G_p(a_p, \sigma(a_{-p})) = \sum_{a_{-p}} \sigma(a_{-p}) G_p(a_p, a_{-p}) \quad (2.39)$$

If all other players are subsumed, the resulting payoff is simply a single-objective vector which could be solved directly.

Best-Response

Suppose we have a mixed joint over other players' strategies $\sigma(a_{-p})$. Player p can calculate the set of

strategies that achieve the greatest possible payoff against such a joint in a game.

$$a_p^{\text{BR}} \in \text{BR}_p^{a_p}(G_p(a), \sigma(a_{-p})) = \underset{a'_p \in A_p}{\operatorname{argmax}} \sum_{a_{-p}} \sigma(a_{-p}) G_p(a'_p, a_{-p}) \quad (2.40)$$

Notice that the best-response operation utilizes the subsume operation discussed above, meaning the best-response only needs to search over a single player's strategy. By definition, there has to be at least one best-response and if there is more than one best-response, they must have identical values. Therefore the value of the best-response is defined independently of any particular pure or mixed strategy.

$$v_p = \text{BRV}_p(G_p(a), \sigma(a_{-p})) = \max_{a'_p} \sum_{a_{-p}} \sigma(a_{-p}) G_p(a'_p, a_{-p}) \quad (2.41)$$

Furthermore, because we are indifferent to the choice of best-response strategy, the convex combination of these pure strategies is equal to the set of possible mixed strategy best-responses. A corollary is that if there exists a mixed strategy best-response, deterministic strategies in the support are also best-responses.

$$\sigma_p^{\text{BR}}(a_p) \in \text{BR}_p^{\sigma_p}(G_p(a), \sigma(a_{-p})) = \underset{\sigma(a'_p) \in \Sigma_p}{\operatorname{argmax}} \sum_{a_{-p}} \sigma(a_{-p}) G_p(a_p, a_{-p}) \quad (2.42)$$

The fact that there are many possible best-responses poses a problem for some algorithms: we have a *best-response selection problem*. Because we have a convex space of best-responses, this means we can use any strongly convex function to select uniquely amongst this space. Commonly we may opt to select the maximum entropy, $\text{MEBR}_p^\sigma(G_p(a), \sigma(a_{-p}))$, or maximum relative entropy from some other mixed strategy, $\text{MREBR}_p^\sigma(G_p(a), \sigma(a_{-p}), \hat{\sigma}(a_p))$. Often the selection criterion used is unimportant and many algorithms will operate with any best-response. In this case we will denote, $\text{AnyBR}_p^\sigma(G_p(a), \sigma(a_{-p}))$ to make this unambiguous. There exists a similar definition for better responses.

2.2.2.2 Dominance Solution Concepts

Dominance solution concepts attempt to reduce the strategy space to an interesting subset of the strategy space by defining rules that define a partial ordering over the strategies.

Pareto Dominance

The most fundamental basic method of reducing the problem space is through the notion of Pareto optimality. We say a strategy profile, a' , Pareto dominates another strategy profile, a , if $G_p(a') \geq G_p(a) \forall p \in [1, N]$, and $\exists G_p(a') > G_p(a)$. Therefore if a strategy profile is Pareto dominated, there exists another strategy in which a player can improve their payoff without hurting other players' payoffs. Although this does not select a single optimal strategy, it at least reduces the space of possible optima. Any strategy profile that is not Pareto dominated by another strategy profile is said to be *Pareto optimal*. Every game has at least one Pareto optimal strategy, and furthermore there always exists at least one pure Pareto optimal strategy. The degree of usefulness of reducing the space of interesting strategies depends on the game class. In zero-sum games, every strategy is Pareto optimal, so reducing Pareto optimal strategies does not reduce the space of interesting strategies. In common-payoff games, all Pareto optimal strategies have identical payoff, and therefore the space of Pareto optimal strategies can be quite small. While Pareto dominance is an interesting starting point, it does not reduce the space of strategies enough in most games to provide useful predictions of behaviour.

Strategic Dominance

Another notion of dominance is defined when a strategy appears better unilaterally to a player according only to their payoff, over all of the other players' strategies. If a player's strategy dominates all other

strategies, it is said to be a *pure dominant strategy*, and a strategy that is dominated by any other strategy is said to be a *pure dominated strategy*. The degree of domination can be *strict* or *weak*. It is possible to introduce an approximation variable ϵ_p , which is commonly defined to be zero. This additional variable will be motivated later in the equilibrium section. Strategy a_p'' is approximately dominated by a_p' if:

$$\text{Strict:} \quad G_p(a_p', a_{-p}) > G_p(a_p'', a_{-p}) + \epsilon_p \quad \forall a_{-p} \in \mathcal{A}_{-p} \quad (2.43a)$$

$$\begin{aligned} \text{Weak:} \quad & G_p(a_p', a_{-p}) > G_p(a_p'', a_{-p}) + \epsilon_p \quad \exists a_{-p}, \text{ and} \\ & G_p(a_p', a_{-p}) \geq G_p(a_p'', a_{-p}) + \epsilon_p \quad \forall a_{-p} \in \mathcal{A}_{-p} \end{aligned} \quad (2.43b)$$

It is also possible to define dominance in terms of mixed strategies. A strategy, a_p'' , is dominated if there exists a mixture, $\sigma(a_p')$, such that:

$$\text{Strict:} \quad \sum_{a_p'} \sigma(a_p') G_p(a_p', a_{-p}) > G_p(a_p'', a_{-p}) + \epsilon_p \quad \forall a_{-p} \in \mathcal{A}_{-p} \quad (2.44a)$$

$$\begin{aligned} \text{Weak:} \quad & \sum_{a_p'} \sigma(a_p') G_p(a_p', a_{-p}) > G_p(a_p'', a_{-p}) + \epsilon_p \quad \exists a_{-p}, \text{ and} \\ & \sum_{a_p'} \sigma(a_p') G_p(a_p', a_{-p}) \geq G_p(a_p'', a_{-p}) + \epsilon_p \quad \forall a_{-p} \in \mathcal{A}_{-p} \end{aligned} \quad (2.44b)$$

It could be argued that in the absence of other information, a rational player should never play a dominated strategy. Therefore dominance describes how a player may act in a game and therefore partly describes a solution to a game. The argument of strategic dominance can be applied iteratively resulting in an algorithm called Iterated elimination of strictly dominated strategies (IESDS). In the scenario where each player only has a single non-dominated strategy, this can be considered the full solution to the game. One criticism of strategic dominance is that under this solution concept, pure joint strategies that have strictly better welfare can be ruled out. The most famous example is Prisoner's dilemma (Table 4.1b).

2.2.2.3 Equilibrium Solution Concepts

A popular class of solution concepts are equilibrium based: joint distributions, $\sigma(a)$, where under certain definitions, no player has incentive to unilaterally deviate. The most well known is Nash equilibrium (Nash, 1951) (NE), which is tractable, interchangeable and unexploitable in two-player, zero-sum games (Shoham and Leyton-Brown, 2009). NEs are always factorizable joint distributions. A related solution concept is correlated equilibrium (Aumann, 1974) (CE) which is more suitable for n-player, general-sum settings where players are allowed to coordinate strategies with each other if it is mutually beneficial. Furthermore, CEs are more compatible with the Bayesian perspective, and arise as a result of learning rules (Cesa-Bianchi and Lugosi, 2006; Foster and Vohra, 1997). The mechanism of implementing a CE is via a correlation device which samples a joint strategy from a known public distribution and recommends the sampled strategy secretly to each player. A distribution is in correlated equilibrium if no player is incentivised to unilaterally deviate from the recommendation after receiving it. CE that are factorizable are also NEs. An additional solution concept, the coarse correlated equilibrium (Moulin and Vial, 1978) (CCE), requires players to commit to the recommendation before it has been sampled. It is less computationally expensive and permits even higher equilibrium payoffs. These sets are related to each other $\epsilon\text{-NE} \subseteq \epsilon\text{-CE} \subseteq \epsilon\text{-CCE}$, $\epsilon\text{-WSNE} \subseteq \epsilon\text{-NE}$. The empirical average policy of no-regret learning algorithms in self-play are known to converge to CCEs (Foster and Vohra, 1997; Hart and Mas-Colell, 2000).

All these equilibria have approximate forms which are parameterized by the approximation parameter ϵ which describes the maximum allowed incentive to deviate to a best-response (across all players). There are two common methods of defining an approximate equilibrium: the standard approximate equilibrium (Shoham and Leyton-Brown, 2009), describes the bound on incentive to deviate under the joint, and the well-supported (WS) approximate equilibrium (Goldberg and Papadimitriou, 2006), describes the

bound on incentive to deviate under the conditionals. When $\epsilon = 0$, these definitions become equivalent. These sets are related to each other ϵ -WSNE $\subseteq \epsilon$ -NE and ϵ -WSCE $\subseteq \epsilon$ -CE. The standard method has the property that any $\epsilon > \epsilon^{\min}$ will permit a full-support equilibrium, where $\epsilon^{\min} \leq 0$ is the minimum ϵ that permits a feasible solution in a game. Each player may have individual tolerances to deviation, ϵ_p .

Well-Supported Correlated Equilibria

Consider a correlation device, a trusted third party that samples, $a'' = (a''_1, \dots, a''_N)$, from a public joint, $\sigma(a)$, and communicates each player's component of the sample (the *recommendation*), a''_p , secretly to each player. After receiving a recommendation, a player can now calculate a posterior probability over what players were recommended $\sigma(a_{-p}|a''_p)$. A player is now free to either play the recommendation or change to another strategy, $\sigma(a'_p)$, a *deviation* strategy. A player is only incentivized to deviate if they believe they will get more payoff in expectation than following the recommendation, assuming the other players follow the recommendation. Suppose that a player will only deviate if they improve their expected payoff by at least ϵ_p after receiving the recommendation. Any distribution $\sigma(a)$ such that no players have incentive to deviate is called a well-supported correlated equilibrium (WSCE) (Aumann, 1974; Czumaj et al., 2014; Goldberg and Papadimitriou, 2006). To develop intuition, consider a normal-form game recast as a two-stage extensive-form game. In the first stage, a chance node represents the selection of strategies from a correlation device. Subsequently, in the second stage, players simultaneously choose their actions².

WSCEs can be defined more directly in terms of deviation gains: the payoff gain of a player, when switching from a recommended strategy, a''_p , to a deviation strategy, a'_p . The gain in payoff obviously depends on the strategies that the other players employed, a_{-p} . The deviation gain can be defined for every deviation strategy, recommended strategy, and other player strategy profile in terms of a tensor with shape $[|\mathcal{A}'_p|, |\mathcal{A}''_p|, \otimes_{q \in -p} |\mathcal{A}_q|]$.

Definition 2.2.1 (Well-Supported Correlated Equilibrium Deviation Gain).

$$A_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) = G_p(a'_p, a_{-p}) - G_p(a''_p, a_{-p}) \quad (2.45)$$

The deviation gains can be used to define linear inequality constraints. The constraints impose that for WSCEs the deviation from a''_p to a'_p must have a payoff gain for player p of at most ϵ_p , after the recommendation is given. Because the bound on the gain is after a recommendation is given, a conditional distribution on the recommendation is used in the WSCE to compute an expected deviation gain for every recommendation-deviation pair.

Definition 2.2.2 (Well-Supported Correlated Equilibrium).

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_{-p}|a''_p) A_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) \leq \epsilon_p \quad \forall a''_p \in \mathcal{A}_p^+, a'_p \in \mathcal{A}_p, p \in [1, N] \quad (2.46)$$

Notice that if $G_p(a'_p, a_{-p}) > G_p(a''_p, a_{-p}) + \epsilon \quad \forall a'_p \in \mathcal{A}_p$, then a''_p must have zero support in any ϵ -WSCE. When $\epsilon = 0$, this is the same as strict strategic dominance. Therefore there is a deep connection between WSCEs and strategic dominance. ϵ -dominated strategies can be pruned from the game without consequence before calculating its WSCE. This argument also holds when $\epsilon > 0$ and $\epsilon < 0$, however in the latter scenario some games may have all of their strategies pruned (implying that there is no feasible solution).

This definition's constraints are only defined for recommendations with positive support, $\mathcal{A}_p^+ := \text{supp}(\sigma(a_p)) = \{a_p | \sigma(a_p) > 0\}$, which can be difficult to know *a priori*. Furthermore, dealing with

²WSCEs corresponds to NEs in this extensive-form game. These terms will be more thoroughly defined in later sections

the conditional $\sigma(a_{-p}|a_p'') = \frac{\sigma(a_{-p}, a_p'')}{\sigma(a_p'')}$ may make the constraint nonlinear, which could complicate computation. Therefore it is easier to work with an equivalent form, which allows us to use the joint distribution. It is found by multiplying each side of the inequality constraint with the marginal $\sigma(a_p'')$, noting that positive scaling an equation of the plane does not change the half-space.

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_{-p}, a_{-p}) A_p^{\text{WSCE}}(a_p', a_p'', a_{-p}) \leq \sigma(a_p'') \epsilon_p \quad \forall a_p'', a_p' \in \mathcal{A}_p, p \in [1, N] \quad (2.47)$$

In a similar argument, summing over $a_{-p} \in \mathcal{A}_{-p}$ can be inconvenient because it means indexing into different joint strategies to compute each constraint, so it is advantageous to define a new WSCE deviation gain as a sparse tensor of shape $[|\mathcal{A}'_p|, |\mathcal{A}''_p|, |\mathcal{A}|]$. This form of deviation gain is actually more naturally related to the correlated equilibrium (CE), which is discussed later. This allows summing all joint strategies without changing the meaning of the WSCE. This is the most mathematically convenient definition of the WSCE.

$$\sum_{a \in \mathcal{A}} \sigma(a) \begin{cases} A_p^{\text{WSCE}}(a_p', a_p'', a_{-p}) & a_p = a_p'' \\ 0 & \text{otherwise} \end{cases} \leq \sigma(a_p'') \epsilon_p \quad \forall a_p'', a_p' \in \mathcal{A}_p, p \in [1, N] \quad (2.48)$$

It is possible to subsume the approximation term directly into the deviation gains to result in approximate deviation gains, $A^{\epsilon\text{-WSCE}}(a_p', a_p'', a_{-p})$.

Definition 2.2.3 (Approximate Well-Supported Correlated Equilibrium Deviation Gain).

$$A_p^{\epsilon\text{-WSCE}}(a_p', a_p'', a_{-p}) = G_p(a_p', a_{-p}) - G_p(a_p'', a_{-p}) - \epsilon_p \quad (2.49a)$$

This results in final equivalent definitions of WSCEs.

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_{-p}|a_p'') A_p^{\epsilon\text{-WSCE}}(a_p', a_p'', a_{-p}) \leq 0 \quad \forall a_p'', a_p' \in \mathcal{A}_p, p \in [1, N] \quad (2.50a)$$

$$\sum_{a \in \mathcal{A}} \sigma(a) \begin{cases} A_p^{\epsilon\text{-WSCE}}(a_p', a_p'', a_{-p}) & a_p = a_p'' \\ 0 & \text{otherwise} \end{cases} \leq 0 \quad \forall a_p'', a_p' \in \mathcal{A}_p, p \in [1, N] \quad (2.50b)$$

Therefore the constraints can be written compactly in matrix form $A_p^{\epsilon\text{-WSCE}} \sigma \leq 0$, where $A_p^{\epsilon\text{-WSCE}}$ is a matrix with shape $[|\mathcal{A}|^2_p, |\mathcal{A}|]$, σ is a vector with shape $|\mathcal{A}|$. It is worth pausing here to consider the geometric interpretation of these constraints. Each inequality constraint corresponds to a plane which partitions the space into a half-space, where one side obeys the constraint and the other side violates the constraint. Each row of the $A_p^{\epsilon\text{-WSCE}}$ matrix corresponds to the normal equation of this plane. Each plane forms a surface, and taken together the planes form a convex polytope set of valid solutions. Any convex combinations of WSCEs is also a WSCE. It is known that for $\epsilon_p \geq 0$ this set is nonempty, meaning that a WSCE exists for every game. Some games have equilibria with $\epsilon < 0$.

Theorem 2.2.4 (Well-Supported Correlated Equilibrium Existence). *For every finite game, with $\epsilon \geq 0$, there exists an ϵ -WSCE.*

The definition of the WSCE can be written in terms of best-response operators. Such equivalent

formulations become important when designing and proving convergence of game-theoretic algorithms.

$$\begin{aligned} \sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_{-p} | a_p'') A_p^{\text{WSCE}}(a_p', a_p'', a_{-p}) &\leq \epsilon_p \quad \forall a_p'' \in \mathcal{A}_p^+, a_p' \in \mathcal{A}_p, p \in [1, N] \\ \max_{a_p' \in \mathcal{A}_p} \sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_{-p} | a_p'') A_p^{\text{WSCE}}(a_p', a_p'', a_{-p}) &\leq \epsilon_p \quad \forall a_p'' \in \mathcal{A}_p^+, p \in [1, N] \end{aligned} \quad (2.51a)$$

$$\begin{aligned} \max_{a_p' \in \mathcal{A}_p} \sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_{-p} | a_p'') (G_p(a_p', a_{-p}) - G_p(a_p'', a_{-p})) &\leq \epsilon_p \quad \forall a_p'' \in \mathcal{A}_p^+, p \in [1, N] \\ \text{BRV}_p^{a_p} (G_p(a), \sigma_p(a_{-p} | a_p'')) - \sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_{-p} | a_p'') G_p(a_p'', a_{-p}) &\leq \epsilon_p \quad \forall a_p'' \in \mathcal{A}_p^+, p \in [1, N] \end{aligned} \quad (2.51b)$$

Correlated Equilibria

An alternative definition, called the Correlated Equilibrium (CE), defines bounds on the overall expected deviation, normalized by the probability of the recommendation. A WSCE, with an approximation per recommendation $\epsilon_p^{\text{WSCE}}(a_p'')$, would relate to a CE with approximation $\epsilon_p^{\text{CE}} = \sigma(a_p'') \epsilon_p^{\text{WSCE}}(a_p'')$. It immediately follows that WSCE is a subset of CE. The ϵ -CE is a weaker version of ϵ -WSCE because a CE permits placing probability on strategies that are arbitrarily low payoff. When $\epsilon = 0$, WSCE and CE are equivalent. The CE can be defined with a small change to the WSCE definition.

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_{-p} | a_p'') A_p^{\text{WSCE}}(a_p', a_p'', a_{-p}) \leq \frac{\epsilon_p}{\sigma(a_p'')} \quad \forall a_p'' \in \mathcal{A}_p^+, a_p' \in \mathcal{A}_p, p \in [1, N] \quad (2.52a)$$

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_p'' | a_{-p}) A_p^{\text{WSCE}}(a_p', a_p'', a_{-p}) \leq \epsilon_p \quad \forall a_p'', a_p' \in \mathcal{A}_p, p \in [1, N] \quad (2.52b)$$

The correlated equilibrium deviation gains, and the correlated equilibrium, can be more naturally defined using the joint form described in the previous section.

Definition 2.2.5 (Correlated Equilibrium Deviation Gain).

$$A_p^{\text{CE}}(a_p', a_p'', a) = \begin{cases} A_p^{\text{WSCE}}(a_p', a_p'', a_{-p}) = G_p(a_p', a_{-p}) - G_p(a_p'', a_{-p}) & a_p = a_p'' \\ 0 & \text{otherwise} \end{cases} \quad (2.53)$$

Definition 2.2.6 (Correlated Equilibrium).

$$\sum_{a \in \mathcal{A}} \sigma(a) A_p^{\text{CE}}(a_p', a_p'', a) \leq \epsilon_p \quad \forall a_p'', a_p' \in \mathcal{A}_p, p \in [1, N] \quad (2.54)$$

Notice that if $G_p(a_p', a_{-p}) > G_p(a_p'', a_{-p}) + \epsilon \quad \forall a_p' \in \mathcal{A}_p$ and $\epsilon \leq 0$, then a_p'' must have zero support in any such ϵ -CE, and strategically dominated strategies can be pruned. However, when $\epsilon > 0$, all strategies can have support, and a full-support ϵ -CE exists. In a similar fashion to before, the approximation parameter can be subsumed into the deviation gain.

Definition 2.2.7 (Approximate Correlated Equilibrium Deviation Gain).

$$A_p^{\epsilon\text{-CE}}(a_p', a_p'', a) = A_p^{\text{CE}}(a_p', a_p'', a) - \epsilon_p \quad (2.55)$$

Therefore the constraints can be written compactly in matrix form $A_p^{\epsilon\text{-CE}} \sigma \leq 0$, if A_p^{CE} is a matrix with shape $[|\mathcal{A}|_p^2, |\mathcal{A}|]$, and σ is a vector with shape $|\mathcal{A}|$. When $\epsilon_p = 0$ the definitions of WSCE and CE are equivalent. Because $\epsilon_p^{\text{WSCE}}(a_p'') = \frac{\epsilon_p^{\text{CE}}}{\sigma(a_p'')}$, for $\epsilon_p > 0$, ϵ -CEs are a superset of ϵ -WSCEs because $\sigma(a_p'') \leq 1$.

CE deviation gains relate to the WSCE deviation gains.

$$A_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) = \sum_{a_p \in \mathcal{A}_p} A_p^{\text{CE}}(a'_p, a''_p, a) \quad (2.56)$$

The definition of the correlated equilibrium can also be written in terms of best-responses.

$$\sum_{a \in \mathcal{A}} \sigma(a) A_p^{\text{CE}}(a'_p, a''_p, a) \leq \epsilon_p \quad \forall a''_p, a'_p \in \mathcal{A}_p, p \in [1, N] \quad (2.57a)$$

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a''_p, a_{-p}) A_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) \leq \epsilon_p \quad \forall a''_p, a'_p \in \mathcal{A}_p, p \in [1, N] \quad (2.57b)$$

$$\max_{a'_p \in \mathcal{A}_p} \sum_{a \in \mathcal{A}} \sigma(a) A_p^{\text{CE}}(a'_p, a''_p, a) \leq \epsilon_p \quad \forall a''_p \in \mathcal{A}_p, p \in [1, N] \quad (2.57c)$$

$$\text{BRV}_p^{a_p} (G_p(a), \sigma_p(a_{-p}, a''_p)) - \sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_{-p}, a''_p) G_p(a''_p, a_{-p}) \leq \epsilon_p \quad \forall a''_p \in \mathcal{A}_p^+, p \in [1, N] \quad (2.57d)$$

Coarse Correlated Equilibria

Another solution concept, coarse correlated equilibrium (CCE) (Moulin and Vial, 1978), is similar to CE but players are required to commit to their recommendation before receiving it. Deviations are therefore considered based only on the prior, which is the public joint distribution, $\sigma(a)$, and no additional posterior can be calculated over the other players' potential recommendations. The deviation gains are then defined either directly, by summing over all possible recommendations of the CE deviation gains, or a reshape of the WSCE deviation gains. There is only a constraint for each possible deviation $a'_p \in \mathcal{A}_p, \forall p$ when defining the CCE, and it is defined directly on the joint distribution.

Definition 2.2.8 (Coarse Correlated Equilibrium Deviation Gains).

$$A_p^{\text{CCE}}(a'_p, a) = G_p(a'_p, a_{-p}) - G_p(a) = \sum_{a''_p} A_p^{\text{CE}}(a'_p, a''_p, a) = A_p^{\text{WSCE}}(a'_p, a_p, a_{-p}) \quad (2.58)$$

Definition 2.2.9 (Coarse Correlated Equilibrium).

$$\sum_{a \in \mathcal{A}} \sigma(a) A_p^{\text{CCE}}(a'_p, a) \leq \epsilon_p \quad \forall a'_p \in \mathcal{A}_p, p \in [1, N] \quad (2.59)$$

There is no “well-supported” version of CCEs. However, an approximate deviation gain can be defined noting the property that $\sum_a \sigma(a) = 1 \implies \sum_a \sigma(a) \epsilon_p = \epsilon_p$.

Definition 2.2.10 (Approximate Coarse Correlated Equilibrium Deviation Gain).

$$A_p^{\epsilon\text{-CCE}}(a'_p, a) = A_p^{\text{CCE}}(a'_p, a) - \epsilon_p \quad (2.60)$$

Therefore the constraints can be written compactly in matrix form $A_p^{\epsilon\text{-CCE}} \sigma \leq 0$, if A_p^{CCE} is a matrix with shape $[|\mathcal{A}|_p, |\mathcal{A}|]$, and σ is a vector with shape $|\mathcal{A}|$. CCEs are a superset of (WS)CEs: $\text{WSCE} \subseteq \text{CE} \subseteq \text{CCE}$. The definition of CCE is also closely related to finding best-responses.

$$\begin{aligned} \sum_{a \in \mathcal{A}} \sigma(a) A_p^{\text{CCE}}(a'_p, a) &\leq \epsilon_p & \forall a'_p \in \mathcal{A}_p, p \in [1, N] \\ \max_{a'_p} \sum_{a \in \mathcal{A}} \sigma(a) A_p^{\text{CCE}}(a'_p, a) &\leq \epsilon_p & \forall p \in [1, N] \end{aligned} \quad (2.61)$$

$$\begin{aligned}
\max_{a'_p} \sum_{a \in \mathcal{A}} \sigma(a) G_p(a'_p, a_{-p}) - \sum_{a \in \mathcal{A}} \sigma(a) G_p(a) &\leq \epsilon_p & \forall p \in [1, N] \\
\max_{a'_p \in \mathcal{A}_{-p}} \sum_{a \in \mathcal{A}} \sigma(a_{-p}) G_p(a'_p, a_{-p}) - \sum_{a \in \mathcal{A}} \sigma(a) G_p(a) &\leq \epsilon_p & \forall p \in [1, N] \\
\text{BRV}_p^{a_p}(G_p(a), \sigma(a_{-p})) - \sum_{a \in \mathcal{A}} \sigma(a) G_p(a) &\leq \epsilon_p & \forall p \in [1, N]
\end{aligned} \tag{2.62}$$

Nash Equilibria

The Nash equilibrium (NE) (Nash, 1951) has a similar definition to (C)CE but have an extra constraint that the joint distribution factorizes $\otimes_p \sigma(a_p) = \sigma(a)$, resulting in nonlinear constraints³. All NEs are also (C)CEs. An NE is always on the boundary of the polytope for non-trivial games (Nau et al., 2004). This factorization means that each player's strategy is independent of all other players' actions. Therefore, a correlation device revealing a signal does not carry any information about the other players recommendations because $\sigma(a_{-p}|a_p) = \otimes_{q \neq p} \sigma(a_q)$, and therefore does not influence a player's own actions. Consequently no correlation device is needed to execute NEs and players can sample their own actions independently. Therefore there is no distinction between the CE and CCE definitions when the joint factorizes and hence ϵ -NE can be defined in term of either ϵ -CE or ϵ -CCE constraints. All of the following are equivalent definitions of NE:

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \otimes_{q \neq p} \sigma(a_q) A_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) \leq \frac{\epsilon_p}{\sigma(a''_p)} \quad \forall a''_p \in \mathcal{A}_p^+, a'_p \in \mathcal{A}_p, p \in [1, N] \tag{2.63a}$$

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a''_p) \otimes_{q \neq p} \sigma(a_q) A_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) \leq \epsilon_p \quad \forall a''_p, a'_p \in \mathcal{A}_p, p \in [1, N] \tag{2.63b}$$

$$\sum_{a \in \mathcal{A}} \otimes_{q \in [1, N]} \sigma(a_q) A_p^{\text{CE}}(a'_p, a''_p, a) \leq \epsilon_p \quad \forall a''_p, a'_p \in \mathcal{A}_p, p \in [1, N] \tag{2.63c}$$

$$\sum_{a \in \mathcal{A}} \otimes_{q \in [1, N]} \sigma(a_q) A_p^{\epsilon\text{-CE}}(a'_p, a''_p, a) \leq 0 \quad \forall a''_p, a'_p \in \mathcal{A}_p, p \in [1, N] \tag{2.63d}$$

$$\sum_{a \in \mathcal{A}} \otimes_{q \in [1, N]} \sigma(a_q) A_p^{\text{CCE}}(a'_p, a) \leq \epsilon_p \quad \forall a'_p \in \mathcal{A}_p, p \in [1, N] \tag{2.63e}$$

$$\sum_{a \in \mathcal{A}} \otimes_{q \in [1, N]} \sigma(a_q) A_p^{\epsilon\text{-CCE}}(a'_p, a) \leq 0 \quad \forall a'_p \in \mathcal{A}_p, p \in [1, N] \tag{2.63f}$$

Nash equilibrium always exists for any finite game. Games can have multiple equilibria and, unlike (C)CEs, these can be isolated. A pure strategy Nash equilibrium does not necessarily exist (for example cyclic games such as matching pennies and rock-paper-scissors). Unlike (C)CEs, there are no NEs with negative approximation, $\epsilon_p < 0$. Because the CCE-based definition of NE has fewer constraints, it is more commonly used.

Definition 2.2.11 (Nash Equilibrium).

$$\sum_{a \in \mathcal{A}} \otimes_{q \in [1, N]} \sigma(a_q) A_p^{\text{CCE}}(a'_p, a) \leq \epsilon_p \quad \forall a'_p \in \mathcal{A}_p, p \in [1, N] \tag{2.64}$$

Theorem 2.2.12 (Nash Equilibrium Existence). *For every finite game, with $\epsilon \geq 0$, there exists an ϵ -Nash equilibrium.*

Lemma 2.2.13. *ϵ -NEs only exist when $\epsilon_p \geq 0$. Equivalently, the largest expected deviation gain is at least zero.*

³This is why NEs are harder to compute than (C)CEs.

Theorem 2.2.14. ϵ -NEs can be equivalently defined in terms of either CE or CCE constraints.

Proof. Using Lemma 2.2.13 for the second last step.

$$\begin{aligned}
\sum_{a \in \mathcal{A}} \otimes_q \sigma(a_q) A_p^{\text{CE}}(a'_p, a''_p, a) &\leq \epsilon_p & \forall a''_p, a'_p \in \mathcal{A}_p, p \in [1, N] \\
\max_{a''_p} \sum_{a \in \mathcal{A}} \otimes_q \sigma(a_q) A_p^{\text{CE}}(a'_p, a''_p, a) &\leq \epsilon_p & \forall a'_p \in \mathcal{A}_p, p \in [1, N] \\
\sum_{a \in \mathcal{A}} \otimes_q \sigma(a_q) \left[\max_{a'_p} A_p^{\text{CE}}(a'_p, a''_p, a) \right] &\leq \epsilon_p & \forall a'_p \in \mathcal{A}_p, p \in [1, N] \\
\sum_{a \in \mathcal{A}} \otimes_q \sigma(a_q) \max [A_p^{\text{CCE}}(a'_p, a), 0] &\leq \epsilon_p & \forall a'_p \in \mathcal{A}_p, p \in [1, N] \\
\max_{a'_p} \sum_{a \in \mathcal{A}} \otimes_q \sigma(a_q) A_p^{\text{CCE}}(a'_p, a) &\leq \epsilon_p & \forall p \in [1, N] \\
\sum_{a \in \mathcal{A}} \otimes_q \sigma(a_q) A_p^{\text{CCE}}(a'_p, a) &\leq \epsilon_p & \forall a'_p \in \mathcal{A}_p, p \in [1, N]
\end{aligned}$$

□

Therefore NEs are simply (C)CEs that have factorizable joints. The definitions of equilibria are tightly related to finding best-responses. This observation is key when designing equilibrium solving algorithms that scale to large games.

$$\begin{aligned}
\sum_{a \in \mathcal{A}} \otimes_q \sigma(a_q) A_p^{\text{CCE}}(a'_p, a) &\leq \epsilon_p & \forall a'_p \in \mathcal{A}_p, p \in [1, N] \\
\max_{a'_p} \sum_{a \in \mathcal{A}} \otimes_q \sigma(a_q) A_p^{\text{CCE}}(a'_p, a) &\leq \epsilon_p & \forall p \in [1, N] \\
\max_{a'_p} \sum_{a \in \mathcal{A}} \otimes_q \sigma(a_q) G_p(a'_p, a_{-p}) - \sum_{a \in \mathcal{A}} \otimes_q \sigma(a_q) G_p(a) &\leq \epsilon_p & \forall p \in [1, N] \\
\max_{a'_p} \sum_{a_{-p} \in \mathcal{A}_{-p}} \otimes_{q \neq p} \sigma(a_q) G_p(a'_p, a_{-p}) - \sum_{a \in \mathcal{A}} \otimes_q \sigma(a_q) G_p(a) &\leq \epsilon_p & \forall p \in [1, N] \\
\text{BR}_p^{a_p}(G_p(a), \sigma(a_{-p})) - \sum_{a \in \mathcal{A}} \otimes_q \sigma(a_q) G_p(a) &\leq \epsilon_p & \forall p \in [1, N] \quad (2.65)
\end{aligned}$$

Well-Supported Nash Equilibria

There also exists well-supported version ϵ -WSNE (Czumaj et al., 2014; Goldberg and Papadimitriou, 2006), which can only be defined in terms of WSCE constraints, because the approximate term needs to be computed after a recommendation. As before, when $\epsilon_p = 0$, ϵ -WSNE and ϵ -NE are equivalent.

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \otimes_{q \neq p} \sigma(a_q) A_p^{(\text{WS})\text{CE}}(a'_p, a''_p, a_{-p}) \leq \epsilon_p \quad \forall a''_p \in \mathcal{A}_p^+, a'_p \in \mathcal{A}_p, p \in [1, N] \quad (2.66a)$$

$$\sum_{a \in \mathcal{A}} \sigma(a''_p) \otimes_{q \neq p} \sigma(a_q) A_p^{(\text{WS})\text{CE}}(a'_p, a''_p, a) \leq \sigma(a''_p) \epsilon_p \quad \forall a''_p, a'_p \in \mathcal{A}_p, p \in [1, N] \quad (2.66b)$$

$$\sum_{a \in \mathcal{A}} \otimes_{q \in [1, N]} \sigma(a_q) A_p^{\epsilon\text{-WSCE}}(a'_p, a''_p, a) \leq 0 \quad \forall a''_p, a'_p \in \mathcal{A}_p, p \in [1, N] \quad (2.66c)$$

Note that for two-player games, the WSNE constraints can be linear (Equation (2.66a)). This means the NE can be directly solved for using a linear program (LP) with the caveat that one knows the support \mathcal{A}_p^+ . The LP is only feasible if a correct support is provided. One could guess a support and attempt to solve the LP. Such techniques enables the two-player general-sum NE solver *support enumeration* (Nisan

et al., 2007).

Connection to Minimax Strategies in Two-Player Games

In n -player general-sum games there is a notion of a *pure maximin strategy*, and a *pure maximin value* which represents a conservative choice where player p maximizes the minimal possible payoff they can receive when the other players play any pure strategy. This is the highest value a player can guarantee without knowing the other players' strategies, or equivalently the lowest value the other players can force for player p when they know player p 's strategy.

$$\text{Pure Maximin Strategy:} \quad a_p = \arg \max_{a_p} \min_{a_{-p}} G_p(a_p, a_{-p}) \quad (2.67a)$$

$$\text{Pure Maximin Value:} \quad \underline{v}_p = \max_{a_p} \min_{a_{-p}} G_p(a_p, a_{-p}) \quad (2.67b)$$

There is an analogous *pure minimax strategy*, and a *pure minimax value*. This represents the lowest value that the other players can force player p to receive without knowing player p 's strategy or equivalently it is the highest value the player can be sure to get when they know the actions of the other players.

$$\text{Pure Minimax Strategy:} \quad a_{-p} = \arg \min_{a_{-p}} \max_{a_p} G_p(a_p, a_{-p}) \quad (2.68a)$$

$$\text{Pure Minimax Value:} \quad \bar{v}_p = \min_{a_{-p}} \max_{a_p} G_p(a_p, a_{-p}) \quad (2.68b)$$

It is also possible to consider mixed strategies to obtain tighter bounds on the values. It is not necessary to optimize over mixed strategies in the outer optimization, because any mixture will be equal to the value of the pure strategies at the optimum.

$$\underline{v}_p = \max_{\sigma(a_p)} \min_{\sigma(a_{-p})} \sum_a \sigma(a_p) \sigma(a_{-p}) G_p(a_p, a_{-p}) = \max_{a_p} \min_{\sigma(a_{-p})} \sum_{a_{-p}} \sigma(a_{-p}) G_p(a_p, a_{-p}) \quad (2.69a)$$

$$\bar{v}_p = \min_{\sigma(a_{-p})} \max_{\sigma(a_p)} \sum_a \sigma(a_p) \sigma(a_{-p}) G_p(a_p, a_{-p}) = \min_{a_{-p}} \max_{\sigma(a_p)} \sum_{a_p} \sigma(a_p) G_p(a_p, a_{-p}) \quad (2.69b)$$

Von Neumann famously proved that these bounds are tight (von Neumann, 1928). This is the theorem that is credited with establishing game theory as a field.

Theorem 2.2.15 (Minimax Theorem). *For any finite n -player general-sum game:*

$$\max_{a_p} \min_{\sigma(a_{-p})} \sum_{a_{-p}} \sigma(a_{-p}) G_p(a_p, a_{-p}) = \min_{a_{-p}} \max_{\sigma(a_p)} \sum_{a_p} \sigma(a_p) G_p(a_p, a_{-p}) \quad (2.70)$$

Therefore finding the mixed minimax value is an easier optimization and is equal to finding the mixed maximin value of a game. We denote the value $v_p = \bar{v}_p = \underline{v}_p$. Now consider a two-player zero-sum game, where $G_1(a) = -G_2(a)$. Note that $\min_i(x_i) = -\max_i(-x_i)$.

$$v_1 = \max_{a_1} \min_{\sigma(a_2)} \sum_{a_2} \sigma(a_2) G_1(a_1, a_2) = \min_{a_2} \max_{\sigma(a_1)} \sum_{a_1} \sigma(a_1) G_1(a_1, a_2) \quad (2.71a)$$

$$\begin{aligned} v_2 &= \max_{a_2} \min_{\sigma(a_1)} \sum_{a_1} \sigma(a_1) G_2(a_1, a_2) = \min_{a_1} \max_{\sigma(a_2)} \sum_{a_2} \sigma(a_2) G_2(a_1, a_2) \\ &= -\min_{a_2} \max_{\sigma(a_1)} \sum_{a_1} \sigma(a_1) G_1(a_1, a_2) = -\max_{a_1} \min_{\sigma(a_2)} \sum_{a_2} \sigma(a_2) G_1(a_1, a_2) = -v_1 \end{aligned} \quad (2.71b)$$

Therefore $v_1 = -v_2$. Importantly, note that the minimax objective for player 1 and player 2 are completely aligned. Therefore the minimax solution of a game is one way to turn a multiobjective multiagent

optimization problem into a single objective saddle point problem.

$$\arg \max_{\sigma(a_1)} \arg \min_{\sigma(a_2)} \sum_{a_1, a_2} \sigma(a_1) \sigma(a_2) G_1(a_1, a_2) \quad (2.72)$$

Theorem 2.2.16. *Minimax equilibria and Nash equilibria are equivalent in two-player zero-sum games.*

Proof. The solutions to player 1's Minimax equilibria can be cast as inequality constraints:

$$\begin{aligned} \max_{\sigma(a'_1)} \min_{\sigma(a_2)} \sum_{a'_1, a_2} \sigma(a'_1) \sigma(a_2) G_1(a'_1, a_2) &= v_1 \\ \max_{\sigma(a'_1)} \sum_{a_1, a_2} \sigma(a'_1) \sigma^*(a_2) G_1(a'_1, a_2) &= \sum_{a_1, a_2} \sigma^*(a_1) \sigma^*(a_2) G_1(a_1, a_2) \\ \max_{a'_1} \sum_{a_2} \sigma^*(a_2) G_1(a'_1, a_2) &= \sum_{a_1, a_2} \sigma^*(a_1) \sigma^*(a_2) G_1(a_1, a_2) \\ \max_{a'_1} \sum_{a_1, a_2} \sigma^*(a_1) \sigma^*(a_2) G_1(a'_1, a_2) &= \sum_{a_1, a_2} \sigma^*(a_1) \sigma^*(a_2) G_1(a_1, a_2) \\ \sum_{a_1, a_2} \sigma^*(a_1) \sigma^*(a_2) G_1(a'_1, a_2) &\leq \sum_{a_1, a_2} \sigma^*(a_1) \sigma^*(a_2) G_1(a_1, a_2) \quad \forall a''_p \\ \sum_{a_1, a_2} \sigma^*(a_1) \sigma^*(a_2) A_p^{\text{CCE}}(a'_p, a) &\leq 0 \quad \forall a''_p \end{aligned}$$

For player 2 we have:

$$- \sum_{a_1, a_2} \sigma^*(a_1) \sigma^*(a_2) A_p^{\text{CCE}}(a'_p, a) \leq 0 \quad \forall a''_p$$

Which is equivalent to the definition of NE. □

This connection allows us to deduce important properties of NE in two-player zero-sum games making it fundamental and prescriptive. If the game is also zero-sum ($G_1(a_1, a_2) = -G_2(a_1, a_2)$), the NE solution is *interchangeable*. This means that if $\sigma(a_1)\sigma(a_2)$ and $\sigma'(a_1)\sigma'(a_2)$ are both NEs, then $\sigma'(a_1)\sigma(a_2)$ and $\sigma(a_1)\sigma'(a_2)$ are also NEs. Furthermore, the expected payoff for every player is equal $v_p = \sum_a \sigma(a_1)\sigma(a_2)G_p(a) = \sum_a \sigma'(a_1)\sigma'(a_2)G_p(a) = \sum_a \sigma(a_1)\sigma'(a_2)G_p(a) = \sum_a \sigma'(a_1)\sigma(a_2)G_p(a)$. Similarly, if there is more than one NE is it also possible to combine any convex combination of the mixed strategies $\sigma''(a_1) = (1-\alpha)\sigma(a_1) + \alpha\sigma'(a_1)$, and the resulting mixed strategy will also be a NE. This means that in two-player zero-sum the space of mixed strategies is convex and is independent of the other player's mixed strategy. Therefore an equilibrium can be easily and uniquely selected using a strictly convex function such as maximum entropy. All these properties mean NE is a particularly suitable solution concept for two-player zero-sum games.

Gap Function

It is possible to define a metric of how close a joint is to an NE using the deviation gains or the best-response operator. First let us define the best-response gain.

$$\text{Gain: } \delta_p^\sigma = \max_{a'_p} \sum_a \sigma(a) A_p^{\text{CCE}}(a'_p, a) = G_p(\text{AnyBR}_p^\sigma, \sigma) \quad (2.73)$$

There are several definitions which are all only zero when $\sigma(a)$ is a NE.

$$\text{Exploitability: } \epsilon = \max_p \epsilon_p = \max_p \delta_p^\sigma \quad (2.74a)$$

$$\text{NEGap: } \delta^\sigma = \sum_p \epsilon_p = \sum_p \delta_p^\sigma \quad (2.74b)$$

The NEGap is sometimes also called NashConv (Lanctot et al., 2017), average deviation incentive (ADI) (Gemp et al., 2021), or Nikaido-Isoda (Nikaidô and Isoda, 1955). Intuitively this says that if there is strategy that a player can deviate to that has better payoff, it is not at a Nash equilibrium. These gap metrics are cheap to compute. Therefore verifying if a distribution is a Nash equilibrium is much cheaper than calculating a Nash equilibrium.

Equilibrium Selection

When a distribution is in equilibrium, no player has incentive to *unilaterally* deviate from it to achieve a better payoff. There can however be many equilibria in a game, choosing amongst these is known as the *equilibrium selection problem* (Harsanyi and Selten, 1988).

(C)CEs form a convex polytope of valid solutions which are defined by their linear inequality constraints. Convex functions can be used to uniquely select from this set. Multiple objectives have been proposed to select from the set of valid solutions including maximum entropy, $-\sum_a \sigma(a) \ln(\sigma(a))$ (ME(C)CE) (Ortiz et al., 2007). Other equilibrium selection objectives like Maximum Welfare, $\sum_p \sum_a \sigma(a) G_p(a)$ (MW(C)CE), are not convex and hence not always unique. For example, all zero-sum games have the same welfare but potentially have many equilibria.

For NEs it has also been suggested to use a maximum entropy criterion (MENE) (Balduzzi et al., 2018), which always exists and is unique in two-player, zero-sum settings. However ME is not unique in general for NEs because the set of NEs is not convex. Another strategy is to regularize the NE of the game with Shannon entropy resulting in the quantal response equilibrium (QRE) (McKelvey and Palfrey, 1995). There exists a continuum of QREs starting at the uniform distribution, finishing at the limiting logit equilibrium (LLE), which is unique for almost all games. Solvers (Gemp et al., 2021) can find LLEs, even in scenarios with stochastic payoffs.

Equilibrium-Invariant and Equilibrium-Symmetric Transforms

Certain transformations to payoffs, $G_p(a) \rightarrow \hat{G}_p(a)$, do not change the set of equilibria, $\sigma(a) \rightarrow \hat{\sigma}(a) = \sigma(a)$, (equilibrium-invariant transforms). The most common invariant transform (Morris and Ui, 2004; Moulin and Vial, 1978; Ostrovski, 2013) is the affine linear transform which consists of an offset over the other players' strategies and a positive scale.

Theorem 2.2.17 (Affine Transformation). *ϵ -NE, ϵ -WSNE, ϵ -CE, ϵ -WSCE, and ϵ -CCE are equivariant under affine transformations of each player's payoff and invariant when $\epsilon_p = 0$. Concretely, when*

$$G_p(a) \rightarrow \hat{G}_p(a) = s_p G_p(a) + b_p(a_{-p}), \quad (2.75)$$

an ϵ_p -equilibrium in the original game is an $s_p \epsilon_p$ -equilibrium in the transformed game: $\sigma(a) \rightarrow \hat{\sigma}(a) = \sigma(a)$ and $\epsilon_p \rightarrow \hat{\epsilon}_p = s_p \epsilon_p$. Where $b_p(a_{-p})$ is any offset, and s_p is any positive scalar.

Proof. Consider the effect of the transformations on the deviation gains.

$$\begin{aligned} A_p^{(\text{WS})\text{CE}}(a'_p, a''_p, a_{-p}) &\rightarrow s_p G_p(a'_p, a_{-p}) + b_p(a_{-p}) - s_p G_p(a''_p, a_{-p}) - b_p(a_{-p}) = s_p A_p^{(\text{WS})\text{CE}}(a'_p, a''_p, a_{-p}) \\ A_p^{(\text{WS})\text{CE}}(a'_p, a''_p, a) &\rightarrow \begin{cases} s_p A_p^{(\text{WS})\text{CE}}(a'_p, a''_p, a_{-p}) & a_p = a''_p \\ 0 & \text{otherwise} \end{cases} = s_p A_p^{(\text{WS})\text{CE}}(a'_p, a''_p, a) \end{aligned}$$

$$A_p^{\text{CCE}}(a'_p, a) \rightarrow s_p G_p(a'_p, a_{-p}) + b_p(a_{-p}) - s_p G_p(a) - b_p(a_{-p}) = s_p A_p^{\text{CCE}}(a'_p, a)$$

Equilibria are entirely defined by their inequality constraints. The affine transform only results in a s_p scale to the LHS of the inequality. If we apply the same positive scale to the RHS of the definition the inequality will still hold. Therefore an ϵ_p -equilibrium in the untransformed game will be an $s_p \epsilon_p$ -equilibrium in the transformed game. If $\epsilon_p = 0$ the equilibria will not change, and therefore an affine transformation will be invariant. \square

This transform can reduce the degrees of freedom in each player's payoff by $|\mathcal{A}_{-p}| + 1$ if $b_p(a_{-p})$ and s_p are chosen to be certain functions of the payoffs themselves. For example, a zero-mean-offset: $b_p(a_{-p}) = \frac{1}{|\mathcal{A}_p|} \sum_{a_p} G_p(a_p, a_{-p})$, and a unit norm scale, $s_p = \frac{1}{\|G_p(a)\|_2}$.

Other transforms, like strategy and player permutation, do not reduce the number of degrees of freedom, but do reduce the volume of games by exploiting symmetry in their definitions (equilibrium-symmetric transforms).

Theorem 2.2.18 (Strategy Permutation Equivalence). *ϵ -NE, ϵ -WSNE, ϵ -CE, ϵ -WSCE and ϵ -CCE equilibrium are equivalent under permutation of each player's strategies. Concretely, permuting each player's strategies in the game,*

$$G_p(a_p, a_{-p}) \rightarrow \hat{G}_p(a) = G_p(\tau_p(a_p), a_{-p}), \quad (2.76)$$

results in an equivalent permutation in the equilibria of the game, $\sigma(a) \rightarrow \hat{\sigma}(a) = \sigma(\tau_p(a_p), a_{-p})$ and $\epsilon_p \rightarrow \hat{\epsilon}_p = \epsilon_p$.

Proof. It is sufficient to prove that permutations of the strategies $a_p \rightarrow \tau(a_p)$, and deviation strategies $a'_p \rightarrow \tau(a'_p)$ are equivalent. Consider the definition of the ϵ -WSCE, ϵ -CE, and, ϵ -CCE:

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_{-p} | a''_p) A_p^{(\text{WS})\text{CE}}(a'_p, a''_p, a_{-p}) \leq \epsilon_p \quad \forall a''_p \in \mathcal{A}_p^+, a'_p \in \mathcal{A}_p \implies \max_{a'_p, a''_p} \sum_{a''_p \in \mathcal{A}_{-p}} \sigma(a_{-p} | a''_p) A_p^{(\text{WS})\text{CE}}(a'_p, a''_p, a_{-p}) \leq \epsilon_p \quad (2.77a)$$

$$\sum_{a \in \mathcal{A}} \sigma(a) A_p^{(\text{WS})\text{CE}}(a'_p, a''_p, a) \leq \epsilon_p \quad \forall a''_p, a'_p \in \mathcal{A}_p \implies \max_{a'_p, a''_p} \sum_{a \in \mathcal{A}} \sigma(a) A_p^{(\text{WS})\text{CE}}(a'_p, a''_p, a) \leq \epsilon_p \quad (2.77b)$$

$$\sum_{a \in \mathcal{A}} \sigma(a) A_p^{\text{CCE}}(a'_p, a) \leq \epsilon_p \quad \forall a'_p \in \mathcal{A}_p \implies \max_{a'_p} \sum_{a \in \mathcal{A}} \sigma(a) A_p^{\text{CCE}}(a'_p, a) \leq \epsilon_p \quad (2.77c)$$

Therefore the definitions do not depend on the order of a'_p and a''_p . It is straightforward to see that if $\sigma(a)$ is in equilibrium, then $\sigma(\tau_p(a_p), a_{-p})$ will be in equilibrium after permutation. \square

Note that all players can independently permute their strategies and the equivariance still holds. There are $\prod_p (|\mathcal{A}_p|!)$ such strategy permutation symmetries of a normal form game. Therefore the number of strategy permutation symmetries grows combinatorially with the number of strategies and exponentially with the number of players.

Theorem 2.2.19 (Player Permutation Equivalence). *ϵ -NE, ϵ -WSNE, ϵ -CE, ϵ -WSCE and ϵ -CCE equilibrium are equivariant under permutation of the players. Concretely, permuting the players in the game,*

$$G_p(a_p, a_{-p}) \rightarrow \hat{G}_p(a) = G_{\tau(p)}(a_{\tau(1)}, \dots, a_{\tau(N)}) \text{ and } \epsilon_p \rightarrow \hat{\epsilon}_p = \epsilon_{\tau(p)}, \quad (2.78)$$

	S	H		G	W		B	S		C	D	
S	4, 4	1, 3		G	−9, −9	1, 0	B	3, 2	0, 0	C	3, 3	1, 4
H	3, 1	2, 2		W	0, 1	0, 0	S	0, 0	2, 3	D	4, 1	2, 2
(a) Stag Hunt			(b) Traffic Lights			(c) Bach or Stravinsky			(d) Prisoner's Dilemma			
	H	T		C	F		A	B		S	W	
H	+1, −1	−1, +1		C	−2, 0	1, 0	A	1, 1	0, 0	S	$\frac{1}{2}, \frac{1}{2}$	1, 0
T	−1, +1	+1, −1		F	−2, 1	−1, 1	B	0, 0	$\frac{1}{2}, \frac{1}{2}$	W	0, 1	$\frac{1}{2}, \frac{1}{2}$
(e) Matching Pennies			(f) Red Dress			(g) Biased Coordination			(h) Dominant			

Table 2.4: Some canonical 2×2 normal form games. The row player chooses between labeled strategies from the left and receives the first payoff of the pair, while the column player chooses between labeled strategies from the top and receives the second payoff of the pair.

results in an equivalent permutation in the equilibria of the game, $\sigma(a_1, \dots, a_N) \rightarrow \hat{\sigma}(a) = \sigma(a_{\tau(1)}, \dots, a_{\tau(N)})$.

Proof. Consider the definition of the ϵ -WSCE, ϵ -CE, and, ϵ -CCE:

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_{-p} | a_p'') A_p^{(\text{WS})\text{CE}}(a_p', a_p'', a_{-p}) \leq \epsilon_p \quad \forall p \in [1, N] \implies \max_p \left[\sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_{-p} | a_p'') A_p^{(\text{WS})\text{CE}}(a_p', a_p'', a_{-p}) - \epsilon_p \right] \leq 0 \quad (2.79a)$$

$$\sum_{a \in \mathcal{A}} \sigma(a) A_p^{(\text{WS})\text{CE}}(a_p', a_p'', a) \leq \epsilon_p \quad \forall p \in [1, N] \implies \max_p \left[\sum_{a \in \mathcal{A}} \sigma(a) A_p^{(\text{WS})\text{CE}}(a_p', a_p'', a) - \epsilon_p \right] \leq 0 \quad (2.79b)$$

$$\sum_{a \in \mathcal{A}} \sigma(a) A_p^{\text{CCE}}(a_p', a) \leq \epsilon_p \quad \forall p \in [1, N] \implies \max_p \left[\sum_{a \in \mathcal{A}} \sigma(a) A_p^{\text{CCE}}(a_p', a) - \epsilon_p \right] \leq 0 \quad (2.79c)$$

Therefore when the order of strategies is transposed, an identical transposition in the joint will ensure that it is also an equilibrium in the transformed game. \square

There are $N!$ such player permutation symmetries of a normal form game. Therefore the number of player permutation symmetries grows combinatorially with the number of players.

2.2.2.4 2×2 Normal-form Games

To intuitively explain some of the concepts discussed, this section examines the simplest class of normal-form games: games with two-players, each with two-strategies. These games are sometimes referred to as 2×2 games. A set of common 2×2 games is given in Table 2.4.

First, consider the space of joint strategies, $\sigma(a)$. The standard constraints on a probability distribution apply: probabilities are nonnegative, $\sigma(a) \geq 0 \forall a \in \mathcal{A}$, and sum to unity, $\sum_{a \in \mathcal{A}} \sigma(a) = 1$. The unity sum constraint means that a distribution over the four strategies of a 2×2 game, which can be denoted with a flat vector $\sigma = [\sigma(a^{AA}), \sigma(a^{AB}), \sigma(a^{BA}), \sigma(a^{BB})]$, where $a^{IJ} = (a_1^I, a_2^J)$, can be expressed with only three variables because one is redundant given the rest (e.g. $\sigma(a^{BB}) = 1 - \sigma(a^{AA}) - \sigma(a^{AB}) - \sigma(a^{BA})$). This means it is possible to visualize a joint distribution with four components in only three dimensions, by ignoring the space of “distributions” that do not sum to unity. This is achieved by specifying four vertices of a tetrahedron (a three dimensional object). Points in the simplex are then described in terms of mixtures of these four vertices (known as a barycentric coordinate system (Möbius, 1827)). Barycentric coordinates, σ , can be converted to Cartesian coordinates, x , via a linear transform, $x = T\sigma$. The columns of T are

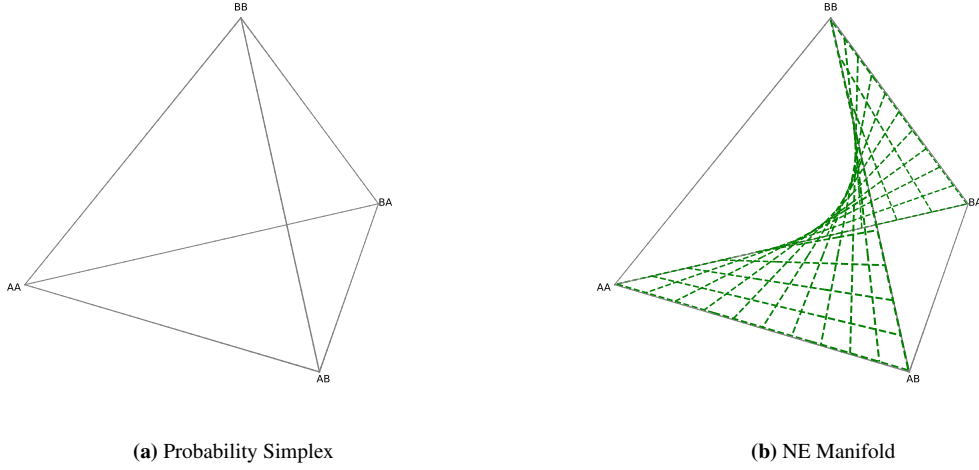


Figure 2.3: Visualization of the space of valid joint distribution, $\sigma(a_1, a_2)$, (tetrahedron) and valid factorizable joint distributions, $\sigma(a_1, a_2) = \sigma(a_1)\sigma(a_2)$, (manifold). The vertices of the tetrahedron correspond to pure joint strategies. The interior of the tetrahedron corresponds to mixed joint strategies.

points of a regular tetrahedron. There are many ways to choose points of a tetrahedron, one is shown in Equation (2.80).

$$x = T\sigma \quad T = \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & 0 & 0 \\ -\frac{\sqrt{3}}{6} & -\frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{6} & 0 \\ -\frac{\sqrt{6}}{12} & -\frac{\sqrt{6}}{12} & -\frac{\sqrt{6}}{12} & \frac{\sqrt{6}}{4} \end{bmatrix} \quad (2.80)$$

The nonnegative inequality constraints geometrically correspond to normal vectors in the equation of a plane (e.g. $[1, 0, 0, 0]\sigma^T \geq 0$). These planes split the space into two halves: those that are feasible distributions and those that are infeasible distributions. Together these four inequality constraints result in a convex polytope with four faces, specifically a regular tetrahedron, when visualized in three dimensions. $\sigma(a_1, a_2)$ corresponds to a full joint distribution. A subset of joints that factorize into their marginals, $\sigma(a_1, a_2) = \sigma(a_1)\sigma(a_2)$, is worth highlighting because of its relationship to NEs which by definition have to factorize. Factorizable joints result in a manifold within the tetrahedron. The tetrahedron and factorizable manifold are shown in Figure 2.3. Before exploring equilibria, note a peculiar property: CEs and CCEs are equivalent in 2×2 games.

Theorem 2.2.20 (Two-Strategy (C)CE equivalence). *For n -player games with two strategies, ϵ -CCEs and ϵ -CEs are equivalent.*

Proof. When $|\mathcal{A}_p| = 2$, where $a_p = \{a_p^A, a_p^B\}$, the number of constraints per player is $|\mathcal{A}_p| = 2$ for CCEs and $|\mathcal{A}_p|(|\mathcal{A}_p| - 1) = 2$ for CEs. By inspection, and a change of variables, we can see that the constraints are equal.

$$\begin{aligned} A_p^{\text{CCE}}(a_p^A, a) &= G_p(a_p^A, a_{-p}) - G_p(a_p, a_{-p}) \\ &= \begin{cases} G_p(a_p^A, a_{-p}) - G_p(a_p^B, a_{-p}) & a_p = a_p^B \\ 0 & \text{otherwise} \end{cases} = A_p^{\text{CE}}(a_p^A, a_p^B, a) \end{aligned} \quad (2.81a)$$

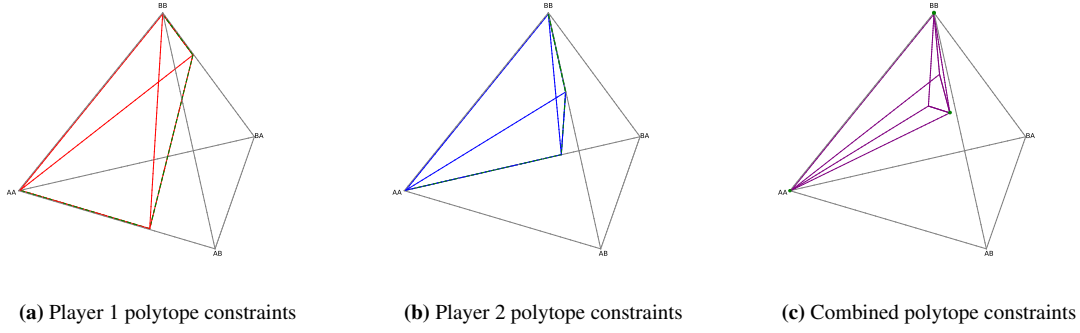


Figure 2.4: (C)CE polytope (purple) and NE points (green) of the biased coordination game. A breakdown of the final (C)CE polytope is given by considering player 1's (red) and player 2's (blue) deviations. For this game, a volume of equilibria are valid (C)CEs. NEs are where the set of (C)CEs intersects with the factorizable manifold which include pure a^{AA} , pure a^{BB} and a full-support mixed NE.

$$\begin{aligned}
 A_p^{\text{CCE}}(a_p'^B, a) &= G_p(a_p'^B, a_{-p}) - G_p(a_p, a_{-p}) \\
 &= \begin{cases} G_p(a_p'^B, a_{-p}) - G_p(a_p''^A, a_{-p}) & a_p = a_p''^A \\ 0 & \text{otherwise} \end{cases} = A^{\text{CE}}(a_p'^B, a_p''^A, a) \quad (2.81b)
 \end{aligned}$$

□

In the context of 2×2 games we will therefore use the notation $A_p^{(\text{C})\text{CE}}(a_p', a)$ to describe the deviation gains for each player and $A^{(\text{C})\text{CE}}$ to describe the constraint matrix with shape $[N|\mathcal{A}_p| = 4, |\mathcal{A}| = 4]$.

Let us study a biased coordination game (Table 2.4g), where players have to coordinate on the best outcome (a_1^A, a_2^A) or the second best outcome (a_1^B, a_2^B) in order to maximize their return. The deviation gains (Equation (2.82)) have 4 columns corresponding to the flattened joint strategies, $(a^{AA}, a^{AB}, a^{BA}, a^{BB})$, and 4 rows corresponding to the deviations, $(a_1^A, a_1^B, a_2^A, a_2^B)$. Player 1 is shown in red and player 2 in blue.

$$A^{(\text{C})\text{CE}} = \begin{bmatrix} G_1(a_1^A, a) - G_1(a) \\ G_1(a_1^B, a) - G_1(a) \\ G_2(a_2^A, a) - G_2(a) \\ G_2(a_2^B, a) - G_2(a) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & -\frac{1}{2} \\ -1 & \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 & -\frac{1}{2} \\ -1 & 0 & \frac{1}{2} & 0 \end{bmatrix} \quad (2.82)$$

Each row of this matrix geometrically corresponds to a normal vector of a plane which splits the joint strategy space in half: one side of the plane are feasible equilibria and the other side are infeasible. Together, and combined with the nonnegative distribution inequalities, these half-spaces result in a convex polytope (Figure 2.4). The vertices of the polytope are given by Equation (2.83). Notice that the probability mass of the mixed NE skews to the weaker pure NE: a^{BB} . This is a property of NE that ensures that players are indifferent between all strategies in the support.

$$\Sigma = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{1}{9} & \frac{2}{9} & \frac{2}{9} & \frac{4}{9} \\ \frac{1}{7} & \frac{2}{7} & 0 & \frac{4}{7} \\ \frac{1}{7} & 0 & \frac{2}{7} & \frac{4}{7} \end{bmatrix} \quad (2.83)$$

Now consider how the polytope changes when the approximation parameter ϵ is altered. Geometrically

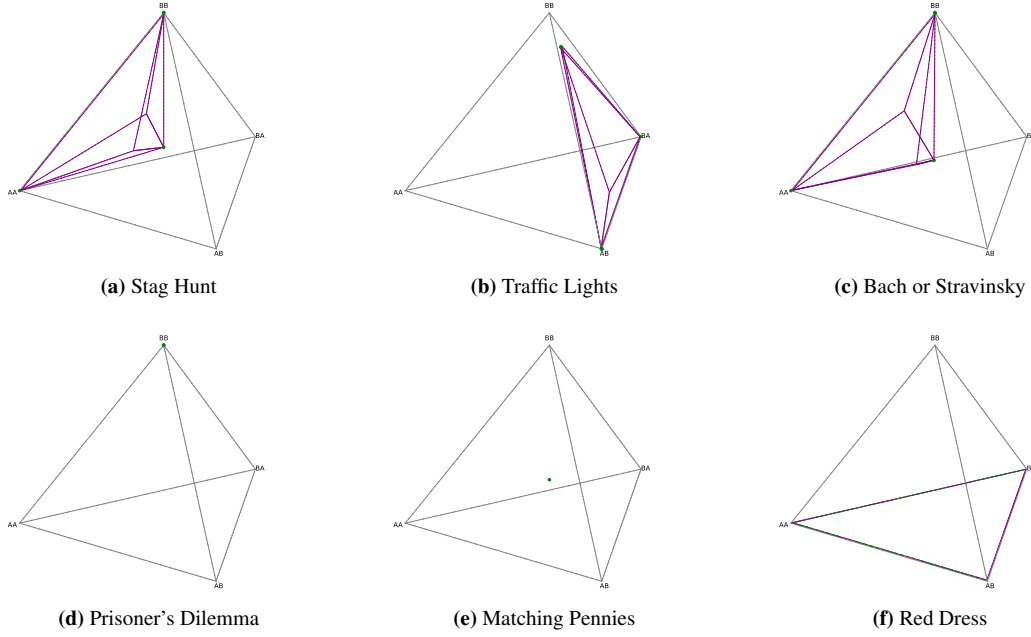


Figure 2.5: Equilibria of canonical 2×2 games. (C)CE polytope is solid purple line. NEs are green dashed lines or points.

the ϵ term in the inequality constraint, $A^{(C)CE}\sigma \leq \epsilon$, corresponds to the offset of the plane. Therefore increasing ϵ_p will expand the volume of the polytope and decreasing ϵ_p will reduce the volume of the polytope (if possible). For some games it is possible to have negative ϵ_p . For WSCEs the region of the polytope changes differently.

Figure 2.5 shows the equilibria of other games including Stag Hunt (Table 2.4a), Traffic Lights (Table 2.4b), Bach or Stravinsky (Table 2.4c), Prisoner's Dilemma (Table 2.4d), Matching Pennies (Table 2.4e), and Red Dress (Table 2.4f).

2.2.2.5 Algorithms

The definition of (WS)(C)CE solutions only comprise of linear inequality constraints. Any objective can be used to select amongst them, however linear and strictly convex objectives are most common. Libraries such as CVXPY (Agrawal et al., 2018; Diamond and Boyd, 2016) use solvers like GLOP (Perron and Furnon), ECOS (Domahidi et al., 2013), OSQP (Stellato et al., 2020), and MOSEK (ApS, 2019), to solve these problems. Finding one (WS)(C)CE is polynomial in complexity (Papadimitriou and Roughgarden, 2008). Finding a specific one depends on the objective and will be harder.

(WS)NE are hard to compute because they have nonlinear constraints and require finding a fixed point (Papadimitriou and Roughgarden, 2008). In general, the complexity was shown to be PPAD-complete (Chen and Deng, 2006; Daskalakis et al., 2006). However, many algorithms exist for finding equilibria in two-player zero-sum games which are easier to solve.

Iterated Elimination of Dominated Strategies

A strategy a_p'' is *dominated* if another strategy a_p' yields a strictly higher payoff, regardless of the opponents' choices: $\exists a_p' \in \mathcal{A}_p : G_p(a_p', a_{-p}) > G_p(a_p'', a_{-p}) \forall a_{-p} \in \mathcal{A}_{-p}$. Iteratively eliminating dominated strategies for each player leads to a subset of strategies containing all Nash equilibria (NEs) and correlated equilibria (CEs). A pure NE emerges when only one strategy remains for each player. Otherwise, further refinement is necessary to obtain a mixed-strategy NE or CE. However, this method cannot identify

Algorithm 2.2 Double Oracle.**Require:** A two-player, constant-sum, payoff matrix, $G(a_1, a_2)$.**Require:** Nonempty starting set of actions, $\emptyset \subset \mathcal{B}_1 \subseteq \mathcal{A}_1$ and $\emptyset \subset \mathcal{B}_2 \subseteq \mathcal{A}_2$.

```

1: function DOUBLEORACLE( $G(a_1, a_2), \mathcal{B}_1, \mathcal{B}_2$ )
2:   while True do
3:      $\tilde{G}(b_1, b_2) \leftarrow G(b_1, b_2) \quad \forall b_1 \in \mathcal{B}_1, b_2 \in \mathcal{B}_2$ 
4:      $\sigma_1(b_1), \sigma_2(b_2) \leftarrow \text{NE}(\tilde{G})$ 
5:      $r_1 \leftarrow \tilde{G}_1(\sigma_1(b_1), \sigma_2(b_2))$ 
6:      $r_2 \leftarrow \tilde{G}_2(\sigma_1(b_1), \sigma_2(b_2))$ 
7:      $a_1 \leftarrow \arg \max_{a_1} \sum_{b_2} \sigma_2(b_2) G_1(a_1, b_2)$  ▷ Best-response against mixture  $\sigma_2(b_2)$ 
8:      $a_2 \leftarrow \arg \max_{a_2} \sum_{b_1} \sigma_1(b_1) G_2(b_1, a_2)$  ▷ Best-response against mixture  $\sigma_1(b_1)$ 
9:     if  $G_1(a_1, \sigma_2(b_2)) \leq r_1 \wedge G_2(\sigma_1(b_1), a_2) \leq r_2$  then
10:      break
11:      $\mathcal{B}_1 \leftarrow \mathcal{B}_1 \cup \{a_1\}$ 
12:      $\mathcal{B}_2 \leftarrow \mathcal{B}_2 \cup \{a_2\}$ 
13:   return  $\sigma_1(b_1), \sigma_2(b_2)$ 

```

CCEs, which may include dominated strategies. Often employed as a low-cost preprocessing step, this algorithm lacks favorable worst-case bounds, requiring $\max_p |A_p|$ iterations. Nonetheless, it tends to converge quickly in practice. While it finds only one NE, NEs are interchangeable in two-player, zero-sum games.

Minimax Optimization

For two-player zero-sum games the NE objective can be formulated as a minimax optimization which can be solved efficiently using linear programming.

$$\sigma^*(a_1) = \min_{a_2} \arg \max_{\sigma(a_1)} \sum_{a_1} G_1(a_1, a_2) \sigma(a_2) \quad (2.84a)$$

$$\sigma^*(a_2) = \max_{a_1} \arg \min_{\sigma(a_2)} \sum_{a_2} \sigma(a_1) G_1(a_1, a_2) \quad (2.84b)$$

Fictitious Play

In fictitious play (Brown, 1951), a player's strategy is the best-response to the average history of other players' actions. The average over all player's histories is known to converge to a Nash equilibrium in two-player zero-sum games.

Double oracle

Double Oracle (DO) (McMahan et al., 2003) (Algorithm 2.2) identifies Nash equilibrium (NE) solutions for large two-player, zero-sum normal-form games. It iteratively solves subgames of increasing size. If both players fail to improve their payoffs at any iteration, the NE of the subgame is also an NE of the full game. DO employs an NE solver as a subroutine at each iteration. While this may appear circular, it becomes valuable when solving large extensive-form games and forms the foundation of the PSRO algorithm.

2.2.3 Markov Games

Markov games (Shapley, 1953), also known as stochastic games, are a generalization of Markov decision processes (MDPs) and repeated games. As a subset of extensive-form games, Markov games differ from MDPs in several ways: a) each player has a distinct reward function, b) players choose joint strategies simultaneously, and c) a joint transition function governs state transitions. Non-terminating Markov games may employ a discount function to account for myopic agent behavior. Perfect information (full observability) or *common observation* is required. *Zero-sum Markov games*, a well-studied subclass, assume a

zero-sum payoff structure. *Single-controller Markov games* feature transition functions dependent on the actions of a single agent, while payoffs still depend on joint actions. For historical context on Markov games, refer to [Solan and Vieille \(2015\)](#). While related, *multistage games* differ in that their states are not necessarily Markovian.

2.2.3.1 Equilibrium Solution Concepts

Equilibria for Markov games no longer concern just a single normal-form stage game, but many normal-form games, where actions taken affect the payoffs of joint decision nodes earlier in the game tree.

Markov Perfect Equilibrium

A *Markov perfect equilibrium* (MPE) is a subgame-perfect equilibrium for perfect-information, simultaneous move extensive form games (i.e. Markov games). In equilibrium, for every state (even those unreachable under joint policies), every subgame is in equilibrium. This can be defined for NEs, CE, or CCEs. Concretely, consider the joint-action-value function. If the joint action distribution, $\pi[s, a]$, is in normal-form equilibrium at all states, it is an MPE. MPE is known to exist for every terminating Markov game, by backward induction. For continuing Markov games, an equilibrium always exists if the rewards are discounted. Such an equilibrium, $\pi[s, a]$, can be implemented in a decentralized fashion if the equilibrium is an NE, because it factorizes. When using (C)CEs, a correlation device is required at every timestep. The extensive-form section will show how it is possible to also define normal-form equilibria for Markov games. These solutions will only require a correlation device to communicate a policy, once, before any actions have been executed.

2.2.3.2 Algorithms

Algorithms used to compute equilibria in Markov games can be borrowed from reinforcement learning.

Model-Based Value-Based Methods

All the tools of MDP Generalized Policy Iteration can be utilized in the multiagent setting. If the dynamics of the game are known, given a *joint policy*, $\pi[s, a_1, \dots, a_N] = \pi[s, a]$, one could use the same policy evaluation techniques discussed previously to calculate the state value function, $v_p^\pi[s]$, and action value function, $q_p^\pi[s, a]$, for each player. Instead of having a single value per state, there is now a value for each player. This means that in order to conduct policy improvement, we need to consider the value of possible competing players: the problem is now multi-objective.

$$v_p^\pi[s] = \sum_a \pi[s, a] q_p^\pi[s, a] \quad \forall p \in [1, N] \quad (2.85a)$$

$$q_p^\pi[s, a] = \sum_{s'} T[s, a, s'] (R_p[s, a, s'] + \gamma v_p^\pi[s']) \quad \forall p \in [1, N] \quad (2.85b)$$

And these have similar recurrent relationships:

$$v_p^\pi[s] = \sum_a \pi[s, a] \sum_{s'} T[s, a, s'] (R[s, a, s'] + \gamma v_p^\pi[s']) \quad \forall p \in [1, N] \quad (2.86a)$$

$$q_p^\pi[s, a] = \sum_{s'} T[s, a, s'] \left(R_p[s, a, s'] + \gamma \sum_{a'} \pi[s', a'] q_p^\pi[s', a'] \right) \quad \forall p \in [1, N] \quad (2.86b)$$

One could proceed by converting this problem to a single objective optimization problem by collapsing the all the players' value functions into a single value function. One common way of doing this is by summing over all players returns, sometimes called the welfare, $v^\pi[s] = \sum_p v_p^\pi[s]$. Thereafter we could employ standard greedy MDP policy improvement techniques to converge to a unique value function. However, this is unsatisfactory for a number of reasons:

1. The scales of the rewards of each player may be mismatched, so combining them is unnatural.
2. The resulting payoffs of the agents may be “unfair” in some sense.
3. There might be an incentive to deviate from such a greedy policy.

Indeed any such arbitrary rule of squashing a multi-objective problem into a single-objective one may have these same issues. There are some edge cases in which squashing to a single-objective function makes sense. For example, in common payoff games, all players will have the same value function $q^\pi[s, a] = q_1^\pi[s, a] = \dots = q_N^\pi[s, a]$ (sometimes called *friend-Q* (Littman, 2001)). In two-player, constant-sum games one can consider the max-min-Q $v_1^\pi[s] = \max_{\sigma(a_1)} \min_{a_2} \sum_{a_1} \sigma[a_1] q_1^\pi[s, a_1, a_2] = -v_2^\pi[s]$ (sometimes called *foe-Q* (Littman, 2001)).

In general, we may wish to utilize equilibrium solution concepts as the policy improvement operator, $f : v[p, s, a] \mapsto \sigma[s, a]$. In fact, the two previous squashing functions arise from the properties of NE in common-payoff and two-player, constant-sum respectively. Instead of the policy improvement condition being based on improving the value, it is based on improving the equilibrium gap.

$$\sum_a \pi_{t+1}[s, a] q_p^{\pi_t}[s, a] \geq \sum_a \pi_t[s, a] q_p^{\pi_t}[s, a] \quad \forall s, p \quad (2.87)$$

The policy improvement operator can be any unique equilibrium concept including NEs, CEs, or CCEs.

$$\pi_{t+1}[s, a] = \text{UniqueEquilibrium}(q_p^{\pi_t}[s, a]) \quad (2.88)$$

If the game is terminating, by backward induction the resulting policy will converge to a Markov perfect equilibrium. If the game is non-terminating, there is no known iterative methods of converging to an NE, CE, or CCE in general. The solutions are known to cycle around a set of so-called cyclic-equilibria (Zinkevich et al., 2005).

Model-Free Value-Based Methods

Of course, we need not know the dynamics of the environment. Model-free MDP methods can be used to learn the value functions in Markov games. When the game is common-payoff, the Q-Learning algorithm is called *friend-Q* (Littman, 2001), when it is two-player constant-sum it is called *foe-Q* (Littman, 2001). When using NE as the policy improvement operator, it can be known as *Nash-Q* (Hu and Wellman, 1998). When using linear CEs, it is known as *CE-Q* (Greenwald and Hall, 2003). A drawback of foe-Q, Nash-Q, and CE-Q is that they require relatively expensive solvers to find a new policy every time the action-values are updated. Furthermore, Nash-Q and CE-Q, suffer from unresolved equilibrium selection issues.

2.2.4 Extensive-Form Games

An extensive-form game (Kuhn and Tucker, 1957) is a temporally extended game where players can take actions sequentially in arbitrary (but defined) order. Players choose between actions at each decision node (equivalent to state in RL terminology) they encounter. As a result, progressing through the game is akin to traversing down a *game tree*. There is no discounting and the payoff is achieved at the leaf nodes of the tree. Note that having payoffs only at leaf nodes is fully general because any rewards at stages of the game could be pushed into the leaf nodes without loss of generality. These games can be imperfect information (equivalent to partially observable in RL terminology), as such some states cannot be differentiated from one another and are called *information states* (equivalent to observations in RL terminology). Furthermore, other players actions are not necessarily observed, and may not necessarily be deduced from an information state. An extensive form game may be stochastic. In such a scenario, sometimes the world is modelled as player 0 which picks between random outcomes according to some distribution. Extensive form games are the most general game formulation. Markov games can be represented as imperfect information extensive-

form games by arbitrarily ordering players actions at each stage, hiding the action that was taken at by each player, and only revealing the action to player after they have all selected an action. And, of course, normal-form games are a subset of Markov games.

There are many examples of extensive form games. Implementations for some games commonly benchmarked in the literature are available in OpenSpiel (Lanctot et al., 2019). *Kuhn Poker*, is a very simplified n-player, zero-sum, imperfect information version of poker. The original two-player game is described in (Kuhn, 1950). An n-player extension is described in (Lanctot, 2014). Additional information about the game (such as equilibrium) can be found in (Hoeft et al., 2005). *Sheriff* (Farina et al., 2019c) is a negotiation game based on a simplified version of the Sheriff of Nottingham board game. *Trade Comm* (Sokota et al., 2021) is a simple two-player common-payoff trading game.

2.2.4.1 Normal-Form Representations of Extensive-Form Games

All extensive-form games can also be represented (albeit inefficiently) as normal-form games, and therefore normal-form solution concepts can also be applied to this representation. The graph structure of the game is lost in this transformation. Therefore subgame perfect equilibrium concepts cannot be applied to the normal-form.

Normal-Form

The normal-form can be obtained by enumerating every possible deterministic action at every information state in the game. Therefore, if there are $|\mathcal{A}_p|$ possible actions at each information state and $|\mathcal{I}_p|$ information states, for each player there are $|\mathcal{A}_p|^{|\mathcal{I}_p|}$ normal-form strategies. An n-player normal-form game can be constructed using this approach. However, the normal-form representation quickly becomes intractably large, even for moderately sized extensive-form games.

Reduced Normal-Form

This can be partially mitigated by considering the *reduced-normal-form* of an extensive-form game. It may be the case that some information states become unreachable after an action has been taken earlier in the game tree. Therefore, any action specified at such information states is irrelevant and need not be considered. In reduced-normal-form, we therefore only consider strategies that are unique up to reachable information states. Using this representation there may be as little as $|\mathcal{A}_p|^T$ distinct strategies, where T is the number of time steps in the environment. The actions at unreachable information states are either left undefined or defined uniquely using an arbitrary rule, such as a uniform distribution.

Meta Normal-Form

Meta games are normal-form games induced from extensive form games by considering the payoffs between a set of (usually stochastic) policies. Because there are infinite stochastic policies in any extensive form game, only a subset of policies are considered. Sometimes these formulations are called empirical games because the payoffs between policies are evaluated by sampling outcomes between policies. Therefore the number of strategies for each player in a meta-game is equal to the number of policies being considered. The advantage of considering such games is that it allows us to leverage simpler normal-form solution concepts to make progress in extensive form games. Methodologies like Empirical game-theoretical analysis (EGTA) (Walsh et al., 2002; Wellman, 2006) make use to the meta normal-form to analyse intractably large temporal multiagent-systems. Equilibrium solvers like Double Oracle (McMahan et al., 2003) PSRO (Lanctot et al., 2017) can be built on this representation to find equilibria for extensive-form games.

Agent-Normal-Form

In an extensive-form game with perfect recall, for each player's information state create a new "agent" player in a normal-form game which has strategies equal to the number of actions available in the information state. This results in the *agent-normal-form*. This is distinct from the standard normal-form representation of an extensive-form game in several ways. The agent-normal-form has many more players,

$N = \sum_p |\mathcal{I}_p|$, than the original extensive-form game but few strategies, $|\mathcal{A}_{I_p}|$. Equilibria in agent-normal-form correspond to perfect equilibrium in extensive-form games.

2.2.4.2 Equilibrium Solution Concepts

The equilibrium solution concepts of normal-form games extend to extensive-form games. However, there are also additional solution concepts, which require more extensive coordination.

Extensive Form Correlated Equilibrium

Extensive form coarse correlated equilibrium (EFCE) (von Stengel and Forges, 2008) is an equilibrium concept for temporally extended games. It requires a coordination device at every timestep to recommend actions to the players. The coordination device samples an entire deterministic joint policy from a publicly known distribution over joint policies. The device then, action by action, secretly recommends strategies to each player. Players are free to deviate after receiving the signal, but if they do so they receive no further recommendations. If no players have incentive to deviate, the distribution is an EFCE.

Extensive Form Coarse Correlated Equilibrium

The extensive form coarse correlated equilibrium (EFCCE) (Farina et al., 2019b), is similar to the EFCE, but players must decide whether to deviate before receiving each action recommendation. The relationship between NFNE \subseteq NFCE \subseteq EFCE \subseteq EFCCE \subseteq NFCCE.

2.2.4.3 Extensive Form Operators

Similar to normal-form games, operators that learning algorithms utilize in their inner-loops are defined in this section.

Mixture Policies

Care should be taken when mixing policies in extensive-form. The method of mixing policies is to sample a policy from a set, $\Pi_p = \{\pi_p^1, \dots\}$, according to a distribution, σ , and then play that policy until environment termination. The space of mixed policies is convex and the payoffs a mixture of policies receives is linear against a fixed opponent. These are useful properties that can be exploited in a number of algorithms.

It is possible to squash that behaviour into a single policy that we denote π^σ . The policy can be calculated by inspecting the reach probabilities $r^\pi(s)$ under each policy, such that the probability of being in initial states $\sum_{s \in \mathcal{S}_0} r^\pi(s_0) = 1$, intermediate states $\sum_{s_t \in \mathcal{S}_t} r^\pi(s_t) = 1$, and terminal states $\sum_{s_T \in \mathcal{S}_T} r^\pi(s_T) = 1$, is always equal to one. These can be computed using the recursive formula starting from the initial states $\mathcal{S}_{t+1} = t(s_t, a)$. The mixture policy is unique, but many possible policies may mix into the same policy. This procedure is more complex when the policies are parameterized by a function approximator. In this scenario one would have to use policy distillation (Rusu et al., 2016) to train a new policy.

Best-Response

The best-response (BR) operator returns a set of policies that maximizes return against a given set of other player's policies, $\{\pi_p\} = \text{BR}_p(\pi_{-p}) = \text{BR}_p(\pi_1, \dots, \pi_{p-1}, \pi_{p+1}, \dots, \pi_n) = \arg \max_{\pi_p} G_p(\pi_1, \dots, \pi_n)$. A best-response always exists, but there can be multiple best-response policies that maximize the return, and therefore the best-response operator does not map to a single unique policy.

We can define a unique best-response operator by selecting for the policy that is closest to another target policy $\tilde{\pi}$ by using the minimum KL divergence $\sum_a \pi(a) \ln \left(\frac{\pi(a)}{\tilde{\pi}(a)} \right)$. The target policy could be chosen in a number of ways, for example one which is closest to human play, or a previously found policy. In the absence of a good target policy one may choose to find the policy closest to the uniform policy. In this case the objective becomes equivalent to finding the maximum entropy policy which is a policy that makes least assumptions, is most difficult for opponents to predict, and maximizes exploration.

Another complication of the BR operator is that it requires searching over all of policy space to find. This space is linear over the mixture of actions and greedy algorithms can be used to solve it. Backward

induction can be used to solve for the BRs by searching over the game tree starting from the terminal leaves. In non-terminating games RL can be used to find BRs. Value iteration, policy iteration, policy gradient and Q-Learning are all suitable BR operators.

2.2.4.4 Algorithms

Fictitious Play

Fictitious play ([Brown, 1951](#); [Heinrich et al., 2015](#)) can also be used to solve extensive-form games. It can also be employed with function approximation ([Heinrich and Silver, 2016](#)).

Empirical Game Theoretic Analysis

Empirical Game Theoretic Analysis (EGTA) ([Walsh et al., 2002](#); [Wellman, 2006](#)) is an approach to applying Double Oracle like algorithms to extensive form games.

Policy-Space Response Oracles

PSRO ([Lanctot et al., 2017](#)) is an agent training framework for two-player zero-sum extensive-form games. PSRO is a generalisation of several algorithms that are known to converge to normal-form Nash equilibria (NFNE). It outputs a set of policies for each player and a distribution over those policies. The distribution that the algorithm uses to compute best-responses is what parameterizes the algorithm. PSRO uses RL as the best-response operator.

Chapter 3

Normal-Form Game Metric Spaces and Embeddings

Equilibrium solution concepts of normal-form games, such as Nash equilibria, correlated equilibria, and coarse correlated equilibria, describe the joint strategy profiles from which no player has incentive to unilaterally deviate. They are widely studied in game theory, economics, and multiagent systems. Equilibrium concepts are invariant under certain transforms of the payoffs. This chapter defines an equilibrium-inspired distance metric for the space of all normal-form games and uncovers a distance-preserving equilibrium-invariant embedding. Furthermore, an additional transform, which defines a better-response-invariant distance metric and embedding, is proposed. The contents of this chapter is part of published work ([Marris et al., 2023](#)).

3.1 Introduction

Equilibrium solutions to normal-form games, such as Nash equilibrium (NE) ([Nash, 1951](#)), correlated equilibrium (CE) ([Aumann, 1974](#)), and coarse correlated equilibrium (CCE) ([Hannan, 1957](#); [Moulin and Vial, 1978](#); [Young, 2004](#)) are ubiquitously used to model the strategic behaviour of rational utility maximizing players in games. In some classes of games, such as two-player zero-sum games, the Nash equilibrium solution concept is considered fundamental ([von Neumann and Morgenstern, 1947](#)), because it is unexploitable and interchangeable in this class. CEs and CCEs are important in n-player general-sum games, which may require coordination facilitated by a correlation device. The set of equilibria in a game is invariant to certain transforms of the payoffs ([Morris and Ui, 2004](#)). The most well known is the affine transform (offset and positive scale of each player's payoff). Such transforms are called *equilibrium-invariant*. In addition, there are symmetries in payoffs which result in equivalent symmetries in the set of equilibria (for example, the order of strategies or players). Transforms over symmetries are called *equilibrium-symmetric*. Finally, there is a weaker notion of better-response invariance, where transforms do not change a player's preference order over responses to other player's joint strategies. These transforms are called *better-response-invariant*. This chapter studies transforms to produce metric spaces and embeddings over n-player general-sum normal-form games.

Let $\mathcal{G} = \mathcal{G}_1 \times \dots \times \mathcal{G}_N$ be the space of games with a particular number of players, $p \in [1, N]$, and strategies, $|\mathcal{A}_p|$. \mathcal{G}_p is the space of payoffs for player p , and equivalent the space of tensors with real elements and of a particular shape, $\mathbb{R}^{|\mathcal{A}_1| \times \dots \times |\mathcal{A}_N|}$. A normal-form game (or equivalently, its payoffs) is an elements of this set $G \in \mathcal{G}$. Equilibria solution concepts can be parameterized with an approximation parameter, ϵ , which describes a larger set of approximate equilibria where not deviating costs at most ϵ to the players. Since the approximation parameter influences with space of acceptable equilibria, in the context of invariant transforms, it is advantageous to consider the approximation parameter as part of the

game definition. Therefore, the approximation parameter (see Equation (2.46)) is included for each player $\mathcal{E} = \mathcal{E}_1 \times \dots \times \mathcal{E}_N$, which is in the real vector space, \mathbb{R}^N , is included in the space of games, defining a new tuple space $(\mathcal{G}, \mathcal{E})$, that is called the approximate game space. It is possible to define a distance metric between two approximate games $d((G^A, \epsilon^A), (G^B, \epsilon^B))$ such that the space of approximate normal-form games is a metric space. Let $\sigma \in \Delta^{|\mathcal{A}|-1}$ be the probability simplex. Let $\sigma^* \in (\text{WS})(\text{C})\text{CE}(G, \epsilon) \subseteq \Delta^{|\mathcal{A}|-1}$ be the subset of joints that are in equilibrium according to either Equation (2.46), (2.54), (2.59) or (2.64). Most commonly the approximation parameter is defined $\epsilon = 0$ and therefore the additional notation around approximate games can be dropped.

Studying the space of normal-form games is important for understanding the a) structure of games, b) computing their equilibria, and c) when applying machine learning techniques to normal-form games. Concretely, a well designed embedding reduces the redundant “degrees of freedom” in a game representation. For example, a distribution over m elements could be represented as logits in \mathbb{R}^m , or equivalently as $m - 1$ probabilities, Δ^{m-1} , where the final probability can be deduced from $p_m = 1 - \sum_{i=1}^{m-1} p_i$, which is a reduction the redundant degrees of freedom by 1. For normal-form games we can reduce the degrees of freedom in a similar way.

To motivate, consider an application of designing a neural network architecture that takes normal-form payoffs as input and outputs an equilibria. A neural network could learn the equilibrium invariances for itself, but only approximately. Building the invariance into the training of the network would ensure perfect equilibrium-invariance and free up resources for the network to learn the non-invariant structure in the problem. Additionally, training such a network would require a dataset. But how would one sample games sensibly and without bias over $\mathbb{R}^{N \times |\mathcal{A}_1| \times \dots \times |\mathcal{A}_N|}$? It turns out that a clever embedding enables easy and uniform sampling over the space of games containing all possible equilibria.

3.2 Equilibrium-Invariant Embedding

This chapter is concerned with studying game transforms, $(G, \epsilon) \rightarrow (\hat{G}, \hat{\epsilon})$, that do not change the set of approximate equilibria, $(\text{WS})(\text{C})\text{CE}(G, \epsilon) = (\text{WS})(\text{C})\text{CE}(\hat{G}, \hat{\epsilon})$. This transformation is an *equilibrium-invariant* transform. The most common such transform (Morris and Ui, 2004; Moulin and Vial, 1978; Ostrovski, 2013) is the *affine transform* which consists of an offset over the other players’ strategies and a positive scale.

Theorem 3.2.1 (Affine Transform). *ϵ -NE, ϵ -WSNE, ϵ -CE, ϵ -WSCE, and ϵ -CCE are equilibrium-invariant under affine transformations of each player’s payoff. Concretely, when*

$$G_p(a) \rightarrow \hat{G}_p(a) = s_p G_p(a_p, a_{-p}) + b_p(a_{-p}), \quad \text{and} \quad \epsilon_p \rightarrow \hat{\epsilon}_p = s_p \epsilon_p, \quad (3.1)$$

an ϵ_p -equilibrium in the original game is an $s_p \epsilon_p$ -equilibrium in the transformed game: $\sigma(a) \rightarrow \hat{\sigma}(a) = \sigma(a)$, where $b_p(a_{-p})$ is any offset, and s_p is any positive scalar.

Proof. Consider the effect of the transforms on the deviation gains.

$$\begin{aligned} A_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) &\rightarrow s_p G_p(a'_p, a_{-p}) + b_p(a_{-p}) - s_p G_p(a''_p, a_{-p}) - b_p(a_{-p}) \\ &= s_p A_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) \end{aligned} \quad (3.2a)$$

$$\begin{aligned} A_p^{\text{CE}}(a'_p, a''_p, a) &\rightarrow \begin{cases} s_p A_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) & a_p = a''_p \\ 0 & \text{otherwise} \end{cases} \\ &= s_p A_p^{\text{CE}}(a'_p, a''_p, a) \end{aligned} \quad (3.2b)$$

$$\begin{aligned} A_p^{\text{CCE}}(a'_p, a) &\rightarrow s_p G_p(a'_p, a_{-p}) + b_p(a_{-p}) - s_p G_p(a) - b_p(a_{-p}) \\ &= s_p A_p^{\text{CCE}}(a'_p, a) \end{aligned} \quad (3.2c)$$

Equilibria are entirely defined by their inequality constraints (Equations (2.46), (2.54), (2.59) and (2.64)). The affine transform only results in a s_p scale to the LHS of the inequality. If the same positive scale is applied to the RHS of the definition the inequality will still hold. Therefore an ϵ_p -equilibrium in the untransformed game will be an $s_p \epsilon_p$ -equilibrium in the transformed game. If $\epsilon_p = 0$ the equilibria will not change. \square

The affine transform can be used to reduce the degrees of freedom in each player's payoff by $|\mathcal{A}_{-p}| + 1$ without changing the equilibria if b_p and s_p are defined in terms of the payoffs themselves. Offsetting by the mean, $b_p(a_{-p}) = -\frac{1}{|\mathcal{A}_p|} \sum_{a_p} G_p(a_p, a_{-p})$, and scaling by the Frobenius norm of the payoff tensor, $s_p = 1/\|G_p\|_F$, are sensible choices. This transform can be used to design a distance metric, d^{equil} , and a distance-preserving equilibrium-invariant embedding. Let $\mathcal{G}^{\text{equil}}$ be an embedding in \mathcal{G} given by a structure preserving mapping, such that $\mathcal{G}^{\text{equil}} \subset \mathcal{G}$. $\mathcal{G}^{\text{equil}}$ is a manifold and a metric space.

Definition 3.2.2 (Equilibrium-Invariant Embedding).

$$G_p^{\text{equil}}(a) = \frac{1}{Z} \left(G_p(a) - \frac{1}{|\mathcal{A}_p|} \sum_{a_p} G_p(a_p, a_{-p}) \right) \quad (3.3a)$$

$$\epsilon_p^{\text{equil}} = \frac{1}{Z_p} \epsilon_p \quad (3.3b)$$

$$Z_p = \left\| G_p - \frac{1}{|\mathcal{A}_p|} \sum_{a_p} G_p(a_p, a_{-p}) \right\|_F \quad (3.3c)$$

This embedding results in N hyper-spheres, sometimes known as the oblique manifold in Riemannian geometry. Distance metrics in this space are extensively studied. The distance of the shortest path between points on a sphere is $\arccos(\sum_a G_p^{\text{A, equil}}(a) G_p^{\text{B, equil}}(a))$. And for the oblique manifold, it is the L_2 norm of each of the spheres distances. We combine the arc lengths with the length of the distance between the approximation parameters to obtain a final distance metric.

Definition 3.2.3 (Equilibrium-Invariant Distance Metric).

$$d^{\text{equil}}((G^A, \epsilon^A), (G^B, \epsilon^B)) = \sqrt{\sum_p \arccos \left(\sum_a G_p^{\text{A, equil}}(a) G_p^{\text{B, equil}}(a) \right)^2 + \sum_p (\epsilon_p^{\text{A, equil}} - \epsilon_p^{\text{B, equil}})^2} \quad (3.4)$$

This definition poses a problem: it is not well defined when a payoff is *trivial* because the normalization constant Z is zero for a trivial game. All joint distributions are in equilibrium in games where all players have trivial payoffs.

Definition 3.2.4 (Trivial Payoff).

$$G_p(a_p, a_{-p}) = b_p(a_{-p}) \quad \forall a_{-p} \in \mathcal{A}_{-p} \quad (3.5)$$

Definition 3.2.5 (Nontrivial Payoff).

$$G_p(a_p, a_{-p}) \neq b_p(a_{-p}) \quad \exists a_{-p} \in \mathcal{A}_{-p} \quad (3.6)$$

This can be simply remedied by defining the norm of a zero vector to be unity, $\|\mathbf{0}\| = 1$, which occurs when the payoff is trivial. For the embedding this is natural. For distances, the inner product, $\sum_a G_p^{A, \text{equil}}(a) G_p^{B, \text{equil}}(a)$, will be zero which is equivalent to the the payoffs being perpendicular. A trivial payoff would be defined to be perpendicular to all other payoffs. This is not an unreasonable definition, this work most commonly deals with nontrivial games, or only compares games with identical triviality structure, which is assumed going forward.

The equilibrium-invariant embedding is an oblique manifold (Trendafilov and Lippert, 2002) (a product manifold of N unit spheres). Each player's payoff embedding, G_p^{equil} , is a point on the surface of one of these spheres. The distance between two games is the norm of the arc lengths between the two points on each of these spheres. Therefore the maximum distance between two $\epsilon = 0$ games is $\sqrt{N}\pi$. The equilibrium-invariant embedding reduces the degrees of freedom in each player's payoff by $|\mathcal{A}_{-p}| + 1$. The linear offset component contributes a $|\mathcal{A}_{-p}|$ portion, while the nonlinear scaling contributes the remaining unit. It turns out that $|\mathcal{A}_{-p}|$ is the largest reduction that can be achieved by a linear function.

Theorem 3.2.6 (Linear Offset Rank Reduction). *The offset component of the equilibrium-invariant embedding reduces a payoff's degrees of freedom by $|\mathcal{A}_{-p}|$. This is the most degrees of freedom that can be reduced with a linear transform without changing the equilibrium.*

Proof. The computation of deviation gains, A , (Equations (2.46), (2.54), and (2.59)) is a linear operation and therefore can be expressed as a matrix multiplication of an operator matrix, T_p , with a player's payoff, G_p . Flattened forms of the payoffs and gains are used.

$$A_p^{\text{WSCE}}(a'_p \otimes a''_p, a_{-p}) = \sum_{a'''} T_p^{\text{WSCE}}(a'_p \otimes a''_p \otimes a_{-p}, a''') G_p(a''') \quad (3.7a)$$

$$A_p^{\text{CE}}(a'_p \otimes a''_p \otimes a) = \sum_{a'''} T_p^{\text{CE}}(a'_p \otimes a''_p \otimes a, a''') G_p(a''') \quad (3.7b)$$

$$A_p^{\text{CCE}}(a'_p \otimes a) = \sum_{a'''} T_p^{\text{CCE}}(a'_p \otimes a, a''') G_p(a''') \quad (3.7c)$$

By inspecting the structure of the operator matrices (Section 3.A.1, Lemma 3.A.1), their rank can be determined.

$$\text{rank}(T_p^{\text{WSCE}}) = \text{rank}(T_p^{\text{CE}}) = \text{rank}(T_p^{\text{CCE}}) = |\mathcal{A}| - |\mathcal{A}_{-p}| \quad \forall p \in [1, N] \quad (3.8)$$

In general, the payoff, G_p , can be full rank, $\mathbb{R}^{|\mathcal{A}|}$. But after matrix multiplying with the operator matrix, which is only rank $|\mathcal{A}| - |\mathcal{A}_{-p}|$, the resulting deviation gains can be at most rank $|\mathcal{A}| - |\mathcal{A}_{-p}|$. Any linear equilibrium-invariant transform that reduces the space of games by more than $|\mathcal{A}_{-p}|$ would imply an operator matrix T_p with rank less than $|\mathcal{A}| - |\mathcal{A}_{-p}|$, thereby reducing the number of inequality constraints that define the equilibrium. Therefore, any linear equilibrium-invariant transform can only reduce the space of games by at most $|\mathcal{A}_{-p}|$ without changing the set of equilibria for all games. The offset component of the affine transform reduces the degrees of freedom by $|\mathcal{A}_{-p}|$ and, by Theorem 3.2.1, does not change the deviation gains. \square

3.2.1 Reversible Deviation Gains

In general, the mapping from payoffs to deviation gains is irreversible because it is not a full-rank linear operation: there are many possible games that result in the same deviation gains. However, in the equilibrium-invariant embedding, there is a one-to-one mapping, and therefore it is possible to reverse the procedure and find the invariant embedding from the deviation gains.

Algorithm 3.1 Equilibrium-Invariant Embedding Sampling

```

1:  $N \leftarrow \text{len}(|\mathcal{A}_1|, \dots, |\mathcal{A}_N|)$ 
2: for  $p \leftarrow [1, N]$  do
3:    $G_p(a) \leftarrow \mathcal{N}(0, 1) \quad \forall a \in \mathcal{A}$ 
4:    $G_p(a) \leftarrow G_p(a) - \frac{1}{|\mathcal{A}_p|} \sum_{a_p \in \mathcal{A}_p} G_p(a_p, a_{-p}) \quad \forall a \in \mathcal{A}$ 
5:    $G_p(a) \leftarrow \frac{G_p(a)}{\|G_p(a)\|_2} \quad \forall a \in \mathcal{A}$ 
6:  $G^{\text{equil}} \leftarrow \text{concat}(G_1, \dots, G_N)$ 
7: return  $G^{\text{equil}}$ 

```

Theorem 3.2.7 (Reversible Deviation Gains). *The equilibrium-invariant embedding can be recovered from the deviation gains.*

$$G_p^{\text{equil}}(a) = -\frac{1}{|\mathcal{A}_p|} \sum_{a'_p} A_p^{\text{WSCE}}(a'_p, a_p, a_{-p}) = -\frac{1}{|\mathcal{A}_p|} \sum_{a'_p, a''_p} A_p^{\text{CE}}(a'_p, a''_p, a) = -\frac{1}{|\mathcal{A}_p|} \sum_{a'_p} A_p^{\text{CCE}}(a'_p, a) \quad (3.9)$$

Proof. Recall the definition of the CCE deviation gains (Equation (2.59)). Take the mean over the player's own deviation strategies, $a'_p \in \mathcal{A}_p$, and rearrange.

$$\begin{aligned} A_p^{\text{CCE}}(a'_p, a) &= G_p(a'_p, a_{-p}) - G_p(a) \implies \\ G_p(a) &= \underbrace{\frac{1}{|\mathcal{A}_p|} \sum_{a'_p} G_p(a'_p, a_{-p})}_{\text{Zero when zero-mean}} - \frac{1}{|\mathcal{A}_p|} \sum_{a'_p} A_p^{\text{CCE}}(a'_p, a) \end{aligned} \quad (3.10)$$

When the payoffs are invariant embeddings, $G_p^{\text{equil}}(a)$, the mean term is zero, and the proof is concluded for CCEs. Noting that $A_p^{\text{CCE}}(a'_p, a) = \sum_{a''_p \in \mathcal{A}_p} A_p^{\text{CE}}(a'_p, a''_p, a)$, there is a similar solution for CEs. Noting that $A_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) = \sum_{a_p \in \mathcal{A}_p} A_p^{\text{CE}}(a'_p, a''_p, a)$, there is a similar solution for WSCEs. \square

This result is not directly utilized in this thesis but is included for several reasons. Firstly, it improves intuition about the space of games and how payoffs relate to the definition of equilibria and their constraints. Secondly, it provides further evidence of the fundamental nature of the equilibrium-invariant embedding. Finally, it opens up a path of research in inverse game theory, which studies the recovery of payoffs from observed equilibrium behaviour.

3.2.2 Sampling Equilibrium-Invariant Embedding

It is easy to uniformly sample over the invariant embedding¹ (Algorithm 3.1) and trivial games (Algorithm 3.2), where $\mathcal{N}(0, 1)$ is a zero-mean unit-variance normal distribution. This sampling approach is principled because it samples games that cover all interesting strategic interactions. Less principled ways of sampling (for example a uniform distribution over entries in the payoff, $G_p(a) = \mathcal{U}(0, 1) \quad \forall a \in \mathcal{A}$), are common in the literature, but may not cover the strategic space of games evenly. This thesis proposes that the equilibrium-invariant embedding should be used for evaluating equilibrium solvers and producing training and testing datasets.

¹Or equivalently, sampling uniformly over the surface of a unit sphere. This is studied in Riemannian geometry.

Algorithm 3.2 Trivial Embedding Sampling

```

1:  $N \leftarrow \text{len}(|\mathcal{A}_1|, \dots, |\mathcal{A}_N|)$ 
2: for  $p \leftarrow [1, N]$  do
3:    $b_p(a_{-p}) \leftarrow \mathcal{N}(0, 1) \quad \forall a_{-p} \in \mathcal{A}_{-p}$ 
4:    $G_p(a) \leftarrow b_p(a_{-p}) \quad \forall a \in \mathcal{A}$ 
5:    $G_p(a) \leftarrow \frac{G_p(a)}{\|G_p(a)\|_2} \quad \forall a \in \mathcal{A}$ 
6:  $G^{\text{tri}} \leftarrow \text{concat}(G_1, \dots, G_N)$ 
7: return  $G^{\text{tri}}$ 

```

3.3 Better-Response Embedding

Previously payoff scaling (within the affine game transform) was shown to be equilibrium-invariant, here it is shown that a novel per-strategy-scale transform is better-response-invariant². This transform results in reciprocal scaled corresponding equilibrium in the transformed game for ϵ -NEs, ϵ -WSNEs, ϵ -CEs, and ϵ -WSCEs.

Definition 3.3.1 (Best-Response-Invariant).

$$\arg \max_{a_p} G_p(a_p, a_{-p}) = \arg \max_{a_p} \hat{G}_p(a_p, a_{-p}) \quad \forall a_{-p} \in \mathcal{A}_{-p} \quad (3.11)$$

Definition 3.3.2 (Better-Response-Invariant³).

$$\arg \text{sort}_{a_p} G_p(a_p, a_{-p}) = \arg \text{sort}_{a_p} \hat{G}_p(a_p, a_{-p}) \quad \forall a_{-p} \in \mathcal{A}_{-p} \quad (3.12)$$

Theorem 3.3.3 (Per-Strategy-Scale Transform). ϵ -NE, ϵ -WSNE, ϵ -CE, ϵ -WSCE are better-response-invariant under positive per-strategy-scale of each player's payoff which results in reciprocal per-strategy-scale ($s_p \rightarrow s_p(a_p)$) of the equilibria. Concretely, when

$$G_p(a) \rightarrow \hat{G}_p(a) = (\otimes_{q \in -p} s_q(a_q)) G_p(a),$$

$$\text{and } \epsilon_p^{\text{WSCE}} \rightarrow \hat{\epsilon}_p^{\text{WSCE}}(a_p'') = \frac{\epsilon_p^{\text{WSCE}}}{Z_{-p}(a_p'')} \text{ or } \epsilon_p^{\text{CE}} \rightarrow \hat{\epsilon}_p^{\text{CE}}(a_p'') = \frac{\epsilon_p^{\text{CE}}}{Z_{s_p}(a_p'')}, \quad (3.13)$$

an equilibrium in the original game has a corresponding equilibrium,

$$\sigma(a) \rightarrow \hat{\sigma}(a) = \frac{1}{Z(\otimes_p s_p(a_p))} \sigma(a), \quad (3.14)$$

in the transformed game, where $Z = \sum_{a \in \mathcal{A}} \frac{\sigma(a)}{\otimes_p s_p(a_p)}$ and $Z_{-p}(a_p'') = \sum_{a_{-p}} \frac{\sigma(a_p'', a_{-p})}{\otimes_{-p} s_p(a_{-p})}$.

Proof. Consider the effect of the transforms on the deviation gains.

$$A_p^{\text{WSCE}}(a_p', a_p'', a_{-p}) \rightarrow \hat{A}_p^{\text{WSCE}}(a_p', a_p'', a_{-p}) = (\otimes_{q \in -p} s_q(a_q)) A_p^{\text{WSCE}}(a_p', a_p'', a_{-p}) \quad (3.15a)$$

$$A_p^{\text{CE}}(a_p', a_p'', a) \rightarrow \hat{A}_p^{\text{CE}}(a_p', a_p'', a) = (\otimes_{q \in -p} s_q(a_q)) A_p^{\text{CE}}(a_p', a_p'', a) \quad (3.15b)$$

Substitute the transformed deviation gains and approximations into the definition of CE (Equation 2.54), which holds $\forall p \in [1, N], a_p'' \neq a_p' \in \mathcal{A}_p$. This can be shown to be equivalent to a definition

²Better-response invariance implies best-response invariance.

³Technically, arg sort is required to split ties between payoffs by action index.

consisting of the untransformed game.

$$\sum_{a \in \mathcal{A}} \hat{\sigma}(a) \hat{A}_p^{\text{CE}}(a'_p, a''_p, a) \leq \hat{\epsilon}_p \quad (3.16a)$$

$$\sum_{a \in \mathcal{A}} \left[\frac{1}{Z(\otimes_p s_p(a_p))} \sigma(a) \right] [(\otimes_{q \in -p} s_q(a_q)) A_p^{\text{CE}}(a'_p, a''_p, a)] \leq \frac{1}{Z s_p(a''_p)} \epsilon_p \quad (3.16b)$$

$$\sum_{a \in \mathcal{A}} \frac{1}{Z s_p(a_p)} \sigma(a) A_p^{\text{CE}}(a'_p, a''_p, a) \leq \frac{1}{Z s_p(a''_p)} \epsilon_p \quad (3.16c)$$

$$\sum_{a \in \mathcal{A}} \sigma(a) A_p^{\text{CE}}(a'_p, a''_p, a) \leq \epsilon_p \quad (3.16d)$$

The $s_p(a_p)$ and $s_p(a''_p)$ terms cancel by substituting the a_p variable for a''_p in the LHS, which is permitted by checking the definition of $A_p^{\text{CE}}(a'_p, a''_p, a)$ (Equation 2.54). Therefore it is proved that if $\sigma(a)$ is an equilibrium in the untransformed game, $\hat{\sigma}(a)$ is an equilibrium in the transformed game, and $\hat{\sigma}(a)$ can be calculated directly from $\sigma(a)$ and $s_p(a_p)$. Now consider the WSCE definition (Equation 2.46), which needs to hold $\forall p \in [1, N], a''_p \neq a'_p \in \mathcal{A}_p$.

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \hat{\sigma}(a_{-p} | a''_p) \hat{A}_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) \leq \hat{\epsilon}_p \quad (3.17a)$$

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \hat{\sigma}(a''_p, a_{-p}) \hat{A}_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) \leq \hat{\sigma}(a''_p) \hat{\epsilon}_p \quad (3.17b)$$

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \frac{1}{Z s_p(a''_p)} \sigma(a''_p, a_{-p}) A_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) \leq \left(\sum_{a_{-p}} \frac{\sigma(a''_p, a_{-p})}{Z s_p(a''_p) (\otimes_{-p} s_p(a_p))} \right) \hat{\epsilon}_p \quad (3.17c)$$

$$\sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a''_p, a_{-p}) A_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) \leq \epsilon_p \quad (3.17d)$$

The definition of NE and WSNE is the same as CE and WSCE, but with the additional constraint that the joint factorizes: $\sigma(a) = \otimes_p \sigma_p(a_p)$. This additional constraint does not affect the proof above, so the result also holds for NE and WSNE. The payoffs are only scaled over other players' strategies, and these scales are positive, which implies better-response-invariance. \square

Notably, Theorem 3.3.3 does not hold for CCEs.

Theorem 3.3.4 (Per-strategy-scaling CCE counterexample). *Per-strategy-scaling of each player's payoff does not result in reciprocal per-strategy-scaling of CCEs.*

Proof. Consider the two-player three-strategy game, scaled-rock-paper-scissors, (G_1, G_2) , which has a CCE (amongst others) at σ defined below.

$$G_1 = \begin{bmatrix} 0 & 4 & -1 \\ -2 & 0 & 1 \\ 2 & -4 & 0 \end{bmatrix} \quad G_2 = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \quad \sigma = \begin{bmatrix} 0 & \frac{1}{4} & \frac{1}{12} \\ \frac{1}{4} & 0 & \frac{1}{12} \\ 0 & 0 & \frac{1}{3} \end{bmatrix} \quad (3.18)$$

Now let's scale this game by $s_1 = [1, 1, 1]$ and $s_2 = [\frac{1}{2}, \frac{1}{4}, 1]$ to arrive at the familiar rock-paper-scissors

game, (\hat{G}_1, G_2) .

$$\hat{G}_1 = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} \quad G_2 = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \quad \hat{\sigma} = \frac{1}{Z} \begin{bmatrix} 0 & 1 & \frac{1}{12} \\ \frac{1}{2} & 0 & \frac{1}{12} \\ 0 & 0 & \frac{1}{3} \end{bmatrix} \quad (3.19)$$

Note that the scaled version of σ , $\hat{\sigma}$, which is calculated according to Theorem 3.3.3, is not an equilibrium in the scaled game. Therefore, by counterexample, Theorem 3.3.3 does not hold for CCEs in general. \square

Fortunately, for two-strategy games, CE and CCEs are equivalent (Monnot and Piliouras, 2017). This property is utilized when deriving embeddings for 2x2 games.

Remark 3.3.5 (Two-Strategy (C)CE Equivalence). For n-player games with two strategies per player, ϵ -CCEs and ϵ -CEs are equivalent.

Embedding and distance metrics for n-player games can be readily defined using the per-strategy-scale transform. However for two-player games, using zero approximation $\epsilon = 0$, there is a natural embedding definition.

Definition 3.3.6 (Two-Player Better-Response-Invariant Embedding).

$$G_1^{\text{res}}(a_1, a_2) = \frac{1}{Z(a_2)} \left(G_1(a_1, a_2) - \frac{1}{|\mathcal{A}_1|} \sum_{a_1} G_p(a_1, a_2) \right) \quad Z(a_2) = \left\| G_1(:, a_2) - \frac{1}{|\mathcal{A}_1|} \sum_{a_1} G_1(a_1, a_2) \right\|_F \quad (3.20a)$$

$$G_2^{\text{res}}(a_1, a_2) = \frac{1}{Z(a_1)} \left(G_1(a_1, a_2) - \frac{1}{|\mathcal{A}_2|} \sum_{a_2} G_p(a_1, a_2) \right) \quad Z(a_1) = \left\| G_1(a_1, :) - \frac{1}{|\mathcal{A}_2|} \sum_{a_2} G_1(a_1, a_2) \right\|_F \quad (3.20b)$$

Definition 3.3.7 (Two-Player Better-Response-Invariant Distance Metric).

$$d^{\text{res}}(G^A, G^B) = \sqrt{\sum_{a_2} \arccos \left(\sum_{a_1} G_1^{A,\text{res}}(a_1, a_2) G_p^{B,\text{res}}(a_1, a_2) \right)^2 + \sum_{a_1} \arccos \left(\sum_{a_2} G_1^{A,\text{res}}(a_1, a_2) G_p^{B,\text{res}}(a_1, a_2) \right)^2} \quad (3.21)$$

Therefore, the better-response-invariant game embedding is an oblique manifold (a product manifold of $|\mathcal{A}_1| + |\mathcal{A}_2|$ unit spheres). For each other player's strategy for each player's payoff, the slice of the embedding is a point on the surface of one of these spheres. The distance between two games is the norm of the arc lengths between the two points on each of these spheres. Therefore the maximum distance between two games is $\sqrt{|\mathcal{A}_1| + |\mathcal{A}_2|} \pi$.

3.4 Symmetric Embedding

The order of the strategies in a normal-form game is arbitrary, therefore games identical up to strategy permutations could be considered equal. Furthermore, if the role of the player is not important⁴, identity up to player permutation can also be considered. If a canonical ordering of the strategies is defined, the area of the equilibrium-invariant embedding that needs to be considered is reduced. One such ordering could be defined as follows. Firstly, for each player, p , independently sort the elements over all other players strategies, a_{-p} , and then lexicographically sort over the player's own strategies, a_p , to obtain an

⁴Role may be important if each player has access to different strategies or different numbers of strategies.

order permutation, $\tau_p^*(a_p)$. Secondly, to get the order of players, sort each player's whole payoff and then lexicographically sort over players to get a player order permutation, $\omega^*(p)$.

$$G'_p(a_p, a_{-p}) = \text{sort}_{a_{-p}} G_p(a_p, a_{-p}) \quad \forall p \quad \tau_p^*(a_p) = \arg \text{lexsort}_{a_p} G'_p(a_p, a_{-p}) \quad \forall p \quad (3.22a)$$

$$G''_p(a) = \text{sort}_a G_p(a) \quad \forall p \quad \omega^*(p) = \arg \text{lexsort}_p G''_p(a) \quad (3.22b)$$

Partial orderings occur when strategies have equal payoff, therefore even if a permutation is only partially ordered, the resulting payoffs will be unique. These permutations can be used to define another embedding: the *symmetric game embedding*.

Definition 3.4.1 (Equilibrium-Symmetric Game Embedding).

$$G_p^{\text{sym}}(a) = G_{\omega^*(p)}(\tau_{\omega^*(p)}^*(a_{\omega^*(p)}), \dots) \quad \epsilon_p^{\text{sym}} = \hat{\epsilon}_{\omega^*(p)} \quad (3.23)$$

Symmetries do not reduce the number of degrees of freedom, but do reduce the volume of games by exploiting symmetry in their definitions. There are $\prod_p (|\mathcal{A}_p|!)$ such strategy permutation symmetries and $N!$ player permutation symmetries in a normal form game. These symmetries should be used in conjunction with either equilibrium-invariant or better-response-invariant embeddings.

Definition 3.4.2 (Equilibrium-Symmetric Distance Metric).

$$d^{\text{sym}}((G^A, \epsilon^A), (G^B, \epsilon^B)) = \min_{\tau_p(a_p), \omega(p)} \sqrt{\sum_p \arccos \left(\sum_a G_{\omega(p)}^{A, \text{equil}}(\tau_{\omega(p)}(a_{\omega(p)}), \dots) G_p^{B, \text{equil}}(a) \right)^2} + \sqrt{\sum_p \left(\epsilon_{\omega(p)}^{A, \text{equil}} - \epsilon_p^{B, \text{equil}} \right)^2} \quad (3.24)$$

3.5 Discussion

This work derived distance metrics and embeddings for n-player general-sum normal-form games, such that they are equilibrium-invariant, equilibrium-symmetric, and better-response-invariant. Similar metrics and embeddings could be readily derived for other succinct representations of games such as polymatrix, symmetric, zero-sum, or common-payoff.

To explore equilibrium-invariance, this chapter focused on the most popular equilibrium solution concepts including NEs, CEs, and CCEs. Other equilibria such as quantal response equilibrium (QRE) (McKelvey and Palfrey, 1995) may also be compatible with the equilibrium-invariant embedding. There may also be interesting connections to evolutionary stable strategies (ESS). Verification of other equilibrium concepts to future work.

Theorem 3.2.7 highlighted the direct connection between the deviation gains and the equilibrium-invariant embedding. Since the deviation gains directly describe the space of equilibria, if one knows the equilibria of a system, one could estimate the deviation gains, and hence the equilibrium-invariant embeddings. Therefore the equilibrium-invariant embedding could be a useful tool in inverse game theory.

A metric-space can be useful in performing perturbation analysis (as hinted in Section 4.5.4). It is not uncommon for payoffs to be estimated from data, such as in empirical game-theoretic analysis (EGTA) (Walsh et al., 2002; Wellman, 2006). Such payoffs have uncertainty, and small changes in the payoffs can cause large changes to the resulting equilibria. A notion of distance between games can help answer questions about how a game's equilibria may change with a different estimate. For example, is it near an equivalence class boundary, or about to switch from being invariant-zero-sum to invariant-common-payoff?

Popular game-theoretic rating methods, like Nash average (Balduzzi et al., 2018), are used to rate and rank strategies in normal-form games. These rankings are invariant to affine transforms of the payoffs

(Section 3.A.2). Therefore the equilibrium-invariant embedding preserves the game-theoretic ranking of strategies, which is further evidence of its fundamental nature.

3.6 Conclusion

This chapter studied payoff transforms that reduce the degrees of freedom in the space of games. This makes them a) easier to understand, b) easier to sample from, and c) easier to visualize. The chapter defined an equilibrium inspired metric-space, an equilibrium-invariant embedding, equilibrium-symmetric embedding, and better-response-invariant embedding for games. It is hoped that this work provides computational insights for game theoretic algorithm developers.

3.A Appendices

3.A.1 Deviation Gains as Linear Operators

The deviation gains (Equations (2.46), (2.54)) and (2.59)) can be found from linear transforms of each player's payoff.

Lemma 3.A.1. *The WSCE, CE and CCE deviation gains can be written as linear operations on the payoffs, $A_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) = \sum_{\tilde{a}} T_p^{\text{CCE}}(a'_p, a''_p, a_{-p}, \tilde{a}) G_p(\tilde{a})$, $A_p^{\text{CE}}(a'_p, a''_p, a) = \sum_{\tilde{a}} T_p^{\text{CE}}(a'_p, a''_p, a, \tilde{a}) G_p(\tilde{a})$, and $A_p^{\text{CCE}}(a'_p, a) = \sum_{\tilde{a}} T_p^{\text{CCE}}(a'_p, a, \tilde{a}) G_p(\tilde{a})$. The rank of the linear operations is $|\mathcal{A}| - |\mathcal{A}_{-p}|$.*

Proof. Flatten the payoff into a vector, $G_p(a''')$, of length $|\mathcal{A}|$, flatten the gain into vectors, $A_p^{\text{CCE}}(a'_p \otimes a)$, $A_p^{\text{WSCE}}(a'_p \otimes a''_p \otimes a_{-p})$ and $A_p^{\text{CE}}(a'_p \otimes a''_p \otimes a)$, of length $|\mathcal{A}_p| |\mathcal{A}|$, $|\mathcal{A}_p|^2 |\mathcal{A}_p|$ and $|\mathcal{A}_p|^2 |\mathcal{A}|$, and use matrix linear operators, $T_p^{\text{CCE}}(a'_p \otimes a, a)$, $T_p^{\text{WSCE}}(a'_p \otimes a''_p \otimes a_{-p}, a''')$ and $T_p^{\text{CE}}(a'_p \otimes a''_p \otimes a, a''')$, with shapes $|\mathcal{A}_p| |\mathcal{A}| \times |\mathcal{A}|$, $|\mathcal{A}_p|^2 |\mathcal{A}_{-p}| \times |\mathcal{A}|$ and $|\mathcal{A}_p|^2 |\mathcal{A}| \times |\mathcal{A}|$. Inspect the block matrix structure of T_1^{CCE} , where I is the identity matrix of shape $|\mathcal{A}_{-1}| \times |\mathcal{A}_{-1}|$, which has $|\mathcal{A}_1|$ block columns and $|\mathcal{A}_1|^2$ block rows. Note the property that the definition of the WSCE is just a reshaped version of the CCE, $A_p^{\text{WSCE}}(a'_p, a''_p, a_{-p}) = A_p^{\text{CCE}}(a'_p, a''_p, a_{-p})$ (Figure 3.1). A similar inspection of T_1^{CE} can be made, which has $|\mathcal{A}_1|$ block columns and $|\mathcal{A}_1|^3$ block rows.

The first block column can be constructed from the negative sum of the remaining block columns. Therefore there are $|\mathcal{A}_{-1}|$ redundant columns. The remaining are linearly independent, resulting in a rank of $|\mathcal{A}| - |\mathcal{A}_{-1}|$. Similar construction patterns can be made for $T_p^{\text{CCE}}(a'_p \otimes a, a)$. Again, one block column, or $|\mathcal{A}_{-p}|$ are not linearly independent, so the rank is $|\mathcal{A}| - |\mathcal{A}_{-1}|$. \square

3.A.2 Game Theoretic Rating

Game-theoretic rating is a method for rating strategies in a normal-form game. Let a strategy rating, $r_p(a_p)$, be a numerical scalar for each player's strategies, and the strategy rank be the sorting order $w_p(a_p) = \arg \text{sort } r_p(a_p)$ defined such that equal ratings are given equal rank (for example, $[1.1, -0.2, 0.3, 0.3] \rightarrow [3, 0, 1, 1]$). A very simple (but not game-theoretic) way of rating strategies in a game would be to examine their average payoff.

Definition 3.A.2 (Average Rating).

$$r_p^{\text{avg}}(a_p) = \frac{1}{|\mathcal{A}_{-p}|} \sum_{a_{-p}} G_p(a_p, a_{-p}) \quad (3.26)$$

A popular game-theoretic rating method, Nash averaging (NA) (Balduzzi et al., 2018), weights the payoffs by mixing over the maximum entropy Nash equilibrium solution. It is most commonly applied in two-player zero-sum where the solution is unique.

Definition 3.A.3 (Nash Average Rating).

$$r_p^{\text{NA}}(a_p) = \sum_{a_{-p}} G_p(a_p, a_{-p}) \left(\otimes_{q \in -p} \sigma_p^{\text{MENE}}(a_q) \right) \quad (3.27)$$

Theorem 3.A.4 (Rank-Invariance). *Nash-average is rank-invariant to affine transformations, $G_p(a) \rightarrow \hat{G}_p(a) = s_p G_p(a) + b_p(a_{-p})$, where $s_p > 0$ and $b_p(a_{-p})$ is arbitrary.*

$$\begin{aligned}
\tilde{T}_1^{\text{WSCE}} = T_1^{\text{CCE}} = & \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ \mathbf{I} & -\mathbf{I} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{I} & 0 & \dots & -\mathbf{I} & 0 \\ \mathbf{I} & 0 & \dots & 0 & -\mathbf{I} \\ \hline -\mathbf{I} & \mathbf{I} & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \mathbf{I} & \dots & -\mathbf{I} & 0 \\ 0 & \mathbf{I} & \dots & 0 & -\mathbf{I} \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline -\mathbf{I} & 0 & \dots & 0 & \mathbf{I} \\ 0 & -\mathbf{I} & \dots & 0 & \mathbf{I} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -\mathbf{I} & \mathbf{I} \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix} \quad (3.25a) \\
T_1^{\text{CE}} = & \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \hline 0 & 0 & \dots & 0 & 0 \\ \mathbf{I} & -\mathbf{I} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline -\mathbf{I} & \mathbf{I} & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \hline 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (3.25b)
\end{aligned}$$

Figure 3.1: Structure of deviation gain operators.

Proof. Consider the ranking after an affine transformation

$$\hat{r}_p^{\text{NA}}(a_p) = \left(\sum_{a_{-p}} s_p G_p(a_p, a_{-p}) + b_p(a_{-p}) \right) (\otimes_{q \in -p} \sigma_p^{\text{MENE}}(a_q)) \quad (3.28a)$$

$$= s_p r_p^{\text{NA}}(a_p) + \underbrace{\sum_{a_{-p}} b_p(a_{-p}) (\otimes_{q \in -p} \sigma_p^{\text{MENE}}(a_q))}_{\text{Constant: does not depend on } a_p.} \quad (3.28b)$$

It is easy to see that $w_p(a_p) = \hat{w}_p(a_p)$. □

The equilibrium-invariant embedding therefore preserves game-theoretic ranking of strategies. Furthermore, the equilibrium-invariant embedding may provide a more natural normalization of payoffs than the approach originally suggested by [Balduzzi et al. \(2018\)](#).

Chapter 4

2×2 Metric Spaces and Embeddings






Metric spaces and embeddings of normal-form games are powerful tools for simplifying and reasoning about the space of games. To demonstrate the utility of these tools, this chapter studies the simplest and most ubiquitous class of normal-form games: 2×2 games. The equilibrium-invariant embedding of 2×2 games has an efficient two variable parameterization (a reduction from eight), where each variable geometrically describes an angle on a unit circle. Interesting properties can be spatially inferred from the embedding, including: equilibrium support, cycles, competition, coordination, distances, best-responses, and symmetries. The best-response-invariant embedding of 2×2 games, after considering symmetries, rediscovers a set of 15 games, and their respective equivalence classes. This chapter proposes that this set of game classes is fundamental and captures all possible interesting strategic interactions in 2×2 games. This work introduces a directed graph representation and name for each class. The contents of this chapter is part of published work ([Marris et al., 2023](#)). Code for producing visualizations is also open source ([Marris et al., 2024](#)).

4.1 Introduction

To explore the importance of equilibrium-invariant metric spaces this chapter focuses on 2×2 games: exploring their properties, visualizing their structure, and rediscovering a set of 15 fundamental games. 2×2 normal-form games have two players, each with two strategies. Players take one of the two strategies (possibly at random) simultaneously. The resulting joint strategy triggers a payoff for each player. The game is played only once. The table of payoffs determines rational behaviour of the players (i.e. the set of equilibria). Popular games are given names based on the payoffs and the resulting behaviour. 2×2 games are utilized so frequently that their names have entered popular culture (Figure 4.1), for example: 🐔 Chicken, 🦔 Prisoner's Dilemma, 🐇 Stag Hunt ([Skyrms, 2004](#)), 🎻 Bach or Stravinsky (battle of the sexes), and 🎲 Matching Pennies¹. The study of such games ([Kelley et al., 2002](#)) is crucial to understanding cooperation ([Gauthier, 1986](#)), competition, coordination, nature ([Wilkinson, 1984](#)), incentive structures ([Sugden, 1986](#)), social dilemmas ([Bruns and Kimmich, 2021](#)), utilitarian behaviour, rational behaviour ([Gintis, 2014](#)), and seemingly irrational behaviour. Games are used to inform economic policy ([Ostrom et al., 1994](#)), social structure ([Bicchieri, 2005](#); [Binmore and Binmore, 1994](#); [Skyrms, 2004](#)), foreign policy ([Schelling, 1966](#)), pandemic response, and environmental treaties ([Brânzei et al., 2021](#); [Breton et al., 2006](#); [Schosser, 2022](#)).

As a result, great effort has been expended in creating parameterizations, taxonomies, topologies, and names for 2×2 games ([Brams, 1993](#); [Bruns, 2015](#); [Kilgour and Fraser, 1988](#); [Rapoport and Guyer,](#)

¹This work accompanies game names with a graphical representation which describes either each player's preference over joint payoffs for ordinal games (e.g. 🎲 Matching Pennies) or their best-response preferences for best-response-invariant embeddings (e.g. 🔄 Cycle). These representations are described more thoroughly in later sections. The visualizations are available in ([Marris et al., 2024](#)).

	C	S		C	D		S	H		M	F		H	T
C	-9, -9	+1, -1	C	-2, -2	-9, +0	S	4, 4	1, 3	M	3, 2	0, 0	H	+1, -1	-1, +1
S	-1, +1	0, 0	D	+0, -9	-5, -5	H	3, 1	2, 2	F	0, 0	2, 3	T	-1, +1	+1, -1

(a) Chicken (b) Prisoner's Dilemma (c) Stag Hunt (d) Bach or Stravinsky (e) Matching Pennies

Figure 4.1: Payoff tables of common 2x2 normal-form games. Player 1 selects a row strategy and player 2 selects a column strategy. Each player respectively receives one of the payoffs in the tuple. The joint payoff preference ordering is shown in the top-left for each player.

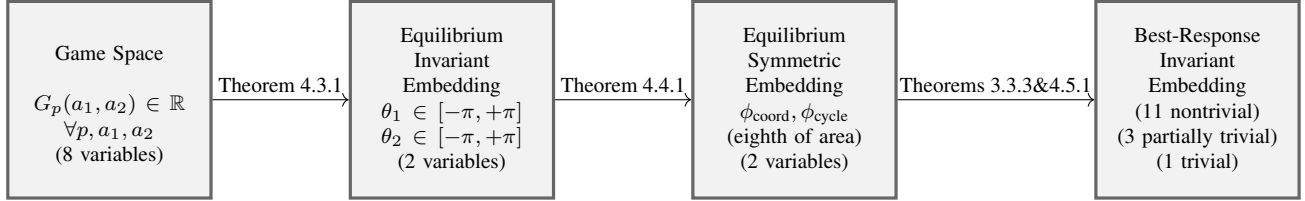


Figure 4.2: Summary of the main contributions of this work showing how all 2x2 games can be transformed to an equilibrium-invariant embedding, an equilibrium-symmetric embedding, and a best-response-invariant embedding.

1966; Rapoport et al., 1976; Robinson and Goforth, 2005; Robinson et al., 2007; Walliser, 1988). The most common of these focus on ordinal games, a type of equivalence that recognises that resulting joint strategies in games are only meaningfully different up to the partial ordering of the elements of each player's payoff. There are 726 such games (Fraser and Kilgour, 1986). Most other work (Böör et al., 2022; Goforth and Robinson, 2005; Harris, 1969; Huertas-Rosero, 2003; Rapoport and Guyer, 1966) only considers subsets (e.g. symmetric or strictly ordinal) of 2x2 games. Borm (1987) classified all 2x2 games into 15 distinct classes studying their NE best-responses. Fishburn and Kilgour (1990) later showed that these classes can be represented with binary games². Neither provide a notion of closeness, distance metric, or satisfying parameterization for these games.

Chapter 3 uncovered metric spaces and embeddings for general-sum n-player normal-form games. This chapter demonstrates these embeddings in 2x2 games (Figure 4.2) to highlight how embeddings can uncover hidden structure by removing degrees of freedom. All nontrivial (Definition 3.2.4) 2x2 games can be transformed to an equilibrium-invariant embedding which can be parameterized using only two variables, a reduction from the eight needed to represent the original payoffs (Table 4.1). Remarkably, this does not change the set of equilibria in the game. Geometrically, each of the two variables describes an angle on a circle and has spatial meaning: similar games are situated near each other. Properties of a game, like equilibrium support, zero-sum-invariance, common-payoff-invariance, whether it is clockwise or anti-clockwise cyclic, whether it is coordination or anti-coordination, and the best-response dynamics, can be easily deduced from this embedding. Using symmetry, the area of equilibrium-invariant embedding can be reduced by a factor of eight to result in the equilibrium-symmetric embedding. Additionally this space can be further reduced to a set of games with a cardinality of 15: the best-response-invariant embedding. Because many equilibrium-invariant embeddings map to the same best-response-invariant embedding, they are also part of equivalence classes. These are the same classes found by Borm (1987), but derived through a different but related argument. To obtain his class, Borm simply enumerates all possible best-response graphs in 2x2 games up to symmetry. This work improves upon Borm's classification by situating them in the equilibrium-invariant embedding, illuminating the relationship between the games. Names and an elegant graphical visualization of the classes are also defined in this chapter.

²Games that either have 0 or 1 payoffs.

	A	B		A	B		A	B
A	g_1^{AA}, g_2^{AA}	g_1^{AB}, g_2^{AB}	A	g^{AA}	g^{AB}	A	0	g^{AB}
B	g_1^{BA}, g_2^{BA}	g_1^{BB}, g_2^{BB}	B	g^{BA}	g^{BB}	B	$-g^{AB}$	0
(a) Full Normal-Form (8 variables)			(b) Symmetric, Common or Zero-Sum (4 variables)			(c) Symmetric Zero-Sum (1 variable)		

Table 4.1: Number of variables needed to describe 2×2 normal-form games and sparser simplifications.

4.2 Preliminaries

Game theory is most developed in a subset of games: those with two players and a restriction on the payoffs, $G_1(a_1, a_2) = -G_2(a_1, a_2)$, known as zero-sum. Particularly in n-player, general-sum games, there is no definitive solution concept. One approach is to consider joints that are in equilibrium: distributions that no player has incentive to unilaterally deviate from. Games are sometimes referred to by their shape, for example: $|\mathcal{A}_1| \times \dots \times |\mathcal{A}_N|$. This primarily focuses on general-sum 2×2 normal-form games.

Two-Player Two-Strategy Games

2×2 games are the smallest possible, but have remarkable strategic depth and explain many real-world interactions. As a result, a rich literature has accumulated on explaining, visualizing, categorizing, parameterizing and naming 2×2 games. Naively a 2×2 game (Table 4.1a) requires 8 variables to define which is too many variables to intuit. Therefore many approaches attempt to reduce this complexity through either using invariances, symmetries and equivalences, or considering a subset of games (for example symmetric, common payoff, or zero-sum games; Tables 4.1b and 4.1c). After this reduction in complexity, either a finite set of games or a reduced space of games remains. This set or space may have a structure that describes how close it is to other games. Approaches that simply bin games into a set are *categorical* approaches. Those that also impose a notion of similarity or closeness between games are *topological*. Those that impose hierarchical structure to the categorization are called *taxonomies*. Those that parameterize the game are *parametric*. Approaches may have multiple properties.

Games can be characterized based on their payoffs. Rapoport and Guyer (1966)’s “taxonomy of games” (and book (Rapoport et al., 1976)) exploits a particular equivalence class where only the order of each player’s payoffs matter. Changes in the magnitude of payoffs that do not change the order result in predictable scaling of the equilibrium of the game. Additionally, if only the subset of games that have strict ordering (*strict ordinal games*) are considered, the payoffs can be represented with permutations of the set of ordinal numbers $\{1, 2, 3, 4\}$. This results in $4! = 24$ ways to strictly order each player’s payoff, which results in $24 \times 24 = 576$ games. Utilizing strategy permutations³ for each player reduces the number of strict ordinal games to 144. When including player permutations⁴ half the non-symmetric games can be removed which reduces the cardinality to 78 strict ordinal games. A drawback of this approach is that it only classifies games with strict payoff orderings. Fraser and Kilgour (1986) introduced the categorization of partially ordered ordinal games (*partial ordinal games*) where payoffs can take on equal values. This results in a total cardinality of 1413 partially ordered ordinal games with strategy symmetries, or 726 if player symmetries are utilized. Partial ordinal games can be visualized using a graphical representation which shows the ordering of joint preferences (Figure 4.1). Goforth and Robinson (2005) improved the categorization of strict ordinal games to produce a “periodic table”. The 144 games were distributed in a 12×12 grid such that adjacent games had similar properties. Robinson et al. (2007) extended this topology to include partially ordered games. Bruns (2015) suggested a formulaic naming scheme for all ordinal

³The literature phrases this as “order graph” equivalence.

⁴Sometimes referred to as “reflection” in the literature. This symmetry does not quite halve the space because all symmetric games are retained. Of the 78 strict ordinal games, 66 are non-symmetric and 12 are symmetric. The literature is shier to utilize this symmetry because it reverses the roles of players.

games as previously only a small subset (often symmetric games) had established common names.

Harris (1969) gives a parameterized classification of symmetric 2x2 games using only two variables which are functions of the payoffs (Table 4.1b): $r_3 = \frac{g^{BB} - g^{AB}}{g^{BA} - g^{AB}}$ and $r_4 = \frac{g^{BA} - g^{AA}}{g^{BA} - g^{AB}}$, with the constraint that $g^{BA} > g^{AB}$. This defines a plane, with regions that correspond to classes of games with similar properties. Huertas-Rosero (2003) also classifies symmetric 2x2 games into 8 base classes, and 12 total classes based on their NE. Böörs et al. (2022) classifies symmetric 2x2 games into 24 classes based on their decomposition into common-payoff and zero-sum parts. Germano (2006) classifies games into various equivalent classes based on their Nash equilibrium geometry. Borm (1987) classifies all 2x2 games into a set of 15 games based on their best-response characteristics. The games can also be parameterized with 4 discrete variables. Borm did not describe any notions of similarity or distance between games in the set. This work rediscovers Borm (1987)'s classification through a scale-based payoff transform. In addition, this work extends this classification to a metric space, provide a much more efficient two variable embedding of these games, and name these games.

4.3 2x2 Equilibrium-Invariant Embedding

Using the embedding discussed in Chapter 3, equilibrium-invariant embeddings are derived for 2x2 games.

4.3.1 Deriving the 2x2 Equilibrium-Invariant Embedding

2x2 games have a particularly efficient equilibrium-invariant embedding parameterized by only two variables. This embedding has a distance metric equivalent to the one defined above, so is also a metric space over 2x2 games.

Theorem 4.3.1 (2x2 Equilibrium-Invariant Embedding). *All nontrivial 2x2 game payoff matrices, G_p , can be mapped to payoff matrices, G_p^{equil} , parameterized by only two variables, without altering the equilibria of the game. The mapping is given by Equations (4.1a) and (4.1b), where $\theta_1 + \frac{\pi}{4} = \arctan2(g_1^{AA} - g_1^{BA}, g_1^{AB} - g_1^{BB})$ and $\theta_2 + \frac{\pi}{4} = \arctan2(g_2^{AA} - g_2^{AB}, g_2^{BA} - g_2^{BB})$.*

$$G_1 = \begin{bmatrix} g_1^{AA} & g_1^{AB} \\ g_1^{BA} & g_1^{BB} \end{bmatrix} \rightarrow G_1^{equil}(\theta_1) = \begin{bmatrix} \frac{1}{\sqrt{2}} \sin(\theta_1 + \frac{\pi}{4}) & \frac{1}{\sqrt{2}} \cos(\theta_1 + \frac{\pi}{4}) \\ -\frac{1}{\sqrt{2}} \sin(\theta_1 + \frac{\pi}{4}) & -\frac{1}{\sqrt{2}} \cos(\theta_1 + \frac{\pi}{4}) \end{bmatrix} \quad (4.1a)$$

$$G_2 = \begin{bmatrix} g_2^{AA} & g_2^{AB} \\ g_2^{BA} & g_2^{BB} \end{bmatrix} \rightarrow G_2^{equil}(\theta_2) = \begin{bmatrix} \frac{1}{\sqrt{2}} \sin(\theta_2 + \frac{\pi}{4}) & -\frac{1}{\sqrt{2}} \sin(\theta_2 + \frac{\pi}{4}) \\ \frac{1}{\sqrt{2}} \cos(\theta_2 + \frac{\pi}{4}) & -\frac{1}{\sqrt{2}} \cos(\theta_2 + \frac{\pi}{4}) \end{bmatrix} \quad (4.1b)$$

Proof. Consider the payoff for player 1, G_1 . First apply the offset of the affine invariant transformation, $G_1(a_1, a_2) \rightarrow \bar{G}_1(a_1, a_2) = G_1(a_1, a_2) + b_1(a_2)$, where the columns (other player strategies) of player 1's payoff matrix are normalized to zero-mean offset, $b_1(a_2) = -\frac{1}{|\mathcal{A}_1|} \sum_{a_1 \in \mathcal{A}_1} G_1(a_1, a_2)$, such that $b_1 = [-\frac{1}{2}g_1^{AA} - \frac{1}{2}g_1^{BA}, -\frac{1}{2}g_1^{AB} - \frac{1}{2}g_1^{BB}]$. Then, make a variable substitution with $g_1^{A-B,A} = g_1^{AA} - g_1^{BA}$ and $g_1^{A-B,B} = g_1^{AB} - g_1^{BB}$, which reduces the number of parameters needed to describe the payoff from $4 \rightarrow 2$.

$$\begin{bmatrix} g_1^{AA} & g_1^{AB} \\ g_1^{BA} & g_1^{BB} \end{bmatrix} \rightarrow \bar{G}_1 = \begin{bmatrix} \frac{1}{2}(g_1^{AA} - g_1^{BA}) & \frac{1}{2}(g_1^{AB} - g_1^{BB}) \\ \frac{1}{2}(g_1^{BA} - g_1^{AA}) & \frac{1}{2}(g_1^{BB} - g_1^{AB}) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}g_1^{A-B,A} & \frac{1}{2}g_1^{A-B,B} \\ -\frac{1}{2}g_1^{A-B,A} & -\frac{1}{2}g_1^{A-B,B} \end{bmatrix} \quad (4.2)$$

Now apply the scale of the affine invariant transform, a unit L_2 normalization, $\bar{G}_1(a_1, a_2) \rightarrow G_1^{equil}(a_1, a_2) = s_1 \bar{G}_1(a_1, a_2)$ where $s_1 = \frac{1}{\|\bar{G}_1\|_2}$, which ensures that the norm over all the elements in the payoff are equal to one. This is a valid transform as long as the payoff is nontrivial (nonzero after

zero-mean offset).

$$\begin{bmatrix} \frac{1}{2}g_1^{A-B,A} & \frac{1}{2}g_1^{A-B,B} \\ -\frac{1}{2}g_1^{A-B,A} & -\frac{1}{2}g_1^{A-B,B} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{1}{\sqrt{2}} \frac{g_1^{A-B,A}}{\sqrt{g_1^{A-B,A^2} + g_1^{A-B,B^2}}} & \frac{1}{\sqrt{2}} \frac{g_1^{A-B,B}}{\sqrt{g_1^{A-B,A^2} + g_1^{A-B,B^2}}} \\ -\frac{1}{\sqrt{2}} \frac{g_1^{A-B,A}}{\sqrt{g_1^{A-B,A^2} + g_1^{A-B,B^2}}} & -\frac{1}{\sqrt{2}} \frac{g_1^{A-B,B}}{\sqrt{g_1^{A-B,A^2} + g_1^{A-B,B^2}}} \end{bmatrix} \quad (4.3)$$

Note that the L_2 norm of the elements of this payoff now equal unity, resulting in the property that:

$$\left(\frac{g_1^{A-B,A}}{\sqrt{g_1^{A-B,A^2} + g_1^{A-B,B^2}}} \right)^2 + \left(\frac{g_1^{A-B,B}}{\sqrt{g_1^{A-B,A^2} + g_1^{A-B,B^2}}} \right)^2 = 1 \quad (4.4)$$

This is the equation of a unit circle, and the opposite and adjacent parameters can be represented as a single angle parameter θ_1 , where $\frac{\pi}{4}$ is an arbitrary offset chosen for visualization purposes. This further reduces the number of parameters needed to describe player 1's payoff from $2 \rightarrow 1$.

$$\tan\left(\theta_1 + \frac{\pi}{4}\right) = \frac{g_1^{A-B,A}}{g_1^{A-B,B}} \implies \theta_1 + \frac{\pi}{4} = \arctan2(g_1^{A-B,A}, g_1^{A-B,B}) \quad (4.5)$$

The function $\arctan2$ is defined by [Organick \(1966\)](#). Since the radius of the unit circle is 1, the elements of the payoff are can be recovered directly from θ_1 .

$$G_1^{\text{equil}}(\theta_1) = \begin{bmatrix} \frac{1}{\sqrt{2}} \sin(\theta_1 + \frac{\pi}{4}) & \frac{1}{\sqrt{2}} \cos(\theta_1 + \frac{\pi}{4}) \\ -\frac{1}{\sqrt{2}} \sin(\theta_1 + \frac{\pi}{4}) & -\frac{1}{\sqrt{2}} \cos(\theta_1 + \frac{\pi}{4}) \end{bmatrix} \quad (4.6)$$

An equivalent set of reductions can be used for player 2, where $\theta_2 + \frac{\pi}{4} = \arctan2(g_2^{A,A-B}, g_2^{B,A-B})$.

$$G_2 = \begin{bmatrix} g_2^{AA} & g_2^{AB} \\ g_2^{BA} & g_2^{BB} \end{bmatrix} \rightarrow G_2^{\text{equil}}(\theta_2) = \begin{bmatrix} \frac{1}{\sqrt{2}} \sin(\theta_2 + \frac{\pi}{4}) & -\frac{1}{\sqrt{2}} \sin(\theta_2 + \frac{\pi}{4}) \\ \frac{1}{\sqrt{2}} \cos(\theta_2 + \frac{\pi}{4}) & -\frac{1}{\sqrt{2}} \cos(\theta_2 + \frac{\pi}{4}) \end{bmatrix} \quad (4.7)$$

□

Therefore all nontrivial 2x2 games can be parameterized using two variables (θ_1, θ_2) which geometrically describe points via angles on two circles ⁵. Several interesting structural properties emerge (Figure 4.3), which will be explained in the following sections. Similarly, the partially trivial invariant embedding can be visualized in a single dimension. Partially trivial games, where one of the players does not have any influence in the game, could be described by a single parameter, either θ_1 or θ_2 , depending on which player contributes to the strategic dynamics. Partially trivial games can therefore be mapped to the *partially trivial equilibrium-invariant embedding manifold*. Trivial games where no player participates are mapped to a singleton set consisting of the \cdot Null game.

4.3.2 Invariant-Zero-Sum and Invariant-Common-Payoff Quadrants

Two-player zero-sum games are well studied in the literature because they are easier to solve than their mixed-motive cousins. In particular for the NE, the problem can be expressed as a min-max optimization. If multiple equilibria exist, all equilibria have the same payoff, and they are known to be interchangeable (it does not matter which equilibrium the opponent chooses). But such games represent a small subset of general-sum games. If a larger set of games could be mapped onto the space of zero-sum games using

⁵Defined as the product manifold of two circles. Topologically this is a torus. Other classifications ([Robinson and Goforth, 2005](#)) also have a similar torus topology.

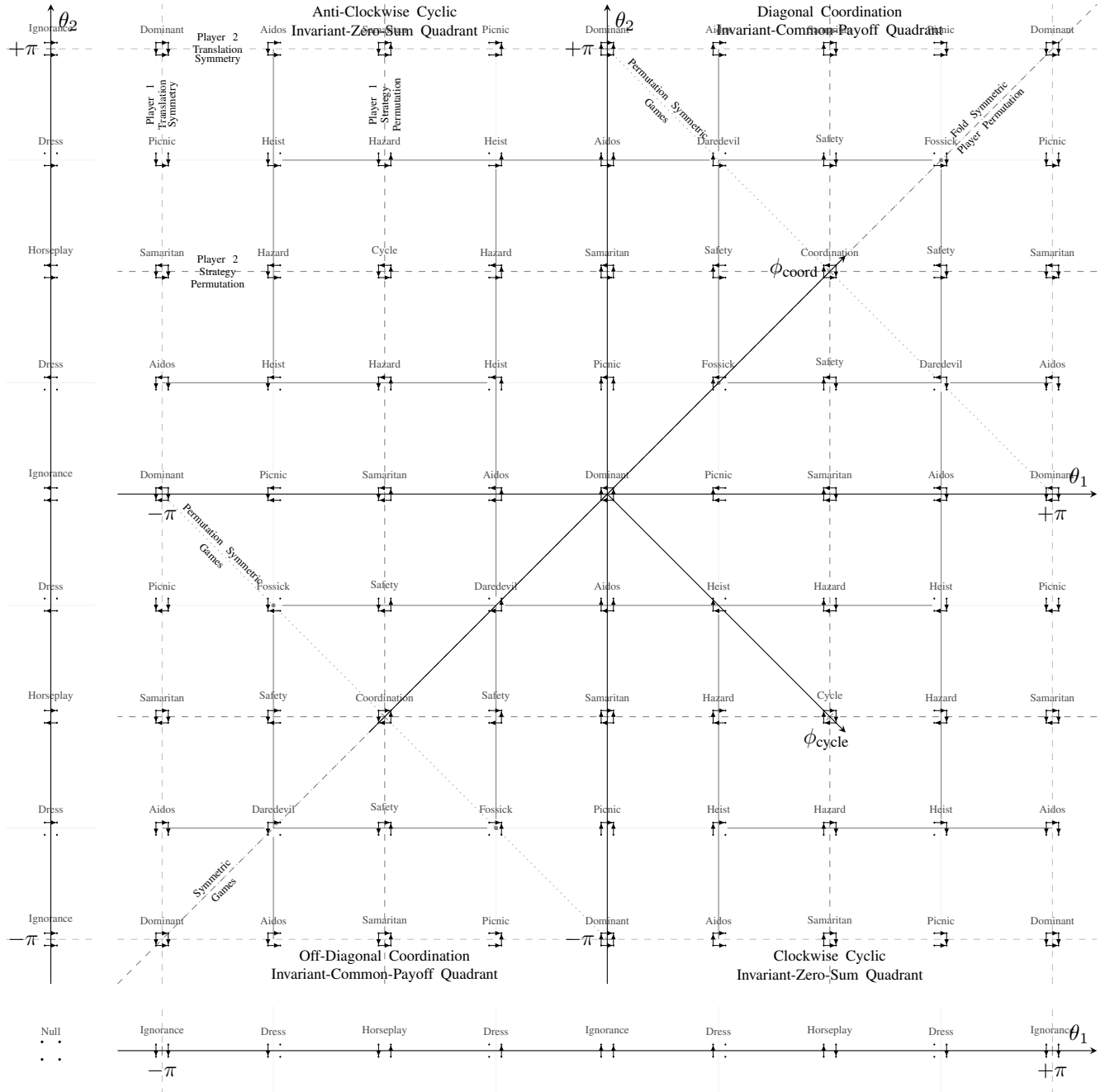


Figure 4.3: The 2x2 equilibrium-invariant embedding. Games can either be parameterized per-player, θ_1 and θ_2 , or over game types, ϕ_{cycle} and ϕ_{coord} . Symmetries (Table 4.2) are indicated by dashed lines and allow reduction in area by a factor of up to 8, depending on which symmetries are utilized. The strategy permutation equilibrium-symmetric embedding is north east patterned (\nearrow). The strategy and player permutation equilibrium-symmetric embedding is north west patterned (\nwarrow). Permutation symmetry with zero-sum assumption is vertical patterned (\parallel). The quadrants separate the space into cyclic and coordination regions. The top-left and bottom-right are cyclic (\square Cycle), invariant-zero-sum regions. The top-right and bottom-left are coordination (\square Coordination), invariant-common-payoff regions. The dotted lines indicate symmetric games or symmetric games with permuted strategies. The light solid lines indicate equivalence class boundaries. The solid gray lines, points, and space between them indicate boundaries in the support of the equilibrium (see also Figure 4.6). The \square Dominant, \square Picnic, and \square Samaritan have pure equilibria, the square regions are full-support (either \square cycle or \square Coordination), the gray line boundaries have either two ($\uparrow\downarrow$ Aidos, $\uparrow\downarrow$ Heist, $\uparrow\downarrow$ Hazard) or three ($\uparrow\downarrow$ Safety, $\uparrow\downarrow$ Daredevil) joint strategy support, and the gray points are diagonal or off-diagonal support ($\uparrow\downarrow$ Fossick). The 11 nontrivial best-response-invariant set of games is annotated with names suffixed with the equilibrium support. \square Dominant, the only game that is both invariant-zero-sum and invariant-common-payoff, is at the origin. [Note: some PDF viewers incompletely render this figure.]

invariant transformations, it could be proved that they would share these interesting properties.

Two such approaches have already been explored in the literature. Firstly, *constant-sum* games are scalar-offset transformations of zero-sum games, $G_p(a) \rightarrow \hat{G}_p(a) = G_p(a) + b_p$, which are well known to share zero-sum game's properties. Secondly, *strictly-competitive* games (Adler et al., 2009) are scalar-offset and player-scale transformations of zero-sum games, $G_p(a) \rightarrow \hat{G}_p(a) = s_p G_p(a) + b_p$. However, there exists a larger space of games that can be mapped to zero-sum games.

Definition 4.3.2 (Invariant-Zero-Sum Games). Invariant-zero-sum games are those that can be mapped to zero-sum games using equilibrium-invariant transformations, $G_p(a) \rightarrow \hat{G}_p(a) = s_p G_p(a) + b_p(a_{-p})$ such that $\sum_p \hat{G}_p(a) = 0 \forall a \in \mathcal{A}$.

By definition, the equilibria of invariant-zero-sum games are identical to their zero-sum counterpart. Furthermore, the equilibria are still interchangeable in the transformed game, however not all equilibria will necessarily have the same payoffs. Other generalizations of zero-sum games including *strategically zero-sum* (Moulin and Vial, 1978), *best-response zero-sum* (Rosenthal, 1974), and *order zero-sum* (Shapley, 1963), are all supersets of invariant zero-sum games. The diagonal quadrants (top-left and bottom-right) in the visualization (Figure 4.3) are *invariant-zero-sum*.

Theorem 4.3.3 (Invariant-Zero-Sum Quadrants). When $\sin(\theta_1) \sin(\theta_2) < 0$ in Equations (4.1a) and (4.1b), the respective game $(\hat{G}_1(\theta_1), \hat{G}_2(\theta_2))$ is invariant-zero-sum.

Proof. For a game to be invariant-zero-sum there have to exist invariant transforms $s_2 > 0$, $b_1(a_2)$, and $b_2(a_1)$ such that $G_1(a_1, a_2) + b_1(a_2) = -s_2 G_2(a_1, a_2) - b_2(a_1) \forall a_1, a_2$. Consider the transforms $s_2 = -\frac{\sin(\theta_1)}{\sin(\theta_2)}$, $b_1 = [-\frac{1}{2} \frac{\sin(\theta_1)}{\tan(\theta_2)}, \frac{1}{2} \frac{\sin(\theta_1)}{\tan(\theta_2)}]$, and $b_2 = [-\frac{1}{2} \cos(\theta_1), \frac{1}{2} \cos(\theta_1)]$, which result in payoffs:

$$\hat{G}_1(\theta_1, \theta_2) = -\hat{G}_2(\theta_1, \theta_2) = \begin{bmatrix} \frac{1}{2} \frac{\sin(\theta_1)}{\tan(\theta_2)} + \frac{1}{\sqrt{2}} \sin(\theta_1 + \frac{\pi}{4}) & -\frac{1}{2} \frac{\sin(\theta_1)}{\tan(\theta_2)} + \frac{1}{\sqrt{2}} \sin(\theta_1 + \frac{\pi}{4}) \\ \frac{1}{2} \frac{\sin(\theta_1)}{\tan(\theta_2)} - \frac{1}{\sqrt{2}} \sin(\theta_1 + \frac{\pi}{4}) & -\frac{1}{2} \frac{\sin(\theta_1)}{\tan(\theta_2)} - \frac{1}{\sqrt{2}} \sin(\theta_1 + \frac{\pi}{4}) \end{bmatrix} \quad (4.8)$$

This is only a valid transformation when $s_2 > 0 \implies \frac{\sin(\theta_1)}{\sin(\theta_2)} < 0 \implies \sin(\theta_1) \sin(\theta_2) < 0$. \square

A similar larger set of cooperative common-payoff games, $G_p(a) = G_q(a) \forall p, q \in [1, N]$, can also be derived using invariant transforms.

Definition 4.3.4 (Invariant-Common-Payoff Games). Invariant-common-payoff games are those that can be mapped to common-payoff games using equilibrium-invariant transformations, $G_p(a) \rightarrow \hat{G}_p(a) = s_p G_p(a) + b_p(a_{-p})$.

These games correspond exactly to the set of 2x2 weighted potential games (Monderer and Shapley, 1996), where the common payoff acts as the potential function. The off-diagonal quadrants (top-right and bottom-left) in the visualization are *invariant-common-payoff*.

Theorem 4.3.5 (Invariant-Common-Payoff Quadrants). When $\sin(\theta_1) \sin(\theta_2) > 0$ in Equations (4.1a) and (4.1b), the respective game $(\hat{G}_1(\theta_1), \hat{G}_2(\theta_2))$ is invariant-common-payoff.

Proof. For a game to be invariant-common-payoff there has to exist invariant transforms $s_2 > 0$, $b_1(a_2)$, and $b_2(a_1)$ such that $G_1(a_1, a_2) + b_1(a_2) = s_2 G_2(a_1, a_2) + b_2(a_1)$. Consider the transforms $s_2 = \frac{\sin(\theta_1)}{\sin(\theta_2)}$, $b_1 = [\frac{1}{2} \frac{\sin(\theta_1)}{\tan(\theta_2)}, -\frac{1}{2} \frac{\sin(\theta_1)}{\tan(\theta_2)}]$, and $b_2 = [\frac{1}{2} \cos(\theta_1), -\frac{1}{2} \cos(\theta_1)]$, which result in payoffs:

$$\hat{G}_1(\theta_1, \theta_2) = \hat{G}_2(\theta_1, \theta_2) = \begin{bmatrix} \frac{1}{2} \frac{\sin(\theta_1)}{\tan(\theta_2)} + \frac{1}{\sqrt{2}} \sin(\theta_1 + \frac{\pi}{4}) & -\frac{1}{2} \frac{\sin(\theta_1)}{\tan(\theta_2)} + \frac{1}{\sqrt{2}} \sin(\theta_1 + \frac{\pi}{4}) \\ \frac{1}{2} \frac{\sin(\theta_1)}{\tan(\theta_2)} - \frac{1}{\sqrt{2}} \sin(\theta_1 + \frac{\pi}{4}) & -\frac{1}{2} \frac{\sin(\theta_1)}{\tan(\theta_2)} - \frac{1}{\sqrt{2}} \sin(\theta_1 + \frac{\pi}{4}) \end{bmatrix} \quad (4.9)$$

This is only a valid transformation when $s_2 > 0 \implies \frac{\sin(\theta_1)}{\sin(\theta_2)} > 0 \implies \sin(\theta_1) \sin(\theta_2) > 0$. \square

The game at the origin (named \updownarrow Dominant) is both invariant-zero-sum and invariant-common-payoff. Along with the permuted versions of this game, this is the only game with this property.

Theorem 4.3.6 (Dominant Special Case). *When $\sin(\theta_1) = 0$ and $\sin(\theta_2) = 0$ in Equations (4.1a) and (4.1b), the respective game $(\hat{G}_1(\theta_1), \hat{G}_2(\theta_2))$ is both invariant-zero-sum and invariant-common-payoff.*

Proof. Consider when $\theta_1 = \theta_2 = 0$. When using transforms $b_1 = [-\frac{1}{2}, \frac{1}{2}]$ and $b_2 = [-\frac{1}{2}, \frac{1}{2}]$ the game becomes zero-sum (Equation (4.10a)). Additionally, when using transforms $b_1 = [\frac{1}{2}, -\frac{1}{2}]$ and $b_2 = [\frac{1}{2}, -\frac{1}{2}]$ the game becomes common-payoff (Equation (4.10b)).

$$\hat{G}_1 = -\hat{G}_2 = \begin{bmatrix} 0 & \frac{1}{2} \\ -\frac{1}{2} & 1 \end{bmatrix} \quad (4.10a)$$

$$\hat{G}_1 = \hat{G}_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (4.10b)$$

\square

Therefore all symmetric 2x2 games are either invariant-zero-sum, invariant-common-payoff, or both. The borders between the quadrants are neither invariant-zero-sum or invariant-common-payoff. \updownarrow Dominant is the only game which is both invariant-zero-sum and invariant-common-payoff.

4.3.3 2x2 Equilibrium-Invariant Distance Metric

Theorem 4.3.7 (2x2 Equilibrium-Invariant Distance Metric). *The distance metric between two 2x2 games is given by:*

$$d(\Theta^A, \Theta^B) = ||[\min(|\theta_1^A - \theta_1^B|, 2\pi - |\theta_1^A - \theta_1^B|), \min(|\theta_2^A - \theta_2^B|, 2\pi - |\theta_2^A - \theta_2^B|)]||_p \quad (4.11)$$

Proof. Consider player 1's pre-norm component of the distance metric (Equation (3.4)) between two parameterized games $\hat{G}_1^A(\theta_1^A)$ and $\hat{G}_1^B(\theta_1^B)$ (Equation (4.1a)).

$$v_1 = \arccos\left(\sin(\theta_1^A + \frac{\pi}{4})\sin(\theta_1^B + \frac{\pi}{4}) + \cos(\theta_1^A + \frac{\pi}{4})\cos(\theta_1^B + \frac{\pi}{4})\right) \quad (4.12a)$$

$$= \arccos(\cos(\theta_1^A - \theta_1^B)) \quad (4.12b)$$

$$= \min(|\theta_1^A - \theta_1^B|, 2\pi - |\theta_1^A - \theta_1^B|) \quad (4.12c)$$

A similar calculation can be made for player 2. Together, this results in Equation (4.11). \square

This definition of a distance metric on the 2x2 equilibrium-invariant embedding of games is natural. As established in Theorem 4.3.1, games with the same equilibria can be embeddings concisely represented by points on two independent unit-circles or equivalently by their angles, i.e., $\Theta = (\theta_1, \theta_2)$. Distance along a unit-circle is measured by arc length. Therefore, the natural way to measure distance between two games is to sum the arc lengths between their embeddings on both circles. More generally, the distances between games' embedding can be measured using any p -norm. In summary, the distance, d , between two games can be found by a) converting the game to the equilibrium-invariant embedding, b) calculating the arc length between each component of the two embeddings, and c) finding a distance using any p -norm, where $\Theta^A = (\theta_1^A, \theta_2^A)$ and $\Theta^B = (\theta_1^B, \theta_2^B)$.

For all norms with $p < \infty$, there is a unique game which maximizes the distance to any other game. Let this game be defined the *opposite game*, which can be calculated by translating the representation by a constant of π around the circle.

$$\theta_1 \rightarrow \text{mod}(\theta_1 + 2\pi, 2\pi) - \pi \quad \theta_2 \rightarrow \text{mod}(\theta_2 + 2\pi, 2\pi) - \pi \quad (4.13)$$





Symmetry	Transformation Description	Transformation
Player 1 Strategy Permutation	Translate on θ_1 , fold over $\theta_1 = 0$	$(\theta_1, \theta_2) \rightarrow (\theta_1 + \pi, -\theta_2)$
Player 2 Strategy Permutation	Fold over $\theta_2 = 0$, translate on θ_2	$(\theta_1, \theta_2) \rightarrow (-\theta_1, \theta_2 + \pi)$
Player Permutation	Fold over $\theta_1 = \theta_2$	$(\theta_1, \theta_2) \rightarrow (\theta_2, \theta_1)$

Table 4.2: Symmetries in 2x2 normal-form games used to derive the equilibrium-symmetric embedding.**Algorithm 4.1** Equilibrium-Symmetric Embedding

```

1:  $\theta_1^*, \theta_2^* \leftarrow \theta_1, \theta_2$ 
2: if  $\theta_1^* \leq -\frac{\pi}{2}$  or  $\frac{\pi}{2} < \theta_1^*$  then                                ▷ Permute Player 1's Strategies
3:    $(\theta_1^*, \theta_2^*) \leftarrow (\theta_1^* + \pi, -\theta_2^*)$ 
4: if  $\theta_2^* \leq -\frac{\pi}{2}$  or  $\frac{\pi}{2} < \theta_2^*$  then                                ▷ Permute Player 2's Strategies
5:    $(\theta_1^*, \theta_2^*) \leftarrow (-\theta_1^*, \theta_2^* + \pi)$ 
6: if  $\theta_2^* > \theta_1^*$  then                                                ▷ Permute Players
7:    $(\theta_1^*, \theta_2^*) \leftarrow (\theta_2^*, \theta_1^*)$ 
8: return  $(\theta_1^*, \theta_2^*)$ 

```

For example, the opposite of a  clockwise Cycle game is an  anti-clockwise Cycle game, and the opposite of a  diagonal Coordination game is an  off-diagonal (“anti”) Coordination game.

4.4 2x2 Equilibrium-Symmetric Embedding

Two common symmetries are strategy permutation and player permutation. The order of strategies and players in normal-form games is arbitrary, and permuting along these dimensions leads to strategically identical games. The area of the equilibrium-invariant embedding can be reduced by considering these symmetries (Table 4.2). This reduced space, called the *equilibrium-symmetric embedding*, is shaded in the visualization (Figure 4.3). It is easy to convert from the equilibrium-invariant embedding to the equilibrium-symmetric embedding (Algorithm 4.1). If the roles of the players are important, only strategy permutation may be used, and the equilibrium-symmetric embedding will only reduce by a factor of four. The symmetries being utilized are usually clear from the context.

4.4.1 Deriving the 2x2 Equilibrium-Symmetric Embedding

Theorem 4.4.1 (Equilibrium-Symmetric Embedding). *Strategy permutation and player permutation can reduce the area of the equilibrium-invariant embedding by a factor of eight when only considering a canonical ordering.*

Proof Sketch. Symmetries (Table 4.2), including player permutation and strategy permutation for each player, can be leveraged to reduce the representation space. Each of these reductions reduces the representation area by a factor of two, and when composed, results in a factor of eight reduction. \square

4.4.2 2x2 Equilibrium-Symmetric Player-Agnostic Embedding

Define a new parameterization of the equilibrium-invariant embedding, $(\phi_{\text{cycle}}, \phi_{\text{coord}})$, which is simply a change in basis from (θ_1, θ_2) . In the equilibrium-symmetric embedding, $0 \leq \phi_{\text{cycle}} \leq 1$ and $-1 < \phi_{\text{coord}} \leq 1$, with $\phi_{\text{cycle}} \pm \phi_{\text{coord}} \leq 1$. This basis describes properties of the game, rather than the payoff of each player. The conversion from the per-player basis and the game-property basis is simple.

$$\phi_{\text{coord}} = \frac{1}{\pi} (\theta_1 + \theta_2) \quad (4.14a) \quad \theta_1 = \frac{\pi}{2} (\phi_{\text{coord}} + \phi_{\text{cycle}}) \quad (4.15a)$$

$$\phi_{\text{cycle}} = \frac{1}{\pi} (\theta_1 - \theta_2) \quad (4.14b) \quad \theta_2 = \frac{\pi}{2} (\phi_{\text{coord}} - \phi_{\text{cycle}}) \quad (4.15b)$$

Therefore games can be described by their coordination (“common-payoff-ness”) and “cyclicness” (“zero-sum-ness”) respectively. The game at the origin, with zero coordination and zero cyclicness, is transitive (the $\begin{smallmatrix} \rightarrow & \rightarrow \\ \leftarrow & \leftarrow \end{smallmatrix}$ Dominant game).

$$\hat{G}_1(\phi_{\text{coord}}, \phi_{\text{cycle}}) = \begin{bmatrix} \frac{1}{\sqrt{2}} \sin\left(\frac{\pi}{2}(\phi_{\text{coord}} + \phi_{\text{cycle}}) + \frac{\pi}{4}\right) & -\frac{1}{\sqrt{2}} \cos\left(\frac{\pi}{2}(\phi_{\text{coord}} + \phi_{\text{cycle}}) + \frac{\pi}{4}\right) \\ -\frac{1}{\sqrt{2}} \sin\left(\frac{\pi}{2}(\phi_{\text{coord}} + \phi_{\text{cycle}}) + \frac{\pi}{4}\right) & -\frac{1}{\sqrt{2}} \cos\left(\frac{\pi}{2}(\phi_{\text{coord}} + \phi_{\text{cycle}}) + \frac{\pi}{4}\right) \end{bmatrix} \quad (4.16a)$$

$$\hat{G}_2(\phi_{\text{coord}}, \phi_{\text{cycle}}) = \begin{bmatrix} \frac{1}{\sqrt{2}} \sin\left(\frac{\pi}{2}(\phi_{\text{coord}} - \phi_{\text{cycle}}) + \frac{\pi}{4}\right) & -\frac{1}{\sqrt{2}} \sin\left(\frac{\pi}{2}(\phi_{\text{coord}} - \phi_{\text{cycle}}) + \frac{\pi}{4}\right) \\ \frac{1}{\sqrt{2}} \cos\left(\frac{\pi}{2}(\phi_{\text{coord}} - \phi_{\text{cycle}}) + \frac{\pi}{4}\right) & -\frac{1}{\sqrt{2}} \cos\left(\frac{\pi}{2}(\phi_{\text{coord}} - \phi_{\text{cycle}}) + \frac{\pi}{4}\right) \end{bmatrix} \quad (4.16b)$$

4.4.3 2x2 Equilibrium-Symmetric Distance Metric

A distance metric between games in the symmetric embedding can also be defined by enumerating all (up to eight) equivalent isomorphic games and finding the minimum distance between the closest pair. For example, some of the distances between points in the symmetric embedding are shown in Table 4.4.

4.5 2x2 Best-Response-Invariant Embedding

Using the better-response-invariant embedding (Definition 3.3.6) a set of 15 fundamental games is derived. For two-strategy games, the concepts of better-response and best-response are synonymous. Therefore this set of games is more simply defined as the 2x2 best-response-invariant embedding. Accompanying this embedding are 15 equivalence classes: many equilibrium-invariant embeddings map to each best-response-invariant embedding. This distinction is subtle but important. Sometimes this work refers to best-response-invariant embeddings, and sometimes this work refers to the equivalence classes they belong to. The best-response-invariant embeddings can be thought of as canonical (fundamental) examples of games within each equivalence class. The equilibria (WSNE, NE, WSCE, CE, and CCE) of all games within each equivalence class can be calculated through simple scaling of the equilibria of these fundamental games. Of the 15 equivalent game classes, 11 are nontrivial, 3 are partially trivial, and 1 is trivial. These 15 classes are the same as those proposed by [Borm \(1987\)](#) and correspond to games with the same best-response dynamics. This work improves upon Borm’s classification because in this work’s derivation, the best-response-invariant embedding inherits the distance metric from the equilibrium-invariant embedding. This allows us to situate these game classes within a metric space, which greatly improves the clarity of the embeddings.

4.5.1 Deriving the Best-Response Embedding

Theorem 4.5.1 (2x2 Best-Response Embedding). *All 2x2 game payoffs can be mapped to a fundamental set consisting of 81 games. After symmetry this reduces to 11 nontrivial games, 3 partially trivial games, and 1 trivial game, resulting in 15 fundamental games. One such mapping is:*

$$G_1 = \begin{bmatrix} g_1^{AA} & g_1^{AB} \\ g_1^{BA} & g_1^{BB} \end{bmatrix} \rightarrow \left\{ \begin{bmatrix} +1 & +1 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} +1 & 0 \\ -1 & 0 \end{bmatrix}, \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}, \right. \\ \left. \begin{bmatrix} 0 & +1 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 0 & +1 \end{bmatrix}, \right. \\ \left. \begin{bmatrix} -1 & +1 \\ +1 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ +1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & -1 \\ +1 & +1 \end{bmatrix} \right\} \quad (4.17a)$$

$$G_2 = \begin{bmatrix} g_2^{AA} & g_2^{AB} \\ g_2^{BA} & g_2^{BB} \end{bmatrix} \rightarrow \left\{ \begin{bmatrix} +1 & -1 \\ +1 & -1 \end{bmatrix}, \begin{bmatrix} +1 & -1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}, \right. \\ \left. \begin{bmatrix} 0 & 0 \\ +1 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ -1 & +1 \end{bmatrix}, \right. \\ \left. \begin{bmatrix} -1 & +1 \\ +1 & -1 \end{bmatrix}, \begin{bmatrix} -1 & +1 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} -1 & +1 \\ -1 & +1 \end{bmatrix} \right\} \quad (4.17b)$$

Proof. Using the per-strategy scale transform, with b_1 defined as the usual zero mean offset and $s_2 = [|g_1^{A-B,A}|, |g_1^{A-B,B}|]$, the player embedding for player 1 is derived.

$$G_1 = \begin{bmatrix} g_1^{AA} & g_1^{AB} \\ g_1^{BA} & g_1^{BB} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{1}{2}g_1^{A-B,A} & \frac{1}{2}g_1^{A-B,B} \\ -\frac{1}{2}g_1^{A-B,A} & -\frac{1}{2}g_1^{A-B,B} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{g_1^{A-B,A}}{|g_1^{A-B,A}|} & \frac{g_1^{A-B,B}}{|g_1^{A-B,B}|} \\ -\frac{g_1^{A-B,A}}{|g_1^{A-B,A}|} & -\frac{g_1^{A-B,B}}{|g_1^{A-B,B}|} \end{bmatrix} \quad (4.18a)$$

$$= \begin{bmatrix} \{+1, +1, +1, 0, 0, 0, -1, -1, -1\} & \{+1, 0, -1, +1, 0, -1, +1, 0, -1\} \\ \{-1, -1, -1, 0, 0, 0, +1, +1, +1\} & \{-1, 0, +1, -1, 0, +1, -1, 0, +1\} \end{bmatrix} \quad (4.18b)$$

Dividing a number by its absolute value results in its sign, and adopting the convention (without consequence) that $\frac{0}{|0|} = 0$, there are 9 payoff possibilities, one of which is the trivial zero payoff. A similar derivation can be followed for player 2. As a result, $9 \times 9 = 81$ games are possible in total. Of these 64 are nontrivial, 16 are partially trivial and 1 is trivial. The permutation symmetries discussed earlier facilitate the observation that only 15 of the games are unique up to symmetric equivalence. \square

The best-response-invariant embedding set could have been defined using binary payoffs (Fishburn and Kilgour, 1990) because it is only necessary to establish preferences over the strategies. However this work opts for ternary payoffs $(-1, 0, +1)$ because they more clearly differentiate the three possibilities: preferring strategy A , preferring strategy B , or being indifferent. This preference ordering is equivalent to the best-response dynamics of each player. For two-strategy games, better-response and best-response dynamics are identical. Better-response invariance is closely related to the affine transform used to derive the equilibrium-invariant embedding (Morris and Ui, 2004; Ostrovski, 2013). Equilibrium-invariance implies better-response-invariance, which in turn implies best-response-invariance.

4.5.2 Equivalence Classes, Graph Representation, and Measure

The equivalence classes of the best-response-invariant embedding are equivalent to considering orderings of each column in player 1's payoff, and each row in player 2's payoff (the best-response dynamics). By convention let the top-left joint strategy be AA and the bottom right be BB , let player 1 be the row player and player 2 be the column player. This ordering can be simply visualized as a directed graph: player 1's column ordering can be indicated with vertical arrows (for example $\downarrow \uparrow$ indicates player 1 prefers strategy B when player 2 plays A , and A when player 2 plays B), and player 2's row ordering can be indicated with horizontal arrows (for example $\rightarrow \leftarrow$). When there is no preference, no edge need be drawn, $\cdot \cdot$. Taken together, any 2x2 game can be represented. For example, Prisoner's Dilemma (Figure 4.1b) would be denoted $\begin{smallmatrix} \rightarrow & \rightarrow \\ \downarrow & \downarrow \end{smallmatrix}$, Chicken (Figure 4.1a), would be denoted $\begin{smallmatrix} \rightarrow & \leftarrow \\ \downarrow & \downarrow \end{smallmatrix}$, and Matching Pennies (Figure 4.1e), would be denoted $\begin{smallmatrix} \rightarrow & \rightarrow \\ \downarrow & \uparrow \end{smallmatrix}$. The set of directed graphs have 15 equivalence classes with respect to graph isomorphism.

Most of the games in the best-response-invariant embedding have measure-zero equivalence classes within the equilibrium-invariant embedding, meaning the probability they are randomly sampled is zero. There is a simple rule of thumb to determine the measure of an equivalence class: a) if no players are indifferent when best-responding, the game is positive measure, b) if one player is indifferent when best responding to one action, the class is zero-measure and is a "boundary class" appearing between two positive

G	Name	N	PN	MN	C	B	θ	ϕ	S	M	T	I	Other Names
	Dominant	Do				5	0, 0	0, 0	✓	$\frac{1}{4}$	N	0	Prisoner's Dilemma
	Coordination	Co				14	$\frac{\pi}{2}, \frac{\pi}{2}$	1, 0	✓	$\frac{1}{8}$	N	0	Battle, Hunt, Chicken
	Cycle	Cy				15	$\frac{\pi}{2}, -\frac{\pi}{2}$	0, 1		$\frac{1}{8}$	N	0	Matching Pennies
	Samaritan	Sm				8	$\frac{\pi}{2}, 0$	$\frac{1}{2}, \frac{1}{2}$		$\frac{1}{2}$	N	0	
	Hazard	Hz				13	$\frac{\pi}{2}, -\frac{\pi}{4}$	$\frac{1}{4}, \frac{3}{4}$		—	N	1	
	Safety	Sf				12	$\frac{\pi}{2}, \frac{\pi}{4}$	$\frac{3}{4}, \frac{1}{4}$		—	N	1	
	Aidos	Ad				7	0, $-\frac{\pi}{4}$	$-\frac{1}{4}, -\frac{1}{4}$		—	N	1	
	Picnic	Pn				6	$\frac{\pi}{2}, 0$	$\frac{1}{4}, \frac{1}{4}$		—	N	1	
	Daredevil	Dd				11	$-\frac{\pi}{4}, -\frac{\pi}{4}$	$-\frac{1}{2}, 0$	✓	·	N	2	
	Fossick	Fo				9	$\frac{\pi}{4}, \frac{\pi}{4}$	$\frac{1}{2}, 0$	✓	·	N	2	
	Heist	Hs				10	$\frac{\pi}{4}, -\frac{\pi}{4}$	0, $\frac{1}{2}$		·	N	2	
	Ignorance	Ig				2				$\frac{1}{2}$	P	2	
	Horseplay	Hp				4				$\frac{1}{2}$	P	2	
	Dress	Dr				3				—	P	3	Red Dress
	Null	Nu				1			✓	1	T	4	Trivial, Zero

Table 4.3: Naming scheme and properties of the set of 2x2 best-response-invariant embeddings. Only Dominant, Coordination, Cycle, Samaritan, and Null have been well studied and have established names. The remaining games (which have indifferences) either do not appear in the literature at all or do so seldomly. Key: graph representation (G), short name (N), pure Nash (PN), mixed Nash (MN), correlated equilibrium support (C), [Borm \(1987\)](#)'s classes (B), symmetric (S), measure (M), triviality (T), and indifferences (I).

measure classes, and c) if both players are indifferent to one of the opponent's actions, the class is a "point class" appearing between four boundary classes. It is easy to visually inspect (Figure 4.6) that only 4 of the games have positive probability of being sampled: Samaritan, Dominant, Coordination, and Cycle (Table 4.3). This provides intuition to why the measure-zero games rarely appear in the literature. This observation matches similar analysis made by [Borm \(1987\)](#).

4.5.3 Naming the Best-Response-Invariant Embedding

[Borm \(1987\)](#) provided numerical classification for the 15 fundamental classes of games. This chapter has already specified parameterized embeddings, (θ_1, θ_2) or $(\phi_{\text{coord}}, \phi_{\text{cycle}})$, and a graphical representation. In addition, the set is small enough to benefit from a standardised naming scheme. This work attempts to, as far as possible, follow established naming conventions and draw inspiration from previous work ([Bruns, 2015](#)). Symmetric games (those that lie on $\theta_1 = \theta_2$ or $\phi_{\text{cycle}} = 0$) are the most well-studied and have established common names. Only some non-symmetric games such as Matching Pennies and Samaritan's Dilemma have been studied. Care was taken not to name classes after games that are similar but subtly different (e.g. Stag Hunt and Chicken were deliberately avoided). The majority of the games have strategies which players are indifferent between. These classes of games are little studied and therefore do not have common names. Table 4.3 shows the names chosen for the 15 fundamental games proposed in this work.

Dominant

In this symmetric game (Figure 4.4a) each player has a strictly dominant strategy. Therefore, this game has a single pure NE and (C)CE . The Dominant embedding is at the origin of the equilibrium-invariant embedding, like other topologies ([Rapoport et al., 1976](#)), and is the only 2x2 game that is both invariant-zero-sum and invariant-common-payoff. Dominant has many common games within its equivalence class including Prisoner's Dilemma, Peace, Deadlock, Total Conflict, Concord, and Compromise. In particular, Prisoner's Dilemma is one of the most studied games in economics. The dominant equivalence class occurs with probability $\frac{1}{4}$ when sampling uniformly over the equilibrium-invariant embedding. Furthermore, this class is the most diverse: games within the class can be anti-clockwise invariant-zero-sum, clockwise

S			W			A			B			H			T			H			N			R			C		
S	+1, +1	+1, -1	A	+1, +1	-1, -1	H	+1, -1	-1, +1	R	4, 3	1, 1	N	0, +1	0, +1															
W	-1, +1	-1, -1	B	-1, -1	+1, +1	T	-1, +1	+1, -1	W	3, 4	2, 2	I	-2, +3	+2, +2															
(a) Dominant			(b) Coordination			(c) Cycle			(d) Samaritan			(e) Hazard																	
C			R			B			P			T			B			N			W			G					
I	+1, +4	-2, +3	N	+1, 0	+1, 0	A	+1, +1	0, -1	N	0, 0	+1, 0	W	+1, 0	+1, 0															
N	0, +1	0, +1	R	-1, +1	-1, -1	D	-1, +1	0, -1	R	0, +1	-1, -1	G	-1, +1	-1, -1															
(f) Safety			(g) Aidos			(h) Picnic			(i) Daredevil			(j) Fossick																	
P			R			A			B			N			C			F			A			B					
N	+1, 0	0, 0	W	+1, 0	+1, 0	N	+1, 0	+1, 0	C	-2, 0	+1, 0	A	0, 0	0, 0															
H	-1, +1	0, -1	L	-1, 0	-1, 0	R	-1, +1	-1, -1	F	-2, +1	-1, +1	B	0, 0	0, 0															
(k) Heist			(l) Ignorance			(m) Horseplay			(n) Dress			(o) Null																	

Figure 4.4: Narrative payoff tables of the 15 best-response-invariant 2x2 games.

invariant-zero-sum, diagonal invariant-common-payoff, off-diagonal invariant-common-payoff, or none.

Coordination

Coordination (Figure 4.4b) is a symmetric common-payoff coordination game where there are two equally desirable outcomes. Players simply have to coordinate to ensure they achieve one of them. Coordination has two pure NEs (and) with equal payoff and a low payoff mixed NE . It is the only nontrivial game with a positive volume of (C)CE equilibria. Coordination has many games within its equivalence class including Stag Hunt, Chicken, Assurance, and Bach or Stravinsky. The Coordination equivalence class occurs with probability $\frac{1}{8}$ when sampling uniformly over the equilibrium-invariant embedding. All games in the coordination equivalence class are invariant-common-payoff.

Cycle

This game is an anti-symmetric zero-sum game (Figure 4.4c), also commonly called Matching Pennies. In this game, the first player prefers both players to play the same strategy (for example, both heads or both tails), and the second player prefers each player to play a different strategy. The best-responses of the pure strategies result in cyclic dynamics. A similar three strategy variant of this game with the same property, Rock Paper Scissors, is a popular children's game. Permuting the strategies of a player will result in changing the game from a clockwise cycle to an anti-clockwise cycle. Cycle has no pure NEs, a single completely mixed NE (with uniform probability) and (C)CE . Cycle is the only game where all equilibria are strictly in the interior of the simplex. All games within Cycle's best-response-invariant equivalence class are also cyclic, but with different biases on the strategies they prefer. This class occurs with probability $\frac{1}{8}$ when sampling uniformly over the equilibrium-invariant embedding.

Samaritan

This game (Figure 4.4d) has a worker player and a Samaritan player. The worker has two strategies: rest (R) or work (W). The Samaritan has two strategies: help (H) or do not help (N). The Samaritan always prefers to help but also prefers it when the worker also works (and so does not free-load). The worker prefers not to work if they receive help but prefers to work to sustain themselves if they do not receive help. Samaritan has a single pure NE and (C)CE . It is named after Samaritan's Dilemma (Schmidtchen, 2002), a game proposed by Buchanan (1975). Samson (Brams, 1993), Alibi (Robinson and Goforth, 2005), Anticipation, Bully, Hamlet, Asymmetric Dilemma, and Called Bluff are all in this game's better-response-invariant equivalence class. The Samaritan embedding is neither zero-sum-invariant nor common-payoff-invariant. However other games within its equivalence class can be either zero-sum-invariant or common-payoff-invariant, or neither. Samaritan equivalence class is special because it occurs with the highest probability $\frac{1}{2}$ when sampling uniformly over the equilibrium-invariant embed-

ding. As well as being the most common class, it is also the class most closely connected to all other games (Section 4.5.4).

⚡ Hazard

In Hazard (Figure 4.4e), player 1 is an insurance company that can either insure (I) or decline (D) player 2's health insurance. Player 2 enjoys drinking at parties, which is risky behaviour (R), but also appreciates the health benefits of not drinking, which is careful behaviour (C). Overall, in the absence of health insurance, player 2 is indifferent to a party lifestyle or a healthy one. However, if player 1 issues medical insurance, player 2 will have a greater risk tolerance and will adopt a party lifestyle. Player 1 does not make any money if it does not issue insurance, makes money if it does and player 2 opts for a healthy lifestyle, and loses money if it issues insurance to an unhealthy lifestyle. Bruns (2015) briefly defines a game that he describes as a moral hazard, with the formulaic name "Middle Hunt \times Low Dilemma". Hazard has a single pure NE \square , a mixed NE \square , and (C)CEs with support \square . The hazard equivalence class has an indifference and so is zero-measure in the equilibrium-invariant embedding. However it is a boundary class which borders the Cycle equivalence class. Of the boundary classes it occurs with probability $\frac{1}{4}$ when uniformly sampling over the equilibrium-invariant embedding. All games within this equivalence class are invariant-zero-sum.

⚡ Safety

This game (Figure 4.4f) is also hinted at by Bruns (2015) in which it was called "Middle Hunt \times Low Concord". It is similar to Hazard, except the insurer now incentivizes player 1 into a healthy lifestyle with free gym membership. This fixes the moral hazard and now player 2 is incentivized to have a healthy lifestyle with insurance. Safety has two pure NEs (\square and \square), a mixed NE \square and (C)CEs \square . The Safety equivalence class has an indifference and so is zero-measure in the equilibrium-invariant embedding. However it is a boundary class which borders the Coordination equivalence class. Of the boundary classes it occurs with probability $\frac{1}{4}$ when uniformly sampling over the equilibrium-invariant embedding. All games within this equivalence class are invariant-common-payoff.

⚡ Aidos

In Aidos (Figure 4.4g), player 1 is a deity and can either reveal themselves (R) or not (N), however they are shy and strictly prefer not to reveal themselves. Player 2 is a human and wants to believe what is true. With lack of evidence either way, the human is indifferent to the existence of the deity, however if the deity reveals themselves they prefer to believe (B) rather than not believe (N). There are two pure NEs (\square and \square), and any mixture of these is also a mixed NE \square and (C)CE. The theme of this game is inspired by Revelation (Brams, 1993), a game with similar dynamics. The Aidos equivalence class has an indifference and so is zero-measure in the equilibrium-invariant embedding. However it is a boundary class which borders the Dominant and Samaritan equivalence classes in the region where the pure equilibrium strategy changes (in contrast to Picnic, described next). Of the boundary classes it occurs with probability $\frac{1}{4}$ when uniformly sampling over the equilibrium-invariant embedding. The games in Aidos' equivalence class can be either zero-sum-invariant, common-payoff-invariant, or neither. The only game in the set that is neither is the Aidos embedding.

⚡ Picnic

In this game (Figure 4.4h) a host can either organise a picnic (P) or order takeaway (T) and they strictly prefer organising a picnic. A guest can either attend (A) or decline (D). They are indifferent about attending when takeaway is ordered but would prefer to attend if a picnic is organised. Picnic has a single pure NE \square . The Picnic equivalence class has an indifference and so is zero-measure in the equilibrium-invariant embedding. However it is a boundary class which borders the Dominant and Samaritan equivalence classes in the region where the pure equilibrium strategy does not change (in contrast to Aidos). Of the boundary classes it occurs with probability $\frac{1}{4}$ when uniformly sampling over the equilibrium-invariant embedding.

The Picnic embedding is neither zero-sum-invariant nor common-payoff-invariant, however games in its equivalence class can be either zero-sum-invariant or common-payoff-invariant.

↖ ↗ **Daredevil**

In the literature, ↖ ↗ Chicken (Figure 4.1a) has two pure anti-coordination NEs (↖ and ↗) and a single mixed NE ↘ which places the majority of the mass on the worst joint outcome. The presence of a mixed NE means that Chicken has a full-support equilibrium and therefore falls into the equivalence class of ↖ ↗ Coordination. However, there is another interesting symmetric, invariant-common-payoff, measure-zero equivalence class game, Daredevil (Figure 4.4i), with properties similar to Chicken. This game has the same two pure anti-coordination NEs (↖ and ↗), and an additional pure NE ↘: crash. There is no completely mixed NE, but there are also two pure-mixed NEs (↖↘ and ↗↘). It also has (C)CEs ↘ which can mix arbitrarily over any of the pure NEs. The game differs from Chicken in the fact that if the other player chooses not to swerve, the player is indifferent to whether they swerve and avoid damage or continue and crash. It has similar dynamics to, and can be thought of as, an extreme edge-case of Chicken. This game has not been studied before in the literature - perhaps unsurprising because it has a measure-zero equivalence class. Daredevil is a point class because the Daredevil embedding is the only game within its class. It is invariant-common-payoff and borders the Aidos and Safety equivalence classes. Of the three point classes, it occurs with probability $\frac{1}{4}$.

↖ ↗ **Fossick**

In the literature, ↖ ↗ Stag Hunt (Figure 4.1c) is a game with two pure coordination NEs (↖ and ↗), where one is preferred over the other, and a completely mixed NE ↘. Stag Hunt is also in the equivalence class of ↖ ↗ Coordination. Fossick (Figure 4.4j) is an extreme version of Stag Hunt which has two pure NEs (↖ and ↗), but no mixed NE. It also has (C)CEs that can mix over only the coordination strategies ↖. In this game players are in a gold rush and can either search for water to sustain themselves in the wilderness or fossick for gold. If the other player searches for water, the player is indifferent to what they search for. However if the other player fossicks for gold, the player would feel left out and would prefer to also fossick for gold. This game has not been studied before in the literature. Fossick is part of a zero-measure equivalence class, and is also a point class (the Fossick embedding is the only game in the equivalence class). Of the three point classes, it occurs with probability $\frac{1}{4}$. Fossick is invariant-common-payoff.

↖ ↗ **Heist**

In Heist (Figure 4.4k), player 1 is a nervous robber and can either do nothing (N) or stage a heist (H). Player 2 is a security guard and can either go on patrol (P) or rest (R). If the robber does not stage a heist, the security guard is indifferent to whether they are on patrol or at rest. However if a heist does occur the guard prefers to be on patrol. If the security guard is at rest, the robber is unsure if the heist is worth the risk and is indifferent. However if the security guard is on patrol, the robber prefers not to stage a heist. This game is most similar to Pursuit (Brams, 1993). Heist has two pure NEs (↖ and ↗), and (C)CEs that mix between them ↘. Heist is part of a zero-measure equivalence class, and is also a point class. Of the three point classes, Heist is the most common and occurs with probability $\frac{1}{2}$. Heist is invariant-zero-sum.

↖ ↗ **Ignorance**

Ignorance (Figure 4.4l) is a partially trivial game where one of the players has no preferences at all. In Ignorance, an informed player wants to win (W) and avoid losing (L) and an ignorant player is indifferent between their strategies and outcomes. Ignorance has two pure NEs (↖ and ↗), a mixed NE ↘ and (C)CE ↘. The equivalence class of Ignorance occurs with probability $\frac{1}{2}$ when sampling uniformly over the partially-trivial equilibrium-invariant embedding.

↖ ↗ **Horseplay**

Horseplay (Figure 4.4m) is a partially trivial game where one of the players has no preferences at all.

Horseplay is a game between an adult and a child. The child can either lift up their arms (A) or curl into a ball (B), they are indifferent to which they do or what the outcome is, they are just happy to play. For convenience, the adult will prefer to throw (T) the child into the air if they have their arms up, or lift them up by their feet (L) if they are curled into a ball. Horseplay has two pure NEs ($\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$ and $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$), three mixed NEs ($\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$, $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$, and $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$), and full support (C)CEs $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$. The equivalence class of Horseplay occurs with probability $\frac{1}{2}$ when sampling uniformly over the partially-trivial equilibrium-invariant embedding.

† ∩ Dress

Described by [Simpson \(2010\)](#), Dress (Figure 4.4n) is a game where two people are going on a date. Player 1 selfishly wishes their partner to dress formally (F), and only if they do so, also lazily prefers to dress for comfort (C). Player 2 also wants their partner to dress formally, but is indifferent to what they wear themselves. Because player 2's preferences are solely based on the other player's strategies, their payoffs are trivial, so Dress is a partially trivial game. This game has three pure NEs, and any mixture between those three NEs ($\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$, $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$, and $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$), two mixed NEs ($\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$ and $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$), and (C)CEs $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$. The equivalence class of Dress is a boundary class between Ignorance and Horseplay. It is measure-zero when sampling uniformly over the partially-trivial equilibrium-invariant embedding.

∩ Null

In the Null game (Figure 4.4o), players have no preferences over strategies which means any pure or mixed strategy is an NE and any joint distribution is a (C)CE $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$. Games of the form $G_1(a_1, a_2) = b_1(a_2)$, $G_2(a_1, a_2) = b_2(a_1)$ are in the Null equivalence class. Null is a point game and is the only class of trivial games. This game is sometimes called the Zero game or Trivial game.

4.5.4 Distance Metric

Distances between the best-response-invariant embeddings can be computed (Table 4.4) using the equilibrium-symmetric distance metric. $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$ Cycle is the most isolated game with the greatest average distance to other games. $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$ Samaritan is the least isolated game with the smallest average distance to other games. No game is more than two steps⁶ away from $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$ Samaritan. Out of the partially trivial games, $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$ Dress is a boundary class between $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$ Ignorance and $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$ Horseplay. Small perturbations in payoffs can change the game and cause step changes in the equilibrium. The distance metrics (and the topology) show the possible adjacent games that small perturbations could result in. Such an analysis could be useful in selecting equilibria that are robust to permutations or dealing with uncertainty in payoffs. This idea is expanded on more in the discussion section.

4.6 Discussion

The novel per-strategy scale better-response-invariant transform can be used to derive a set of 2x2 best-response-invariant embeddings. After symmetry, this results in 15 equivalence classes. This same set has been identified previously by studying best-responses of 2x2 games ([Borm, 1987](#)). There is a deep connection between best-response-invariance and equilibrium-invariance ([Morris and Ui, 2004](#)). However, this work also provides an equilibrium-invariant embedding, distance measure, efficient parameterization, graph representation, and naming scheme for the best-response-invariant embeddings.

note that the majority of the games in the best-response-invariant embedding have indifferences (11 out of 15). Indifferences cannot be captured in ordinal payoffs: partially ordinal payoffs are required. However, popular topologies and classifications for 2x2 games only focus on ordinal games. This work argues that such a choice both a) limits the space of interesting games that are studied, and b) places too much prominence on games without indifferences which are highly redundant, comprising of only 4 out of 15 of the best-response-invariant equivalence classes.

⁶Steps or "hops" are measured over the planar grid visualized in Figure 4.3.

	Dominant	Coordination	Cycle	Samaritan	Hazard	Safety	Aidos	Picnic	Daredevil	Fossick	Heist	Ignorance	Horseplay	Dress	Null
Dominant	0	4	4	2	3	3	1	1	2	2	2				
Coordination	4	0	4	2	3	1	3	3	2	2	4				
Cycle	4	4	0	2	1	3	3	3	4	4	2				
Samaritan	2	2	2	0	1	1	1	1	2	2	2				
Hazard	3	3	1	1	0	2	2	2	3	3	1				
Safety	3	1	3	1	2	0	2	2	1	1	3				
Aidos	1	3	3	1	2	2	0	2	1	3	1				
Picnic	1	3	3	1	2	2	2	0	3	1	1				
Daredevil	2	2	4	2	3	1	1	3	0	2	2				
Fossick	2	2	4	2	3	1	3	1	2	0	2				
Heist	2	4	2	2	1	3	1	1	2	2	0				
Ignorance												0	2	1	
Horseplay												2	0	1	
Dress												1	1	0	
Zero															0

Table 4.4: Table showing the L_1 distance between games in the best-response-invariant embedding. Distances between nontrivial and trivial games are left undefined.

The 2×2 best-response-invariant embeddings can be used quickly to calculate equilibria for any 2×2 game, simply by storing the extreme points of the (C)CE polytope in a lookup table for the 15 best-response-invariant embeddings. The equilibria can be calculated for any 2×2 game by a) calculating the equilibrium-invariant embedding b) identifying the equivalence class the game belongs to, and c) scaling the equilibria in the lookup table according to Theorem 4.5.1.

Limitations

To motivate the metric spaces and embeddings this chapter focused on 2×2 games. This work derived an efficient parameterization of the 2×2 equilibrium-invariant embedding, which requires only two variables. This is achieved by making an assumption: only the equilibria, or similarly, the strategic interactions (best-response dynamics), of games are important. Most solution concepts are equilibrium or best-response based and the majority of the study of games is devoted to finding these solutions. However, this assumption does have a consequence: the preference ordering of *joint strategy* payoffs is not necessarily maintained after equilibrium-invariant transforms. This means that equilibrium selection methods such as maximum welfare could select for different equilibria in the equilibrium-invariant embedding⁷. Furthermore, the narrative of games that are motivated based on the ordering of joint payoffs may unravel. For example the most studied game, Prisoner's Dilemma, has dominant strategies that do not result in a welfare maximizing equilibrium. Prisoner's Dilemma is transformed to Dominant which is intuitive from an equilibrium preserving perspective, but less so from a welfare or social dilemma perspective. It is important to stress that this is a feature of the methods described in this work. If one is only concerned with the resulting rational behaviour of players, simplifying analysis of the game to only its equilibrium-invariant embedding or best-response-invariant embedding is a valuable way of removing redundant features of games.

⁷ Although, with knowledge of the mean and scale used to make the transformation, a trivial modification to the objective of the linear program could preserve the selection

4.7 Conclusion

This chapter explored equilibrium-invariant embeddings in 2×2 games which are ubiquitously studied in the literature. An efficient two variable parameterization of the 2×2 equilibrium-invariant embedding was uncovered. These variables geometrically represent angles on unit circles which allows them to be clearly visualized in two dimensions. Several properties, including equilibrium support, cyclicity, competitiveness, distances, and symmetries, can be read from this visualization. A new equilibrium-payoff transform was applied to 2×2 games that enabled further simplification, resulting in a rediscovery of a set of 15 equivalence classes of games. This set is fundamental, and covers all interesting 2×2 strategic interactions. Names and properties of these classes were explored. It is hoped that this work builds intuition for normal-form games, champions the set of 15 2×2 game classes

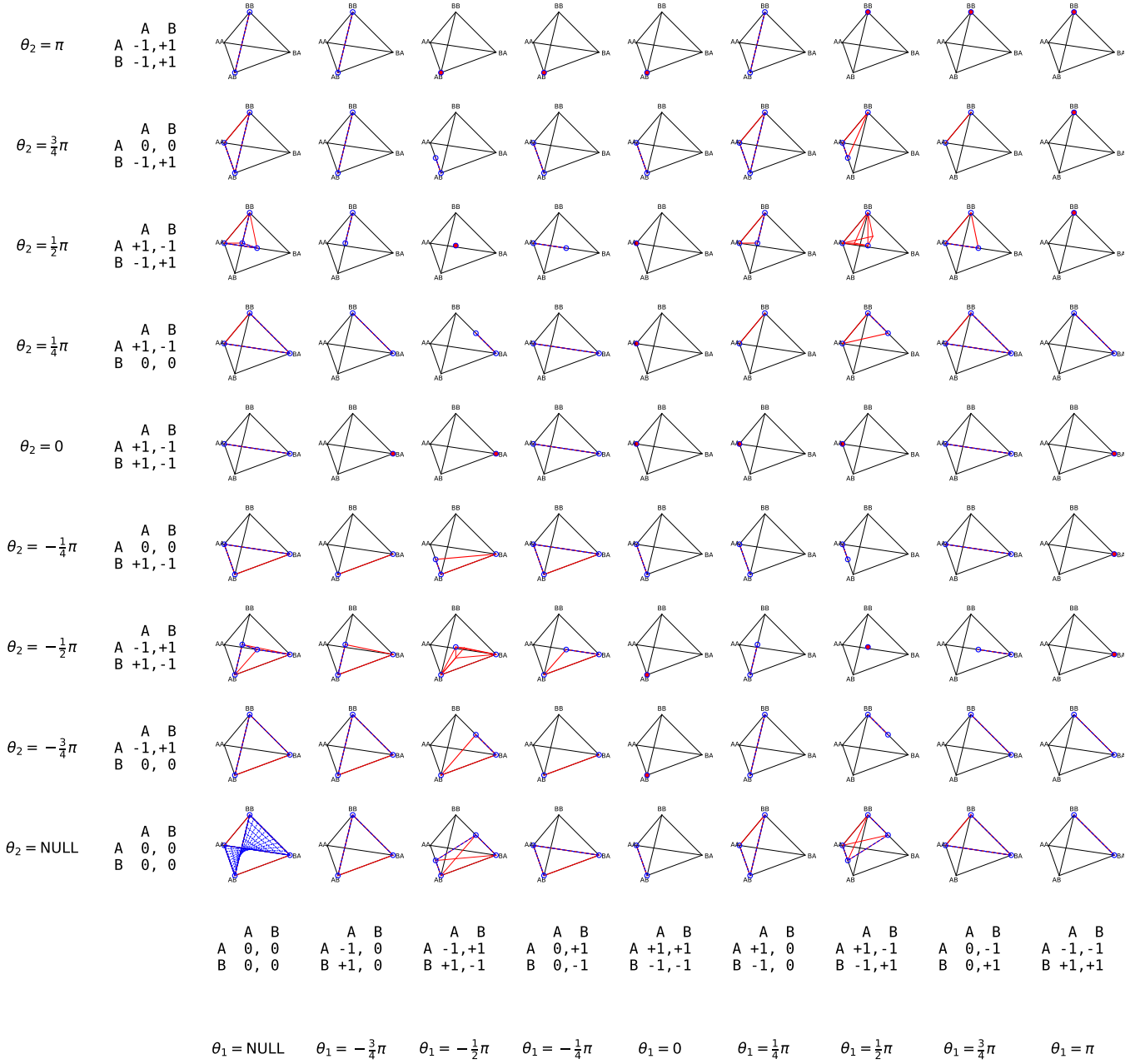


Figure 4.5: The space of NEs (dashed blue lines) and (C)CEs (solid red polytopes) equilibria for 2x2 games. For two-strategy games CE and CCE are identical. The left column contains games where player 1 has trivial payoff. The bottom row contains games where player 2 has trivial payoff. The intersection of the polytopes in these games results in the polytopes of the non-null games. In \emptyset Null, all joints are (C)CEs and all factorizable joints are NEs. The NEs of the diagonal quadrants have convex (interchangeable) solutions because they can be mapped onto zero-sum games. The off-diagonal games are non-interchangeable because they can be mapped onto common-payoff games.

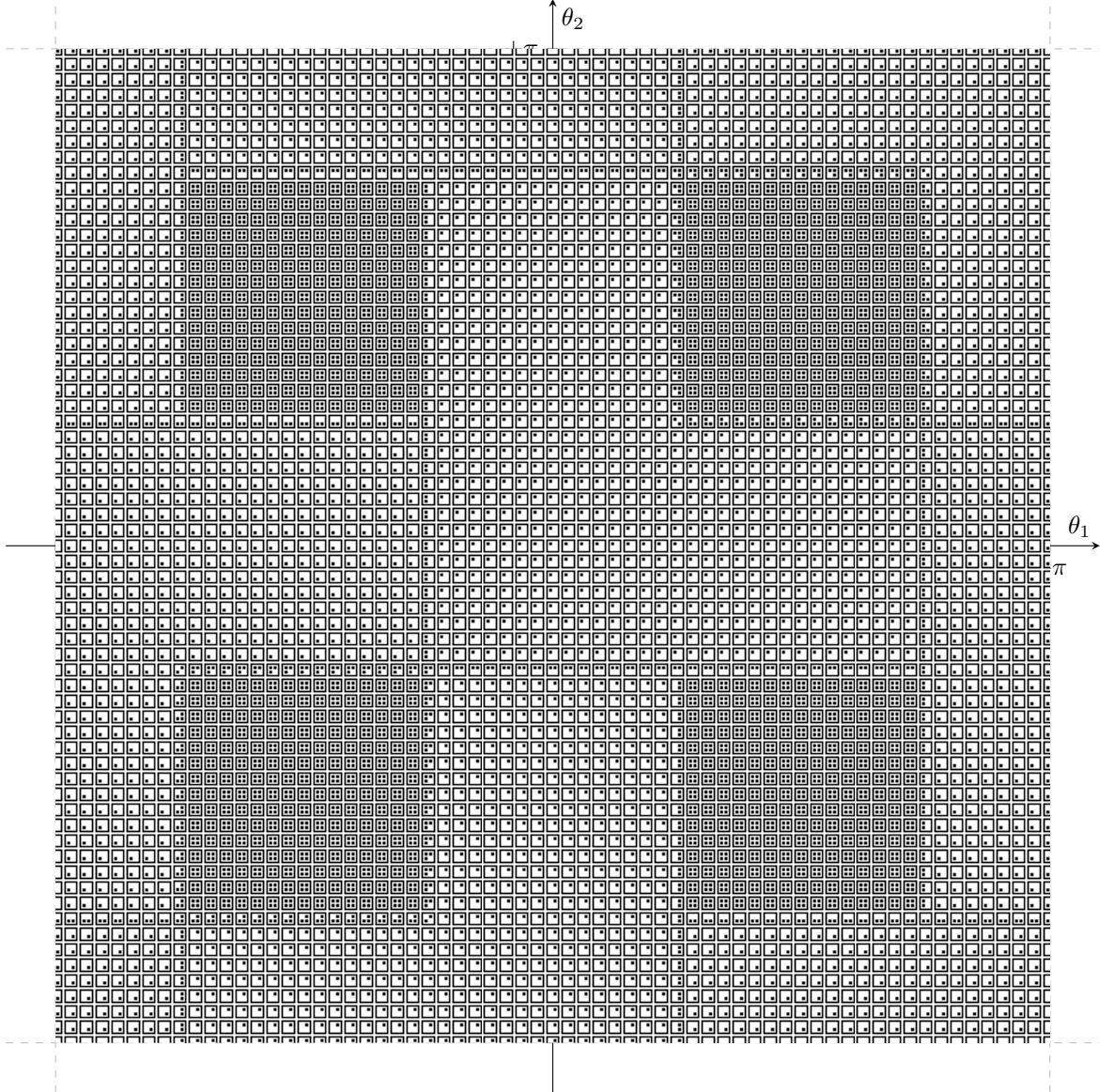


Figure 4.6: Shows the joint strategies that can have support in equilibrium over the space of 2×2 games. The tiling indicates the support of a joint distribution (square) with four optionally shaded quadrants that correspond to the four joint strategies. All nonzero combinations $2^4 - 1 = 15$ of supports are possible. For example, $\begin{smallmatrix} \rightarrow & \rightarrow \\ \rightarrow & \rightarrow \end{smallmatrix}$ Cycle and $\begin{smallmatrix} \rightarrow & \rightarrow \\ \rightarrow & \rightarrow \end{smallmatrix}$ Coordination have full support $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$, $\begin{smallmatrix} \rightarrow & \rightarrow \\ \rightarrow & \rightarrow \end{smallmatrix}$ Dominant has pure support $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$, $\begin{smallmatrix} \rightarrow & \rightarrow \\ \rightarrow & \rightarrow \end{smallmatrix}$ Fossick has diagonal support $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$, and $\begin{smallmatrix} \rightarrow & \rightarrow \\ \rightarrow & \rightarrow \end{smallmatrix}$ Daredevil has support over all but one joint $\begin{smallmatrix} \blacksquare & \blacksquare \\ \blacksquare & \blacksquare \end{smallmatrix}$. The majority of space either has a pure joint strategy or permits full-support mixed equilibrium. Other equilibria are possible but are measure-zero and exist on the boundaries between the pure and full-support games.

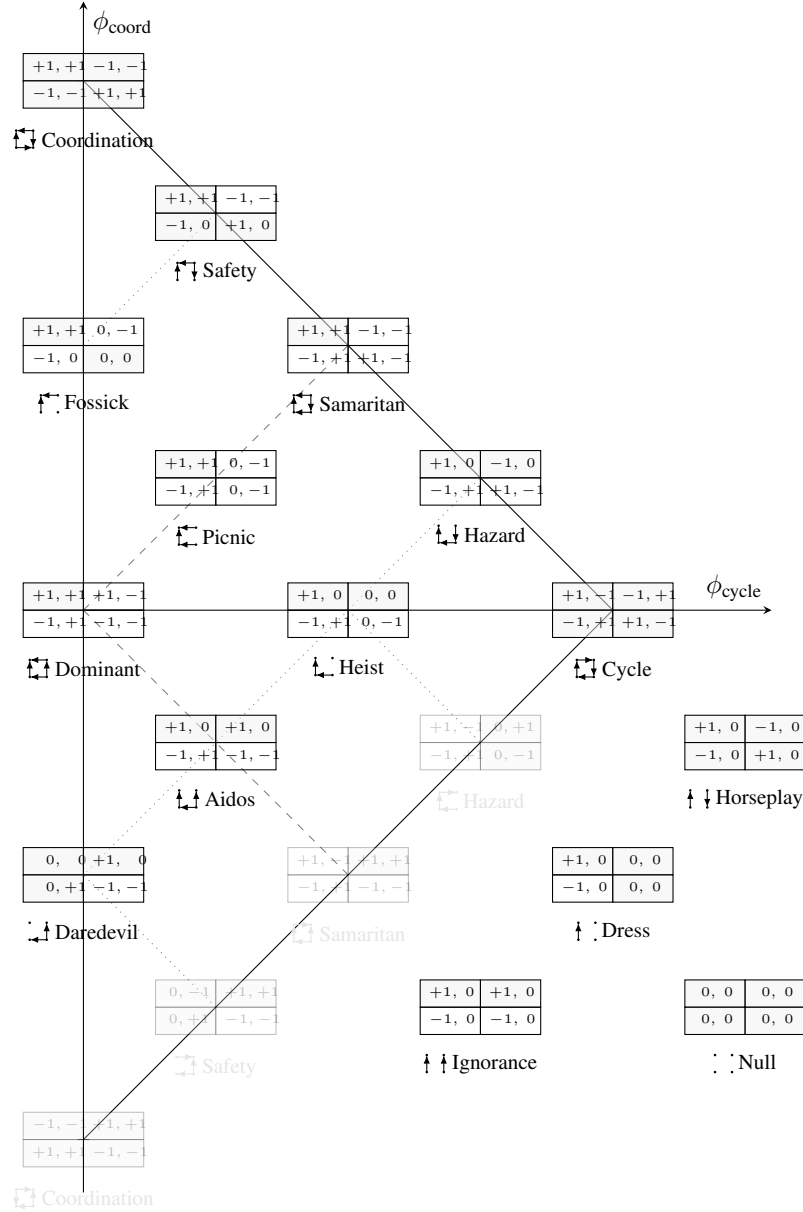


Figure 4.7: Shows 15 fundamental two-player two-strategy best-response-invariant embeddings. Payoffs are shown in the table, with shaded joint strategies appearing in an equilibrium. The faded games are symmetric permutations of other games. All 2x2 games will map onto one of these games. The dashed gray lines show the quadrants: the right quadrant is zero-sum clockwise cyclic and left half-quadrants show coordination games. The dotted lines show the equilibrium solution region which, going clockwise from the top, are diagonal-coordination, top-left dominant, clockwise-coordination, top-right dominant, and off-diagonal-coordination. Games outside the equilibrium-symmetric embedding are trivial.

Chapter 5

Visualizing Large Games

2x2 equilibrium-invariant embeddings can be parameterized using only two variables. This makes them particularly easy to be spatially represented in two dimensions. This property is leveraged to develop game-theoretic visualizations of large many-player many-action normal-form and extensive-form games. These visualisations aim to fingerprint the strategic interactions that occur within these games, which has previously been considered intractably large to represent. The contents of this chapter is part of published work ([Marris et al., 2023](#)).

5.1 Introduction

Tools developed in Chapter 4 for studying 2x2 games are amenable to games with more players and strategies, including large normal-form and extensive-form games. Historically, these games have been considered intractable to visualize or summarize. This work develops game-theoretic visualizations that fingerprint strategic interactions within these games. The field of machine learning has enjoyed such simplified visualizations of high dimensional complex data. Principled techniques like PCA ([Hotelling, 1936](#); [Pearson, 1901](#)) aim to reduce dimensionality, while maintaining the maximum amount of information. Other less principled techniques like t-SNE ([Hinton and Roweis, 2003](#); [van der Maaten and Hinton, 2008](#)) are also very popular. 2x2 equilibrium-invariant embeddings can be utilized to produce visualizations of 2x2, $|\mathcal{A}_1| \times |\mathcal{A}_2|$, two-player extensive-form, n-player polymatrix, and n-player extensive-form games (Figure 5.1). Equilibrium and payoff properties can be directly deduced from these visualizations. It is hoped that these visualizations prove useful for game theory practitioners, where such visualization tools are underdeveloped.

5.2 2x2 Game Visualization

First, consider the space of joint strategies, $\sigma(a)$, which can be denoted with a flat vector $\sigma = [\sigma(a^{AA}), \sigma(a^{AB}), \sigma(a^{BA}), \sigma(a^{BB})]$, where $a^{IJ} = (a_1^I, a_2^J)$. The standard constraints on a probability distribution apply: probabilities are nonnegative, $\sigma(a) \geq 0 \forall a \in \mathcal{A}$, and sum to unity, $\sum_{a \in \mathcal{A}} \sigma(a) = 1$. The unity sum constraint means that a distribution over the four strategies of a 2x2 game, can be expressed with only three variables because one is redundant given the rest (e.g. $\sigma(a^{BB}) = 1 - \sigma(a^{AA}) - \sigma(a^{AB}) - \sigma(a^{BA})$). This means it is possible to visualize a joint distribution with four components in only three dimensions, by ignoring the space of distributions that do not sum to unity. Typically this is accomplished by specifying four vertices of a tetrahedron (a three dimensional object). Points in the simplex are then described in terms of mixtures of these four vertices (known as a barycentric coordinate system ([Möbius, 1827](#))). Barycentric coordinates, σ , can be converted to Cartesian coordinates, x , via a linear transform, $x = T\sigma$. The columns of T are points of a regular tetrahedron. There are many ways to choose points of a

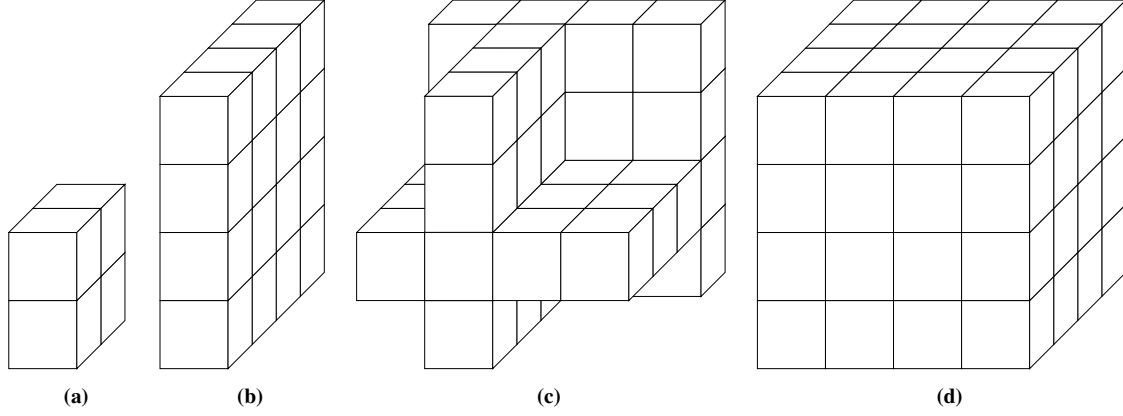


Figure 5.1: Examples of different games that can be visualized using the techniques in this work: (a) 2x2 games, (b) two-player games, and (c) polymatrix games. Each cube represents a joint strategy in the game. Normal-form games with more than two players (d) cannot be directly visualized with the techniques described and have to be approximated around a joint strategy using a local polymatrix approximation.

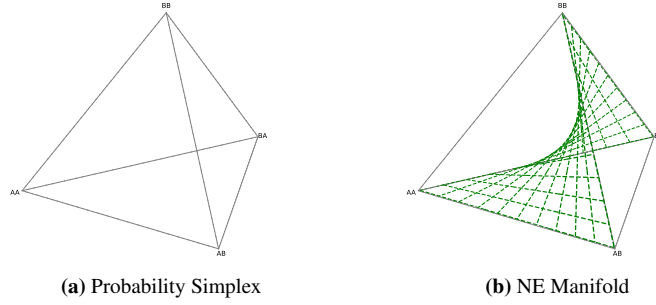


Figure 5.2: Visualization of the space of valid joint distribution, $\sigma(a_1, a_2)$, (tetrahedron) and valid factorizable joint distributions, $\sigma(a_1, a_2) = \sigma(a_1)\sigma(a_2)$, (manifold). The vertices of the tetrahedron correspond to pure joint strategies. The interior of the tetrahedron corresponds to mixed joint strategies.

tetrahedron, one is given in Equation (5.1).

$$\mathbf{x} = T\boldsymbol{\sigma} \quad T = \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & 0 & 0 \\ -\frac{\sqrt{3}}{6} & -\frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{6} & 0 \\ -\frac{\sqrt{6}}{12} & -\frac{\sqrt{6}}{12} & -\frac{\sqrt{6}}{12} & \frac{\sqrt{6}}{4} \end{bmatrix} \quad (5.1)$$

Each nonnegative inequality constraint geometrically corresponds to a normal vector in the equation of a plane (e.g. $[1, 0, 0, 0]\boldsymbol{\sigma}^T \geq 0$). These planes split the space into two halves: those that are feasible distributions and those that are infeasible distributions. Together the four nonnegative probability inequality constraints result in a convex polytope with four faces, specifically a regular tetrahedron, when visualized in three dimensions (Figure 5.2a). $\sigma(a_1, a_2)$ corresponds to a full joint distribution. A subset of joints that factorize into their marginals, $\sigma(a_1, a_2) = \sigma(a_1)\sigma(a_2)$, is worth highlighting because of its relationship to NEs which by definition have to factorize. Factorizable joints result in a manifold within the tetrahedron (Figure 5.2b).

Now consider a player's payoff, $G_p(a_1, a_2)$. It is known that each player's 2x2 equilibrium-invariant embedding is parameterized by an angle, $G_p^{\text{equil}}(\theta_p)$, on a circle. This circle can be meaningfully traced in three-dimensions: $x(\theta_p) = Tg_p^{\text{equil}}(\theta_p)$. A particular payoff for a player, G_p^{equil} , can be represented by drawing an arrow from the origin to a point on this circle. The direction of this arrow conveys meaning: it

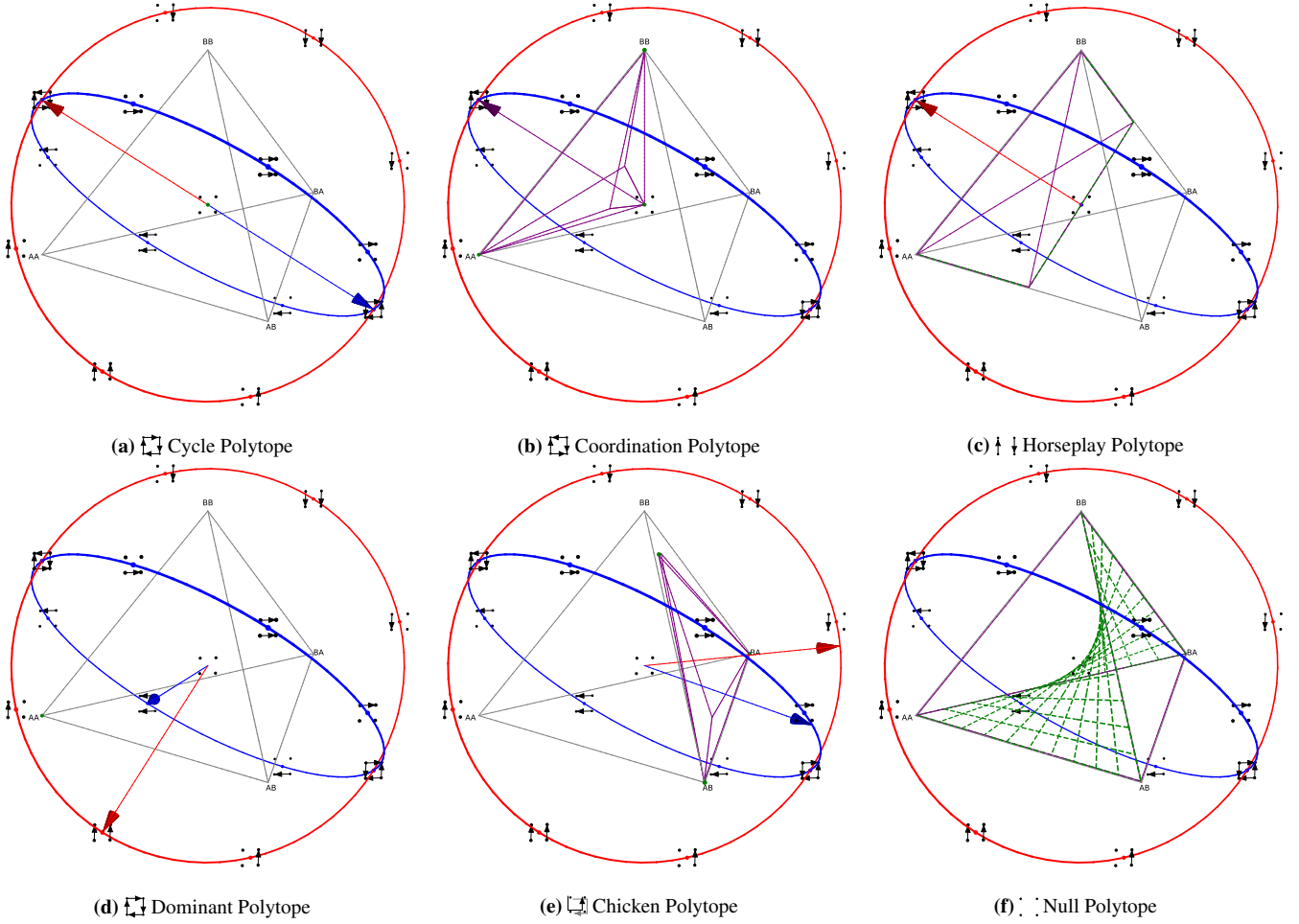


Figure 5.3: Visualization of 2x2 games. The θ_1 (red) and θ_2 (blue) parameters are shown on their unit circles. The arrows are the invariant unit vectors of each player's payoff so the direction indicates joints that will linearly increase payoff. The (C)CE polytope of feasible equilibria is shown in purple. Green shows NEs.

is the direction which linearly increases the equilibrium-invariant embedding payoff received under a joint, and it points to the region where equilibria are likely to reside. The original payoffs, $Tg_p(\theta_p)$, could be any vectors perpendicular to the plane that the circle lies on.

The deviation gains (Equations (2.46), (2.54), and (2.59)) of the various equilibrium concepts can also be visualized. Each row of the deviation gain matrix corresponds to a half plane. The rows in aggregate make a convex polytope, which can also be visualized in three dimensions. NEs are where this polytope intersects with the factorizable joint manifold.

This chapter proposes a visualization of 2x2 games that includes their equilibrium-invariant embedding, best-response-invariant embedding, (C)CE polytope and NE set (Figure 5.3). When the arrows are near opposite, the game is invariant-zero-sum (for example $\begin{smallmatrix} \uparrow & \downarrow \\ \downarrow & \uparrow \end{smallmatrix}$ Cycle, Figure 5.3a). When the arrows are near alignment, the game is invariant-common-payoff (for example $\begin{smallmatrix} \uparrow & \uparrow \\ \downarrow & \downarrow \end{smallmatrix}$ Coordination, Figure 5.3b). Dominant games have near perpendicular vectors. Sometimes a continuum of NEs is feasible, when that is the case it will be shown with a dashed line (for example $\begin{smallmatrix} \uparrow & \downarrow \\ \downarrow & \uparrow \end{smallmatrix}$ Horseplay, Figure 5.3c).

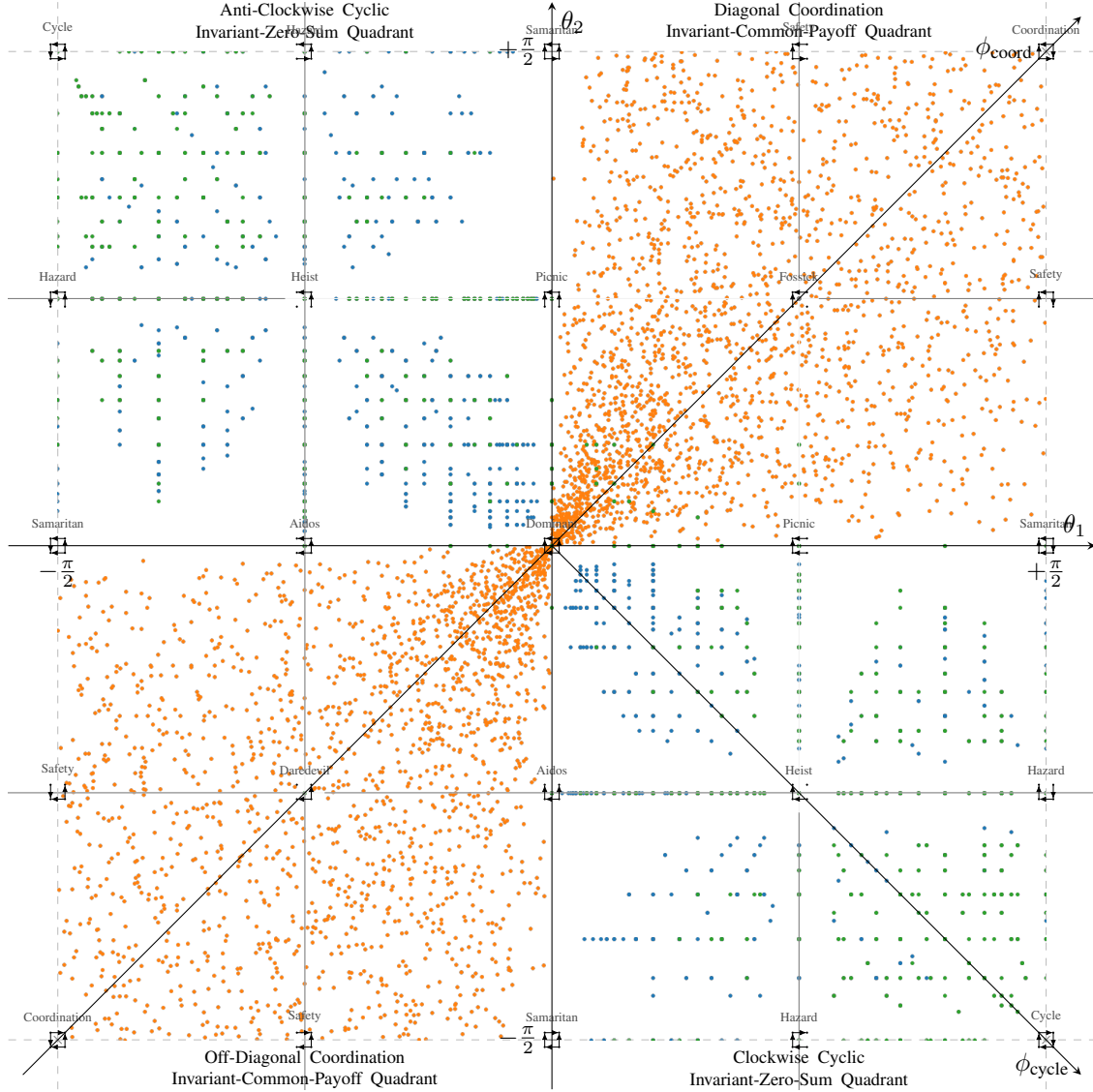


Figure 5.4: Visualization of the strategy space of asymmetric, two-player, extensive-form games: ■ zero-sum Kuhn Poker, ■ common-payoff Tiny Bridge, and ■ mixed-motive Sheriff. Zero-sum games only have interactions in the diagonal quadrants. Common-payoff games only have interactions in the off-diagonal quadrants. Mixed motive games can have interactions in all quadrants. The games are asymmetric so strategy permutation (but not player permutation) can be utilized to plot the interactions on a reduced invariant embedding.

5.3 Two-Player Game Visualization

A two-player game with more than two strategies ($|\mathcal{A}_1| \times |\mathcal{A}_2|$), many strategies (such as an extensive-form¹ game with deterministic policies), or even infinitely many strategies (such as an extensive-form game with stochastic policies), can be summarized by considering either every possible 2×2 subgame or approximated by sampling subgames, \tilde{G}_p . To produce a single point, uniformly sample strategies \tilde{a}_1^A , \tilde{a}_1^B , \tilde{a}_2^A , and \tilde{a}_2^B , to produce a sampled 2×2 payoff. Then find the equilibrium-invariant embedding $(\tilde{\theta}_1, \tilde{\theta}_2)$, using symmetries

¹ All extensive-form games have an equivalent normal-form representation. However, subgame perfection is not representable in normal-form.

Algorithm 5.1 Two-Player Global Visualization.

Require: Two-player payoffs, $G(a_1, a_2)$.

- 1: **function** SAMPLEGLOBALEMBEDDING(G)
- 2: $\tilde{a}_1^A, \tilde{a}_1^B \leftarrow \text{Cat}(|\mathcal{A}_1|)$
- 3: $\tilde{a}_2^A, \tilde{a}_2^B \leftarrow \text{Cat}(|\mathcal{A}_2|)$
- 4: **for** $s_1 \in [A, B]$ **do**
- 5: **for** $s_2 \in [A, B]$ **do**
- 6: $\tilde{G}_1(s_1, s_2) \leftarrow G_1(\tilde{a}_1^{s_1}, \tilde{a}_2^{s_2})$
- 7: $\tilde{G}_2(s_1, s_2) \leftarrow G_2(\tilde{a}_1^{s_1}, \tilde{a}_2^{s_2})$
- 8: $\theta_1 \leftarrow \text{Embed}(\tilde{G}_1)$
- 9: $\theta_2 \leftarrow \text{Embed}(\tilde{G}_2)$
- 10: **return** θ_1, θ_2

Algorithm 5.2 N-Player Local Visualization.

Require: Two-player payoffs, $G(a_1, \dots, a_N)$.

Require: Background strategy, a .

Require: Focus players, r and c .

- 1: **function** SAMPLELOCALEMBEDDING(G)
- 2: $\tilde{a}_r^A \leftarrow a_r$
- 3: $\tilde{a}_c^A \leftarrow a_c$
- 4: $\tilde{a}_r^B \leftarrow \text{Cat}(|\mathcal{A}_r|)$
- 5: $\tilde{a}_c^B \leftarrow \text{Cat}(|\mathcal{A}_c|)$
- 6: **for** $s_r \in [A, B]$ **do**
- 7: **for** $s_c \in [A, B]$ **do**
- 8: $\tilde{G}_1(s_r, s_c) \leftarrow G_r(\tilde{a}_r^{s_r}, \tilde{a}_c^{s_c}, a_{-p})$
- 9: $\tilde{G}_2(s_r, s_c) \leftarrow G_c(\tilde{a}_r^{s_r}, \tilde{a}_c^{s_c}, a_{-p})$
- 10: $\theta_1 \leftarrow \text{Embed}(\tilde{G}_1)$
- 11: $\theta_2 \leftarrow \text{Embed}(\tilde{G}_2)$
- 12: **return** θ_1, θ_2

if appropriate.

$$\tilde{G}_p^{\text{global}}(\tilde{a}_1^A, \tilde{a}_1^B, \tilde{a}_2^A, \tilde{a}_2^B) = \begin{bmatrix} G_p(\tilde{a}_1^A, \tilde{a}_2^A) & G_p(\tilde{a}_1^A, \tilde{a}_2^B) \\ G_p(\tilde{a}_1^B, \tilde{a}_2^A) & G_p(\tilde{a}_1^B, \tilde{a}_2^B) \end{bmatrix} \rightarrow (\tilde{\theta}_1^{\text{global}}, \tilde{\theta}_2^{\text{global}}) \quad (5.2)$$

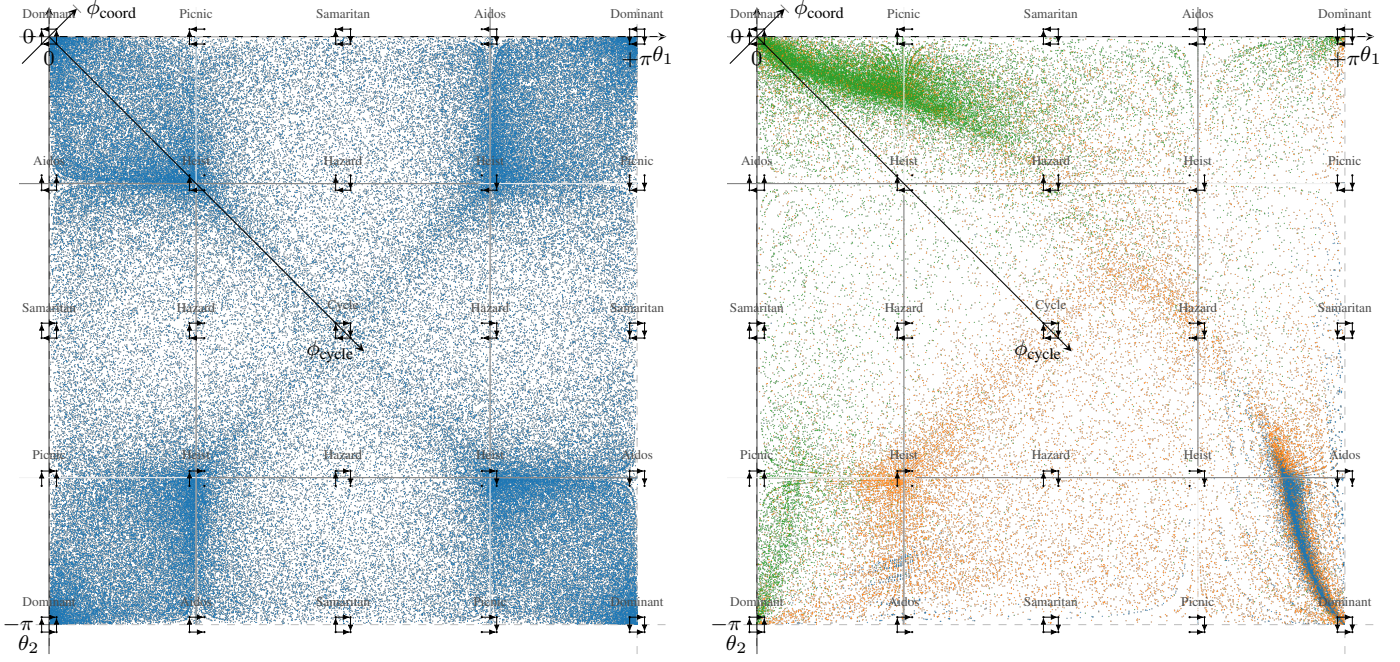
Large two-player games can be represented as a point cloud of sampled equilibrium-invariant embeddings. Properties of the game can be deduced from the positions and density of the the embeddings in this plot. This work names this sampling scheme *global* because all strategies are uniformly sampled over the full normal-form game being analysed.

5.3.1 Extensive-Form Games

Three two-player extensive-form games taken from the OpenSpiel library (Lanctot et al., 2019) are visualized (Figure 5.4). All extensive-form games can be converted to normal-form games by considering the enumeration of pure-strategy policies available to each player. To produce the visualization, it is not required to enumerate all policies; simply sample random pure joint policies to approximate the visualization. This enables scaling the analysis to large extensive-form games. Because all the games considered in this visualization are asymmetric, player symmetry is not utilized. Strategies are sampled arbitrarily from the payoffs, so the order of strategies is also arbitrary, so strategy symmetry is utilized. Therefore, the plots cover the domain: $-\frac{\pi}{2} \leq \theta_p < \frac{\pi}{2}$.

Kuhn Poker (Kuhn, 1950) is an asymmetric, zero-sum simplified poker variant. First, notice that all points lie in invariant-zero-sum quadrants, which is what the theory predicts for a two-player zero-sum extensive-form game. Interestingly, many boundary classes ($\uparrow\downarrow$ Aidos, $\uparrow\leftarrow$ Picnic, and $\uparrow\downarrow$ Hazard) appear. This shows that this game has situations where players are indifferent between strategies. Kuhn poker does have a high concentration of points in the $\leftarrow\rightarrow$ Dominant and $\leftarrow\rightarrow$ Samaritan classes indicating that in most situations there are obvious strategies that either both or at least one of the players should choose. There are also a significant number of $\leftarrow\rightarrow$ Cycle interactions which indicates that this game has some interesting strategic depth.

Two-player Cooperative Tiny Bridge (Lockhart et al., 2020) is a common-payoff game. As expected, all points lie in the invariant-common-payoff quadrants of the visualization. The game also has a high concentration of points near the origin indicating that most strategic interaction is $\leftarrow\rightarrow$ Dominant: both



(a) Approximate summary of the full 888×888 AlphaStar league (Vinyals et al., 2019). Each strategy corresponds to a learned policy in the extensive-form game and the payoff was calculated empirically. The plot is redundant outside of the domain $0 \leq \theta_1 \leq \frac{\pi}{2}$ and $-\frac{\pi}{2} \leq \theta_2 \leq 0$.

(b) Local approximation of the AlphaStar league. The first strategy is fixed for both players, and only the second sampled. Key: ■ early training, ■ middle training, and ■ late training.

Figure 5.5: Analysis of a large empirical normal-form game. In early and late training, the local strategic space is mostly transitive (Dominant), and in mid training the space is cyclic (Cycle). This matches Czarniecki et al. (2020)’s “spinning top” hypothesis in games of skill.

players have a strict preference between strategies. Coordination also features prominently, indicating there are situations that require players to coordinate to maximize payoffs. It seems no boundary classes appear in Tiny Bridge meaning that it is rare for players to be indifferent over their strategies.

Sheriff (Farina et al., 2019c) is an asymmetric, mixed-motive game. The point cloud of this game covers both invariant-common-payoff and invariant-zero-sum quadrants, however it predominately occupies the invariant-zero-sum quadrant indicating that the game is more competitive than cooperative. Again, boundary classes (Aidos, Picnic, and Hazard) appear, indicating times when players are indifferent.

5.3.2 AlphaStar League

The AlphaStar league is a symmetric zero-sum, 888×888 , empirical normal-form game. It consists of payoffs between policies learned by AlphaStar (Vinyals et al., 2019). The visualization of this game is shown in Figure 5.5a. The game is symmetric, so player symmetry is utilized. The game is zero-sum, so only zero-sum quadrants need be plotted. This plot is also strategy invariant, so strategy symmetry could have been utilized. However, to have a simpler comparison to Figure 5.5b, it is not used. Therefore the plot is over the domain: $0 \leq \theta_1 \leq \pi$ and $-\pi \leq \theta_2 \leq 0$, although outside of $0 \leq \theta_1 \leq \frac{\pi}{2}$ and $-\frac{\pi}{2} \leq \theta_2 \leq 0$ the plot is redundant. The empirical game mainly has Dominant strategic interaction, although there are significant Samaritan and Cycle components.

Czarniecki et al. (2020) hypothesised that games of skill have few cyclic interactions early in training as policies can easily transitively improve. In mid training, there are numerous reasonably competent and

diverse policies, which results in interesting cyclic interactions. However, in late training, the interactions between policies become less cyclic, as the skills needed to play the game are perfected. [Czarnecki et al. \(2020\)](#) describes this phenomenon as a “spinning top” shaped distribution of cycles. This hypothesis is tested in the visualization by using an alternative game sampling scheme: fixing the first strategy, a^A , for both players, and sampling the second strategy for each player randomly, \tilde{a}_p^B .




$$\tilde{G}_p^{\text{local}}(\tilde{a}_1^B, \tilde{a}_2^B) = \begin{bmatrix} G_p(a^A, a^A) & G_p(a^A, \tilde{a}_2^B) \\ G_p(\tilde{a}_1^B, a^A) & G_p(\tilde{a}_1^B, \tilde{a}_2^B) \end{bmatrix} \rightarrow (\tilde{\theta}_1^{\text{local}}, \tilde{\theta}_2^{\text{local}}) \quad (5.3)$$

The AlphaStar league game is symmetric, so it is more natural to use the same fixed strategy for both players. Note that in general, the analysis could have different fixed strategies for each player.

$$\tilde{G}_p^{\text{local}}(\tilde{a}_1^B, \tilde{a}_2^B) = \begin{bmatrix} G_p(a_1^A, a_2^A) & G_p(a_1^A, \tilde{a}_2^B) \\ G_p(\tilde{a}_1^B, a_2^A) & G_p(\tilde{a}_1^B, \tilde{a}_2^B) \end{bmatrix} \rightarrow (\tilde{\theta}_1^{\text{local}}, \tilde{\theta}_2^{\text{local}}) \quad (5.4)$$

This sampling scheme is called *local* because it is sampling over deviation strategies around a fixed background strategy $a^{AA} = (a_1^A, a_2^A)$. Such a sampling strategy will provide an indication of the strategic dynamics locally around these background strategies.

The strategies in the AlphaStar league game correspond to policies roughly ordered according to their training time. Later strategies are policies trained against distributions over previous policies. Therefore, later policies have more training time and have trained against more diverse opponents, and will more likely be stronger. But what does the strategic landscape of the game look like at each stage of training? By analysing the game around background strategies which correspond to early, middle and late training, [Czarnecki et al. \(2020\)](#)’s hypothesis (Figure 5.5b) can be tested.

Unsurprisingly, ■ early training primarily occupies a  Dominant region, where players strictly wish to deviate from their weakly trained fixed background policy. In contrast, ■ late training primarily occupies the opposite  Dominant region, where players strictly prefer to stick with their fixed background strategies. ■ Mid training has a varied strategic space, but has more mass in the  Cycle region than the other background policies. Broadly, this analysis supports the “spinning top” hypothesis. A deeper analysis of the training timeline could uncover more structure.

5.4 N-Player Polymatrix Game Visualization

It is possible to extend the two-player local subgame sampling technique to n-player games by defining a fixed background strategy for all players, $a^{A\dots A} = (a_1^A, \dots, a_N^A)$, and then visualizing all pairwise player interactions.

$$\tilde{G}_{pq}^{\text{local}}(\tilde{a}_p^B, \tilde{a}_q^B) = \begin{bmatrix} G_p(a_p^A, a_q^A, a_{\dots p-q}^A) & G_p(a_p^A, \tilde{a}_q^B, a_{\dots p-q}^A) \\ G_p(\tilde{a}_p^B, a_q^A, a_{\dots p-q}^A) & G_p(\tilde{a}_p^B, \tilde{a}_q^B, a_{\dots p-q}^A) \end{bmatrix} \rightarrow (\tilde{\theta}_p^{\text{local}}, \tilde{\theta}_q^{\text{local}}) \quad \forall p \neq q \in [1, N] \quad (5.5)$$


















To avoid ambiguity with the player slots of the original n-player game the representation parameters are renamed from θ_1 and θ_2 to θ_r and θ_c to indicate the row and column players. Because the order of the players and strategies matter in this pairwise approximation of an n-player game, it is necessary to use the full invariant space: $-\pi \leq \theta_r \leq +\pi$ and $-\pi \leq \theta_c \leq +\pi$. It is unnecessary to plot both permutations of the player pairs: the point cloud will be the same but mirrored over $\theta_r = \theta_c$.

Considering only pairwise player interaction is an established succinct game representation called the *polymatrix approximation* ([Janovskaja, 1968](#)). This representation is described by the tuple

$(G_{1,2}(a_1, a_2), G_{1,3}(a_1, a_3), \dots, G_{N-1,N}(a_N, a_{N-1}))$. By fixing the background strategies in an n-player normal-form game a *local polymatrix approximation* is created around these background strategies. One need not restrict themselves to strategies in a normal-form game. Entire policies, $\tilde{\pi}_p^B$, can be sampled in extensive-form games around fixed background policies, $\pi^{A \dots A} = (\pi_1^A, \dots, \pi_N^A)$. Either stochastic or deterministic policies can be sampled (this work considers deterministic policies in its experiments). Therefore the visualization can be used to approximate the strategic dynamics of a game around salient policies, such as the policies players are currently executing in a game, or perhaps ones that have been found using a learning method. It is expected that the local polymatrix approximation landscape to be different around different background policies, as policies influence transition dynamics in the game.

$$\tilde{G}_{pq}^{\text{local}}(\tilde{\pi}_p^B, \tilde{\pi}_q^B) = \begin{bmatrix} G_p(\pi_p^A, \pi_q^A, \pi_{-p-q}^{\dots}) & G_p(\pi_p^A, \tilde{\pi}_q^B, \pi_{-p-q}^{\dots}) \\ G_p(\tilde{\pi}_p^B, \pi_q^A, \pi_{-p-q}^{\dots}) & G_p(\tilde{\pi}_p^B, \tilde{\pi}_q^B, \pi_{-p-q}^{\dots}) \end{bmatrix} \rightarrow (\tilde{\theta}_p^{\text{local}}, \tilde{\theta}_q^{\text{local}}) \quad \forall p \neq q \in [1, N] \quad (5.6)$$

5.4.1 Three-Player Leduc Poker

Leduc poker (Southey et al., 2005), implemented in OpenSpiel (Lanctot et al., 2019), is a simplified Texas Hold'em implementation with N players, two suits and $2(N + 1)$ cards. This experiment studies the three-player game (Figure 5.6), with each player respectively having the background policies: always raise, always call, and always fold. If any of these actions are infeasible at an information state, the policy will fall back to the call action. Unsurprisingly, given players must pay a blind, always fold is a losing strategy. This is most apparent when analysing  call vs fold where call is almost always the best strategy to play. The majority of the points are in the basins of games along the θ_c axis:   Dominant and   Samaritan. This indicates two properties. Firstly, that call is almost always a dominant strategy for the row player (in the context of the other background policies), regardless of the column player's strategy. Secondly, the strength of the fold strategy depends on what the other players play. It is sometimes good ( Dominant and  Samaritan), and sometimes bad ( Dominant and  Samaritan). Fold is also poor in the context of  fold vs raise. Deviating from fold is sometimes a better strategy ( Dominant and  Samaritan) but there are situations where it is not ( Dominant and  Samaritan). It is very rare for fold to be the worse strategy when playing against the other background policies, but be the preferred strategy against the deviation strategy ( Samaritan). Generally, clustering around the θ_c axis indicates a strong row player and clustering around the θ_r axis indicates a strong column player. The  raise vs call dynamics are interesting: the majority of the points lie in the anti-clockwise invariant-zero-sum region. Player 3, who always folds, plays such a weak strategy that they are an irrelevant player and the local raise vs call two-player game is almost perfectly zero-sum. Call is usually a good strategy against raise and a poor strategy against deviations from raise, and while raise is a poor strategy against call it is usually a good strategy against deviations, resulting in a  Cycle. The majority of the points for all the local games lie in the invariant-zero-sum quadrants, which is unsurprising because this is a zero-sum game. Only two-player invariant-zero-sum games lie completely in the invariant-zero-sum quadrants.

5.4.2 Tiny Bridge 2vs2

Tiny Bridge is an extensive-form, two versus two, team game (Lockhart et al., 2020) implemented in OpenSpiel (Lanctot et al., 2019) with zero-sum dynamics between teams and common-payoff dynamics within the team. The game is a simplified version of the popular card game Contract Bridge, but consists of only two suits each with four cards. Using the polymatrix visualization tools described above this experiment studies the dynamics of Tiny Bridge. The study verifies that the visualization a) captures interesting dynamics between teammates and opponents, and b) is able to differentiate between dynamics under different

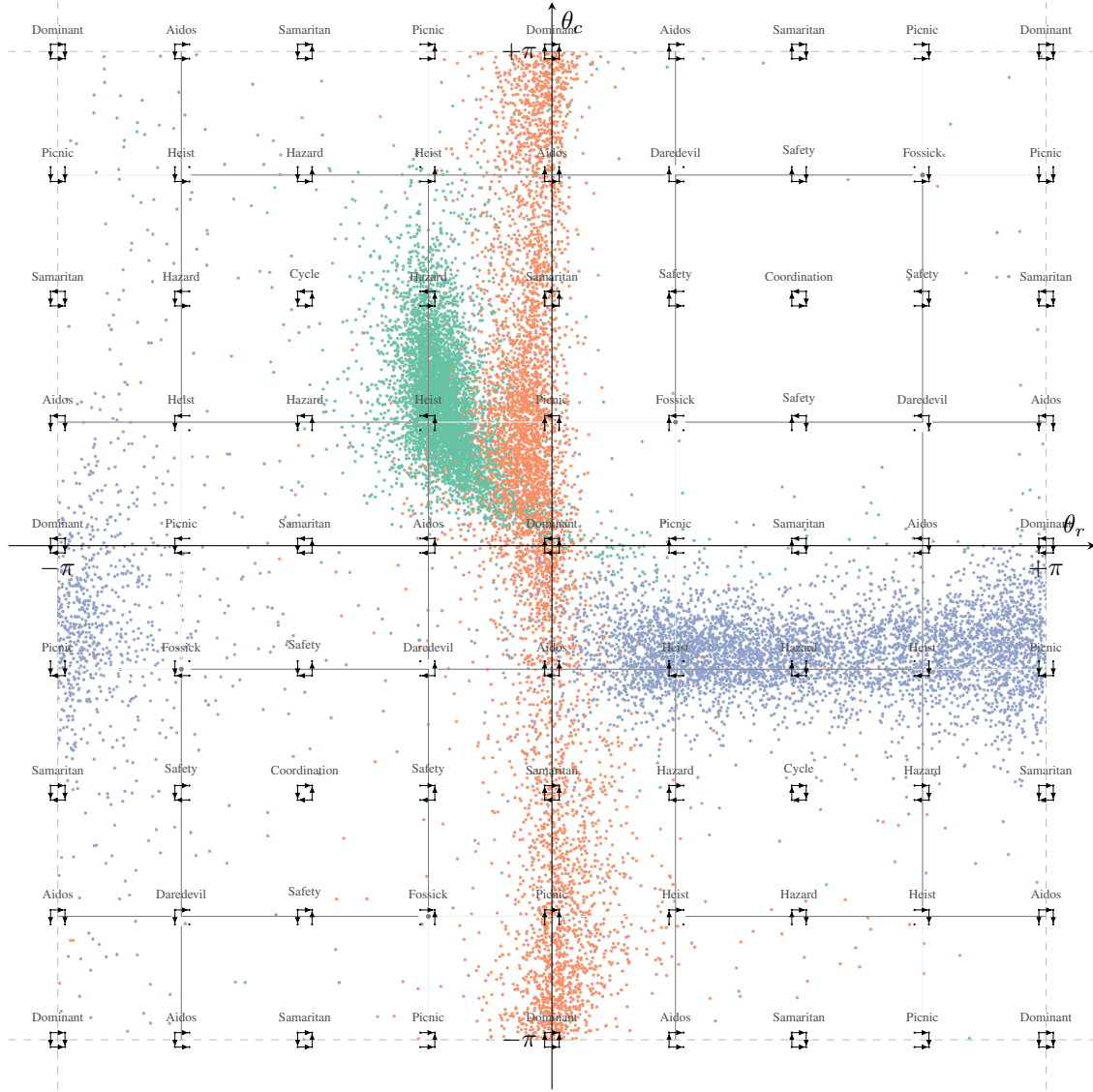


Figure 5.6: Local polymatrix approximation visualization of three-player Leduc poker around the background strategies: always raise, always call, and always fold. Key: ■ raise vs call, ■ call vs fold, ■ fold vs raise.

background policies. To produce quantifiably different policies, twelve random deterministic policies are generated for each player, compute an empirical normal-form game from their expected returns (Wellman, 2006) (Figure 5.7b), and rate them using a game theoretic rating scheme (Marris et al., 2022b) (Figure 5.7a). The ratings scheme is calculated on expected payoff under a joint equilibrium distribution (Figure 5.7c). The maximum entropy criterion is used to choose a CCE. The strongest policies are selected as the “best vs best” background set (Figure 5.8a) and the strongest of team 1 with the weakest of team 2 as the “best vs worst” background set (Figure 5.8b). The local polymatrix approximation of each set is visualized using a different colour for each of the pairwise interactions. The mixed set contains (relatively) strong policies for team 1: they are good at countering their opponents and good at coordinating with each other. The strong set contains competent policies for all players.

The two teams, labeled 1 and 2, each have two players, labeled A and B. For both sets, as expected, the common-payoff intra-team dynamics (■ 1A vs 1B and ■ 2A vs 2B) only lie in the off-diagonal quadrants and the zero-sum inter-team dynamics (■ 1A vs 2A, ■ 1A vs 2B, ■ 1B vs 2A, and ■ 1B vs 2B) lie in the



Figure 5.7: Twelve randomly generated policies for each of the four players in Tiny Bridge were rated using a game theoretic rating technique (Marris et al., 2022b) based on a CCE. The ratings, payoffs, and CCE have been reordered from lowest to highest rating. The CCE and payoffs are visualized in two dimensions as team vs team for convenience; they remain four player games. Green, yellow and red indicate high, zero, and low team 1 payoff.

diagonal quadrants. Focusing on intra-team dynamics (■ 1A vs 1B and ■ 2A vs 2B), for the best vs best background set the point cloud is concentrated around ↻↻ Dominant and ↻↻ Samaritan. This indicates that both teams have good intra-team cooperation. However for the best vs worst background set, the worse team’s policies (■ 2A vs 2B) have poor intra-team cooperation. Points are clustered around ↻↻ Dominant and ↻↻ Samaritan, indicating that deviating away from the background policy is advantageous. Focusing on inter-team dynamics, the best vs worst background set shows that team 1 out-competes team 2 (■ 1A vs 2A, ■ 1A vs 2B, ■ 1B vs 2A, and ■ 1B vs 2B). The points are clustered along the θ_c axis which indicates a stronger row player. Furthermore, the points are primarily in the basins of ↻↻ Samaritan and ↻↻ Dominant: showing that the row player prefers its background policy over deviations and the column player prefers deviations over background policy. Player 1A’s dynamics (■ 1A vs 2A and ■ 1A vs 2B) are closer to the θ_c axis than player 1B’s dynamics (■ 1B vs 2A and ■ 1B vs 2B) indicating that out of the two players on team 1, A is more competitive. This is not surprising as 1A plays first which has a natural advantage in Bridge as they have first opportunity to bid and convey information. The inter-team dynamics in the best vs best background policies are more balanced.

5.5 Discussion

Summarizing and visualizing large datasets with high dimensionality is an important area of research in machine learning. Principled techniques like PCA (Hotelling, 1936; Pearson, 1901) that reduce dimensionality, while maintaining the maximum amount of information, are ubiquitous. PCA allows data to be visualized in fewer dimensions. Other less principled techniques like t-SNE (Hinton and Roweis, 2003; van der Maaten and Hinton, 2008) are also a very popular tool for inspecting datasets. While the fields of statistics and machine learning have benefited from such tools for decades, game theory has lacked such analysis tools, although some attempts have been made to visualize games (Czarnecki et al., 2020; Omidshafiei et al., 2020, 2022). Large games are extremely complex: they have many possible equilibria and complicated better-response dynamics. The visualization tools in this work build on fundamental principles

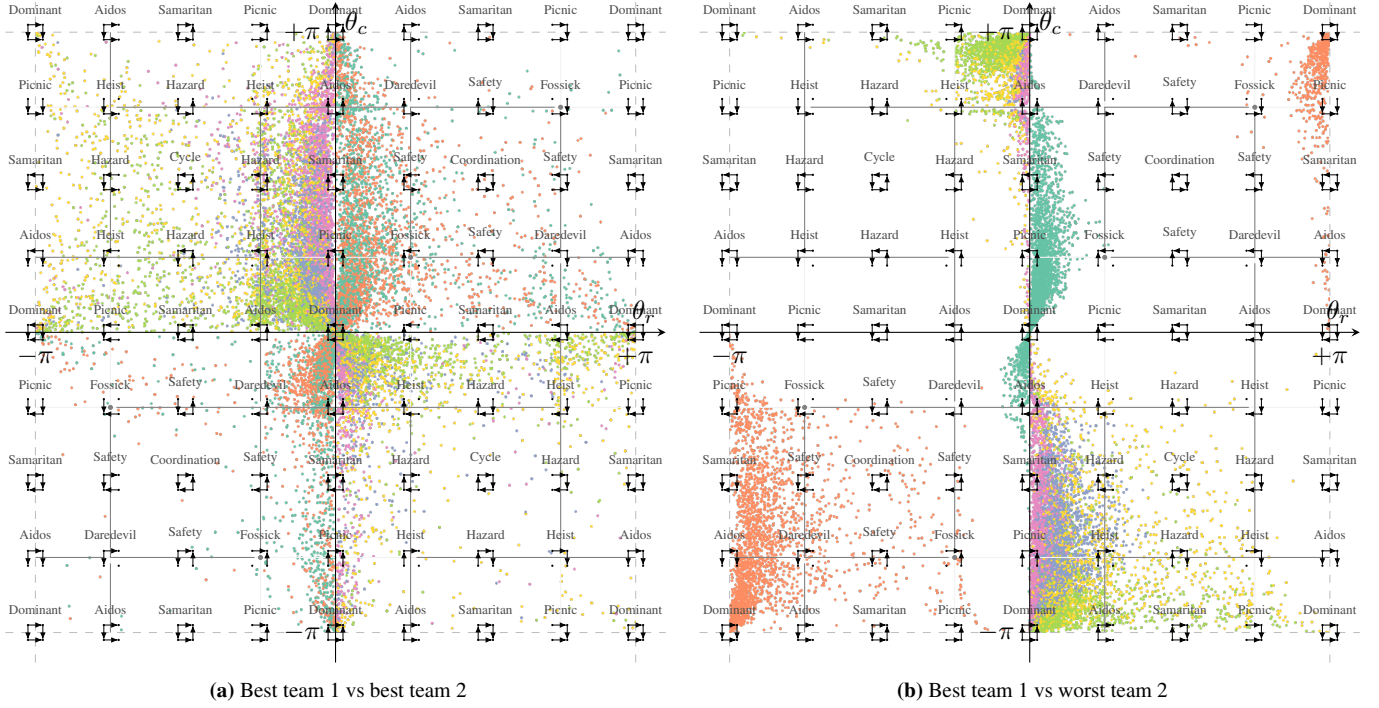


Figure 5.8: Visualization of local polymatrix approximation of the 2vs2 team extensive-form Tiny Bridge with two different background policies sets. There are two teams, 1 and 2, each with two players, A and B. Key: ■ 1A vs 1B, ■ 2A vs 2B, ■ 1A vs 2A, ■ 1A vs 2B, ■ 1B vs 2A, and ■ 1B vs 2B.

and could be an important first step to developing such analysis techniques for large games.

The 2×2 equilibrium-invariant embedding can be leveraged to produce visualizations of larger games. Payoff structure, equilibrium properties, and other details can be read from the visualizations by observing where the point cloud spreads and how the density of points land in best-response-invariant equivalence classes. Although evidence is provided that visualizations can give an insight into how cyclic a game is, it is only capable of showing cycles of length two, longer cycles may not be captured. Fortunately, longer cycles still result in points that appear in the cyclic region. For example, consider Rock-Paper-Scissors, a 2×2 cyclic game, visualized using a 2×2 point cloud (Figure 5.9).

5.6 Conclusion

This chapter develops visualization tools for 2×2 and $|A_1| \times |A_2|$ normal-form games and arbitrary dimensional polymatrix games. Since all extensive-form games have a normal-form representation and a local polymatrix approximation representation, this chapter explored visualizing the strategic space of large games that have, until now, been considered intractable to visualize. It is hoped that this work provides useful visualization tools for game theorists.

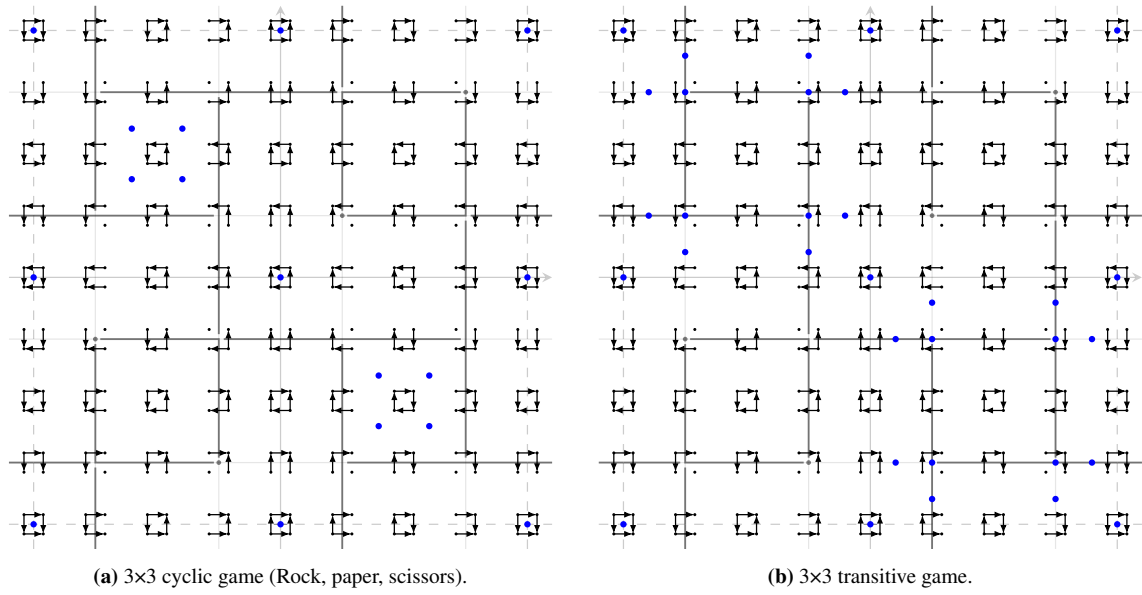


Figure 5.9: Point cloud for two different 3x3 normal-form games.

Chapter 6

Equilibrium Selection of Correlated Equilibria and Coarse Correlated Equilibria

Equilibrium selection is the problem of selecting a single equilibrium from a set of possible equilibria. Solving selection in a principled way allows broad application of game-theoretic solution concepts. This includes situations where a unique target is necessary. This chapter focuses on exploring unique and computationally tractable equilibrium selection criteria. The work in this chapter is combined from parts of two publications (Marris et al., 2021b, 2022a).

6.1 Introduction

Recent success in tackling two-player, zero-sum games (Silver et al., 2016; Vinyals et al., 2019) has out-paced progress in n-player, general-sum games despite a lot of interest (Anthony et al., 2020b; Berner et al., 2019; Brown and Sandholm, 2019; Gray et al., 2020; Jaderberg et al., 2019; Lockhart et al., 2020). One reason is because Nash equilibrium (NE) (Nash, 1951) is tractable and interchangeable in the two-player, zero-sum setting but becomes intractable (Daskalakis et al., 2009) and potentially non-interchangeable¹ in n-player and general-sum settings. The problem of selecting from multiple solutions is known as the equilibrium selection problem (Avis et al., 2010; Goldberg et al., 2013; Harsanyi and Selten, 1988).

Outside of normal form (NF) games, this problem setting arises in multiagent training when dealing with empirical games (also called meta-games (Lanctot et al., 2017; Walsh et al., 2002; Wellman, 2006)), where a game payoff tensor is populated with expected outcomes between agents playing an extensive form (EF) game, for example the StarCraft League (Vinyals et al., 2019) and Policy-Space Response Oracles (PSRO) (Lanctot et al., 2017), a recent variant of which reached state-of-the-art results in Stratego Barrage (McAleer et al., 2020).

This work proposes using correlated equilibrium (CE) (Aumann, 1974) and coarse correlated equilibrium (CCE) as a suitable target equilibrium space for n-player, general-sum games. The (C)CE solution concept has three main benefits over NE. Firstly, it provides a mechanism for players to correlate their actions to arrive at mutually higher payoffs. Secondly, it is computationally tractable to compute solutions for n-player, general-sum games (Daskalakis et al., 2009). And finally, the set of (C)CEs in a game is a convex polytope (defined using linear inequality constraints, for example see Equation (2.2.9)), and therefore is amenable to equilibrium selection.

Maximum entropy correlated equilibrium (MECE) (Ortiz et al., 2007) is the canonical example of such an equilibrium selection procedure. It has an efficient dual parameterization and, because it selects based on the principle of maximum entropy (Jaynes, 1957), it is somewhat justified. An important area of related

¹That is, there are no longer any guarantees on the expected utility when each player plays their part of some equilibrium; guarantees only hold when all players play *the same* equilibrium. Since players cannot guarantee what others choose, they cannot optimize independently, so the Nash equilibrium loses its appeal as a prescriptive solution concept.

work is α -Rank (Omidshafiei et al., 2019) which also aims to provide a tractable alternative solution in normal form games. It gives similar solutions to NE in the two-player, zero-sum setting, however it is not directly related to NE or (C)CE. α -Rank has also been applied to ranking agents and as a meta-solver for PSRO (Muller et al., 2020).

This chapter proposes a novel equilibrium selection concept called Maximum Gini (Coarse) Correlated Equilibrium (MG(C)CE). This concept is inspired by MECE (Ortiz et al., 2007)²: it has a similarly efficient dual parameterization, and selects an equilibrium that maximizes the Gini impurity (Equation (6.1a)). It can be formulated as a quadratic program (QP), which is an optimization class that has known efficient convex solvers. This work thoroughly explores MG(C)CE's properties including tractability, scalability, invariance, and a parameterized family of solutions.

The Gini impurity is a form of entropy and therefore an approximation to the Shannon entropy. Consider the definitions of the Gini impurity and Shannon entropy.

$$\text{Gini Impurity:} \quad G(p) = 1 - \sum_i^k p_i^2 \quad (6.1a)$$

$$\text{Shannon Entropy:} \quad S(p) = - \sum_i^k p_i \log(p_i) \quad (6.1b)$$

Furthermore, consider families of parameterized entropies, Tsallis and Renyi, which are parameterized by q and α respectively.

$$\text{Tsallis Entropy:} \quad T(p, q) = \frac{1}{1-q} \left(1 - \sum_i^k p_i^q \right) \quad (6.2a)$$

$$\text{Renyi Entropy:} \quad R(p, \alpha) = \frac{1}{1-\alpha} \log \left(\sum_i^k p_i^\alpha \right) \quad (6.2b)$$

Both the Gini impurity and the Shannon entropy are instantiations of these families of entropies.

$$G(p) = T(p, q = 2) = 1 - \exp(-R(p, \alpha = 2)) \quad (6.3a)$$

$$S(p) = T(p, q \rightarrow 1) = R(p, \alpha \rightarrow 1) \quad (6.3b)$$

6.1.1 Equilibrium Selection and Correlation Devices

There are two levels of coordination; first is selecting an equilibrium before play commences, and second is implementing this equilibrium during play. Both NEs and (C)CEs require agreement on what equilibrium is being played (Avis et al., 2010; Goldberg et al., 2013; Harsanyi and Selten, 1988): for (C)CEs this is a joint action probability distribution, and for NEs this is also a joint action probability distribution that is a product of marginal distributions for each player. Therefore, at this level of coordination, both NEs and (C)CEs are similar. This coordination problem is the *equilibrium selection problem* (Harsanyi and Selten, 1988). When it comes to implementing an equilibrium, NEs and (C)CEs differ. Only (C)CEs require further coordination. NEs are factorizable and therefore each player can sample independently without further coordination. (C)CEs rely on a central correlation device that will recommend actions from the equilibrium that was previously agreed upon.

This means that neither NEs nor (C)CEs can be directly used prescriptively in n-player, general-sum games. These solution concepts specify what subsets of joint strategies are in equilibrium, but do not specify

²A selection procedure that used Shannon entropy.

how decentralized agents should select amongst these. Furthermore, the presence of a correlation device does not make (C)CEs prescriptive because the agents still need a mechanism to agree on the distribution the correlation device samples from³. This coordination problem can be cast as one that is more computational in nature: what rules allow an equilibrium to be uniquely (and perhaps de-centrally) selected?

This highlights the main drawback of MW(C)CE, which does not select for unique solutions (for example, in zero-sum games all solutions have maximum welfare (MW)). One selection criterion for NEs is maximum entropy Nash equilibrium (MENE) (Balduzzi et al., 2018), however outside of the two-player zero-sum setting, these are generally not easy to compute (Daskalakis et al., 2009). CE exist in a convex polytope, so any convex function can select among them. Maximum entropy correlated equilibrium (MECE) (Ortiz et al., 2007) is limited to full-support solutions, which may not exist when $\epsilon = 0$, and can be hard to solve in practice. Therefore, there is a gap in the literature for a computationally tractable, unique, solution concept.

6.1.2 Desired Properties

Prior research emphasizes the significance of a unique objective for equilibrium selection. However, justifying the superiority of one unique solution over others remains challenging. Occam's razor and the principle of maximum entropy (Jaynes, 1957) have been invoked to support specific selection functions. While maximum entropy solutions offer uniqueness, they often lead to equilibria with low payoffs. In contrast, maximum welfare is generally favored as a selection criterion due to its tendency to yield high-payoff equilibria and its linear nature. However, it is a not unique selection criterion. Equilibrium selection can be defined as a function that maps the set of games and their equilibria to a single equilibrium: $(\mathcal{G}, \Sigma^*) \rightarrow \sigma^*$. There are a number of desired properties an equilibrium selection problem may have.

Unique: Uniqueness is useful for consistency and stability. Fortunately, any strictly convex function can select from a convex set. Linear functions are not strictly convex, hence MW is not in general a unique selection.

Invariant to Equilibrium-Invariant Transforms: As discussed in Chapter 3, some payoff transforms are equilibrium-invariant. An equilibrium selection could select for the same equilibria for all games in an embedding. MW does not necessarily preserve this property as payoffs are shifted and scaled, however modified versions of MW could be made to be equilibrium-invariant. ME is invariant to payoff transformations because it is only a function of the joint.

Value Maximizing: The selection criteria should prefer solutions that give high value to players. MW is a criterion that directly attempts to do this, however it does not necessarily distribute the value amongst the players. On the other hand, the ME criterion gives very low value.

Tractable: Linear solutions can be solved with LPs in polynomial time. Maximum entropy is a more difficult class of problems to solve, requiring exponential cone programming.

Principled: Ideally the criterion should be principled or simple in some way. ME is known to maximize the uncertainty of other player's actions while maintaining equilibrium (Ortiz et al., 2007). It also makes the fewest assumptions (Jaynes, 1957).

Robust: Small changes in the payoffs can cause step changes in the equilibrium. Some equilibria may be more robust than others to perturbations in the payoffs.

6.2 MG(C)CE and its Computation

The set of (C)CEs forms a convex polytope, and therefore any strictly convex function could uniquely select a solution. The literature only provides one such example: MECE (Ortiz et al., 2007) which has a number

³This is true if the correlation device is not considered as part of the game. If it was part of the game (for example traffic lights at a junction) the solution concept can appear prescriptive.

of appealing properties, but was found to be slow to solve large games. There is a gap in the literature for a more tractable approach, and this work proposes to use the Gini impurity (GI) (Bishop, 2006; Breiman et al., 1984). GI is a member of the Tsallis entropy family, a generalized entropy that is equivalent to GI under a certain parameterization. It is maximized when the probability mass function is uniform $\sigma = \frac{1}{|\mathcal{A}|}$ and minimized when all mass is on a single outcome. GI is popular in decision tree classification algorithms because it is easy to compute (Breiman et al., 1984). The resulting solution concept is called maximum Gini (coarse) correlated equilibrium (MG(C)CE). This approach has connections to maximum margin (Cortes and Vapnik, 1995) and maximum entropy (Jaynes, 1957). The derivations (Section 6.A.2.2) follow standard optimization theory.

6.2.1 Quadratic Program

The Gini impurity is defined as $1 - \sigma^T \sigma$, and the MG(C)CE is denoted σ^* . This work uses an equivalent standard form objective $-\frac{1}{2} \sigma^T \sigma$. The most basic form of the problem can be expressed directly as a quadratic program (QP), consisting of a quadratic objective function (Equation 6.4a) and linear constraints (Equations 6.4b and 6.4c).

$$\text{Gini objective:} \quad \max_{\sigma} -\frac{1}{2} \sigma^T \sigma \quad \text{s.t.} \quad (6.4a)$$

$$\text{(C)CE constraints:} \quad A_p \sigma \leq \epsilon \quad \forall p \quad (6.4b)$$

$$\text{Probability constraints:} \quad \sigma \geq 0 \quad e^T \sigma = 1 \quad (6.4c)$$

QPs are a well studied problem class and many techniques may be used to solve them, including convex and quadratic optimization software, such as CVXPY (Agrawal et al., 2018; Diamond and Boyd, 2016) and OSQP (Stellato et al., 2020).

6.2.2 Primal and Dual Forms

The primal objective to be optimized is $\min_{\sigma} \max_{\alpha, \beta, \lambda} L(\sigma, \alpha, \beta, \lambda) = L_{\sigma}^{\alpha, \beta, \lambda}$, where $L_{\sigma}^{\alpha, \beta, \lambda}$ is the primal Lagrangian function, $\alpha_p \geq 0$ are the dual variable vectors corresponding to the ϵ -(C)CE inequality constraints (Equation 6.4b), $\beta \geq 0$ is the dual variable vector corresponding to the distribution inequality constraints (Equation 6.4c), and λ is the dual variable corresponding to the distribution equality constraint (Equation 6.4c). By augmenting the dual variables $\alpha = [\alpha_1, \dots, \alpha_n]$ and constraints matrix $A = [A_1, \dots, A_n]$, the primal objective can be written compactly as:

$$L_{\sigma}^{\alpha, \beta, \lambda} = \frac{1}{2} \sigma^T \sigma + \alpha^T (A \sigma - \epsilon) - \beta^T \sigma + \lambda (e^T \sigma - 1), \quad (6.5)$$

where the constant vector of ones with appropriate size is denoted by e , and ϵ is a vector populated with the approximation parameter. A simplified dual version of the optimization can be formulated:

$$L^{\alpha, \beta} = -\frac{1}{2} \alpha^T A C A^T \alpha + b^T A^T \alpha - \epsilon^T \alpha - \frac{1}{2} \beta^T C \beta - b^T \beta + \alpha^T A C \beta + \frac{1}{2} b^T b,$$

where $C = I - e b^T$ normalizes by the mean, and $b = \frac{1}{|\mathcal{A}|} e$ is the uniform vector. The optimal primal solution σ^* can be recovered from the optimal dual variables α_p^* and β_p^* using

$$\sigma^* = b - C A^T \alpha^* + C \beta^*. \quad (6.6)$$

The full-support assumption states that all joint probabilities have some positive mass, $\sigma > 0$. In this scenario, the dual variable vector corresponding to the non-negative probability constraint is zero, $\beta = 0$.

Therefore, simplified primal and dual objectives can be defined.

$$L_{\sigma}^{\alpha, \lambda} = -\frac{1}{2}\sigma^T \sigma + \alpha^T (A\sigma - \epsilon) + \lambda(e^T \sigma - 1) \quad (6.7a)$$

$$L^{\alpha} = -\frac{1}{2}\alpha^T A C A^T \alpha + b^T A^T \alpha - \epsilon^T \alpha + \frac{1}{2}b^T b \quad (6.7b)$$

$$\sigma^* = b - C A^T \alpha^* \quad (6.7c)$$

6.3 Properties of MG(C)CE

This section discusses some of the properties of ϵ -MG(C)CE⁴. Section 6.A.2 contains the proofs for this section.

6.3.1 Uniqueness

Theorem 6.3.1 (Uniqueness and Existence). *MG(C)CE provides a unique solution to the equilibrium solution problem and always exists.*

6.3.2 Scalable Representation

MG(C)CE can provide solutions in general-support and, similar to MECE, MG(C)CE permits a scalable representation when the solution is full-support. Under this scenario, the distribution inequality constraint variables, β , are inactive, are equal to zero, can be dropped, and the α variables can fully parameterize the solution.

Theorem 6.3.2 (Scalable Representation). *The MG(C)CE, σ^* , has the following forms:*

$$\text{General Support: } \sigma^* = b - C A^T \alpha^* + C \beta^* \quad (6.8a)$$

$$\text{Full Support: } \sigma^* = b - C A^T \alpha^* \quad (6.8b)$$

Where e is a vector of ones, $|\mathcal{A}| = \prod_p |\mathcal{A}_p|$, $C = I - e^T b$, and $b = \frac{1}{|\mathcal{A}|} e$ are constants. $\alpha^* \geq 0$ and $\beta^* \geq 0$ are the optimal dual variables of the solution, corresponding to the (C)CE and distribution inequality constraints respectively.

Let $|\mathcal{A}_p|$ correspond to the number of actions available to player p , and the total number of joint actions, σ , is $|\mathcal{A}| = \prod_p |\mathcal{A}_p|$. For each value of σ , there is a corresponding β dual variable. The number of α dual variables is no more than the number of pair permutations $\sum_p |\mathcal{A}_p|(|\mathcal{A}_p| - 1)$ for CEs or actions $\sum_p |\mathcal{A}_p|$ for CCEs. Clearly, games with three or more players and many actions, $\sum_p |\mathcal{A}_p|(|\mathcal{A}_p| - 1) \ll \prod_p |\mathcal{A}_p|$ for CEs and $\sum_p |\mathcal{A}_p| \ll \prod_p |\mathcal{A}_p|$ for CCEs, allow for a very scalable parameterization if the full-support assumption holds. Furthermore, optimal α^* are sparse so rows can be discarded from A , in a similar spirit to SVMs (Cortes and Vapnik, 1995).

For CEs, full-support is not possible when an action is strictly dominated by another. This case can be easily mitigated by iterated elimination of strictly dominated strategies (IESDS) (Fudenberg and Tirole, 1991). This also has the desirable property of simplifying the optimization. In a similar argument, when actions are repeated (having the same payoffs), only one needs be retained with appropriate modifications to the optimization.

Among the set of ϵ -MG(C)CE there always exists one with full-support. Note that any infinitesimal positive ϵ will permit a full-support (C)CE, but ϵ -MG(C)CE does not necessarily select these. An upper bound on ϵ which permits a full-support solution is given by Theorem 6.3.3.

⁴Some of the properties discussed here also apply to MECE (Ortiz et al., 2007).

Theorem 6.3.3 (Existence of Full-Support ϵ -MG(C)CE). *For all games, there exists an $\epsilon \leq \max(Ab)$ such that a full-support, ϵ -MG(C)CE exists. A uniform solution, b , always exists when $\max(Ab) \leq \epsilon$. When $\epsilon < \max(Ab)$, the solution is non-uniform.*

6.3.3 Family of Solutions

ϵ -MG(C)CE provides an intuitive way to control the strictness of the equilibrium via the approximation parameter, ϵ , which parameterizes a family of unique solutions. Positive ϵ expands the solution set and results in a higher Gini impurity solution, at the expense of lower payoff, and approximate equilibrium. Negative ϵ shrinks the solution set to achieve a strict equilibrium and higher payoff at the expense of Gini impurity. This might also be a more robust solution (Ben-Tal et al., 2009; Wald, 1939, 1945) if the payoff is uncertain.

It is worth emphasizing a set of particularly interesting solutions within this family. Firstly the standard MG(C)CE, with $\epsilon = 0$, provides a weak equilibrium for non-trivial games (Theorem 6.3.4). Secondly, an edge case with positive ϵ is $\max(Ab)$ - ϵ -MG(C)CE which guarantees a uniform distribution solution. Converging to uniform when increasing ϵ is a desirable property (principle of insufficient reason) (Jaynes, 1957; Leonard J. Savage, 1954; Sinn, 1980). Thirdly, note that all $\epsilon < \max(Ab)$ are guaranteed to have a non-uniform distribution (Theorem 6.3.3), therefore, a $\frac{1}{2} \max(Ab)$ - ϵ -MG(C)CE could be an interesting way to regularise a MGCE towards a uniform distribution. Fourthly, because the algorithms are particularly scalable when full-support, working out the minimum ϵ such that a full-support solution exists, full- ϵ -MG(C)CE, would be useful. Finally, the solution with the smallest feasible ϵ is the min ϵ -MG(C)CE. This solution has the lowest entropy of the family, but the highest payoff, and constitutes the strictest equilibrium. Refer to Figure 7.1 for the family of solutions for the traffic lights game.

Theorem 6.3.4. *For non-trivial games (Nau et al., 2004), the MG(C)CE lies on the boundary of the polytope and hence is a weak equilibrium.*

Since the ϵ is deterministically known for the $\max(Ab)$ - ϵ -MG(C)CE, $\frac{1}{2} \max(Ab)$ - ϵ -MG(C)CE and MG(C)CE solutions, one can solve for these using the standard solvers discussed in Section 6.2. For the min ϵ -MG(C)CE one can tweak the optimization procedure to solve for this case directly by simply including a $c\epsilon$ term to minimize, where $c > 1$. Bisection search can be used to find full- ϵ -MG(C)CE.

6.3.4 Invariance

An important concept in decision theory, called cardinal utility (Mas-Colell et al., 1995), is that offset and positive scale of each player's payoff does not change the properties of the game. A notable solution concept that does not have this property is MW(C)CE.

Theorem 6.3.5 (Affine Payoff Transformation Invariance). *When a payoff is transformed $(G_p, \epsilon_p) \rightarrow (s_p G_p + b_p(a_{-p}), s_p \epsilon_p)$ for some positive s_p , the MG(C)CE of the transformed game is invariant.*

6.3.5 Computationally Tractable

In general, finding NEs is a hard problem (Daskalakis et al., 2009). While solving for any valid (C)CE is simple (basic feasible solution of a linear constraint problem) (Matouek and Gärtner, 2006), and finding a (C)CE with a linear objective is an LP, solving for a particular (C)CE can be hard. For example, MECE (Ortiz et al., 2007) requires optimizing a constrained nonlinear objective. α -Rank can be solved in cubic time in the number of pure joint strategies, $O(|\mathcal{A}|^3)$.

MG(C)CE, however, is the solution to a quadratic program, and therefore can be solved in polynomial time. Furthermore, if the assumption is made that the solution is full-support, the algorithm's variables scale better than the number of σ parameters. Space requirements are dominated by the storage of the

advantage matrix A , which requires a space of $O(n|\mathcal{A}_p||\mathcal{A}|)$ when exploiting sparsity. Computation is also on the order $O(n|\mathcal{A}_p||\mathcal{A}|)$ for gradient computation, exploiting sparsity. The number of variables depends on whether the equilibrium is general-support, $|\mathcal{A}| + n|\mathcal{A}_p|^2$, or full-support, $n|\mathcal{A}_p|^2$. It is possible to make use of sparse matrix implementations and only efficient matrix-vector multiplications are required to compute the derivatives.

6.4 Conclusion

There has been significant recent interest in solving the equilibrium selection problem ([Omidshafiei et al., 2019](#); [Ortiz et al., 2007](#)). This chapter provides a novel approach which is computationally tractable, supports general-support solutions, and has favourable scaling properties when the solution is full-support. The new solution concept MG(C)CE is rooted in the powerful principles of entropy and margin maximisation. Therefore it is a simple solution that makes limited assumptions, and is robust to many possible counter strategies ([Jaynes, 1957](#)). The MG(C)CE defines a family of unique solutions parameterized by ϵ , that can control for the properties of the distribution.

6.A Appendices

6.A.1 Generalized Entropy

Shannon's Entropy (Shannon, 1948), I_S , is a familiar quantity and is described as a measure of "information gain". The Gini Impurity (Bishop, 2006; Breiman et al., 1984) is a measurement of the probability of misclassifying a sample of a discrete random variable, if that sample were randomly classified according to its own probability mass function, $I_G = \sum_i^N \sigma_i \sum_{j \neq i} \sigma_j = 1 - \sum_i^N \sigma_i^2$. Both Shannon's entropy and Gini Impurity are maximized when the probability mass function is uniform $\sigma_i = \frac{1}{|\mathcal{A}|}$ and minimized when all mass is on a single outcome. Both metrics are used in decision tree classification algorithms, with Gini being more popular because it is easier to compute (Breiman et al., 1984).

In physics, there has been recent interest in non-extensive entropies which have been found to better model certain physical properties. One such entropy is called the Tsallis entropy, $I_T = \frac{1 - \sum_i \sigma_i^q}{q-1}$, (Havrda et al., 1967; Kaur and Buttar, 2019; Tsallis, 1988; Wang and Xia, 2017) and is parameterized by real q . A notable property of the Tsallis entropy is that it is non-additive. Assume that there are two independent variables A and B , with joint probability $P(A, B) = P(A)P(B)$, then the combined Tsallis entropy of this system is $I_T(A, B) = I_T(A) + I_T(B) + (1 - q)I_T(A)I_T(B)$. Therefore it can be seen that the $(1 - q)$ quantity is a measure of the departure from additivity, with additivity being recovered in the limit when $q \rightarrow 1$. This corresponds to the additive Shannon's entropy. The Gini impurity is recovered when $q = 2$. Therefore, the Gini impurity is a non-extensive generalized entropy.

6.A.2 Proofs of MG(C)CE Properties

6.A.2.1 Uniqueness and Existence

Theorem 6.A.1 (Uniqueness and Existence). *MG(C)CE provides a unique solution to the equilibrium solution problem and always exists.*

Proof. The problem is a concave maximization problem with linear constraints so therefore has a unique solution. Existence follows from the fact that a CE always exists. \square

6.A.2.2 Scalable Representation

Theorem 6.A.2 (Scalable Representation). *The maximum Gini (C)CE, σ^* , has the following forms:*

$$\text{General Support: } \sigma^* = b - CA^T \alpha^* + C\beta^* \quad (6.9a)$$

$$\text{Full Support: } \sigma^* = b - CA^T \alpha^* \quad (6.9b)$$

Where e is a vector of ones, $|\mathcal{A}| = \prod_p |\mathcal{A}_p|$, $C = I - e^T b$, and $b = \frac{1}{|\mathcal{A}|}e$ are constants. $\alpha^* \geq 0$ and $\beta^* \geq 0$ are the optimal dual variables of the solution, corresponding to the CE and distribution inequality constraints respectively.

Proof. Start with the primal Lagrangian form.

$$L_{\sigma}^{\alpha, \beta, \lambda} = \frac{1}{2} \sigma^T \sigma + \alpha(A\sigma - e) - \beta^T \sigma + \lambda(e^T \sigma - 1) \quad (6.10)$$

We wish to find the saddle point, $\min_{\sigma} \max_{\alpha, \beta, \lambda} L_{\sigma}^{\alpha, \beta, \lambda}$. To construct the dual Lagrangian, first take derivatives with respect to the primal variables σ , and set them equal to zero.

$$\frac{\partial L_{\sigma}^{\alpha, \beta, \lambda}}{\partial \sigma} = \sigma^* + (A^T \alpha - \beta + \lambda) = 0 \implies \sigma^* = -A^T \alpha + \beta - \lambda e \quad (6.11)$$

These can be substituted back into the primal Lagrangian.

$$L^{\alpha,\beta,\lambda} = -\frac{1}{2} [A^T \alpha - \beta + \lambda e]^T [A^T \alpha - \beta + \lambda e] - \alpha^T \epsilon - \lambda$$

Taking derivatives with respect to λ .

$$\frac{\partial L^{\alpha,\beta,\lambda}}{\partial \lambda} = -|\mathcal{A}| \lambda^* - e^T A^T \alpha_p + e^T \beta - 1 = 0 \implies \lambda^* = \frac{1}{|\mathcal{A}|} (-e^T A^T \alpha + e^T \beta - 1) \quad (6.12)$$

Substituting λ back. Remember that there are non-negative constraints on $\alpha \geq 0$ and $\beta \geq 0$. Therefore, one cannot easily solve for β to reduce this expression further. By defining $C = I - eb^T$, and $b^T = \frac{1}{|\mathcal{A}|} e^T$ (the uniform distribution), noting $b^C = 0$ and $C^T C = C$, we arrive at the general support dual Lagrangian form.

$$\begin{aligned} L^{\alpha,\beta} &= -\frac{1}{2} \left[CA^T \alpha - C\beta - \frac{1}{|\mathcal{A}|} e \right]^T \left[CA^T \alpha - C\beta - \frac{1}{|\mathcal{A}|} e \right] - \alpha^T \epsilon + b^T A^T \alpha - b^T \beta + \frac{1}{|\mathcal{A}|} \\ &= -\frac{1}{2} \alpha^T A C A^T \alpha + b^T A^T \alpha - \alpha^T \epsilon - \frac{1}{2} \beta^T C \beta - b^T \beta + \alpha^T A C \beta + \frac{1}{2} b^T b \end{aligned}$$

By combining Equations 6.11 and 6.12, we can arrive at an equation that describes the relationship between the primal and dual parameters.

$$\sigma^* = b - CA^T \alpha^* + C\beta^* \quad (6.13)$$

It is advantageous to try and obtain a more compact representation. This is achievable if σ has full support. In this case, $\beta = 0$, because none of the $\sigma \geq 0$ constraints are active which results in Equation 6.14a, the full support dual Lagrangian form.

$$L^\alpha = -\frac{1}{2} \alpha^T A C A^T \alpha + b^T A^T \alpha - \epsilon^T \alpha + \frac{1}{2} b^T b \quad (6.14a)$$

$$\sigma^* = b - CA^T \alpha^* \quad (6.14b)$$

□

Theorem 6.A.3 (Existence of Full-Support ϵ -MG(C)CE). *For all games, there exists an $\epsilon \leq \max(Ab)$ such that a full-support, ϵ -MG(C)CE exists. A uniform solution, b , always exists when $\max(Ab) \leq \epsilon$. When $\epsilon < \max(Ab)$, the solution is non-uniform.*

Proof. Note, $A\sigma \leq \epsilon \iff AC\sigma + Ab \leq \epsilon$, $Cb = 0$ and that b is the uniform distribution with maximum possible Gini impurity. Note that when $\max(Ab) \leq \epsilon$ the inequality will always hold with $\sigma = b$. And the inequality cannot hold with $\sigma = b$ when $\epsilon \leq \max(Ab)$. □

6.A.2.3 Family

Theorem 6.A.4. *For non-trivial games, the MG(C)CE lies on the boundary of the polytope and hence is a weak equilibrium.*

Proof. MG(C)CE is attempting to be near the uniform distribution. If the uniform distribution is not a (C)CE the MG(C)CE lies on the boundary of the (C)CE polytope, and by definition is weak. If the uniform distribution is a (C)CE, then it is also an NE (because it factorizes). It therefore lies on the polytope if it is a non-trivial game by (Nau et al., 2004). □

Table 6.1: Family of MG(C)CE solutions.

MG(C)CE	ϵ	Properties
$\max(Ab)\epsilon$ -MG(C)CE	$\max(Ab)$	Uniform, highest entropy, lowest payoff
$\frac{1}{2} \max(Ab)\epsilon$ -MG(C)CE	$\frac{1}{2} \max(Ab)$	Between uniform and (C)CE
full ϵ -MG(C)CE	$\leq \max(Ab)$	Minimum ϵ such that MG(C)CE is full-support
MG(C)CE	0	Weak (C)CE, NE in two-player constant sum
$\min \epsilon$ -MG(C)CE	≤ 0	Strictest (C)CE, lowest entropy, highest payoff

Table 6.1 summarizes the family of solutions that make up MG(C)CE. Note that a similar family can be defined for ME(C)CE.

6.A.2.4 Invariance

Theorem 6.A.5 (Affine Payoff Transformation Invariance). *If σ^* is the ϵ -MG(C)CE of a game, \mathcal{G} , then for each player p independently we can transform the payoff tensors $\tilde{G}_p = c_p G_p + d_p$ and approximation vector $\tilde{\epsilon}_p = a_p \epsilon_p$ for some positive c_p and real d_p scalars, without changing the solution. Furthermore, if a game, \mathcal{G} , has (C)CE constraint matrix, A , and bound vector, ϵ , then each row can be scaled independently without changing the MG(C)CE.*

Proof. The only way that a game's payoff, G , influences the solution is via the (C)CE constraint matrices A_p . Recall that these are defined as the difference between action payoffs $a_p \neq a'_p \in \mathcal{A}_p$. It is easy to see that the constant d_p will cancel immediately.

$$\begin{aligned} \tilde{A}_{p,i,j} &= \tilde{G}_p(a'_p, a_{-p}) - \tilde{G}_p(a_p, a_{-p}) \\ &= c(G_p(a'_p, a_{-p}) - G_p(a_p, a_{-p})) \end{aligned} \quad (6.15)$$

Notice that A always appears alongside the dual variables α . Therefore any scale in $\tilde{A}\tilde{\alpha} = cA\tilde{\alpha}$ can be counteracted by $\tilde{\alpha} = \frac{\alpha}{c}$, without changing the nature of the optimization.

Similar to above, not only does α_p appear alongside A_p , each element appears alongside a particular row of A_p . Therefore not only can a whole A_p be scaled by a positive factor, each row of A_p can be scaled individually. Intuitively, each row of the (C)CE constraint matrix defines an equation of a plane in the simplex, and planes are not altered when scaled by a positive factor. We may exploit this property to better condition the optimization problem. \square

6.A.3 MGCE Computation

There are several tricks that can be employed to simplify the nature of the computation problem.

6.A.3.1 Bounded Gradient Methods

It is easy to formulate gradient algorithms to solve for the MG(C)CE. It is most convenient to work in the reduced dual form of the problem as it enforces the probability equality constraint automatically, allows for making the full-support assumption, and does not require any projection routines. The computations involve sparse matrices, so appropriate sparse data structures should be used. The dual variables have a non-negative constraint, which is also sometimes referred to as a box or bound constraints in the literature. For gradient ascent, initialize $\alpha^0 = 0$, $\beta^0 = 0$, and update the variables according to their gradient, where $\text{NN}(\sigma) = \max(0, \sigma)$, ensures the variables remain non-negative.

$$\alpha^{t+1} \leftarrow \text{NN} [\alpha^t + \gamma(-ACA^T \alpha^t + Ab - \epsilon + AC\beta^t)] \quad (6.16a)$$

$$\beta^{t+1} \leftarrow \text{NN} [\beta^t + \gamma(-C\beta^t - b + C^T A^T \alpha^t)] \quad (6.16b)$$

If we assume the solution is full-support, we can simplify the dual version even further by dropping the β variable updates.

$$\alpha^{t+1} \leftarrow \text{NN} [\alpha^t + \gamma(-ACA^T \alpha^t + Ab - \epsilon)] \quad (6.17)$$

Second order derivatives are also easily computed, allowing use of bounded second order linesearch optimizers, such as L-BFGS-B (Byrd et al., 1995). Other techniques such as momentum (Rumelhart et al., 1986), preconditioning the rows of the A matrix, and iterated elimination of strictly dominated strategies of the payoff matrix will also help. An efficient conjugate gradient method can be adapted from Polyak's algorithm (O'Leary, 1980; Polyak, 1969), which is a conjugate gradient method modified to support solving problems with bounds and is proven to converge in finite iterations.

6.A.3.2 Payoff Reductions

There are two methods which could be used to reduce the size of the payoff tensor and hence reduce the complexity of the game that is required to be solved; repeated action elimination, and dominated action elimination.

Repeated Action Elimination: Consider a payoff which has repeated strategies (identical payoffs). This represents a redundancy in the game formulation and we can therefore keep only one of these actions and appropriately modify the objective to account for this alteration. Let r_p be the number of repeats for each action after elimination (i.e. $r_p = e$ if all were unique). Define $r = \otimes_p r_p$ as the flattened repeat count which is the same size as σ and $\tilde{r}_p = \otimes_{p'} \{e \text{ if } p' = p \text{ else } r_{p'}\}$. Then the constraints now become $r^T \sigma = 0$ and $A_p(\sigma \cdot \tilde{r}_p) \leq \epsilon_p$, and the objective becomes $1 - \sigma^T(\sigma \cdot r)$. This has the dual effect of reducing the number of variables and constraints in the problem and, more importantly, breaks the symmetry of repeated terms which several solvers can struggle with. It is important to run this procedure before eliminated dominated actions, because repeated actions by definition do not dominate one another.

Dominated Action Elimination: Strictly dominated strategies can be pruned from the payoff without affecting the results because dominated strategies can never have non-zero support in CEs where $\epsilon \leq 0$. Any CE solution with non-positive ϵ can exploit this reduction.

Empirically, it is common for actions to be repeated and actions to be strictly dominated by others.

6.A.3.3 Eigenvalue Normalization

Some methods, such as gradient methods, benefit from the eigenvalues of the problem being similar in magnitude. We found empirically that re-normalizing by the L_2 norm of the rows of the constraint matrix resulted in eigenvalues close to 1. This is allowed by Theorem 6.A.5.

6.A.3.4 Dual Optimal Learning Rate

For the dual form of the objective there is an optimal constant learning rate we can use which is based on the eigenvalues of the Hessian. Calculating the eigenvalues exactly may be too computationally expensive. We can instead obtain an upper bound. A good choice of learning rate that is guaranteed to converge is $\gamma = \frac{2}{\sigma_{\max} + \sigma_{\min}^+} \geq \frac{2}{\max_j \sum_i |D_{ij}| + \min_j \sum_i |D_{ij}|}$, where D is the Hessian of the dual form. A proof follows below.

Proof. C is idempotent and positive semi-definite. For any B , BB^T is positive semi-definite, therefore $(AC)(AC)^T = ACA^T$ is positive semi-definite. This is the first part of the block diagonals of the Hessian, D , which is therefore singular symmetric positive semi-definite.

It is known that the best choice of constant learning rate in this setting is $\gamma = \frac{2}{\sigma_{\max} + \sigma_{\min}^+}$. Because the Hessian is not full rank and positive semi-definite, $\sigma_{\min} = 0$. One needs to find the smallest non-zero

eigenvalue. One possible upper bound on the maximal eigenvalues of a positive semi-definitive matrix, by the Gerschgorin circle Theorem ([Gerschgorin, 1931](#)), is:

$$\sigma_{max} \leq \max_j \sum_i |D_{ij}| = \max_i \sum_j |D_{ij}| \quad (6.18)$$

$$\sigma_{min}^+ \leq \min_j \sum_i |D_{ij}| = \min_i \sum_j |D_{ij}| \quad (6.19)$$

□

Chapter 7

Joint Policy-Space Response Oracles

A popular framework called Policy-Space Response Oracles (PSRO) unifies the Nash equilibrium solving algorithms self-play, fictitious self-play, and double oracle. In addition, it proposes using reinforcement learning (RL) as an approximate best-response oracle. Go (Silver et al., 2018) and StarCraft (Vinyals et al., 2019) were solved using this framework. However there has been limited progress outside of the two-player zero-sum setting. This work proposes Joint Policy-Space Response Oracles (JPSRO), an algorithm for training agents in n-player general-sum extensive-form games, which provably converges to a correlated equilibrium (CE) or coarse correlated equilibrium (CCE). More generally, CEs and CCEs are demonstrated to be promising meta-solvers. Several experiments are conducted using (C)CE meta-solvers with JPSRO which demonstrate convergence on n-player general-sum games. This work has been published at ICML (Marris et al., 2021a,b). The convergence proofs were contributed by Paul Muller.

7.1 Introduction

Recent success in tackling two-player, zero-sum games (Silver et al., 2016; Vinyals et al., 2019) has out-paced progress in n-player general-sum games despite a lot of interest in the space (Anthony et al., 2020b; Berner et al., 2019; Brown and Sandholm, 2019; Gray et al., 2020; Jaderberg et al., 2019; Lockhart et al., 2020). One reason is because Nash equilibrium (NE) (Nash, 1951) is tractable and interchangeable in the two-player, zero-sum setting but becomes intractable (Daskalakis et al., 2009) and non-interchangeable¹ in n-player and general-sum settings. As well as being a problem in normal-form (NF) games, this problem setting arises in multiagent training when dealing with empirical games (also called meta-games (Walsh et al., 2002; Wellman, 2006)), where a game payoff tensor is populated with expected outcomes between agents playing an extensive-form (EF) game, for example the StarCraft League (Vinyals et al., 2019) and Policy-Space Response Oracles (PSRO) (Lanctot et al., 2017), a recent variant of which reached state-of-the-art results in Stratego Barrage (McAleer et al., 2020).

In this work correlated equilibrium (CE) (Aumann, 1974) and coarse correlated equilibrium (CCE) (Hannan, 1957; Moulin and Vial, 1978) are proposed as a suitable target equilibrium space for n-player general-sum games². The (C)CE solution concept has two main benefits over NE; firstly, it provides a mechanism for players to correlate their actions to arrive at mutually higher payoffs and secondly, it is computationally tractable to compute solutions for n-player general-sum games (Daskalakis et al., 2009).

¹That is, there are no longer any guarantees on the expected utility when each player plays their part of some equilibrium; guarantees only hold when all players play *the same* equilibrium. Since players cannot guarantee what others choose, they cannot optimize independently, so the Nash equilibrium loses its appeal as a prescriptive solution concept.

²The most general games (also called environments) are targeted: extensive-form games, multiagent MDPs and POMDPs (stochastic games), and imperfect information games are all solvable with this approach.

This work proposes a novel training framework, Joint Policy-Space Response Oracles (JPSRO), that converges to (C)CEs in extensive-form games. A variety of (C)CE meta-solvers are proposed which have different trade-offs for finding high value equilibria or exploring the space of policies. An empirical study shows convergence rates and social welfare across a variety of games including n-player general-sum, and common-payoff games. The result is a set of tools for theoretically solving any complete information³ multiagent problem. These tools are amenable to scaling approaches; including utilizing reinforcement learning, function approximation, and online solution solvers.

There have been numerous extensions proposed for PSRO (McAleer et al., 2020, 2021). Furthermore, several different approaches have been suggested for meta-solvers in PSRO, including α -Rank (Muller et al., 2020), Projected Replicator Dynamics (PRD) (Lanctot et al., 2017), uniform distribution (Brown, 1951; Heinrich et al., 2015), and the Nash equilibrium (McMahan et al., 2003). Another important area of related work concerns optimization based approaches (Dudik and Gordon, 2012; Farina et al., 2019a; von Stengel and Forges, 2008) and no-regret approaches (Celli et al., 2019, 2020; Morrill et al., 2021). These approaches identify specific subsets or supersets of (C)CE in the extensive-form game by constructing constraint programs or by local regret minimization using the full representation of the information state space. In contrast, the oracle approach can iteratively identify meta-games with smaller support that summarize the strategic complexity of the game compactly.

7.2 Preliminaries

7.2.1 Normal-Form and Extensive-Form Equilibria

(C)CEs provide a richer set of solutions than NEs. The maximum social welfare in (C)CEs is at least that of any NE. In particular, this allows more intuitive solutions to anti-coordination games such as chicken and traffic lights. Consider the traffic lights example; a symmetric, general-sum, two-player game consisting of two actions *go*, (G), and *wait*, (W). (G, G) results in a crash, in (W, W) no progress is made, and (G, W) and (W, G) result in progress for one of the players. Figure 7.1 shows the NE and CE solution space for the traffic lights game. The mixed NE solution (G, W) = $(\frac{1}{11}, \frac{10}{11})$ is clearly unsatisfactory ($\frac{1}{121}$ crashing and $\frac{100}{121}$ waiting). One could argue that the best solution is to have players flip a coin to decide who waits and who goes. It turns out that this solution is a valid CE and is in fact the unique solution of $\min \epsilon$ -MGCE, a novel solution concept introduced in Section 6.2.

The solution concepts discussed so far apply to normal-form (NF) games, and therefore are sometimes prefixed as such in the literature (NFCE and NFCCE) to disambiguate them from their extensive-form (EF) counterparts: EFCE (von Stengel and Forges, 2008) and EFCCE (Farina et al., 2019a). This distinction is important because although EF solutions are a natural choice in EF games: NF solutions can also be applied in EF games by using whole policies $\pi_p \in \Pi_p$ in place of strategies $a_p \in \mathcal{A}_p$. These solutions are subsets of one another; $\text{NFCE} \subseteq \text{EFCE} \subseteq \text{EFCCE} \subseteq \text{NFCCE}$ (von Stengel and Forges, 2008), therefore NFCE is the most restrictive correlation device while NFCCE is the least restrictive and is therefore capable of achieving the highest welfare. The best correlation device to use is a matter of debate in the literature. However, note that NF solutions are interesting in EF games because a) they permit the highest welfare, and b) they only require communicating recommendations once before the game starts (as opposed to EF(C)CEs which require communication at every timestep). (J)PSRO trains sets of policies and converges to an NF equilibrium. Therefore, all equilibria discussed in this work are NF and this is assumed going forward.

7.2.2 Policy-Space Response Oracles (PSRO)

Policy-Space Response Oracles (PSRO) (Lanctot et al., 2017) (Algorithm 7.1) is an iterative population based training method for multiagent learning that generalizes other well known algorithms such as ficti-

³Payoffs for all players are required for the correlation device.

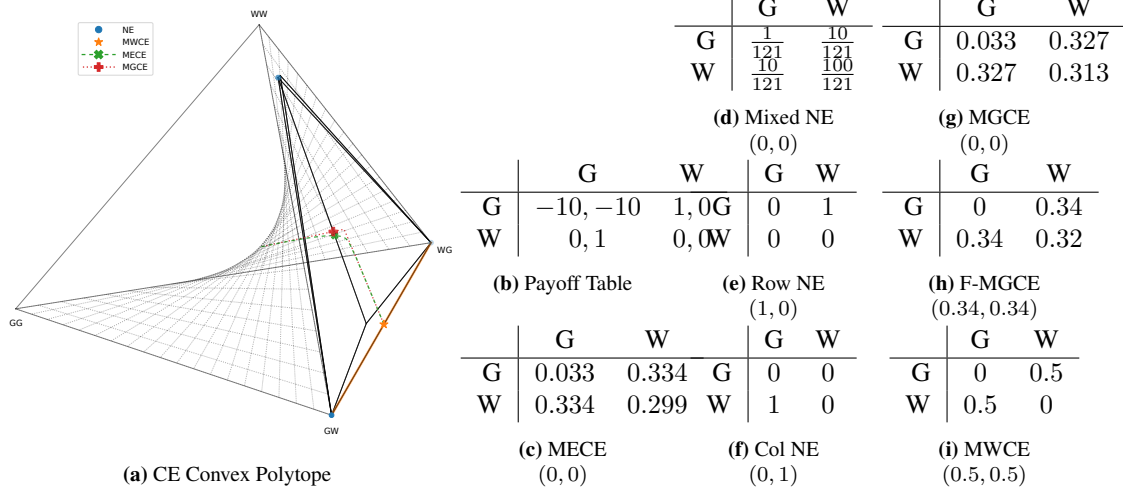


Figure 7.1: The solution landscape for the traffic lights game. The solid polytope shows the space of CE joint strategies, and the dotted surface shows factorizable joint strategies. NEs are where the surface and polytope intersect. There are three unsatisfying NEs: mixed spends most of its time waiting and does not avoid crashing, the others favour only the row or column player. One MWCE provides a better solution (note that Row NE and Col NE, and any mixture of the two, are also MWCE solutions). The center of the tetrahedron is the uniform distribution and the MECE and MGCE attempt to be near this point. The dashed lines correspond to the family of solutions permitted by MGCE and MECE when varying the approximation parameter ϵ . Both have $(GW, WG) = (0.5, 0.5)$ as the min ϵ solution. Player payoffs are given in parenthesis.

tious self play (FSP) (Brown, 1951; Heinrich et al., 2015) and double oracle (DO) (McMahan et al., 2003).

PSRO (Algorithm 7.1) operates by iteratively growing a set of policies for each player, $(\pi_p \in \Pi_p)_{p=1..n}$. At each iteration, a new policy is added to each player's set by calculating the policy that best-responds to a distribution over the set of policies found so far, $(\sigma_p)_{p=1..n}$. The best-response can be calculated exactly, or approximated using scalable RL techniques. If at any point, best-response cannot unilaterally improve the payoff over the expected payoff under the distribution, by definition the distribution is a Nash equilibrium, and the algorithm can terminate. If best-responses are restricted to pure strategies (without loss of generality), there are finite such best-responses, and therefore the algorithm will always terminate.

The important part of the algorithm is the choice of distribution to best-respond to. One choice is to best-respond to a uniform distribution over the policies. Another choice is to use the Nash distribution over the policies found so far. This is calculated by calculating Nash equilibrium of the normal-form game populated with payoffs between the set policies found so far. Either the uniform or Nash distribution converges to an NE in two-player, zero-sum games, and has recently been extended to convergence to other types of equilibria (McAleer et al., 2021; Muller et al., 2020). The work in this chapter is in line with these developments, studying convergence of a variant of PSRO with joint policy distributions and (C)CE meta-solvers in n-player general-sum games.

PSRO consists of a response oracle that estimates the best response (BR) to a joint distribution of policies. Commonly the response oracle is either an RL agent or a method that computes the exact BR. The component that determines the distribution of policies that the oracle responds to is called the meta-solver (MS). The MS operates on the meta-game (MG), which is a payoff tensor estimated by measuring the expected return (ER) of policies against one another. This is an NF game, but instead of strategies corresponding to actions, a , they correspond to policies, π . The set of deterministic policies can be huge and that of stochastic policies is infinite, therefore PSRO only considers a subset of game policies: the ones found by the BR over all iterations so far. Different MSs result in different algorithms: the uniform

Algorithm 7.1 Two-Player PSRO

```

1:  $\Pi_1^0, \Pi_2^0 \leftarrow \{\pi_1^0\}, \{\pi_2^0\}$ 
2:  $G^0 \leftarrow \text{ER}(\Pi^0)$ 
3:  $\sigma_1^0, \sigma_2^0 \leftarrow \text{MS}(G^0)$ 
4: for  $t \leftarrow \{1, \dots\}$  do
5:    $\pi_1^t, \Delta_1^t \leftarrow \text{BR}(\Pi_2^{t-1}, \sigma_2^{t-1})$ 
6:    $\pi_2^t, \Delta_2^t \leftarrow \text{BR}(\Pi_1^{t-1}, \sigma_1^{t-1})$ 
7:    $\Pi_1^t, \Pi_2^t \leftarrow \Pi_1^{t-1} \cup \{\pi_1^t\}, \Pi_2^{t-1} \cup \{\pi_2^t\}$ 
8:    $G^t \leftarrow \text{ER}(\Pi^t)$ 
9:    $\sigma_1^t, \sigma_2^t \leftarrow \text{MS}(G^t)$ 
10:  if  $\Delta_1^t + \Delta_2^t = 0$  then
11:    break
return  $(\Pi_1^{0:t}, \Pi_2^{0:t}), (\sigma_1^t, \sigma_2^t)$ 

```

Algorithm 7.2 JPSRO

```

1:  $\Pi_1^0, \dots, \Pi_n^0 \leftarrow \{\pi_1^0\}, \dots, \{\pi_n^0\}$ 
2:  $G^0 \leftarrow \text{ER}(\Pi^0)$ 
3:  $\sigma^0 \leftarrow \text{MS}(G^0)$ 
4: for  $t \leftarrow \{1, \dots\}$  do
5:   for  $p \leftarrow \{1, \dots, n\}$  do
6:      $\{^1\pi_p^t, \dots\}, \{^1\Delta_p^t, \dots\} \leftarrow \text{BR}_p(\Pi^{0:t-1}, \sigma^{t-1})$ 
7:      $\Pi_p^{0:t} \leftarrow \Pi_p^{0:t-1} \cup \{^1\pi_p^t, \dots\}$ 
8:      $G^{0:t} \leftarrow \text{ER}(\Pi^{0:t})$ 
9:      $\sigma^t \leftarrow \text{MS}(G^{0:t})$ 
10:    if  $\sum_{p,c} {}^c\Delta_p^t = 0$  then
11:      break
return  $\Pi^{0:t}, \sigma^t$ 

```

distribution results in FSP, and using the NE distribution results in an extension of DO.

7.3 Joint PSRO

JPSRO (Algorithm 7.2) is a novel extension to Policy-Space Response Oracles (PSRO) (Lanctot et al., 2017) (Algorithm 7.1) with full mixed joint policies to enable coordination among policies. Although a conceptually straightforward extension, careful attention is needed to a) develop suitable best response (BR) operators, b) develop tractable joint distribution meta-solvers (MS), c) evaluate the set of policies found so far, and d) develop convergence proofs.

This work uses notation similar notation to normal-form games, but with policies instead of strategies. Let $(\Pi_p^*)_{p=1..n}$ be the set of all policies of the extensive-form game available for each player, and $\Pi^* = \otimes_p \Pi_p^*$ be the set of all joint policies. JPSRO is an iteration-based algorithm, let $\{^c\pi_p^t, \dots\} = \Pi_p^t$ be the set of new policies found at iteration t for player p with $c \in \mathcal{C}$ indexing an individual policy within that set. The set of all policies found so far for player p is denoted $\Pi_p^{0:t}$ and the set of joint policies is denoted $\Pi^{0:t} = \otimes_p \Pi_p^{0:t}$. The expected return (ER) of an NF game, $(G_p^{0:t})_{p=1..n}$, is tracked for each joint policy found so far such that $G_p^{0:t}(\pi)$ is the expected return to player p when playing joint policy π . G_p^* is defined to be the payoff over all possible joint policies.

The MS is a function taking in the ER and returning a joint distribution, σ^t , over $\Pi^{0:t}$, such that $\sigma^t(\pi)$ is the probability to play joint policy $\pi \in \Pi^{0:t}$ at iteration t . The BR operator finds a policy which maximizes the expected return over the opponent mixed joint policies, $\pi_{-p} \in \Pi_{-p}^{0:t}$. This mixture is defined in terms of the MS joint distribution, σ^t .

7.3.1 Meta-Game Estimation

The meta-game is a normal-form game, empirically derived from the expected returns of policies in complex, potentially temporally extended, extensive-form games. Such meta-games are common in empirical game-theoretic analysis (EGTA) (Walsh et al., 2002; Wellman, 2006). Broadly, there are two approaches for calculating the meta-game: exactly via traversing the game tree or approximately by averaging over many sampled returns.

In exact policy evaluation, the exact expected return is computed for each player by traversing the entire game tree. The game tree contains all chance outcomes and randomness derived from the potentially stochastic policies. Therefore the resulting return is the true expected payoff of each player. This approach is only tractable for either small games which have small game trees, or when using deterministic policies that only require visiting a small subset of the game tree.

In larger games, or situations where the policy cannot be easily queried (for example when using a policy that depends on internal state like an LSTM), the return has to be estimated through sampling many matchups between policies. In such a setting, the meta-game will be an approximation, and will be noisy. Unfortunately, small changes in the payoffs can result in step-changes in the equilibrium. This is a known limitation with PSRO when estimating meta-games via sampling. In such a setting PSRO may only converge to an approximate NE. JPSRO inherits this limitation.

This work focuses on the exact policy evaluation regime for two reasons. Firstly, it allows empirical validation that the algorithms introduced converge exactly to (C)CEs given enough iterations. Secondly, it removes a confounding factor when comparing MSs, allowing exact conclusions to be drawn from the performance of different MSs. Although this work focuses on this exact case, JPSRO can utilize approximate meta-games, if necessary, to scale to larger problems.

7.3.2 Meta-Solvers

The purpose of a meta-solver is to calculate a joint distribution, (usually an equilibrium) from the meta-game which can be used for two purposes. Firstly, the joint can be used as a target that players will attempt to best-respond to in order to expand the set of policies. Secondly, the joint can be used to evaluate the strength of the set of policies found so far and to provide the final output distribution of the algorithm.

Historically, many of the traditional PSRO meta-solvers are factorizable solutions. Equivalently, their joint probabilities can be marginalized without losing any information. For example, the meta-solver that outputs a uniform distribution results in a parameterization of PSRO known as fictitious self play (FSP) (Heinrich et al., 2015). This approach is proven to slowly converge to an NE in two-player zero-sum games. A key advantage of this approach is that it is not necessary to compute the meta-game to obtain this distribution. An approach that converges faster in practice uses NE as the meta-solver, which is a parameterization of PSRO known as Double Oracle (DO). A drawback is that NE is hard to compute in n -player general-sum games which this work focuses on. Projected replicator dynamics (PRD) (Lanctot et al., 2017) is an evolutionary method of approximating NE, which has also been used in PSRO. α -Rank is a full joint solution concept based on the stationary distribution of a Markov chain (Omidshafiei et al., 2019). It is the one prior example of a full distribution being deployed in PSRO (Muller et al., 2020), however the authors marginalize over the distribution so that it works with PSRO.

This work proposes that (C)CEs are good candidates for meta-solvers (MSs). They are more tractable to compute than NEs and can enable coordination to maximize payoff between cooperative agents. In particular, three flavours of equilibrium MSs are proposed in this work. Firstly, greedy approaches, such as MW(C)CE, select highest payoff equilibria. In general, maximum welfare is a non-unique linear formulation that maximizes the sum of payoffs over all players. In case there are multiple (C)CEs with MW, they can be further selected using a maximum entropy criterion (MEMW(C)CE), or by randomly selecting amongst them. Secondly, maximum entropy (such as MG(C)CE and ME(C)CE) attempt to be robust against many policies through spreading weight. ME(C)CE is a unique nonlinear convex formulation that maximizes the Shannon entropy of the resulting distribution (Ortiz et al., 2007), however it is an exponential programming problem. MG(C)CE is also unique but is a quadratic programming problem. Finally, random samplers, such as Random Vertex (C)CE (RV(C)CE) attempt to explore by probing the extreme points of equilibria. RV(C)CE is a linear formulation of the standard linear (C)CE problem, using a linear cost function sampled from the unit ball. Note that this selects a random vertex on the (C)CE polytope and is not sampling from within the polytope volume or elsewhere on the polytope surface. Note that these MSs search through the equilibrium subspace, not the full policy space, and this restriction is a powerful way of achieving convergence. Note that since $\text{CEs} \subseteq \text{CCEs}$, one can also use CE MSs with JPSRO(CCE). In addition to (C)CE meta-solvers, this work also uses a couple of other (naive) baselines. Firstly, *random*

Dirichlet samples a whole joint from a flat Dirichlet distribution. Secondly *random joint* samples a random pure joint policy from the set.

7.3.3 Best-Response Operators

At iteration $t + 1$ each set, $\Pi_p^{0:t}$, can be expanded using either a CCE or CE best response (BR) operator. The type of BR operator used determines whether JPSRO converges to a CE or a CCE.

Definition 7.3.1 (JPSRO(CCE) Best-Response Operator).

$$\text{BR}_p^{t+1}(G_p(\pi_p^*, \pi_{-p}), \sigma(\pi_{-p})) \in \underset{\pi_p^* \in \Pi_p^*}{\operatorname{argmax}} \sum_{\pi_{-p} \in \Pi_{-p}^{0:t}} \sigma^t(\pi_{-p}) G_p^*(\pi_p^*, \pi_{-p}) \quad (7.1a)$$

$$\Pi_p^{0:t+1} = \Pi_p^{0:t} \cup \{\text{BR}_p^{t+1}(G_p(\pi), \sigma(\pi_{-p}))\} \quad (7.1b)$$

Compare this definition to the normal-form best-response definition (Equation (2.40)). Of course, the JPSRO(CCE) BR searches over policies, π_p^{BR} , rather than strategies, a_p^{BR} , in a normal-form game, however this difference is semantic. The important difference is that although the best-response operator only responds to the limited set of policies found so far, $\Pi_{-p}^{0:t}$, it searches for a best-response over all possible policies, Π_p^* . This is equivalent to player p finding a normal-form best-response in a game with shape $[|\Pi_1^{0:t}|, \dots, |\Pi_p^*|, \dots, |\Pi_N^{0:t}|]$, where $|\Pi_p^*|$ is the number of deterministic policies in a game. If there are multiple deterministic best-response policies it is possible to arbitrarily mix over them to produce stochastic policies.

Importantly, note that it is not necessary to know the entire normal-form game with shape $[|\Pi_1^{0:t}|, \dots, |\Pi_p^*|, \dots, |\Pi_N^{0:t}|]$ to search over $|\Pi_p^*|$. Using the framework mentioned in Section 2.2.2.1, the other players can be subsumed into the environment. The problem then becomes a single-agent single-objective optimization problem. This problem can be solved exactly by traversing the game tree using a depth-first search to find one or all deterministic best-responses. Alternatively, this problem can be approximately solved at scale with RL, which is suited to solving single-agent environments. In this setting, the learning algorithms train against randomly sampled joint-policies according to $\sigma(\pi)$, and do standard value maximization. Both on-policy (such as Policy Gradient) and off-policy (such as Q-Learning) are suitable learning algorithms. Function approximation may also be used. PSRO championed the idea of using RL (Lanctot et al., 2017). The experiments in this work use the exact tree search best-responses so that inexact best-responses do not become a confounding variable, and the exact convergence of the algorithm can be verified.

Definition 7.3.2 (JPSRO(CE) Best-Response Operator).

$$\text{BR}_p^{t+1}(G_p(\pi_p^*, \pi_{-p}), \sigma(\pi_{-p}|\pi_p'')) \in \underset{\pi_p^* \in \Pi_p^*}{\operatorname{argmax}} \sum_{\pi_{-p} \in \Pi_{-p}^{0:t}} \sigma_p^t(\pi_{-p}|\pi_p'') G_p^*(\pi_p^*, \pi_{-p}) \quad \forall \pi_p'' \in (\Pi_p^{0:t})^+ \quad (7.2a)$$

$$\Pi_p^{0:t+1} = \Pi_p^{0:t} \cup \{\text{BR}_p^{t+1}(G_p(\pi_p^*, \pi_{-p}), \sigma(\pi_{-p}|\pi_p'')) , \dots\} \quad (7.2b)$$

In contrast to the CCE BR, for CEs each player exploits each policy conditioned on possible positive-support recommended policies, π_p'' . Therefore the CE BR is similar to the CCE best-response, and can be calculated using the same approaches discussed above, however there are multiple distributions to best-respond against at each iteration. If the distribution is factorizable (like NE), then the CE BR distributions are equal for all recommended policies. Additionally, they are also equal to the CCE BR distribution. This is the exact distribution PSRO uses. Therefore JPSRO((C)CE) is a strict generalisation PSRO: it handles full joint distributions and is identical to PSRO when the joint factorizes.

7.3.4 Gap and Convergence

The relationship between the best responses and the definitions of (C)CEs can be understood by recalling Equation (2.62) for CCE, and Equation (2.51b) for WSCE.

CCE Gap and Convergence

Recalling and modifying the definition for normal-form CCEs, Equation (2.62), a distribution, $\sigma(\pi)$, over policies, $\pi \in \Pi$, is in ϵ -CCE when:

$$\text{BRV}_p^{\sigma_p^*}(G_p(\pi^*, \pi_{-p}), \sigma(\pi_{-p})) - \sum_{\pi \in \Pi_p} \sigma(\pi) G_p(\pi) \leq \epsilon_p \quad \forall p \in [1, N] \quad (7.3)$$

The first part of the LHS is known as the *best-response value*, and the second part of the LHS is known as the *expected value*. JPSRO(CCE) is concerned with finding distributions and policies such that $\epsilon_p = 0$ is feasible, so that a CCE is achieved. It is easy to measure the distance to a CCE (the CCE gap). The per-player CCE gap can be combined into a single gap by either summing or taking the maximum.

Definition 7.3.3 (CCE Gap).

$$\text{CCEGap}_p(G_p, \sigma) = \max \left[\text{BRV}_p^{\sigma_p^*}(G_p(\pi^*, \pi_{-p}), \sigma(\pi_{-p})) - \sum_{\pi \in \Pi_p} \sigma(\pi) G_p(\pi), 0 \right] \quad (7.4a)$$

$$\text{CCEGap}^\Sigma(G_p, \sigma) = \sum_p \text{CCEGap}_p(G_p, \sigma) \quad (7.4b)$$

$$\text{CCEGap}^{\max}(G_p, \sigma) = \max_p \text{CCEGap}_p(G_p, \sigma) \quad (7.4c)$$

Note that the zero-clipping is necessary because the term inside can be negative if the distribution falls within the CCE polytope (equivalently a strict CCE). This is in contrast to the NE gap which can never have such a negative component.

It is now straightforward to spot a necessary and sufficient condition for convergence. When the CCE gap is zero, the distribution is a CCE. Equivalently when the best-response value does not improve upon the expected value of the distribution for all players, the distribution is a CCE. From this condition, a vacuous proof follows.

Theorem 7.3.4 (CCE Convergence). *When using a CCE meta-solver and CCE best-response operator in JPSRO(CCE) the mixed joint policy converges to a CCE under the meta-solver distribution.*

Proof. Given a CCE distribution, $\sigma^t(\pi)$, over the meta-game, $G_p(\pi)$, at time step t , a best-response is computed for all players. At this stage there are two possibilities:

1. All players have zero CCE gap, and the distribution is at a CCE in the full game.
2. At least one player has nonzero gap, and the distribution is not a CCE in the full game.

In the latter case the policies found by best-responses with the nonzero gap will be novel. This is a property of the CCE of the meta-game. If it is possible to deviate to a strategy already found, and increase the payoff, then the distribution is not actually a CCE of the meta-game, which is a contradiction. Therefore, if the distribution over policies found so far is not a CCE of the full game, the best-response operators will produce at least one novel policy at each iteration. There are a finite number of deterministic policies, therefore JPSRO(CCE) will eventually converge. \square

PSRO and JPSRO converge, in the worst case, in a number of time steps which is exponential to the number of information states in the game (McAleer et al., 2021). This is because the unreduced normal-form representation of an extensive form game has a number of strategies equal to $|\mathcal{A}_p|^{|\mathcal{I}_p|}$, where $|\mathcal{A}_p|$ is

the number of strategies at each time step, and the proof relies on enumerating all strategies in the worst case. Fortunately, the algorithm performs much better in practice.

CE Gap and Convergence

Recalling and modifying the definition for normal-form WSCEs, Equation (2.51b), a distribution, $\sigma(\pi)$, over policies, $\pi \in \Pi$, is in ϵ -WSCE when:

$$\text{BRV}_p^{\sigma_p^*}(G_p(\pi_p^*, \pi_{-p}), \sigma(\pi_{-p}|\pi_p'')) - \sum_{\pi_{-p} \in \Pi_{-p}} \sigma(\pi_{-p}|\pi_p'') G_p(\pi_p'', \pi_{-p}) \leq \epsilon_p \quad \forall \pi_p'' \in \Pi_p, p \in [1, N] \quad (7.5)$$

The per-player CE gap can be combined into a single gap by either summing or taking the maximum.

Definition 7.3.5 (CE Gap).

$$\text{CEGap}_p(G_p, \sigma) = \sum_{\pi'' \in \Pi_p} \sigma(\pi_p'') \max \left[\text{BRV}_p^{\sigma_p^*}(G_p(\pi_p^*, \pi_{-p}), \sigma(\pi_{-p}|\pi_p'')) - \sum_{\pi_{-p} \in \Pi_{-p}} \sigma(\pi_{-p}|\pi_p'') G_p(\pi_p'', \pi_{-p}), 0 \right] \quad (7.6a)$$

$$\text{CEGap}^\Sigma(G_p, \sigma) = \sum_p \text{CEGap}_p(G_p, \sigma) \quad (7.6b)$$

$$\text{CEGap}^{\max}(G_p, \sigma) = \max_p \text{CEGap}_p(G_p, \sigma) \quad (7.6c)$$

Theorem 7.3.6 (CE Convergence). *When using a CE meta-solver and CE best-response operator in JPSRO(CE) the mixed joint policy converges to a CE under the meta-solver distribution.*

Proof. The same proof argument as Theorem 7.3.4 can be used. \square

7.3.5 Evaluation

Simply measuring convergence to NE (NE Gap, (Lanctot et al., 2017)) is suitable in two-player zero-sum games as the solution concept is prescriptive. However, it is not rich enough in mixed-motive settings. Outside of this narrow setting it is unclear how to fairly evaluate the policies that have been found. This is true for a number of reasons including: there being multiple equilibria, and equilibria not necessarily having good payoff. A combination of high payoff and stability is indicative of a strong set of policies. To get a more holistic measurement of the quality of equilibria JPSRO finds, this work measures:

- Convergence to (C)CE via the CCE gap and CE gap.
- The expected value obtained by each player.
- The number of unique policies found.

Both gap and value metrics need to be evaluated under a distribution. Using the same distribution for evaluation and exploration may be unsuitable because MSs are necessarily equilibria, may be random, or may maximize entropy. Therefore evaluation may be under other distributions such as MW(C)CE, because it constitutes an equilibrium and maximizes value. If using a (C)CE MS and the gap is positive, it is guaranteed to find a novel BR policy. Each iteration of JPSRO(CCE) produces N policies (one for each player), and JPSRO(CE) produces up to the number of policies found so far. The number of unique policies found so far could be a good indicator of how efficiently a meta-solver is exploring policy space.

7.4 Experiments CEs and CCEs as Joint Meta-Solvers

A number of (C)CE MSs are evaluated in JPSRO on Kuhn Poker, Trade Comm, and Sheriff. These cover three-player, general-sum, and common-payoff games. Implementations of all the games are available in

OpenSpiel (Lanctot et al., 2019). An exact BR oracle is used to exactly evaluate policies in the meta-game by traversing the game tree to precisely isolate the MS’s contribution to the algorithm.

The (C)CE MSs are compared against common MS including uniform, α -Rank (Muller et al., 2020; Omidshafiei et al., 2019), Projected Replicator Dynamics (PRD) (Lanctot et al., 2017) which is an NE approximator, and random vertex (coarse) correlated equilibrium (RV(C)CE) which randomly selects a solution on the vertices of (C)CE polytope. Random joint and random Dirichlet solvers are also included as baselines. The solutions to the MSs are treated as full joint distributions. Random solvers were evaluated with five seeds and the mean is plotted. When evaluating, equilibrium gaps are measured under their own MS distribution and MW(C)CE to provide a consistent and value maximizing comparison. Experiments were run for up to 6 hours, after which they were terminated.

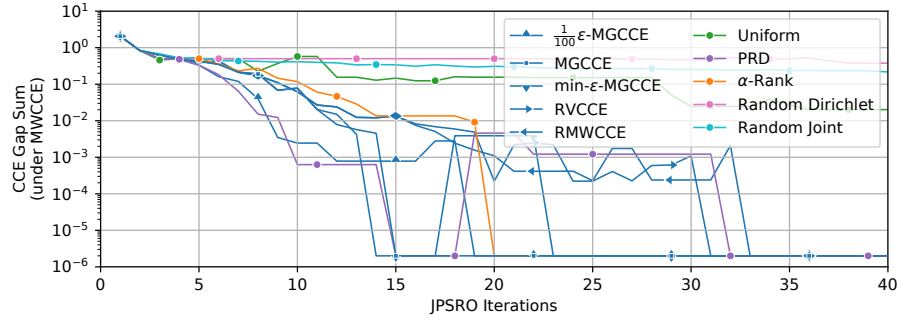
Kuhn Poker (Kuhn, 1950; Lanctot, 2014; Southey et al., 2009) is a simplified n -player zero-sum, sequential, imperfect information version of poker. It consists of $n + 1$ playing cards. In each round of the game, every player remaining *antes* one chip. One card is dealt to each player. Each player has two choices, *bet* one chip or *check*. If a player bets, other players have the option to *call* or *fold*. Out of the players that bet, the one with the highest card wins. If all players check, the player with the highest card wins. The original two-player game is described by Kuhn (1950). An n -player extension is described by Lanctot (2014). Additional information about the game (such as equilibrium) can be found in Hoehn et al. (2005). The two-player variant is solvable with PSRO, however the three-player version benefits from JPSRO. The results in Figure 7.2a show rapid convergence to equilibrium.

Trade Comm (Sokota et al., 2021) is a two-player, common-payoff trading game, where players attempt to coordinate on a compatible trade. In this game each player (in secret) receives one of I different items. The first player can then make one of I utterances to the second agent, and vice versa. Then each agent chooses one of I^2 trades in private, if the trade is compatible both agents receive 1 reward, otherwise both receive 0. The goal of the agents is therefore to find a bijection between the items and utterances and the trade proposal. There are I^4 deterministic policies per player, and good learning algorithms will be able to search over these policies. Because the game is common-payoff, it is very transitive, and has many dominated strategies, however there are multiple strategies with equal payoff, and therefore many equilibria in partially explored policy space. It is for this reason many learning algorithms get stuck exploiting sub-optimal policies they have already found. Figure 7.2b shows a remarkable dominance of CCE MSs. It is clear that traditional PSRO MSs cannot cope with this cooperative setting.

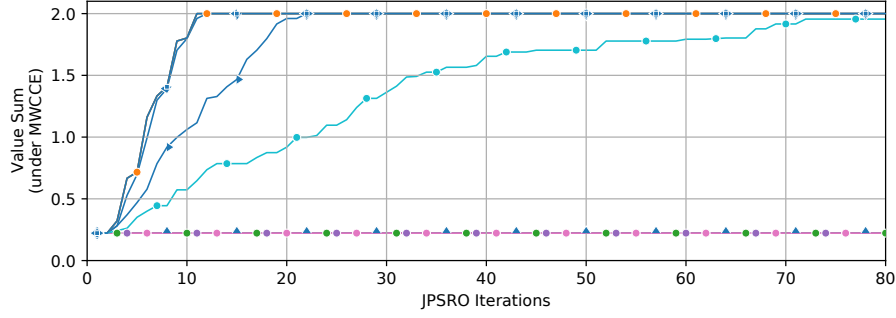
Sheriff (Farina et al., 2019c) is a simplified two-player general-sum version of the board game Sheriff of Nottingham (Farina et al., 2019c). This negotiation game consists of a smuggler, who is motivated to import contraband without getting caught, and a sheriff, who is motivated to either find contraband or accept bribes. The players negotiate a bribe over several rounds after which the bribe is accepted or rejected. If the sheriff finds contraband, the smuggler pays a fine, otherwise if no contraband is found the sheriff must pay compensation to the smuggler. The smuggler also gets value from smuggling goods. The game has different optimal values for NFCCE, EFCCE, EFCE, and NFCE solutions concepts. Figure 7.2c shows that JPSRO is capable of finding the optimal value.

7.5 Discussion

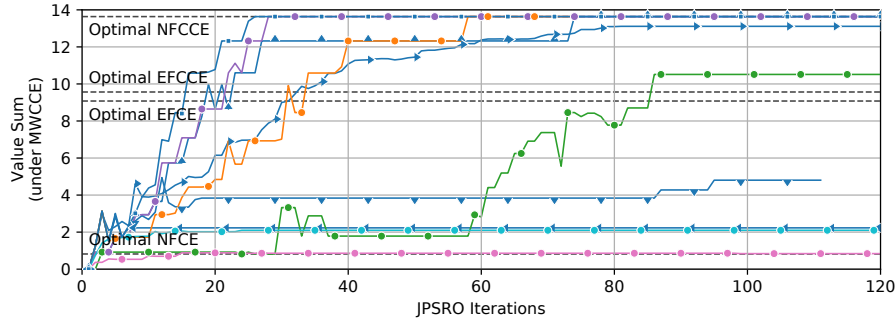
PSRO has proved to be a formidable learning algorithm in two-player zero-sum games. However, in the absence of a correlating signal, a single joint policy is, in general, insufficient to represent a correlated equilibrium. To see this, let us consider the Traffic Light game (Figure 7.1b). One possible correlated equilibrium consists of recommending (G, W) half of the time, and (W, G) the other half. Let us now consider this game as an extensive-form, partial-information game, where the row player first chooses their action, and the column player then chooses their own without knowing the action chosen by the row



(a) CCE Gap on three-player Kuhn Poker. Several MSs converge to within numerical accuracy (data is clipped) of a CCE.



(b) Value sum on three-item Trade Comm. The approximate CCE MS was not sufficient to converge in this game, however all valid CCE MSs were able to converge to the optimal value sum.



(c) Value sum on Sheriff. The optimal maximum welfare of other solution concepts are included to highlight the appeal of using NFCCE.

Figure 7.2: JPSRO(CCE) on various games. Additional metrics can be found in Section 7.A.2. MGCCE is consistently a good choice of MS over the games tested.

player. In the absence of a correlating signal, it is impossible for the column player to know which action the row player has played, and therefore playing (G, W) or (W, G) becomes impossible, as the column player is unable to change their action as a function of the action taken by the row player. Therefore, without modifying the game and observation space to add a correlating signal, convergence to a correlated equilibrium necessarily requires a distribution over joint policies. Population based training (PBT), a set of methods that slowly grow the space of (joint) policies, therefore appears to be the appropriate framework to converge to (C)CEs without adding correlating signals to the considered game.

JPSRO, with (C)CE MSs, is able to correlate behaviour and shows promising results on n-player general-sum games. The secret to the success of these methods seems to lie in (C)CEs ability to compress the search space of opponent policies to an expressive and non-exploitable subset. For example, no dominated policies are part of CE, and during execution there are no policies a player would rather deviate to.

For (C)CE MSs, if there is a value-improving BR it is guaranteed to be a novel policy.

There is a rich polytope of possible equilibria to choose from, however, an MS must pick one at each time step. There are three competing properties which are important in this regard: exploitation, robustness, and exploration. For exploitation, maximum welfare equilibria appear to be useful. However, to prevent JPSRO from stalling in a local equilibrium it is essential to randomize over multiple solutions satisfying the maximum welfare criterion. To produce robust BRs, entropy maximizing MSs (such as MG(C)CE) have better empirical value and convergence than the uniform MS. For exploration, randomly selecting a valid equilibrium at each iteration outperforms random joint and random Dirichlet by a significant margin (similar to AlphaStar’s “exploiter policies” (Vinyals et al., 2019)). Furthermore, one could also switch between MSs at each iteration to achieve the best mix of exploitation and exploration. Another strength of (C)CE MSs is that they appear to perform well across many different games, with different numbers of players and payoff properties.

7.6 Conclusion

This chapter invents a new algorithm, JPSRO, and proves that JPSRO converges to an NF(C)CE over joint policies in extensive-form games. Furthermore, there is empirical evidence that some MSs also result in high value equilibria over a variety of games. (C)CEs are important in evaluating policies in n-player general-sum games and thoroughly evaluate several MSs. Finally, this work proposes that JPSRO can scale to large problems by exploiting function approximation and RL.

7.A Appendices

7.A.1 JPSRO Hyper-parameters

There are several ways of implementing JPSRO in practice through various hyper-parameters.

Best Response: This work uses an exact best response calculation that assigns uniform probability over valid actions for states with zero reach probability. However, other best response approaches will also work including reinforcement learning.

Pool Type: The data structure used to store the policies found so far can either be a set or a multi-set. Using a set ensures that all policies are unique and only appear once even if multiple iterations produce the same best response policy. Some meta-solvers rely on repeated policies being present for convergence (for example, the uniform meta-solver can converge in two-player, zero-sum because the repeated policies trend to an NE over repeats). In this case using a multi-set is more suitable. This parameterization is only relevant when using tabular policies which can be checked for equality.

Player Updates Per Iteration: It is not necessary to find the best response for all players at every iteration. Other strategies such as cycling through players or randomly selecting a player will work too. It is sufficient that over time all players should be updated. Updating a single player at a time is more efficient when minimizing the number of best responses necessary for convergence, however updating all can be done in parallel.

Best Responses Per Iteration: When computing the CE best response, each player has several best responses to calculate. It is not necessary to compute them all and, even if they are all computed, it is not necessary to add them all to the pool of policies. The best responses can be calculated at random. And only best responses with nonzero gap need be added, or perhaps only the one with the largest gap. In order to measure convergence to a CE, all best responses (and their gaps) must be computed.

Policy Initialization: Policies can be initialized in any manner and the algorithm will converge to an equilibrium under any initial condition. However, the initial policies do determine the space of equilibrium reachable (so for example it may not be possible to find the MWCE from all initial policies). JPSRO works, without limitation, using only deterministic policies, however stochastic policies are supported too. A stochastic uniform policy over valid actions is a reasonable setting.

Best Response Type: The most important parameterization is picking one of the two best response types: CE and CCE. The resulting algorithm is named either JPSRO(CE) or JPSRO(CCE) respectively.

Meta-Solvers: The second most important parameterization is the type of meta-solver to use (Table 7.1). An important constraint is that JPSRO(CE) is only guaranteed to converge under CE meta-solvers. JPSRO(CCE) must use CCE meta-solvers (noting that CEs are a subset of CCEs).

7.A.2 Extended Experiments

Extended experiments were conducted over three extensive-form games to demonstrate the versatility of the algorithm over n-player general-sum games. For each game both JPSRO(CCE) and JPSRO(CE) algorithms are evaluated under all suitable meta-solvers and baselines.

For JPSRO(CCE): initialize using uniform policies, update all players at every iteration, and use multi-sets for the pool. For JPSRO(CE): initialize using uniform policies, update all players at every iteration, only add the highest-gap BR to the pool for each player at each iteration, and use multi-sets for the pool. For random meta-solvers the experiment is repeated five times and the average is plotted, otherwise the experiment is deterministic. The experiments were run for 6 hours, after which any that had not finished were truncated.

In order to measure performance, five metrics were tracked:

1. The gap to equilibrium under a maximum welfare equilibrium (MW(C)CE) distribution. This de-

Table 7.1: Summary of meta-solvers used during experiments and their properties. The normalized ϵ is assumed, for example $\frac{1}{100}\epsilon$ -MGCE means $\frac{1}{100} \max(Ab)\epsilon$ -MGCE.

Meta-Solver	Joint	CCE	CE	Max Val	Max Ent	Rand
Uniform					✓	
PRD						
α -Rank	✓					
Rand Dirichlet	✓					✓
Rand Joint	✓					✓
RMWCCE	✓	✓		✓		✓
RVCCE	✓	✓				✓
$\frac{1}{100}\epsilon$ -MGCE	✓	ϵ			✓	
MGCE	✓	✓			✓	
min ϵ -MGCE	✓	✓			✓	
RMWCE	✓	✓	✓	✓		✓
RVCE	✓	✓	✓			✓
$\frac{1}{100}\epsilon$ -MGCE	✓	ϵ	ϵ		✓	
MGCE	✓	✓	✓		✓	
min ϵ -MGCE	✓	✓	✓		✓	

scribes how close the algorithm is to finding a set of joint policies that are in exact equilibrium in the extensive-form game.

2. The gap to equilibrium under the meta-solver's distribution. This is the gap that JPSRO theoretically converges to when using (C)CEs.
3. The value of the game to the players under the MW(C)CE distribution.
4. The value of the game to the players under the meta-solver's distribution.
5. The number of unique policies found so far.

Ultimately, the algorithm should be finding high-value joint policies that are in equilibrium, over a variety of games. The first game is a purely competitive, three-player game called Kuhn Poker (Figure 7.3). The second game is a purely cooperative, common-payoff game called Trade Comm (Figure 7.4). The final game is a general-sum game called Sheriff (Figure 7.5).

7.A.3 Open Source Code

An open source implementation of JPSRO is available in OpenSpiel (Lanctot et al., 2019) under https://github.com/deepmind/open_spiel/blob/master/open_spiel/python/examples/jpsro.py.

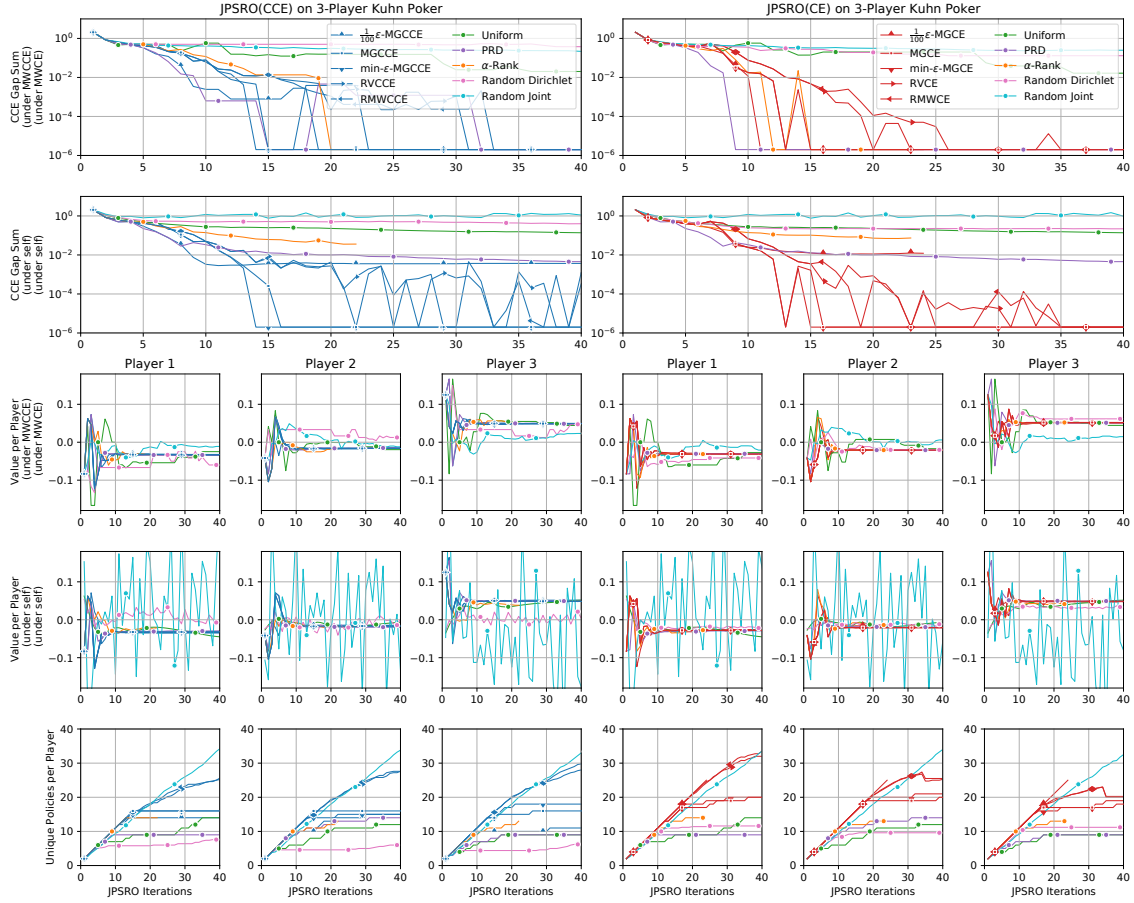


Figure 7.3: JPSRO(CCE) and JPSRO(CE) on three-player Kuhn Poker. All (C)CE MSs, PRD and α -Rank find joint policies capable of supporting equilibrium (although α -Rank was slow and was terminated after 6 hours). This is some evidence that classic MSs designed for the two-player, zero-sum setting can generalize well to the three-player, zero-sum.

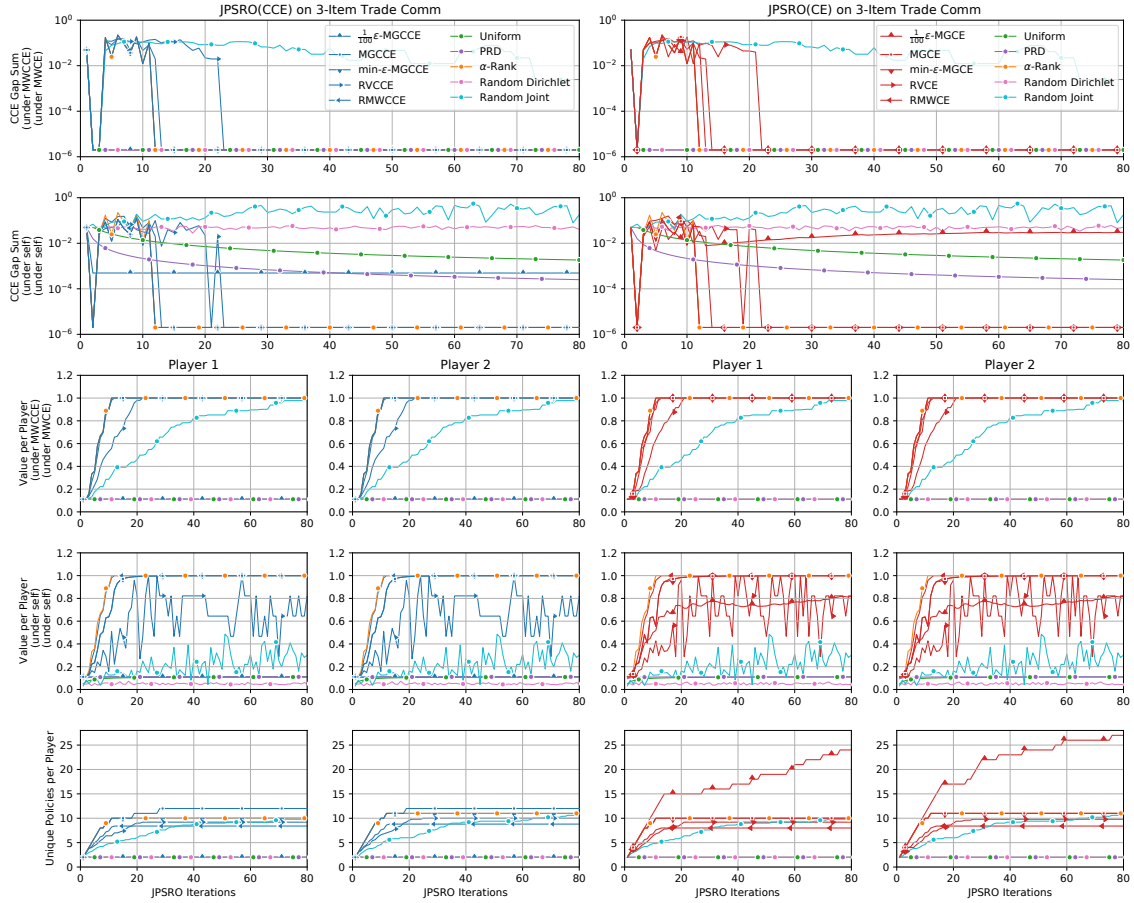


Figure 7.4: JPSRO(CCE) and JPSRO(CE) on three-item Trade Comm. In JPSRO(CCE), $\frac{1}{100}$ min-MGCCE fails to find the maximum welfare equilibrium, however, all other (C)CE MSs find the maximum welfare equilibrium. Unexpectedly, α -Rank performs well on this game, while all other classic MSs fail to make progress on this purely cooperative game. Performing well on this game requires exploration, so the random joint MS is able to make progress, albeit naively and slowly.

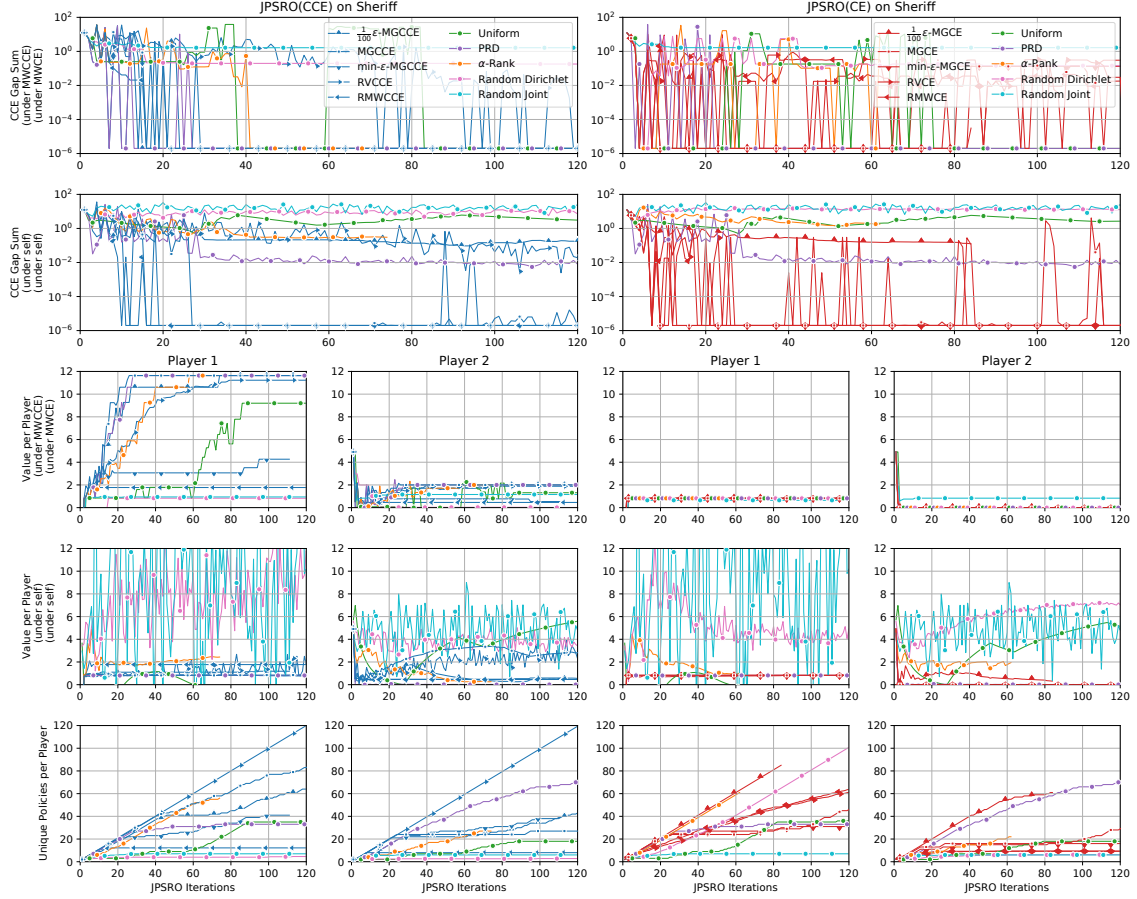


Figure 7.5: JPSRO(CCE) and JPSRO(CE) on Sheriff. This game is interesting because it is general-sum and different solution concepts have different optimal maximum welfare values. The maximum welfare NFCCE is 13.64 for the smuggler and 2.0 for the sheriff which JPSRO(CCE) successfully finds, while the maximum welfare NFCE is 0.82 for the smuggler and 0.0 for the sheriff which JPSRO(CE) successfully finds. This demonstrates the appeal of using NFCCE as a target equilibrium. Interestingly, for this game, $\frac{1}{100}\epsilon$ -MG(C)CE was able to produce BRs of high enough quality to converge which is evidence that scaled methods that only approximate (C)CEs may be enough in some settings. RMWCCE converged to an equilibrium, but not the welfare maximizing one, providing evidence that greedy MSs are not always suitable. In a similar argument, min- ϵ -MGCE did not reach the maximum welfare solution within the allocated number of iterations. RV(C)CE is efficient at finding novel policies but ones of limited utility. PRD and α -Rank perform well and find the maximum welfare (C)CE equilibria.

Chapter 8

Neural Equilibrium Solvers

Normal-form solution concepts such as Nash equilibrium, correlated equilibrium, and coarse correlated equilibrium are potentially useful components for many multiagent machine learning algorithms. Unfortunately, solving a normal-form game requires using an iterative solver that could take a prohibitive amount of time, a non-deterministic amount of time, or could fail. This work develops a special equivariant neural network architecture to accurately predict equilibria in all games of fixed shape. The network is capable of uniquely selecting an equilibrium that maximizes relative entropy, welfare, stability, or a combination thereof. The network is trained without needing to generate any supervised training data. It shows remarkable zero-shot generalization to larger games. Furthermore, this framework can be used to find vertices of the convex solution polytope, enabling approximation of the entire equilibrium space of any fixed shape game. This work is published at NeurIPS ([Marris et al., 2022a](#)).

8.1 Introduction

Normal-form solution concepts such as Nash equilibrium (NE) ([Nash, 1951](#)), correlated equilibrium (CE) ([Aumann, 1974](#)), and coarse correlated equilibrium (CCE) ([Moulin and Vial, 1978](#)) are useful components and subroutines for many multiagent machine learning algorithms. For example, value-based reinforcement learning algorithms for solving Markov games, such as Nash Q-learning ([Hu and Wellman, 2003](#)) and Correlated Q-Learning ([Greenwald and Hall, 2003](#)) maintain state action values for every player in the game. These action values are equivalent to per-state normal-form games, and policies are equilibrium solutions of these games. Critically, this policy will need to be recomputed each time the action-value is updated during training, and for large or continuous state-space Markov games, every time the agents need to take an action. Another class of multiagent algorithms are those in the space of empirical game-theoretic analysis (EGTA) ([Walsh et al., 2002](#); [Wellman, 2006](#)) including PSRO ([Lanctot et al., 2017](#); [McMahan et al., 2003](#)), JPSRO ([Marris et al., 2021a](#)), and NeuPL ([Liu et al., 2022a,b](#)). These algorithms are capable of training policies in extensive-form games, and require finding equilibria of empirically estimated normal-form games as a subroutine (the “meta-solver” step). In particular, these algorithms have been critical in driving agents to superhuman performance in Go ([Silver et al., 2016](#)), Chess ([Silver et al., 2018](#)), and StarCraft ([Vinyals et al., 2019](#)).

Unfortunately, solving for an equilibrium can be computationally complex. NEs are known to be PPAD ([Chen et al., 2009](#); [Daskalakis et al., 2009](#)). (C)CEs are defined by linear constraints and, if a linear objective is used to select an equilibrium, can be solved by linear programs (LPs) in polynomial time. However, in general, the solutions to LPs are non-unique (e.g. zero-sum games), and therefore are unsuitable equilibrium selection methods for many algorithms, and unsuitable for training neural networks which benefit from unambiguous targets. Objectives such as maximum Gini (a quadratic program (QP)

introduced in Chapter 6), and maximum entropy (Ortiz et al., 2007) (a nonlinear program), are unique but are more complex to solve.

As a result, solving for equilibria often requires deploying iterative solvers, which theoretically can scale to large normal-form games but may (i) take an unpredictable amount of time to converge, (ii) take a prohibitive amount of time to do so, and (iii) may fail unpredictably on ill-conditioned problems. Furthermore, classical methods (Fargier et al., 2022; Lemke and Howson, 1964; McKelvey et al., 2016) (i) do not scale, and (ii) are non-differentiable. This limits the applicability of equilibrium solution concepts in multiagent machine learning algorithms.

Therefore, there exists an important niche for approximately solving equilibria in medium sized normal-form games, quickly, in batches, reliably, and in a deterministic amount of time. With appropriate care, this goal can be accomplished with a neural network which amortizes up-front training cost to map normal-form payoffs to equilibrium solution concepts quickly at deployment time. This work proposes the Neural Equilibrium Solver (NES). This network is trained to optimize a composite objective function that weights accuracy of the returned equilibrium against auxiliary objectives that a user may desire such as maximum entropy, maximum welfare, or minimum distance to some target distribution. This work introduces several innovations into the design and training of NES so that it is efficient and accurate. Unlike most supervised deep learning models, NES avoids the need to explicitly construct a labeled dataset of (game, equilibrium) pairs. Instead, a loss function that can be minimized in an unsupervised fashion from only game inputs is derived. The loss also exploits the duality of the equilibrium problem. Instead of solving for equilibria in the primal space, NES solves for them in the dual space, which has a much smaller representation. This work utilizes a training distribution that efficiently represents the space of all normal-form games of a desired shape and uses an invariant preprocessing step to map games at test time to this space. The network architecture consists of a series of layers that are equivariant to symmetries in games such as permutations of players and strategies, which reduces the number of training steps and improves generalization performance. The network architecture is independent of the number of strategies in the game and demonstrates interesting zero-shot generalization to larger games. This network can either be pre-trained before being deployed, trained online alongside another machine learning algorithm, or a mixture of both. Note that training the network is itself an iterative optimization problem, but once a model has been found, approximating an equilibrium is achieved with a fixed number of computations.

8.2 Preliminaries

Game Theory

Most concepts for normal-form game theory are described in Section 2.2.2. In addition, let a *cubic* game be one where every player has an equal number of strategies: $|\mathcal{A}_1| = |\mathcal{A}_p| \forall p$. This property is rarely relevant from a game theoretic perspective but becomes notable when designing equivariant neural network architectures.

Equilibrium Solution Concepts

Normal-form equilibrium solution concepts are described thoroughly in Section 2.2.2.3. Note an additional property about NEs in two-player zero-sum games.

Theorem 8.2.1 (Two-Player Zero-Sum Marginal of CCEs). *For two-player zero-sum games, the marginal, $\sigma^*(a_p) = \sum_{a_{-p}} \sigma^*(a)$, of any exact ($\epsilon = 0$) CCE, $\sigma^*(a)$, is also an NE.*

Therefore the CCE machinery can be used to solve for NEs in two-player zero-sum games.

Neural Network Solvers

Approximating NEs using neural networks is known to be agnostic PAC learnable (Duan et al., 2021). There is also work learning (C)CEs (Bai et al., 2020; Jin et al., 2021) and training neural networks to approximate

NEs (Duan et al., 2021; Hartford et al., 2016) on subclasses of games. Learned NE meta-solvers have been deployed in PSRO (Feng et al., 2021). Differentiable neural networks have been developed to learn QREs (Ling et al., 2018). NEs for contextual games have been learned using fixed point (deep equilibrium) networks (Heaton et al., 2021). A related field, L2O (Chen et al., 2021), aims to learn an iterative optimizer more suited to a particular distribution of inputs, while this work focuses on learning a direct mapping. No work exists training a general approximate mapping from the full space of games to (C)CEs with flexible selection criteria.

8.3 Maximum Welfare Minimum Relative Entropy (C)CEs

Previous work has argued that having a unique objective to solve for equilibrium selection is important. The principle of maximum entropy (Jaynes, 1957) has been used to find unique equilibria (Ortiz et al., 2007). In maximum entropy selection, payoffs are ignored and selection is based on minimizing the distance to the uniform distribution. This has two interesting properties: (i) it makes defining unique solutions easy, (ii) the solution is invariant transformations (such as offset and positive scaling) of the payoff tensor. While these solutions are unique, they both result in weak and low payoff equilibria because they find solutions on the boundary of the polytope. Meanwhile, the literature tends to favour maximum welfare (MW) because it results in high value for the agents and is a linear objective, however in general it is not unique. The composite objective function is composed of (i) minimum relative entropy (MRE, also known as Kullback-Leibler divergence) between a target joint, $\hat{\sigma}(a)$, and the equilibrium joint, $\sigma(a)$, (ii) distance between a target approximation, $\hat{\epsilon}_p$, and the equilibrium approximation, ϵ_p , (iii) maximum of a linear objective, $\sum_{a \in \mathcal{A}} \sigma(a)W(a)$, where $W : \mathcal{A} \rightarrow \mathbb{R}$. The objective is constrained by the (i) distribution constraints ($\sum_a \sigma(a) = 1$ and $\sigma(a) \geq 0$) and, (ii) either CCE constraints (Equation (2.59)) or CE constraints (Equation (2.54)).

$$\arg \max_{\sigma, \epsilon_p} \mu \sum_{a \in \mathcal{A}} \sigma(a)W(a) - \sum_{a \in \mathcal{A}} \sigma(a) \ln \left(\frac{\sigma(a)}{\hat{\sigma}(a)} \right) - \rho \sum_p (\epsilon_p^+ - \epsilon_p) \ln \left(\frac{1}{\exp(1)} \frac{\epsilon_p^+ - \epsilon_p}{(\epsilon_p^+ - \hat{\epsilon}_p)} \right) \quad (8.1)$$

The approximation weight, ρ , and welfare weight, μ , are hyperparameters that control the balance of the optimization. The maximum approximation parameter, ϵ_p^+ , is another constant that is usually chosen to be equal to the payoff scale (Section 8.4.1). The approximation term is designed to have a similar form to the relative entropy and is maximum when $\hat{\epsilon}_p = \epsilon_p$. This equilibrium selection framework is named Target Approximate Maximum Welfare Minimum Relative Entropy ($\hat{\epsilon}$ -MWMRE).

In particular, a distance to a target approximation, $\hat{\epsilon}_p$, is used for two reasons. Firstly, it allows soft-targeting of negative approximations which find strict equilibria. Only equilibria with nonnegative approximations are guaranteed to exist (Theorem 2.2.12). Therefore the network can be trained to target strict equilibria, without needing strict equilibria to exist for all games. Secondly, the definitions of the primal variables (discussed below) favour full-support solutions. Having a soft-target approximation parameter allows the optimization to behave for non-full-support solutions.

8.3.1 Dual of ϵ -MWMRE (C)CEs

Rather than performing a constrained optimization, it is easier to solve the dual problem, $\arg \min_{\alpha_p} L^{(C)CE}$ (derived in Section 8.A.1), where $\alpha_p^{CE}(a'_p, a''_p) \geq 0$ are the dual deviation gains corresponding to the CE constraints, and $\alpha_p^{CCE}(a'_p) \geq 0$ are the dual deviation gains corresponding to the CCE constraints. Note that

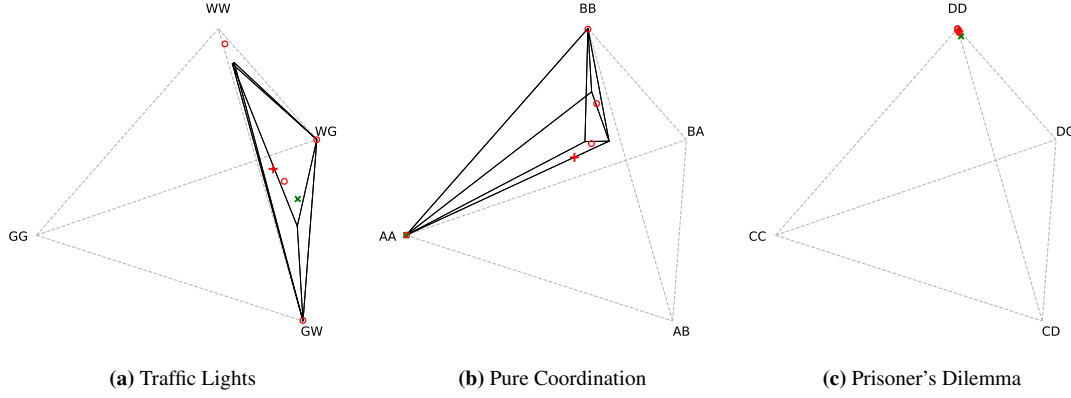


Figure 8.1: Diagrams for three 2×2 normal-form games, showing their (C)CE solution polytope on the joint simplex (in two-strategy games CEs and CCEs are equivalent). An **MWME NES**, trained by sampling over the space of payoffs and welfare targets, is used to approximate the MW(C)CE solution (\times). An **MRE NES**, trained by sampling over the space of payoffs and joint targets, is used to approximate the ME(C)CE ($+$), and all pure joint target MRE(C)CEs (\circ). The networks have never trained on these games.

it is not necessary to optimize over the primal joint, $\sigma(a)$. The Lagrangian is defined:

$$L^{\text{CE}} = \ln \left(\sum_{a \in \mathcal{A}} \hat{\sigma}(a) \exp \left(l^{\text{CE}}(a) \right) \right) + \sum_p \epsilon_p^+ \sum_{a'_p, a''_p} \alpha_p^{\text{CE}}(a'_p, a''_p) - \rho \sum_p \epsilon_p^{\text{CE}} \quad (8.2a)$$

$$L^{\text{CCE}} = \ln \left(\sum_{a \in \mathcal{A}} \hat{\sigma}(a) \exp \left(l^{\text{CCE}}(a) \right) \right) + \sum_p \epsilon_p^+ \sum_{a'_p} \alpha_p^{\text{CCE}}(a'_p) - \rho \sum_p \epsilon_p^{\text{CCE}} \quad (8.2b)$$

The logits $l^{(\text{C})\text{CE}}(a)$ are defined:

$$l^{\text{CE}}(a) = \mu \sum_a W(a) - \sum_p \sum_{a'_p, a''_p} \alpha_p^{\text{CE}}(a'_p, a''_p) A_p^{\text{CE}}(a'_p, a''_p, a) \quad (8.3a)$$

$$l^{\text{CCE}}(a) = \mu \sum_a W(a) - \sum_p \sum_{a'_p} \alpha_p^{\text{CCE}}(a'_p) A_p^{\text{CCE}}(a'_p, a) \quad (8.3b)$$

The primal joint and primal approximation parameters are defined:

$$\sigma^{(\text{C})\text{CE}}(a) = \frac{\hat{\sigma}(a) \exp \left(l^{(\text{C})\text{CE}}(a) \right)}{\sum_{a \in \mathcal{A}} \hat{\sigma}(a) \exp \left(l^{(\text{C})\text{CE}}(a) \right)} \quad (8.4a)$$

$$\epsilon_p^{(\text{C})\text{CE}} = (\hat{\epsilon}_p - \epsilon_p^+) \exp \left(-\frac{1}{\rho} \left\{ \sum_{a'_p, a''_p} \alpha_p^{\text{CE}}(a'_p, a''_p), \sum_{a'_p} \alpha_p^{\text{CCE}}(a'_p) \right\} \right) + \epsilon_p^+ \quad (8.4b)$$

8.4 Neural Network Training

The network maps the payoffs of a game, $G_p(a)$, and the targets $(\hat{\sigma}(a), \hat{\epsilon}_p, W(a))$ to the dual deviation gains, $\alpha_p^{(\text{C})\text{CE}}$, that define the equilibrium. The duals are a significantly more space-efficient objective target ($\sum_p |\mathcal{A}_p|^2$ for CEs and $\sum_p |\mathcal{A}_p|$ for CCEs) than the full joint ($\prod_p |\mathcal{A}_p|$), particularly when scaling the number of strategies and players. The joint, $\sigma(a)$, and approximation, ϵ_p , can be computed analytically

from the dual deviation gains and the inputs using Equations (8.4a) and (8.4b). The network is trained by minimizing the loss, $L^{(\text{CCE})}$ (Equations (8.2a) and (8.2b)), over a training distribution of games and game properties, $\mathbb{E}_{G \sim \mathcal{G}, \dots} [L^{(\text{CCE})}]$. The resulting model is called a Neural Equilibrium Solver (NES).

8.4.1 Training Distribution and Input Preprocessing

Most commonly, in literature favours sampling normal-form games from the uniform or normal distribution. When training neural networks, this introduces two problems:

1. It biases the distribution of games solvable by the network. For example, if $G_p(a) \sim \mathcal{U}(0, 1)$, any payoff with values greater than 1 or smaller than 0 will be out of distribution. Similarly, a normal distribution technically has domain over all real numbers, but values close to the mean will be more frequently sampled. Therefore a payoff with zero mean, $\bar{G}_p(a)$, may be better approximated than a payoff with identical equilibria, $\bar{G}_p(a) + b_p$, further from the mean.
2. It unnecessarily requires the network to learn offset and scale equilibrium-invariance. It is known in theory (Theorem 3.2.1) that these transforms do not change the equilibria. Having the network learn this invariance is an unnecessary burden and would only do so approximately.

To mitigate these issues, this work carefully selects the training distribution and input preprocessing to ensure the resulting network can handle all possible normal-form games of a fixed shape and is not biased to certain scales or offsets of the payoffs.

For the training distribution payoffs are sampled from the equilibrium-invariant embedding (EIE) (Chapter 3). There is an algorithm for sampling uniformly over this space (Algorithm 3.1). Although this space is an embedding of the full space of games, it contains games with all possible nontrivial equilibria. The embedding is a product manifold over N spheres, where sphere has a dimensionality of $|\mathcal{A}|$. A modification is made to this embedding: instead of sampling from a unit-sphere, a sphere with radius $\sqrt{|\mathcal{A}|}$ is used. This can be simply implemented by multiplying the sample by the constant $\sqrt{|\mathcal{A}|}$. This ensures that each element in the sampled payoff has unit variance, which is a useful property that complements weight initialization strategies (Glorot and Bengio, 2010) in neural networks.

$$\tilde{G}_p(a) \sim \sqrt{|\mathcal{A}|} \text{EIE}(|\mathcal{A}_1|, \dots, |\mathcal{A}_N|) \quad (8.5)$$

It is known that $-\frac{1}{2}\sqrt{|\mathcal{A}|} \leq \tilde{G}_p(a) \leq \frac{1}{2}\sqrt{|\mathcal{A}|}$ for any norm, and therefore $-\sqrt{|\mathcal{A}|} \leq A_p(\dots, a) \leq \sqrt{|\mathcal{A}|}$. Furthermore $-\sqrt{|\mathcal{A}|} \leq \sum_a \sigma(a) A_p(a'_p, a''_p, a) \leq \sqrt{|\mathcal{A}|}$, which means the only the values of approximation parameters that need to be considered are $-\sqrt{|\mathcal{A}|} \leq \tilde{\epsilon}_p \leq \sqrt{|\mathcal{A}|}$. This is because values larger than $\sqrt{|\mathcal{A}|}$ will always permit any joint and values less than $-\sqrt{|\mathcal{A}|}$ will always be infeasible. Therefore, $\epsilon_p^+ = \sqrt{|\mathcal{A}|}$.

$$\tilde{\epsilon}_p \sim \mathcal{U}(-\sqrt{|\mathcal{A}|}, \sqrt{|\mathcal{A}|}) \quad (8.6)$$

The target joint is sampled from the flat Dirichlet distribution, and welfare is sampled from a unit sphere.

$$\tilde{\sigma}_p \sim \text{Dir}(1) \quad (8.7)$$

$$W_p(a) \sim \text{Sphere}(|\mathcal{A}|) \quad (8.8)$$

This results in a natural training distribution for the network. However, the network will have only been trained on the equilibrium-invariant embedding. Fortunately, any payoff can be projected to the equilibrium-invariant embedding, without changing its equilibrium. The approximation parameter is also scaled and

clipped within the relevant range. The welfare and joint are also normalized to be unit variance.

$$G_p(a) = \sqrt{|\mathcal{A}|} \frac{G_p(a) - \frac{1}{|\mathcal{A}_p|} \sum_{a_p} G_p(a_p, a_{-p})}{\left\| G_p(a) - \frac{1}{|\mathcal{A}_p|} \sum_{a_p} G_p(a_p, a_{-p}) \right\|_2} \quad (8.9a)$$

$$\hat{\epsilon}_p = \text{clip} \left(\frac{\epsilon_p}{\left\| G_p(a) - \frac{1}{|\mathcal{A}_p|} \sum_{a_p} G_p(a_p, a_{-p}) \right\|_2}, -\hat{\epsilon}^+ = -\sqrt{|\mathcal{A}|}, +\hat{\epsilon}^+ = +\sqrt{|\mathcal{A}|} \right) \quad (8.9b)$$

$$W(a) = \sqrt{|\mathcal{A}|} \frac{W(a) - \frac{1}{|\mathcal{A}|} \sum_a W(a)}{\left\| W(a) - \frac{1}{|\mathcal{A}|} \sum_a W(a) \right\|_m} \quad (8.9c)$$

$$\hat{\sigma}(a) = |\mathcal{A}| \sqrt{\frac{|\mathcal{A}| + 1}{|\mathcal{A}| - 1}} \left(\hat{\sigma}(a) - \frac{1}{|\mathcal{A}|} \right) \quad (8.9d)$$

The selection criterion is invariant to all of these transforms, however the welfare is sensitive to the transforms. Care should be taken to ensure that welfare is modified so that it selects for the same equilibrium as before transformation. Finally, the inputs $(G_p(a), \hat{\sigma}(a), \hat{\epsilon}_p, W(a))$ are then broadcast and concatenated together so that they result in an input of shape $[C, N, |\mathcal{A}_1|, \dots, |\mathcal{A}_N|]$, where the channel dimension $C = 4$, if all inputs are required.

Note that sampling from the equilibrium-symmetric embedding (Chapter 3) and incorporating a sorting preprocessing step, is an option to further reduce the space of payoffs. However, instead the network exploits symmetries in the equilibrium definitions using an equivariant architecture (Section 8.4.3).

8.4.2 Gradient Calculation

The gradient update is found by taking the derivative of the loss (Equations (8.2a)-(8.2b)) with respect to the dual variables, α . Note that computing a gradient does not require knowing the optimal joint $\sigma^*(a)$, so the network can be trained in an unsupervised fashion, from randomly generated inputs, $G_p(a)$, $\hat{\sigma}(a)$, $\hat{\epsilon}_p$, and $W(a)$.

$$\frac{\partial L^{\text{CE}}}{\partial \alpha_p^{\text{CE}}(a'_p, a''_p)} = - \sum_a A_p(a'_p, a''_p, a) \sigma^{\text{CE}}(a) + \epsilon_p^{\text{CE}} \quad (8.10a)$$

$$\frac{\partial L^{\text{CCE}}}{\partial \alpha_p^{\text{CCE}}(a'_p)} = - \sum_a A_p(a'_p, a) \sigma^{\text{CCE}}(a) + \epsilon_p^{\text{CCE}} \quad (8.10b)$$

The dual variables, $\alpha_p^{\text{CE}}(a'_p, a''_p)$ or $\alpha_p^{\text{CCE}}(a'_p)$, are outputs of the neural network, with learned parameters θ . Gradients for these parameters can be derived using the chain rule:

$$\begin{aligned} \frac{\partial L^{\text{CE}}}{\partial \theta} &= \frac{\partial L^{\text{CE}}}{\partial \alpha_p^{\text{CE}}(a'_p, a''_p), \alpha_p^{\text{CCE}}(a'_p)} \frac{\partial \alpha_p^{\text{CE}}(a'_p, a''_p)}{\partial \theta} \\ \frac{\partial L^{\text{CCE}}}{\partial \theta} &= \frac{\partial L^{\text{CCE}}}{\partial \alpha_p^{\text{CCE}}(a'_p)} \frac{\partial \alpha_p^{\text{CCE}}(a'_p)}{\partial \theta} \end{aligned}$$

Backprop efficiently calculates these gradients, and many powerful neural network optimizers (Duchi et al., 2011; Kingma and Ba, 2014; Tieleman et al., 2012) and ML frameworks (Abadi et al., 2015; Bradbury et al., 2018; Paszke et al., 2019) can be leveraged to update the network parameters.

8.4.3 Equivariant Architectures

The ordering of strategies and players in a normal-form game is unimportant, therefore the output of the network should be equivariant under two types of permutation; (i) strategy permutation, and (ii) player

permutation. Specifically, for some strategy permutation $\tau_p(1), \dots, \tau_p(|\mathcal{A}_p|)$ applied to each element of a player's inputs (payoffs, target joint, and welfare), the outputs must also have permuted dimensions: $(\alpha_p(a_p) = \alpha_p(\tau_p(a_p)))$ and $\sigma^\tau(a_1, \dots, a_N) = \sigma(\tau_1(a_1), \dots, \tau_N(a_N))$. Likewise, for some player permutation $\tau(1), \dots, \tau(N)$, the outputs must be transposed: $(\alpha_p^\tau(a_p) = \alpha_{\tau(p)}(a_{\tau(p)}))$ and $\sigma^\tau(a_1, \dots, a_N) = \sigma(a_{\tau(1)}, \dots, a_{\tau(N)})$. The latter equivariance can only be exploited by a network if all players have the same number of strategies ("cubic games"). There are $|\mathcal{A}_p|!$ possible strategy permutations for each player and $N!$ player permutations, resulting in $N! (|\mathcal{A}_p|!)^N$ possible equivariant permutations of each sampled payoff. Note that this is much greater than the number of joint strategies in a game, $N! (|\mathcal{A}_p|!)^N \gg |\mathcal{A}_p|^N$, which is an encouraging observation when considering how this approach will scale to large games. Utilizing an equivariant architecture (Shawe-Taylor, 1993; Wood and Shawe-Taylor, 1996; Zaheer et al., 2018) is therefore crucial to scale to large games because each sample represents many possible inputs. Equivariant architectures have been used before for two-player games (Hartford et al., 2016).

Payoffs Transformations

The main layers of the architecture consists of activations with shape $[C, N, |\mathcal{A}_1|, \dots, |\mathcal{A}_N|]$, which is the same shape as a payoff tensor (with a channel dimension). Layers with this shape are referred to as "pay-off" layers even when after being transformed they cease to semantically be payoffs. When transforming payoffs, functions of the form are considered:

$$g_{l+1}(c_{l+1}, p, a_1, \dots, a_N) = f \left(\sum_{c_l^i}^{IC_l} w(c_{l+1}, c_l^i) \text{Con}_i^I [\phi_i(g_l(c_l, p, a_1, \dots, a_N))] + b(c_{l+1}) \right) \quad (8.11)$$

Where f is any equivariant nonlinearity¹, w are learned network weights, b are learned network biases, Con is the concatenate function along the channel dimension, and ϕ_i is one of many possible equivariant pooling functions. Some equivariant functions which map payoff structures to payoff structures are:

$$\begin{aligned} g(p, a_1, \dots, a_N) & \quad (8.12a) & \phi_{p, a_q} g(p, a_1, \dots, a_N) & \quad (8.12e) & \phi_{a_p} g(q, a_1, \dots, a_N) & \quad (8.12i) \\ \phi_{a_1, \dots, a_N} g(p, a_1, \dots, a_N) & \quad (8.12b) & \phi_{p, a_{-q}} g(p, a_1, \dots, a_N) & \quad (8.12f) & \phi_{a_{-p}} g(q, a_1, \dots, a_N) & \quad (8.12j) \\ \phi_{p, a_1, \dots, a_N} g(p, a_1, \dots, a_N) & \quad (8.12c) & \phi_{a_q} g(p, a_1, \dots, a_N) & \quad (8.12g) & \phi_{a_q} g(q, a_1, \dots, a_N) & \quad (8.12k) \\ \phi_p g(p, a_1, \dots, a_N) & \quad (8.12d) & \phi_{a_{-q}} g(p, a_1, \dots, a_N) & \quad (8.12h) & \phi_{a_{-q}} g(q, a_1, \dots, a_N) & \quad (8.12l) \end{aligned}$$

For example, consider one such function, $\phi_i = \sum_{a_1}$, which is invariant across any permutation of a_1 (similar to sum-pooling in CNNs), and equivariant over permutations of p, a_2, \dots, a_N . In general one can use $\phi_{\subseteq \{p, a_1, \dots, a_N\}} g(p, a_1, \dots, a_N)$, where ϕ can perform mean-pooling, max-pooling, or another operation. If all players have an equal number of strategies, for some functions, weights can be shared over all $p \in [1, N]$ because of symmetry (Shawe-Taylor, 1994). Note that the number of trainable parameters scales with the number of input and output channels, and not with the size of the game (Figure 8.2), therefore it is possible for the network to generalize to games with different numbers of strategies. The basic layer, g_{l+1} , therefore comprises of a linear transform of a concatenated, broadcasted set of pooling functions.

Payoffs to CCE Duals Transformations

Payoffs can be transformed to CCE duals, $\alpha_p^{\text{CCE}}(c_{l+1}, a'_p)$, by using a combination of a subset of the equivariant functions ϕ_i discussed above that sum over at least $-p$. If the number of strategies are equal for each player, the transformation weights can be shared and the duals can be stacked into a single object for more efficient computation in later layers: $\alpha_p^{\text{CCE}}(c_{l+1}, p, a'_p) = \text{Stack}_p (\alpha_p^{\text{CCE}}(c_{l+1}, a'_p))$.

¹ Common nonlinearities such as element-wise (ReLU, tanh, sigmoid), and SoftMax are all equivariant.

Payoffs to CE Duals Transformations

The transformation to produce the CE duals is more complex. CE duals, $\alpha_p^{\text{CCE}}(c_{l+1}, a_p'', a_p')$, need to be *symmetrically equivariant*. This property can be obtained by (i) independently generating two CCE duals and, (ii) taking outer operations, \boxdot (for example sum or product), over them.

$$\alpha_p^{\text{CE}}(c_{l+1}, a_p'', a_p') = \hat{f}(\alpha_p^{\text{CCE}}(c_{l+1}, a_p'') \boxdot \alpha_p^{\text{CCE}}(c_{l+1}, a_p')) \quad (8.13)$$

Where \hat{f} is any equivariant nonlinearity *with zero diagonal*². The diagonal is zero because it represents the dual of the deviation gain when deviating from a strategy to itself, which is zero, and therefore cannot be violated. This is a useful property which will be exploited in later dual layers. These can also be stacked if players have an equal number of strategies: $\alpha^{\text{CCE}}(c_{l+1}, p, a_p', a_p'') = \text{Stack}_p(\alpha_p^{\text{CCE}}(c_{l+1}, a_p', a_p''))$.

CCE Duals Transformations

Because the payoff activations are high-dimensional, it is worthwhile to operate on them in dual space. When transforming CCE duals the following mapping can be used:

$$\alpha_{l+1}^{\text{CCE}}(c_{l+1}, p, a_p') = f\left(\sum_{c_l^i}^{IC_l} w(c_{l+1}, c_l^i) \text{Con}_i^I[\phi_i(\alpha_l^{\text{CCE}}(c_l, p, a_p'))] + b(c_{l+1})\right) \quad (8.14)$$

where f is any equivariant nonlinearity, and ϕ_i is from a set of only two possible equivariant transformation functions (and two more, Equations (8.15c) and (8.15d), if the game is cubic). A SoftPlus nonlinearity is used on the final layer to ensure the output is nonnegative and has gradient everywhere.

$$\alpha_p(a_p') \quad (8.15a) \quad \phi_{a_p'} \alpha_p(a_p') \quad (8.15b) \quad \phi_p \alpha_p(a_p') \quad (8.15c) \quad \phi_{p, a_p'} \alpha_p(a_p') \quad (8.15d)$$

CE Duals Transformations

When transforming CE duals, these functions are suitable:

$$\alpha_{l+1}^{\text{CE}}(c_{l+1}, p, a_p', a_p'') = \hat{f}\left(\sum_{c_l}^{IC_l} w(c_{l+1}, c_l^i) \text{Con}_i^I[\phi_i(\alpha_l^{\text{CE}}(c_l, p, a_p', a_p''))] + b(c_{l+1})\right) \quad (8.16)$$

For the CE case, the equivariant linear transformations are more complex: they are symmetric over the recommended and deviation strategies. Fortunately this is a well studied equivariance class (Thiede et al., 2020), which can be fully covered by combining seven transforms which comprise of different sums and transpositions of the input. All zero-diagonal equivariant CE dual pooling functions are:

$$\alpha_p(a_p', a_p'') \quad (8.17a) \quad \phi_{a_p'} \alpha_p(a_p', a_p'') \quad (8.17c) \quad \phi_{a_p''} \alpha_p(a_p', a_p'') \quad (8.17e) \quad \phi_{a_p', a_p''} \alpha_p(a_p', a_p'') \quad (8.17g)$$

$$\alpha_p(a_p'', a_p') \quad (8.17b) \quad \phi_{a_p'} \alpha_p(a_p'', a_p') \quad (8.17d) \quad \phi_{a_p''} \alpha_p(a_p'', a_p') \quad (8.17f) \quad \phi_{p, a_p', a_p''} \alpha_p(a_p', a_p'') \quad (8.17h)$$

Activation Variance

Because the equivariant network possibly involves summing over dimensions of the inputs, activations are no longer independent of one another, so extra care needs to be taken when initializing the network to avoid variance explosion. Three techniques are used to combat this: (i) inputs are scaled to unit variance as described previously, (ii) the network is randomly initialized with variance scaling to ensure the variance at every layer is one, and (iii) use BatchNorm (Ioffe and Szegedy, 2015) between every layer. Weight decay is used to regularize the network.

²Masking is sufficient: e.g. $\hat{f}(a_p', a_p'') = (1 - I(a_p', a_p''))f(a_p', a_p'')$, where I is the identity matrix.

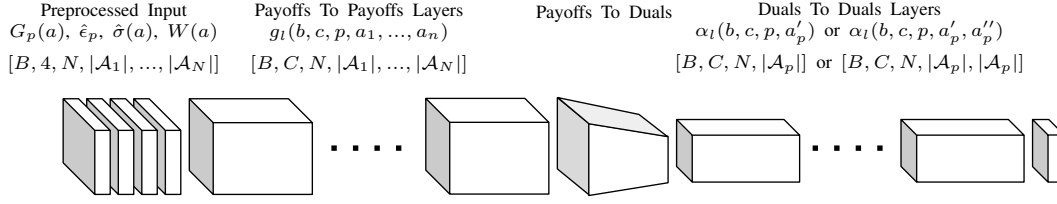


Figure 8.2: Network architecture showing the name, indices and shape (**B**atch, **C**hannels, **N**umber of players, **A**ctions per player) of each layer. Other architectures are possible, for example some of the inputs (target approximation, target joint, or welfare) could be passed in at a later layer.

Table 8.1: Neural network solution parameterizations.

	$G_p(a)$	$\hat{\sigma}(a)$	$\hat{\epsilon}_p$	$\hat{\epsilon}^+$	$W(a)$	ρ	μ
ME	$\sim L_2$	$\frac{1}{ \mathcal{A} }$	0	1	0	$\gg 1$	0
MT	$\sim L_2$	$\hat{\sigma}(a)$	0	1	0	$\gg 1$	0
MU	$\sim L_2$	$\frac{1}{ \mathcal{A} }$	0	1	$\sum_p G_p(a)$	$\gg 1$	$\gg 1$
MWE	$\sim L_2$	$\frac{1}{ \mathcal{A} }$	0	1	$\sim L_2$	$\gg 1$	$\gg 1$
MRE	$\sim L_2$	$\sim \text{Dir}(1)$	0	1	0	$\gg 1$	0
MS	$\sim L_2$	$\frac{1}{ \mathcal{A} }$	-1	1	0	$\gg 1$	0
$\hat{\epsilon}_p$ -ME	$\sim L_2$	$\frac{1}{ \mathcal{A} }$	$\sim \text{U}(-1, 1)$	1	0	$\gg 1$	0
$\hat{\epsilon}_p$ -MWE	$\sim L_2$	$\frac{1}{ \mathcal{A} }$	$\sim \text{U}(-1, 1)$	1	$\sim L_2$	$\gg 1$	$\gg 1$
$\hat{\epsilon}_p$ -MRE	$\sim L_2$	$\sim \text{Dir}(1)$	$\sim \text{U}(-1, 1)$	1	0	$\gg 1$	0

Advanced Architectures

More advanced architectures such as ResNet (He et al., 2015) or Transformers (Vaswani et al., 2017) are possible. An example of the final architecture is summarized in Figure 8.2.

8.4.4 Parameterizations

The composite objective framework allows us to define a number of combinations of auxiliary objectives. Several interesting specifications are highlighted in Table 8.1. The most basic is maximum entropy (ME) which simply finds the unique equilibrium closest to the uniform distribution according to the relative entropy distance. This distribution need not be uniform, it could be any target distribution (MT). A welfare objective parameterized on the payoffs could be used to find a maximum welfare (MW) solution. The two previous solutions can be generalized to solve for any welfare (maximum welfare and entropy (MWE)) or any target (minimum relative entropy (MRE)). Furthermore nonzero approximation are possible, for example finding the minimum possible approximation parameter results in the Maximum Strength (MS) solution. Finally, equilibria for any approximation parameter, ϵ , for the objectives discussed so far can be parameterized ($\hat{\epsilon}$ -MWME and $\hat{\epsilon}$ -MRE).

8.5 Performance Experiments

Traditionally performance of NE, and (C)CE solvers has focused on evaluating time to converge to a solution within some tolerance. Feedforward neural networks can produce *batches* of solutions quickly³ and deterministically. For non-trivial games this is much faster than what an iterative solver could hope to achieve. The experiments therefore focus evaluation on the trade-offs of the neural network solver, namely (i) how long it takes to train, and (ii) how accurate the solutions are. For the latter, there are two useful

³Inference, $\frac{\text{step time}}{\text{batch size}}$, is around $1\mu\text{s}$ on the hardware used for this experiment.

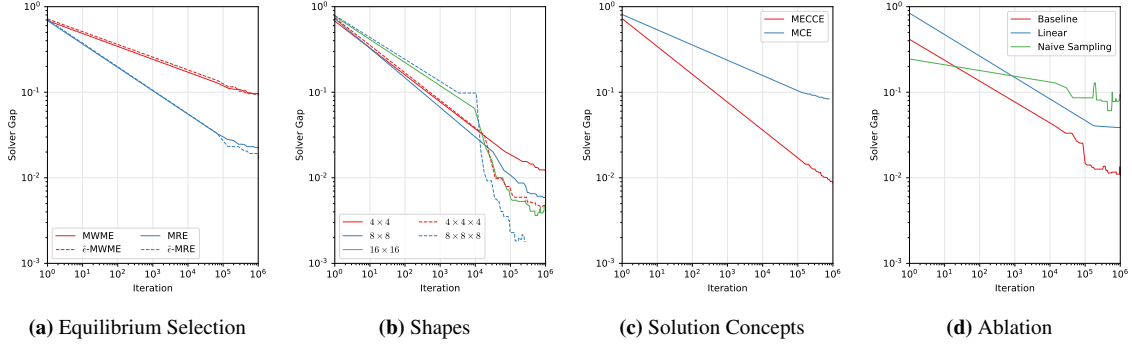


Figure 8.3: Sweeps and ablation studies showing the average solver gap of three experiment seeds evaluated over 512 sampled games against the number of train steps. Subfigure (a) shows 4×4 games over different equilibrium selection, (b) shows MECCE over games with different numbers of players and strategies, (c) shows CE and CCE concepts on 8×8 games, and (d) shows ablation experiments on MECCE $4 \times 4 \times 4$ games.

metrics:

$$\text{Solver Gap: } \frac{1}{2} \sum_a |\sigma^*(a) - \sigma(a)| \quad (8.18a)$$

$$\text{(C)CE Gap: } \sum_p \left[\max_a \sum_a (A_p(\cdot, a) - \epsilon_p) \sigma(a) \right]^+ \quad (8.18b)$$

The first (Solver Gap) measures the distance to the exact unique solution found by an iterative solver⁴, $\sigma^*(a)$, and is bounded between 0 and 1, and is zero for perfect prediction. The second ((C)CE Gap) measures the distance to the equilibrium solution polytope, and is zero if it is within the polytope.

Parameterization Sweeps

The experiment shows performance across a number of parameterizations, including (i) different equilibrium selection criteria (Figure 8.3a), (ii) different shapes of games (Figure 8.3b), and (iii) different solution concepts (Figure 8.3c).

Classes of Games

It is known that some distributions of game payoffs are harder for some methods to solve than others (Porter et al., 2008b). This experiment compares performance across a number of classes of transfer games (Appendix Table 8.2) for a single MECCE and a single MECE (Ortiz et al., 2007) Neural Equilibrium Solver trained on 8×8 games. Figure 8.4 shows the worst, mean, and best performance over 512 samples from each class in terms of (i) distance to any equilibrium, and (ii) distance to the target equilibrium found by an iterative solver. The performance of a uniform joint is used as a naive baseline, as the gap can be artificially reduced by scaling the payoffs. With regards to equilibrium violation, ME is tricky because it lies on the boundary of the polytope, so some violation is expected in an approximate setting. The plots showing the failure rate and run time of the iterative solver are to intuit difficult classes. The baseline iterative solver takes about 0.05s to solve a single game, the network can solve a batch of 4096 games in 0.0025s. For most classes, the NES is very accurate with a solver gap of around 10^{-2} . Some classes of games are indeed more difficult and these align with games that iterative equilibrium solvers struggle with. This hints that difficult games are ill-conditioned.

Ablations

The ablation experiments show the performance (Figure 8.3d) of the proposed method compared with (i)

⁴Implemented in CVXPY (Agrawal et al., 2018; Diamond and Boyd, 2016), which leverages the ECOS (Domahidi et al., 2013) solver.

Table 8.2: Classes of random games: three equilibrium-invariant embeddings (with L_1 , L_2 , and L_∞ norms), and a subset (Porter et al., 2008a) of GAMUT (Nudelman et al., 2004) games using the functions in parenthesis and parameterized with the `-random.params` flag.

Name	Game Description
L_1	L_1 Invariant
L_2	L_2 Invariant
L_∞	L_∞ Invariant
D1	Bertrand Oligopoly (BertrandOligopoly)
D2	Bidirectional LEG, Complete Graph (BidirectionalLEG-CG)
D3	Bidirectional LEG, Random Graph (BidirectionalLEG-RG)
D4	Bidirectional LEG, Star Graph (BidirectionalLEG-SG)
D5	Covariance Game, $\rho = 0.9$ (CovariantGame-Pos)
D6	Covariance Game, $\rho \in [-1/(N-1), 1]$ (CovariantGame)
D7	Covariance Game, $\rho = 0$ (CovariantGame-Zero)
D8	Dispersion Game (DispersionGame)
D9	Graphical Game, Random Graph (GraphicalGame-RG)
D10	Graphical Game, Road Graph (GraphicalGame-Road)
D11	Graphical Game, Star Graph (GraphicalGame-SG)
D12	Graphical Game, Small-World (GraphicalGame-SW)
D13	Minimum Effort Game (MinimumEffortGame)
D14	Polymatrix Game, Complete Graph (PolymatrixGame-CG)
D15	Polymatrix Game, Random Graph (PolymatrixGame-RG)
D16	Polymatrix Game, Road Graph (PolymatrixGame-Road)
D17	Polymatrix Game, Small-World (PolymatrixGame-SW)
D18	Uniformly Random Game (RandomGame)
D19	Travelers Dilemma (TravelersDilemma)
D20	Uniform LEG, Complete Graph (UniformLEG-CG)
D21	Uniform LEG, Random Graph (UniformLEG-RG)
D22	Uniform LEG, Star Graph (UniformLEG-SG)

a linear network, and (ii) no invariant pre-processing with naive payoff sampling (each element sampled using a uniform distribution). Both result in significant reduction in performance.

Scaling

Due to the size of the representation of the payoffs, $G_p(a)$, the inputs and therefore the activations of the network grow significantly with the number of joint strategies in the game. Therefore without further work on sparser payoff representation, NES is limited by size of payoff inputs. For further discussion see Section 8.7. Nevertheless, Table 8.3 shows good performance when scaling to moderately sized games. Note that the “solver gap” metric is incomplete on larger games because the ECOS evaluation solver fails to converge.

Generalization

An interesting property of the NES architecture is that its parameters do not depend on the number of strategies in the game. Therefore the generalization ability of the network is tested, zero-shot, on games with different numbers of strategies (Table 8.4). There are two observations: (i) NES only weakly generalizes to other game sizes under the solver gap metric, and (ii) NES strongly generalizes to larger games under the CCE gap, remarkably achieving zero violation. Therefore the network retains the ability to reliably find CCEs in larger games, but does struggle to accurately select the target MWMRE equilibrium. This could be mitigated by training the network on a mixture of game sizes, which is left for future work.

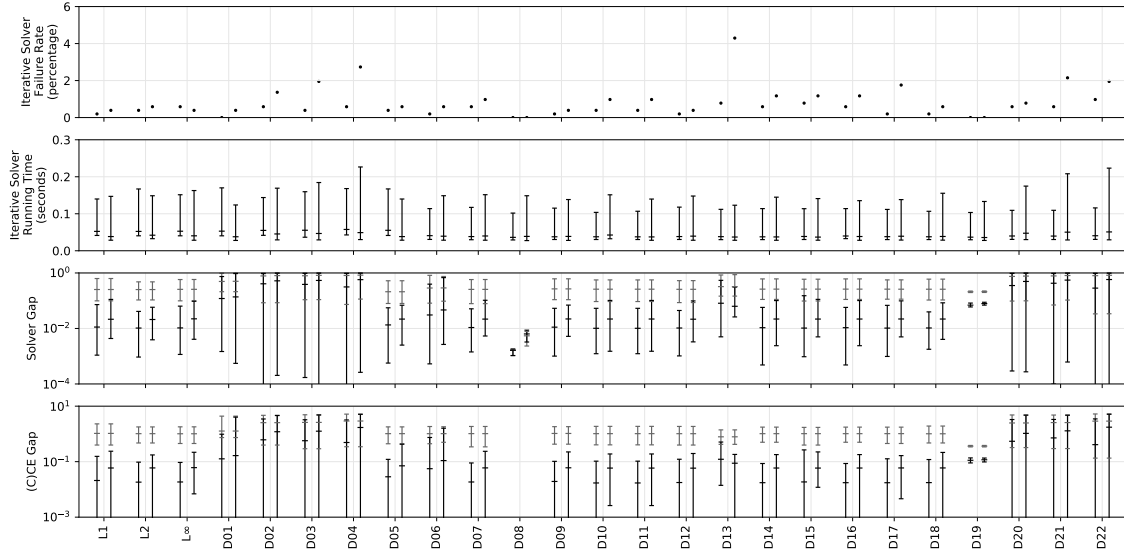


Figure 8.4: Worst, mean, and best performance of MECCE (left in pair) and MECE (right in pair) over 512 samples on the three equilibrium-invariant embeddings, and on a subset (Porter et al., 2008a) of transfer GAMUT (Nudelman et al., 2004) games (Appendix Table 8.2). The network was only trained on the “ L_2 invariant subspace” distribution of games. The gray range indicates the performance under a uniform distribution baseline.

Table 8.3: Scaling experiments showing the gaps of five NES models for larger games, with a uniform baseline, over 128 samples. The ECOS solver used to evaluate “solver gap” fails on large games.

Game	CCE Gap under uniform	CCE Gap under NES	Solver Gap under uniform	Solver Gap under NES	Success Fraction
4×4	1.1006	0.0274	0.3552	0.0120	100%
8×8	1.0043	0.0163	0.2513	0.0054	99%
16×16	0.8861	0.0173	0.2014	0.0034	98%
32×32	0.7376	0.0215	—	—	0%
64×64	0.5864	0.0288	—	—	0%

8.6 Applications

With Neural Equilibrium Solvers (NES) it is possible to quickly find approximate equilibrium solutions using a variety of selection criteria. This allows the application of solution concepts in areas that would otherwise be too time-expensive and are not as sensitive to approximations.

Inner Loop of MARL Algorithms

For algorithms (Greenwald and Hall, 2003; Hu and Wellman, 2003, 1998; Lanctot et al., 2017; Liu et al., 2022b; Marris et al., 2021b) where speed is critical, and the size of games is modest, but numerous, and approximations can be tolerated.

Warm-Start Iterative Solvers

Many iterative solvers start with a guess of the parameters and refine them over time to find an accurate solution (Duan et al., 2021). It is possible to use NES to warm-start iterative solver algorithms, potentially significantly improving convergence.

Polytope Approximation

The framework can be used to approximate the full space of solutions by finding extreme points of the convex polytope. Because of convexity, any convex mixture of these extreme points is also a valid solution. Two approaches could be used to find extreme points (i) using different welfare objectives, or (ii) using

Table 8.4: Generalization experiments showing how an 8×8 network generalizes to games with a different number of strategies, over 128 samples.

Game	CCE Gap under uniform	CCE Gap under NES	Solver Gap under uniform	Solver Gap under NES	Success Fraction
4×4	1.1006	4.4445	0.3552	0.1500	100%
8×8	1.0043	0.0163	0.2513	0.0054	99%
16×16	0.8861	0.0000	0.2014	0.1089	98%
32×32	0.7376	0.0000	—	—	0%
64×64	0.5864	0.0000	—	—	0%

different target joint objectives. For example, using pure joint targets:

$$W(a) = \begin{cases} 1, & \text{if } a = \hat{a} \\ 0, & \text{otherwise} \end{cases} \quad \hat{\sigma}(a) = \begin{cases} 1^-, & \text{if } a = \hat{a} \\ 0^+, & \text{otherwise} \end{cases}$$

These could be computed in a single batch, and would cover a reasonably large subset of full polytope of solutions (the latter approach is demonstrated in Figure 8.1). It would be easy to develop an algorithm that refines the targets at each step to gradually find all vertices of the polytope, if desired.

Differentiable Model and Mechanism Design

Mechanism design (MD) is a sub-field of economics often described as “inverse game theory”, where instead of studying the behavior of rational payoff maximizing agents on a given game, the task is designing a game so that rational payoff maximizing participants will exhibit behaviours *at equilibrium* that are deemed desirable. The field has a long history to which it is near impossible to do justice; see (Maskin, 2008) for a review. The work presented here could impact MD in two ways. First, by making it easy to compute equilibrium strategies, NES could widen the class of acceptable output games, relaxing the restrictive requirements (e.g. strategic dominance) often imposed of the output games out of concern more permissive solution concepts could be hard for participants to compute. Second, NES maps payoffs to joint strategies and is differentiable, one could imagine turning a mechanism design task to an optimization problem that could be solved using standard gradient descent (e.g. design a general-sum game where strategies at equilibrium maximize some non-linear function of welfare and entropy, with payoff lying in a useful convex and closed subset). A related idea is to find a game that produces a certain equilibrium (Ling et al., 2018). Given an equilibrium, a payoff could be trained through the differentiable model that results in the desired specific behaviour.

8.7 Discussion

The main limitation of this approach is that the activation space of the network is large, particularly with a large number of players and strategies which limits the size of games that can be tackled. Future work could look at restricted classes of games, such as polymatrix games (Deligkas et al., 2016, 2017), or graphical games (Jiang et al., 2011), which consider only local payoff structure and have much smaller payoff representations. This is a promising direction because NES otherwise has good scaling properties: (i) the dual variables are space-efficient, (ii) there are relatively few parameters, (iii) the number of parameters is independent of the number of strategies in the game, (iv) equivariance means each training sample is equivalent to training under all payoff permutations, and (v) there are promising zero-shot generalization results to larger games.

Solving for equilibria has the potential to promote increased cooperation in general-sum games, which could increase the welfare of all players. However, if a powerful and unethical actor had influence on the game being played, welfare gains of some equilibria could unfairly come at the expense of other players.

8.A Appendices

8.A.1 Approximate Target Maximum Welfare Minimum Relative Entropy Equilibria

This section derives Minimum Relative Entropy (RME) (also known as minimum KL divergence) $\sum_a \sigma(a) \ln \left(\frac{\sigma(a)}{\hat{\sigma}(a)} \right)$, where $\hat{\sigma}(a) > 0$ is a full-support joint such that, $\sum_a \hat{\sigma}(a) = 1$. This objective is similar to maximum entropy correlated equilibrium (MECE) (Ortiz et al., 2007), and the proofs here are similar to the framework set out there. A drawback of MECE is that it is not easy to determine the minimum ϵ_p permissible. If ϵ_p is chosen so that it does not permit a valid solution, then the parameters will diverge. This problem can be circumvented by optimizing the distance to a target $\hat{\epsilon}_p$. This target, $\min_{\epsilon_p} \rho \sum_p (\epsilon_p^+ - \epsilon_p) \ln \left(\frac{1}{\exp(1)} \frac{\epsilon_p^+ - \epsilon_p}{(\epsilon_p^+ - \hat{\epsilon}_p)} \right)$, is engineered to have a global minimum at $\epsilon_p = \hat{\epsilon}_p$, where $0 < \rho < \infty$ is a hyper-parameter used to control the balance between the distance to the target distribution and the distance to the target approximation parameter. And μ is for balancing the linear objective.

8.A.1.1 CEs

Theorem 8.A.1 (ϵ -MWMRE CE). *The $\hat{\epsilon}$ -MWMRE CE solution is equivalent to minimizing the loss:*

$$L^{CE} = \ln \left(\sum_{a \in \mathcal{A}} \hat{\sigma}(a) \exp \left(l(a)^{CE} \right) \right) + \sum_p \epsilon_p^+ \sum_{a'_p, a''_p} \alpha_p^{CE}(a'_p, a''_p) - \rho \sum_p \epsilon_p^{CE}$$

With logits defined as:

$$l(a)^{CE} = \mu \sum_a W(a) - \sum_{p, a'_p, a''_p} \alpha_p^{CE}(a'_p, a''_p) A_p^{CE}(a'_p, a''_p, a)$$

And primal variables defined:

$$\sigma(a)^{CE} = \frac{\hat{\sigma}(a) \exp \left(l(a)^{CE} \right)}{\sum_{a \in \mathcal{A}} \hat{\sigma}(a) \exp \left(l(a)^{CE} \right)} \quad \epsilon_p^{CE} = (\hat{\epsilon}_p - \epsilon_p^+) \exp \left(-\frac{1}{\rho} \sum_{a'_p, a''_p} \alpha_p^{CE}(a'_p, a''_p) \right) + \epsilon_p^+$$

Proof. Construct a Lagrangian, $\max_{\sigma, \epsilon} \min_{\alpha, \beta, \lambda} L_{\alpha, \beta, \lambda}^{\sigma, \epsilon_p}$, where the primal variables are being maximized and the dual variables are being minimized.

$$\begin{aligned} L_{\alpha, \beta, \lambda}^{\sigma, \epsilon_p} &= - \sum_a \sigma(a) \ln \left(\frac{\sigma(a)}{\hat{\sigma}(a)} \right) + \mu \sum_{a \in \mathcal{A}} W(a) \sigma(a) - \rho \sum_p (\epsilon_p^+ - \epsilon_p) \ln \left(\frac{1}{\exp(1)} \frac{\epsilon_p^+ - \epsilon_p}{(\epsilon_p^+ - \hat{\epsilon}_p)} \right) \\ &\quad + \sum_a \beta(a) \sigma(a) - \lambda \left(\sum_a \sigma(a) - 1 \right) - \sum_{p, a'_p, a''_p} \alpha_p(a'_p, a''_p) \left(\sum_a \sigma(a) A_p(a'_p, a''_p, a) - \epsilon_p \right) \\ &= \sum_a \sigma(a) \left(-\ln \left(\frac{\sigma(a)}{\hat{\sigma}(a)} \right) + \mu W(a) + \beta(a) - \lambda - \sum_{p, a'_p, a''_p} \alpha_p(a'_p, a''_p) A_p(a'_p, a''_p, a) \right) \\ &\quad + \sum_{p, a'_p, a''_p} \alpha_p(a'_p, a''_p) \epsilon_p + \lambda - \rho \sum_p (\epsilon_p^+ - \epsilon_p) \ln \left(\frac{1}{\exp(1)} \frac{\epsilon_p^+ - \epsilon_p}{(\epsilon_p^+ - \hat{\epsilon}_p)} \right) \end{aligned}$$

Taking the derivatives with respect to the joint distribution $\sigma(a)$, and setting to zero.

$$\frac{\partial L_{\alpha_p, \beta, \lambda}^{\sigma, \epsilon_p}}{\partial \sigma(a)} = -\ln \left(\frac{\sigma(a)}{\hat{\sigma}(a)} \right) - 1 + \mu W(a) + \beta(a) - \lambda - \sum_{p, a'_p, a_p} \alpha_p(a'_p, a_p) A_p(a'_p, a_p, a) = 0$$

$$\sigma^*(a) = \hat{\sigma}(a) \exp \left(-\lambda - 1 + \mu W(a) + \beta(a) - \sum_{p, a'_p, a_p} \alpha_p(a'_p, a_p) A_p(a'_p, a_p, a) \right)$$

Substituting back in:

$$L_{\alpha_p, \beta, \lambda}^{\epsilon_p} = \sum_a \sigma^*(a) + \sum_{p, a'_p, a_p} \alpha_p(a'_p, a_p) \epsilon_p + \lambda - \rho \sum_p (\epsilon_p^+ - \epsilon_p) \ln \left(\frac{1}{\exp(1)} \frac{\epsilon_p^+ - \epsilon_p}{(\epsilon_p^+ - \hat{\epsilon}_p)} \right)$$

Taking the derivative with respect to λ and setting to zero.

$$\frac{\partial L_{\alpha_p, \beta, \lambda}^{\epsilon_p}}{\partial \lambda} = -\sum_a \sigma^*(a) + 1 = 0$$

$$\exp(\lambda^* + 1) = \sum_a \hat{\sigma}(a) \exp \left(\mu W(a) + \beta(a) - \sum_{p, a'_p, a''_p} \alpha_p(a'_p, a''_p) A_p(a'_p, a''_p, a) \right)$$

Substituting back in:

$$L_{\alpha_p, \beta}^{\epsilon_p} = \ln \left(\sum_a \hat{\sigma}(a) \exp \left(\mu W(a) + \beta(a) - \sum_{p, a'_p, a''_p} \alpha_p(a'_p, a''_p) A_p(a'_p, a''_p, a) \right) \right)$$

$$+ \sum_{p, a'_p, a''_p} \alpha_p(a'_p, a''_p) \epsilon_p - \rho \sum_p (\epsilon_p^+ - \epsilon_p) \ln \left(\frac{1}{\exp(1)} \frac{\epsilon_p^+ - \epsilon_p}{(\epsilon_p^+ - \hat{\epsilon}_p)} \right)$$

Noting that the term is minimized when $\beta(a) = 0$, the $\hat{\sigma}(a)$ term can be lifted into the exponential.

$$L_{\alpha_p}^{\epsilon_p} = \ln \left(\sum_a \hat{\sigma}(a) \exp \left(\mu W(a) - \sum_{p, a'_p, a''_p} \alpha_p(a'_p, a''_p) A_p(a'_p, a''_p, a) \right) \right)$$

$$+ \sum_{p, a'_p, a''_p} \alpha_p(a'_p, a''_p) \epsilon_p - \rho \sum_p (\epsilon_p^+ - \epsilon_p) \ln \left(\frac{1}{\exp(1)} \frac{\epsilon_p^+ - \epsilon_p}{(\epsilon_p^+ - \hat{\epsilon}_p)} \right)$$

Taking the derivatives with respect to the approximation parameter ϵ_p , and setting to zero.

$$\frac{\partial L_{\alpha_p}^{\epsilon_p}}{\partial \epsilon_p} = \rho \ln \left(\frac{\epsilon_p^+ - \epsilon_p^*}{\epsilon_p^+ - \hat{\epsilon}_p} \right) + \sum_{a'_p, a_p} \alpha_p(a'_p, a_p) = 0 \implies \epsilon_p^* = (\hat{\epsilon}_p - \epsilon_p^+) \exp \left(-\frac{1}{\rho} \sum_{a'_p, a_p} \alpha_p(a'_p, a_p) \right) + \epsilon_p^+$$

Therefore:

$$\ln \left(\frac{1}{\exp(1)} \frac{\epsilon_p^+ - \epsilon_p^*}{(\epsilon_p^+ - \hat{\epsilon}_p)} \right) = \ln \left(\frac{1}{\exp(1)} \exp \left(-\frac{1}{\rho} \sum_{a'_p, a_p} \alpha_p(a'_p, a_p) \right) \right) = -\frac{1}{\rho} \sum_{a'_p, a_p} \alpha_p(a'_p, a_p) - 1$$

Substituting back in:

$$L_{\alpha_p} = \ln \left(\sum_a \hat{\sigma}(a) \exp \left(\mu W(a) - \sum_{p, a'_p, a''_p} \alpha_p(a'_p, a''_p) A_p(a'_p, a''_p, a) \right) \right) \\ + \sum_{p, a'_p, a''_p} \epsilon_p^+ \alpha_p(a'_p, a''_p) - \rho \sum_p (\hat{\epsilon}_p - \epsilon_p^+) \exp \left(-\frac{1}{\rho} \sum_{a'_p, a''_p} \alpha_p(a'_p, a''_p) \right)$$

□

8.A.1.2 CCEs

Theorem 8.A.2 (CCE). *The $\hat{\epsilon}$ -MWMRE CCE solution is equivalent to minimizing the loss:*

$$L^{CCE} = \ln \left(\sum_{a \in \mathcal{A}} \hat{\sigma}(a) \exp \left(l^{CCE}(a) \right) \right) + \sum_p \epsilon_p^+ \sum_{a'_p} \alpha_p^{CCE}(a'_p) - \rho \sum_p \epsilon_p^{CCE}$$

With logits defined as:

$$l^{CCE}(a) = \mu \sum_a W(a) - \sum_{p, a'_p} \alpha_p^{CCE}(a'_p) A_p(a'_p, a)$$

And primal variables defined:

$$\sigma^{CCE}(a) = \frac{\hat{\sigma}(a) \exp \left(l^{CCE}(a) \right)}{\sum_{a \in \mathcal{A}} \hat{\sigma}(a) \exp \left(l^{CCE}(a) \right)} \quad \epsilon_p^{CCE} = (\hat{\epsilon}_p - \epsilon_p^+) \exp \left(-\frac{1}{\rho} \sum_{a'_p} \alpha_p^{CCE}(a'_p) \right) + \epsilon_p^+$$

Proof. Similar proof to Theorem 8.A.1. □

8.A.2 Equivariant Pooling Functions

Functions which maintain equivariance over player and strategy permutations are useful building blocks for neural network architectures. These comprise of two components; (i) an equivariant pooling function ϕ , such as mean sum, min, or max, and (ii) the reduction dimensions.

An *equivariant pooling function* has three properties: (i) it collapses one or more of the dimensions of a tensor, (ii) the operation is invariant to the order of the elements in the collapsed dimensions, and (iii) the operation is equivariant to the order of the elements in the non-collapsed dimensions. For example, the reduction, $\sum_{a_2} G_1(a_1, a_2) = R_1(a_1)$, (i) reduces the dimensionality from $|\mathcal{A}_1| \times |\mathcal{A}_2|$ to $|\mathcal{A}_1|$, (ii) reordering the columns of $G_1(a_1, a_2)$ does not change the calculation, but (iii) reordering the rows of $G_1(a_1, a_2)$ results in an equivariant output, where $R_1(a_1)$ is reordered in the same way.

Such combinations of pooling functions and reduction dimensions can be combined to construct a network that is equivariant to strategy and player permutation. Consider several such pooling functions composed together (Figure 8.5).

8.A.3 Experiment Architecture and Hyper-Parameters

The architecture and hyper-parameters were chosen from a coarse sweep. The performance of architecture was not very sensitive to parameterization: similar settings will work well, or even better. Nevertheless the details of the exact architecture used in the experiments are provided.

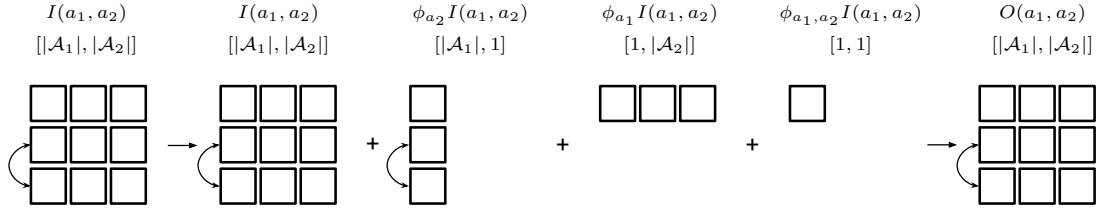


Figure 8.5: Equivariant pooling functions, mapping an input $I(a_1, a_2)$ to an output $O(a_1, a_2)$. Swapping the second and third row (for example) of the input results in the same swap in the outputs.

Architecture

All experiments use the same network architecture, with either CCE or CE dual parameterization, implemented in JAX (Bradbury et al., 2018) and Haiku (Hennigan et al., 2020). The network used pooling functions (Equations (8.12a)-(8.12d) and (8.12k)-(8.12l)) for the payoffs to payoffs layers, and used all the pooling functions for dual layers. For ϕ , both mean and max pooling together are used. The network consists of 5 payoffs to payoffs layers, each with 32 channels, a payoffs to duals layer with 64 channels and 2 duals to duals layers with 32 channels, which are denoted $[(32, 32, 32, 32, 32), 64, (32, 32)]$. The network has 79,905 parameters. All nonlinearities are ReLUs apart from the final layer which is a Softplus. Batch-Norm (Ioffe and Szegedy, 2015) is used between every layer with learned scale and variance correction. The network was initialized such that the variance of activations at every layer is unity. This was done empirically by passing a dummy batch of data through the network and calculating the variance.

Hyper-Parameters

The network was trained with a batch size of 4096, the Optax (Hessel et al., 2020) implementation of Adam (Kingma and Ba, 2014) (learning rate 4×10^{-4}) optimizer with adaptive gradient clipping (Brock et al., 2021) (clipping 10^{-3}). The experiments used a learning rate schedule with (iteration, factor) pairs of $[(1 \times 10^5, 1.0), (1 \times 10^6, 0.6), (4 \times 10^6, 0.3), (7 \times 10^6, 0.1), (1 \times 10^7, 0.06), (1 \times 10^8, 0.03)]$. Finally, a weight decay loss (learning rate 1×10^{-7}) is used for regularization.

8.A.3.1 Hardware

The network is trained on a 32 core TPU v3 (Jouppi et al., 2020), and evaluated on an 8 core TPU v2 (Jouppi et al., 2020). For intuition, the 8×8 network trains at around 400 batches per second (1,638,400 examples per second). Evaluation is even faster. Bigger games take longer, and scale approximately linearly with the number of joint actions in the game.

8.A.4 Relative Entropy and Welfare Objectives

Any solution can be realised by some relative entropy objective, since if a joint distribution $\sigma(a)$ is a (C)CE, then the solution with a relative entropy objective to $\hat{\sigma}(a) = \sigma(a)$ itself is optimised by $\sigma(a)$. One might imagine therefore that a relative entropy objective could be chosen to induce a maximum welfare solution, based on the payoffs. If possible, this would allow the MWMRE to be simplified. However, it is not straightforward to determine a priori which relative entropy objective(s) will lead to the maximum welfare. This means that relative entropy objectives are insufficient for finding Maximum Welfare solutions.

For example, consider a welfare $W(a) = \sum_p G_p(a)$. One might try to induce a welfare maximising solution by choosing a target joint $\hat{\sigma}(a) = \lim_{T \rightarrow \infty} \text{SoftMax}(TW(a))$, where T is the temperature parameter. Finding a CE that minimizes relative entropy to $\hat{\sigma}$ would place high mass on the highest welfare joint action, but is not equivalent to maximizing the linear objective $\sum_a \sigma(a)W(a)$, for either CCEs or CEs.

Consider game (a) in Table 8.5. This game consists of two games of chicken side-by-side, which are

Table 8.5: Payoffs for two games that show that maximum welfare (MW) cannot be discovered via an MW objective to the distribution given by a softmax of welfare. Both games are symmetric, payoffs are given for the row player

(a) CE MW Counterexample					(b) CCE MW Counterexample				
	1	2	3	4		1	2	3	4
1	-4	2, -2	-999	-999	1	2	0	0	0, 2
2	-2, 2	1	-999	-999	2	0	3	0, 3	-10, 7
3	-999	-999	-3	2, -2	3	0	3, 0	-10	-10, -6
4	-999	-999	-2, 2	1.1	4	2, 0	7, -10	-6, -10	0

mutually incompatible (i.e. the players must co-ordinate to play the same game of chicken to avoid a very large negative payoff for both). The softmax relative entropy objective will prefer the action pair (4, 4) over all others, as it gives slightly higher payoffs than (2, 2). Notice that a CE that recommends (4, 4) must also recommend the action pair (4, 3) some of the time in order to disincentivise the row player from deviating from action 4 to action 3. Similarly, a CE that recommends (2, 2) must also recommend (2, 1) some of the time to disincentivise the row player from deviating from action 2 to action 1.

Crucially, because (1, 1) has a lower payoff for the row player than (3, 3), in CEs consisting (2, 2) the mediator doesn't have to recommend (2, 1) as often as it has to recommend (4, 3) to form an effective disincentive. The result is that a CE that plays exclusively the joint actions (1, 2), (2, 1) and (2, 2) can achieve higher welfare than any that plays (4, 4), despite never playing the welfare maximising joint action.

Game (b) in Table 8.5 provides a counterexample for the CCE case. It works in a similar way to the CE counterexample: there are two high welfare joint strategies, (1, 1) and (2, 2). The latter has higher welfare, but if played too much a deviation to strategy 4 is incentivised. In the limit of T , the relative entropy objective selects whichever equilibrium has the highest probability of the maximum welfare joint.

To disincentivise the row player's deviation to strategy 4, the column player must either play only strategies 1 and 4, because the strategies 1 and 4 have the same payoff for the row player, or play strategy 3 sufficiently frequently that the benefit of deviating from strategy 2 to strategy 4 is nullified.

The first option gives rise to the maximum welfare CCE, which plays (1, 1) with probability 1. The second gives rise to the CCE that plays (2, 2) with the highest possible probability: 0.2. It plays (2, 3) and (3, 2) with probability 0.4 each. This is chosen by the relative entropy objective, but gives each player an average payoff of 1.8, which is equal to the payoff for deviating to action 4 in this equilibrium, but lower than the payoff of (1, 1).

Chapter 9

Game Theoretic Rating

Rating strategies in a game is an important area of research in game theory and artificial intelligence, and can be applied to any real-world competitive or cooperative setting. Traditionally, only transitive dependencies between strategies have been used to rate strategies (e.g. Elo (Elo, 1978)), however recent work has expanded ratings to utilize game theoretic solutions to better rate strategies in non-transitive games. This work generalizes these ideas and proposes novel algorithms suitable for n-player general-sum rating of strategies in normal-form games according to the payoff rating system. This enables well-established solution concepts, such as equilibria, to be used to rate strategies in games. These ratings summarize the complex strategic interactions which arise in multiagent training and real-world scenarios. The methods are empirically validated on real world normal-form game data (Premier League) and multiagent reinforcement learning agent evaluation.

9.1 Introduction

Traditionally, rating systems assume transitive dependencies of strategies in a game (such as Elo (Elo, 1978) and TrueSkill (Herbrich et al., 2007)). That is, there exists an unambiguous ordering of all strategies according to their relative strengths. This ignores all other interesting interactions between strategies, including cycles where strategy S beats P beats R beats S in the classic game of Rock, Paper, Scissors (Table 9.1). Many interesting games have this so-called “strategic” dimension (Czarnecki et al., 2020), or “gamescapes” (Balduzzi et al., 2018), that cannot be captured by pairwise transitivity constraints.

Game theoretic rating of strategies is an emerging area of study which seeks to overcome some of these drawbacks. These methods can be employed in normal-form games or in empirical games. Empirical games are normal-form games estimated from extensive-form games, where strategies are policies competing in a multiagent interaction (e.g. a simulation or a game) and the payoffs are approximate expected returns of the players employing these policies (Tuyls et al., 2020; Walsh et al., 2002; Wellman, 2006).

The Nash Average (NA) (Balduzzi et al., 2018) algorithm proposed a way of rating strategies in two-player, zero-sum, normal-form games. This approach is known as maximal lottery (Fishburn, 1984; Kreweras, 1965) in social choice theory, where it first arose, and is so fundamental it has been rediscovered across many fields (Brandt, 2017). In particular, NA proposed two applications of rating: agent-vs-agent interactions and agent-vs-task interactions. NA possesses several interesting properties: its ratings are invariant to strategy duplication, and it captures interesting non-transitive interactions between strategies. However, the technique is difficult to apply outside of two-player, zero-sum domains due to computational tractability and equilibrium selection difficulties. More recent work, α -Rank (Omidshafiei et al., 2019), sought to remedy this by introducing a novel computationally feasible solution concept based on the stationary distribution of a discrete-time evolutionary process. Its main advantages are its uniqueness and efficient computation in

n-player and general-sum games.

This work expands game theoretic rating techniques to established equilibrium concepts correlated equilibrium (CE) (Aumann, 1974), and coarse-correlated equilibrium (CCE) (Moulin and Vial, 1978). Section 9.2 defines a novel general rating definition: payoff rating, which is equivalent to NA if the game is two-player zero-sum. Payoff rating is the expected payoff under a joint distribution, conditioned on taking a certain strategy. The choice of joint distribution is what provides payoff ratings with its interesting properties. Section 9.3 suggests joint distributions to parameterize game theoretic rating algorithms. Section 9.5 tests these algorithms on instances of n-player, general-sum games using real-world data. Finally, Section 9.6 is a discussion of the connections of this work to other areas of machine learning and the relevance of the work to machine learning.

9.2 Game-Theoretic Rating

This chapter introduces a novel generalized rating for n-player, general-sum: the *payoff rating*. The definition functions for arbitrary joint strategy distributions, however the key idea of this work is to use equilibrium distributions, which have interesting game-theoretic properties.

9.2.1 Payoff Rating

The rating is defined in terms of the payoff, G_p , and the joint distribution players are assumed to be playing under, σ .

$$\begin{aligned} r_p^\sigma(a_p) &= \frac{\partial}{\partial \sigma(a_p)} \sum_{a \in \mathcal{A}} G_p(a) \sigma(a) = \frac{\partial}{\partial \sigma(a_p)} \sum_{a_{-p} \in \mathcal{A}_{-p}} G_p(a_p, a_{-p}) \sigma(a_{-p} | a_p) \sigma(a_p) \\ &= \sum_{a_{-p} \in \mathcal{A}_{-p}} G_p(a_p, a_{-p}) \sigma(a_{-p} | a_p) \end{aligned} \quad (9.1)$$

Theorem 9.2.1. (*Nash Average Equivalence*) When using an maximum entropy Nash equilibrium (MENE) for the joint strategy distribution in two-player, zero-sum games, payoff rating is equivalent to Nash Average (NA).

Proof. For NE, a player's strategies are independent from the other player's strategies, $\sigma(a_2 | a_1) = \sigma(a_2)$. Therefore $r_1^\sigma(a_1) = \sum_{a_2 \in \mathcal{A}_2} G_1(a_1, a_2) \sigma(a_2)$ and $r_2^\sigma(a_2) = \sum_{a_1 \in \mathcal{A}_1} G_2(a_1, a_2) \sigma(a_1)$, which is the definition of NA. \square

This definition has two interpretations: a) the change in the player's payoff under a joint strategy distribution, $\sum_{a \in \mathcal{A}} G_p(a) \sigma(a)$, with respect to the probability of selecting that strategy, $\sigma(a_p)$ b) the expected strategy payoff under a joint strategy distribution conditioned on that strategy. When defined, the payoff rating is bounded between the minimum and maximum values of a strategy's payoff, $\min_a G_p(a_p, a_{-p}) \leq r_p^\sigma(a_p) \leq \max_a G_p(a_p, a_{-p})$.

Note the mathematical edge case that strategies with zero marginal probability, $\sigma(a_p) = 0$, have undefined conditional probability, $\sigma(a_{-p} | a_p)$, and therefore have undefined payoff rating. Consider a symmetric two-player zero-sum transitive game where strategy S dominates A , and A dominates W . Many game theoretic distributions (including NE, CE and CCE) will place all probability mass on (S, S) , leaving strategies A and W with undefined rating. This may be unsatisfying for two reasons; firstly there could be further ordering between A and W such that $S > A > W$ is reflected in the ranking, and secondly, that all strategies should receive a rating value. It could be argued that if a strategy dominates all others then an ordering over the rest is redundant. However there are ways to achieve ordering; a) with approximate equilibria, certain joint strategies (such as $\epsilon^{\min+}$ -MECCE) are guaranteed to place at least some mass on all strategies, b) assign $r_p^\sigma(a) = \min_a G_p(a)$ for undefined values, and c) rate using a sub-game with dominating strategies

pruned.

9.2.2 Joint Strategy Distributions

Consider some joint distributions that the rating could be measured under. The most ubiquitous approach is the uniform distribution which is equivalent to calculating the mean payoff across all opponent strategies. As discussed previously, this approach does not consider any interesting dynamics of the game. It is, however, the distribution with maximum entropy and therefore makes the fewest assumptions (Jaynes, 1957).

In order to be more game theoretic, using distributions that are in certain types of *equilibrium* is beneficial. Firstly, consider the definitions of several equilibria (Equations (2.64), (2.54), and (2.59)). These equations are linear¹ inequality constraints between strategies, so already closely resemble a partial ordering. Rankings are nothing more than partial orderings between elements. Secondly, values of the payoff ratings depend entirely on the payoffs under distributions that all players are not incentivized to deviate from. Therefore this set of joint distributions are representative of ones which rational agents may employ in practice. In contrast, the uniform distribution is rarely within an equilibrium set. Therefore, this work argues, equilibrium distributions are a much more natural approach. Further mathematical justification is given in Section 9.A.3.

It is possible to mix the opinionated properties of an equilibrium with the zero-assumption properties of the uniform: there exists a principled continuum between the uniform distribution and an equilibrium distribution (Marris et al., 2021a) to achieve this balance. The uniform distribution is recovered when using a large enough approximation parameter $\epsilon_p \geq \epsilon_p^{\text{uni}}$. The value of ϵ_p^{uni} depends on the solution concept, and can be determined directly from a payoff.

Theorem 9.2.2 (Minimum Approximation Uniform). *The uniform distribution is an ϵ -equilibrium, when using approximation parameter $\epsilon_p \geq \epsilon_p^{\text{uni}}$.*

$$\text{WSCE / WSNE: } \epsilon_p^{\text{uni}} = \max_{a'_p, a''_p} \sum_{a_{-p} \in \mathcal{A}_{-p}} \frac{1}{|\mathcal{A}_{-p}|} A_p^{\text{WSCE}}[a'_p, a''_p, a_{-p}] \quad (9.2a)$$

$$\text{CE / NE: } \epsilon_p^{\text{uni}} = \max_{a'_p, a''_p} \sum_{a \in \mathcal{A}} \frac{1}{|\mathcal{A}|} A_p^{\text{CE}}[a'_p, a''_p, a] \quad (9.2b)$$

$$\text{CCE: } \epsilon_p^{\text{uni}} = \max_{a'_p} \sum_{a \in \mathcal{A}} \frac{1}{|\mathcal{A}|} A_p^{\text{CCE}}[a'_p, a] \quad (9.2c)$$

Proof. Note that the uniform distribution factorizes, therefore the distinction between NE and its CE-based definition can be ignored. Simply consider the maximum possible deviation possible for each player. \square

For completeness, there is a similar approximation parameter that permits any distribution to be in equilibrium. Due to convexity it is only necessary to consider the most extreme points: pure strategies. This is the same over all solution concepts.

Theorem 9.2.3 (Minimum Approximation All). *All distributions are ϵ -WSNE, ϵ -NE, ϵ -WSCE, ϵ -CE, and ϵ -CCE, when using approximation parameter $\epsilon_p \geq \epsilon_p^{\text{all}}$.*

$$\epsilon_p^{\text{all}} = \max_{a'_p, a''_p, a_{-p}} A_p^{\text{WSCE}}[a'_p, a''_p, a_{-p}] = \max_{a'_p, a''_p, a} A_p^{\text{CE}}[a'_p, a''_p, a] = \max_{a'_p, a} A_p^{\text{CCE}}[a'_p, a] \quad (9.3)$$

These theorems are utilized in Figure 9.2 to explore relationship between rating under a uniform distribution and a game theoretic rating by smoothly adjusting the approximation parameter.

¹ In joint distribution space.

9.2.3 Properties of Equilibria Ratings

Naively, one may want a rating strategy to differentiate the strategies it is rating. Game-theoretic rating does the opposite: it groups strategies into similar ratings that *should not* be differentiated, such as strategies that are in a cycle with one another (Tables 9.2a, and 9.2b). We call this phenomenon the *grouping property*. It is well-known that Nash equilibria have this property: players should be indifferent between strategies in the equilibrium support. Other properties, such as strategic dominance resulting in dominated ratings, consistent ratings over repeated strategies, and consistency between players in a symmetric game, can also be achieved when using the maximum entropy criterion (Section 9.A.3).

9.3 Rating Algorithms

A generalized payoff rating algorithm (Algorithm 9.1) is therefore parameterized over an equilibrium concept, and an equilibrium selection criterion. This section makes some recommendations on suitable parameterizations.

Algorithm 9.1 Generalized Payoff Rating

```

1:  $\sigma(a) \leftarrow \text{CONCEPTANDSELECTION}(G(a), \epsilon)$ 
2: for  $p \leftarrow 1 \dots n$  do
3:    $r_p^\sigma(a_p) \leftarrow \sum_{a_{-p} \in \mathcal{A}_{-p}} G_p(a_p, a_{-p}) \sigma(a_{-p} | a_p)$ 
4: return  $(r_1^\sigma(a_1), \dots, r_n^\sigma(a_n))$ 

```

9.3.1 $\epsilon^{\min+}$ -MECCE Payoff Rating

This work recommends using Coarse Correlated Equilibrium (CCE) as the joint strategy distribution, maximum entropy (ME) for the equilibrium selection function. Consider the solution when $\epsilon \rightarrow \epsilon^{\min}$ (or equivalently with a sufficiently small $\epsilon = \epsilon^{\min+}$), where $\epsilon^{\min} \leq 0$ is the minimum approximation parameter that permits a feasible solution (Section 9.A.2) (Marris et al., 2021b). The resulting rating is called $\epsilon^{\min+}$ -MECCE Payoff Rating.

Using CCEs as the solution concept has a number of advantages: a) full joint distributions allow cooperative as well as competitive games to be rated; factorizable distributions such as NE struggle with cooperative components of a game b) CCEs are more tractable to compute than CE and NEs, c) full-support CCEs only require a single variable per strategy to define², d) they are amenable to equilibrium selection because it permits a convex polytope of solutions, e) under a CCE, no player has incentive to deviate from the joint (possibly correlated) distribution to any of their own strategies unilaterally since it would not result in higher payoff, and f) the empirical joint strategy of no-regret algorithms in self-play converge to a CCE.

In combination with CCEs, ME with any $\epsilon > \epsilon^{\min}$ (Section 9.A.2) spreads at least some mass over all joint strategies (“full support” (Ortiz et al., 2007)) meaning that the conditional distribution, and hence the payoff rating, is always well defined. This equilibrium selection method is also invariant under affine transforms (Marris et al., 2021a) of the payoff, scales well to large numbers of players and strategies, and is principled in that it makes minimal assumptions about the distribution (Jaynes, 1957). Empirically, it groups strategies within strategic cycles with each other. Using a solution near ϵ^{\min} allows for a strong, high value equilibrium to be selected which is particularly important for coordination games.

9.3.2 $\frac{\epsilon}{\epsilon_{\text{uni}}}$ -MECCE Payoff Rating

A drawback of using $\epsilon = \epsilon^{\min+}$ is that sometimes (usually when strategies are strictly dominated by others) the distribution needs to be computed to a very high precision, otherwise numerical issues will complicate

²With the payoff tensor.

Table 9.1: Dwayne, Pen, Sword, Rock, Paper, Scissors (DPSRPS) symmetric, two-player, zero-sum game. Where Dwayne beats pen (dominance in arts), pen beats sword, sword beats Dwayne. When the first three strategies interact with the second three strategies they retain the usual properties of the RPS game resulting in those three quadrants having identical payoff. Note that the top left quadrant has a reversed cycle to the usual RPS game. Note the sub-games: DRPS and RPS.

	D	Pe	Sw	R	P	S
D	$\frac{1}{2}, \frac{1}{2}$	1, 0	0, 1	$\frac{1}{2}, \frac{1}{2}$	0, 1	1, 0
Pe	0, 1	$\frac{1}{2}, \frac{1}{2}$	1, 0	1, 0	$\frac{1}{2}, \frac{1}{2}$	0, 1
Sw	1, 0	0, 1	$\frac{1}{2}, \frac{1}{2}$	0, 1	1, 0	$\frac{1}{2}, \frac{1}{2}$
R	$\frac{1}{2}, \frac{1}{2}$	0, 1	1, 0	$\frac{1}{2}, \frac{1}{2}$	0, 1	1, 0
P	1, 0	$\frac{1}{2}, \frac{1}{2}$	0, 1	1, 0	$\frac{1}{2}, \frac{1}{2}$	0, 1
S	0, 1	1, 0	$\frac{1}{2}, \frac{1}{2}$	0, 1	1, 0	$\frac{1}{2}, \frac{1}{2}$

the calculation of the conditional distributions.

In order to mitigate this problem let us use an approximate equilibrium distribution which will spread more mass. It is advantageous to normalize the approximation parameter (Marris et al., 2021a), $\frac{\epsilon}{\epsilon_{\text{uni}}}$, where ϵ_{uni} is the minimum ϵ that permits the uniform distribution in the feasible set. When $\frac{\epsilon}{\epsilon_{\text{uni}}} = 1$ the uniform distribution is selected by ME, when $\frac{\epsilon}{\epsilon_{\text{uni}}} = 0$ the MECCE solution is recovered. For some games, it is possible to set $\frac{\epsilon}{\epsilon_{\text{uni}}} \leq 0$ to produce ratings with very robust distributions. This is similar in idea to the continuum of QREs (McKelvey and Palfrey, 1995). Figure 9.2 shows how ratings change with $\frac{\epsilon}{\epsilon_{\text{uni}}}$ for a two-player, zero-sum game.

9.4 Implementation Considerations

There are two additional considerations that may need to be handled when implementing game theoretic ratings algorithms: uncertain payoffs and repeated strategies.

9.4.1 Uncertainty in Payoffs

Often the outcome of a game is stochastic and it may not be able to query the exact expected return of a joint strategy. Instead, the expected return may have to be estimated through sampling each element of the payoff. Furthermore, there may be scenarios where elements of a payoff tensor are missing. There has been significant work on estimating solution concepts in uncertain or incomplete information settings (Du et al., 2021; Rashid et al., 2021; Rowland et al., 2019). The work presented here does not offer involved solutions for handling uncertain payoffs, but it will make one recommendation: when the payoff is uncertain, an appropriately large ϵ should be used. This is because small changes in payoff can result in large changes in the equilibrium set. Larger ϵ mitigates the size of those changes.

9.4.2 Repeated Strategy Problem

Consider the Rock, Paper, Scissors (RPS) game (Table 9.1). For RPS, each strategy is clearly distinct as they have different payoffs. Now let us consider a similar game Dwayne, Rock, Paper, Scissors (DRPS). In this case strategies D and R have identical payoffs, but does that mean they are repeated instances of the same strategy?

Strategies with the same payoffs are mathematically identical. The only situation where one may want to differentiate between strategies with identical payoffs is when one is in a sub-game regime: where only a subset of the strategies of a full game are known. This scenario is common in EGTA, and this problem arises in multiagent training algorithms like Double Oracle (DO) (McMahan et al., 2003) and Policy-Space Response Oracles (PSRO) (Lanctot et al., 2017). In this scenario strategies may become non-identical when additional opponent strategies are added to the sub-game. For example, consider the Dwayne, Pen, Sword, Rock, Paper, Scissors (DPWRPS) game (Table 9.1).

However, there is still additional information that can be attached to each strategy to aid differentiation.

For example it is common that strategies represent policies from an extensive form game. In this case one could differentiate strategies with the same payoffs in a sub-game by examining their policies. Identical policies imply identical payoffs, but identical payoffs do not imply identical policies.

A desirable property of rating algorithms is that they are invariant under strategy repeats. An algorithm with this property is particularly useful when rating sub-games where distinctly different strategies may appear to have the same payoffs in the sub-game. For example, if you were studying an RPS tournament and found that 50% of participants played rock, one may come away thinking that paper is the strongest strategy. However, while this may be the case in this particular tournament, it does not paint an accurate portrayal of the underlying RPS game, where each strategy is equally good and equally exploitable. Payoff ratings derived from NEs are invariant to strategy repeats (Balduzzi et al., 2018). (C)CEs and α -Rank are not automatically invariant to strategy repeats. Retaining this property for these other solution concepts is advantageous.

This could be simply achieved by eliminating repeated strategies from a game. When the payoffs are exactly known, this can be implemented by testing for equality between all elements of a slice of a payoff tensor and eliminating duplicate slices. When the payoffs are noisy estimates, *soft strategy elimination* may need to be used. After solving the remaining sub-game after elimination, the joint distribution for the non-eliminated game can be reconstructed by spreading any mass equally over repeated strategies.

9.5 Experiments

In order to build intuition and demonstrate the flexibility of the rating algorithms presented, this section shows ratings for several standard and real world data games. Experiments compare against uniform and α -Rank rating methods.

9.5.1 Standard Normal Form Games

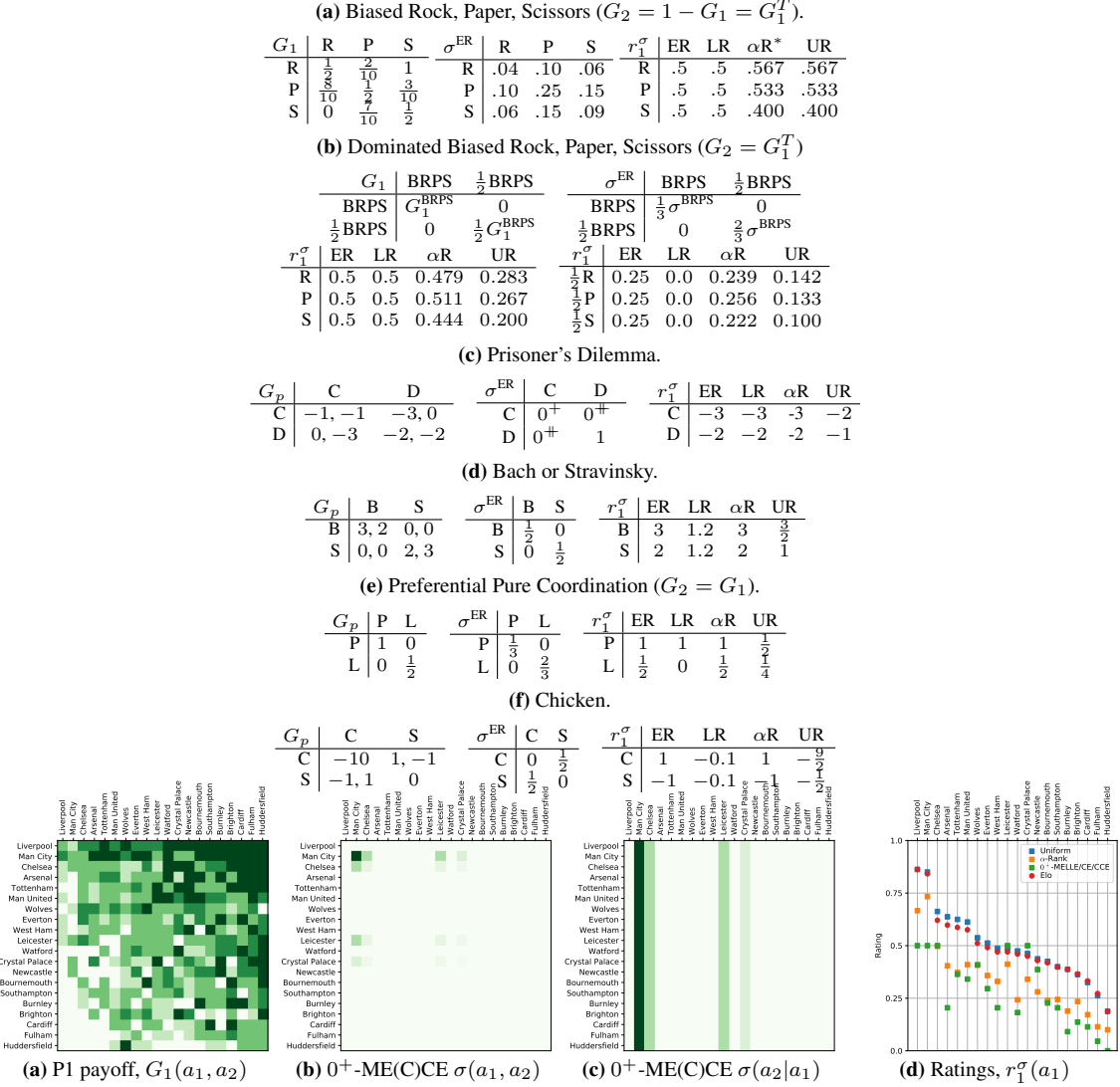
First let us consider the payoff, equilibrium and payoff ratings of some two-player normal-form games (Table 9.2). RPS has three strategies in a cycle, and therefore equilibrium ratings dictate that these strategies' ratings should be equal. This is true even if the cycles are biased (Table 9.2a) for the MECCE rating. In this case the probability mass is spread unevenly but the resulting payoff rating is equal for all the strategies in the cycle, grouping them together. α -Rank does not produce equal payoff ratings for BRPS, but *does* spread mass equally (not shown in the table). The uniform payoff incorrectly ungroups these strategies. It is also possible to construct a general-sum game with two sets of cycles where one cycle "dominates" the other (Table 9.2b). MECCE successfully groups the ratings of each of the cycles.

In prisoner's Dilemma (Table 9.2c) the dominant joint strategy receives all the mass (using slightly above ϵ^{\min} means (C, D) also gets some mass). Uniform rating results in the correct ordering in transitive games, but with less intuitive values. In Bach or Stravinsky (Table 9.2d) and the coordination game (Table 9.2e) MECCE is able to perfectly correlate actions to give better mutual payoffs. The limiting logit equilibrium (LLE) (McKelvey and Palfrey, 1995) is unsatisfactory on coordination games because factorizable distributions cannot exploit coordination opportunities. Interestingly, in the chicken game (Table 9.2f) C has better payoff rating than S because it is the strategy that gives the highest payoff when the other player swerves. This is an example of when the uniform rating gives a different ordering to the payoff rating.

9.5.2 Constant-Sum Premier League Ratings

Consider a two-player, zero-sum, symmetric win probability³ representation of clubs playing against each other in the 2018/2019 season of the premier league. Figure 9.1a shows win rates between clubs. For example, observe that Liverpool is very strong and beats every club apart from Man City. Although Man

³In practice, the Premier League is general-sum with 3 points for a win, 1 point each for a draw, and 0 points for a loss.

Table 9.2: Player 1's payoff ratings for standard games. ER: $\epsilon^{\min+}$ -MECCE. LR: LLE. αR : α -Rank. UR: uniform.**Figure 9.1:** Symmetric, two-player, zero-sum Premier League game where players pick between clubs as strategies. The clubs are ordered according to their average win probability. The conditional distribution is recovered from very small mass present in the joint distribution (MENE/CE/CCE shown). Log scales of the joint distribution can be found in Figure 9.2a.

City can beat Liverpool, Man City draws against four other clubs: Chelsea, Leicester, Crystal Palace and Newcastle, the latter three being middle of the table. Furthermore, Chelsea beats Crystal Palace, Crystal Palace beats Leicester, and Leicester beats Chelsea. Newcastle at best draws against Leicester. Therefore there is a weak cycle including Man City, Chelsea, Leicester and Crystal Palace, where Man City can threaten Liverpool. Because of this cycle, all these clubs have strategic relevance, and therefore should be rated equally highly by a game theoretic rating technique.

The 0^+ -ME(C)CE rating spreads the majority of the mass (Figure 9.1b) over clubs within this cycle. Note that for two-player, zero-sum games, exact CCE, CE and NE distributions are identical, and are therefore factorizable. Although it cannot be observed in the figure, there is nonzero support for every joint strategy and the conditional distribution (Figure 9.1c) reflects this. The payoff rating (Figure 9.1d) is identical for all strategies with nonzero NE support (see Section 9.A.3.1 for an explanation).

Furthermore, this work studies the $\frac{\epsilon}{\epsilon_{\text{uni}}}$ -MECCE mass (Figure 9.2a) and payoff (Figure 9.2b) ratings when varying $\frac{\epsilon_{\text{min}}}{\epsilon_{\text{uni}}} = 0^+ \leq \frac{\epsilon}{\epsilon_{\text{uni}}} \leq 1$. Some clubs (including Leicester, Crystal Palace and Newcastle which are in a weak cycle with Man City) improve their rankings as the joint distribution nears the 0^+ -MECCE solution. When using $\frac{\epsilon}{\epsilon_{\text{uni}}} = 1$ the uniform distribution is selected, and the payoff ratings are simply the mean performances against other clubs. When $\frac{\epsilon}{\epsilon_{\text{uni}}}$ is reduced towards zero, the rating becomes more game theoretic and the ratings change to reflect this. In particular, Leicester, Crystal Palace, and Newcastle all climb in rankings because they are in a weak cycle with Man City. Furthermore, the marginal masses, $\sigma(a_p)$, of many of the strategies tend to zero, with only a handful of clubs maintaining positive mass.

9.5.3 General-Sum Two-Player Premier League Ratings

Consider the general-sum points game, where each club plays each other twice and score 3 points for each win, 1 for each draw and 0 for each loss (Figure 9.3). This game is studied because it is not purely competitive: coordination exists because players would prefer mixing over two win-loss joint strategies (1.5 points each) rather than a single draw-draw joint strategy (1 point each).

Consider a particular NE equilibrium, the limiting logit equilibrium (LLE), which has which has a factorizable joint distribution (Figure 9.3d). A key drawback of factorizable distributions is that they cannot coordinate with other players, and therefore miss out on opportunities to increase the value of the game. Observe that LLE has the lowest value of the solution concepts tested (Figure 9.3l), even lower than uniform, while CCE has the highest, as the theory predicts. Therefore CCEs have the property that they can handle both competitive and cooperative rating.

9.5.4 Three-Player Premier League Ratings

Using the same data, this experiment introduces another player, the location player, which has two strategies: home or away. The location player gets a point if the club playing in the location it selects wins. The clubs get a point if they win irrespective of what the location player plays. This results in a three-player, general-sum game: location vs home club vs away club (Figure 9.4a).

This time, consider ratings using an approximate equilibrium with $\frac{\epsilon}{\epsilon_{\text{uni}}} = 0.1$. As expected, the location player's ratings (Figure 9.4b) favour the home strategy (reflecting the well-known home advantage phenomenon). The performance of clubs at home (Figure 9.4c) is high. The away performance (Figure 9.4d) of Leicester and Crystal Palace earn them top payoff ratings even though they are in the middle of the table.

9.5.5 Three-Player ATP Tennis Ratings

Using data from 2000-2020 ATP Tennis tournaments, this work studied the ratings of three competitors (Djokovic, Federer and Nadal) and the surfaces they play on (Hard, Clay and Grass), resulting in a three-player game: surface vs competitor vs competitor.

The surface a player competes on is a large factor of the game, however Elo, the traditional method of rating players, ignores this dependency. Out of the 144 games between these competitors in the dataset, 84 were on hard, 48 were on clay, and 12 were on grass surfaces. A transitive rating system, like Elo, is susceptible to this distribution and therefore favours players who have a strong hard surface game.

This experiment used an imagined game (Figure 9.5a) where the surface player gets the “win” if the match goes to tiebreak, shares half a point with the winning player if there is a single set difference between the winning and losing competitor, otherwise the winning competitor gets the win. Pairings between competitors and themselves are given zero points. Intuitively, if the match is sufficiently close, the surface “wins” because it is too difficult for the competitors to break.

The grass surface results in the closest matches and therefore provides the most points to the sur-

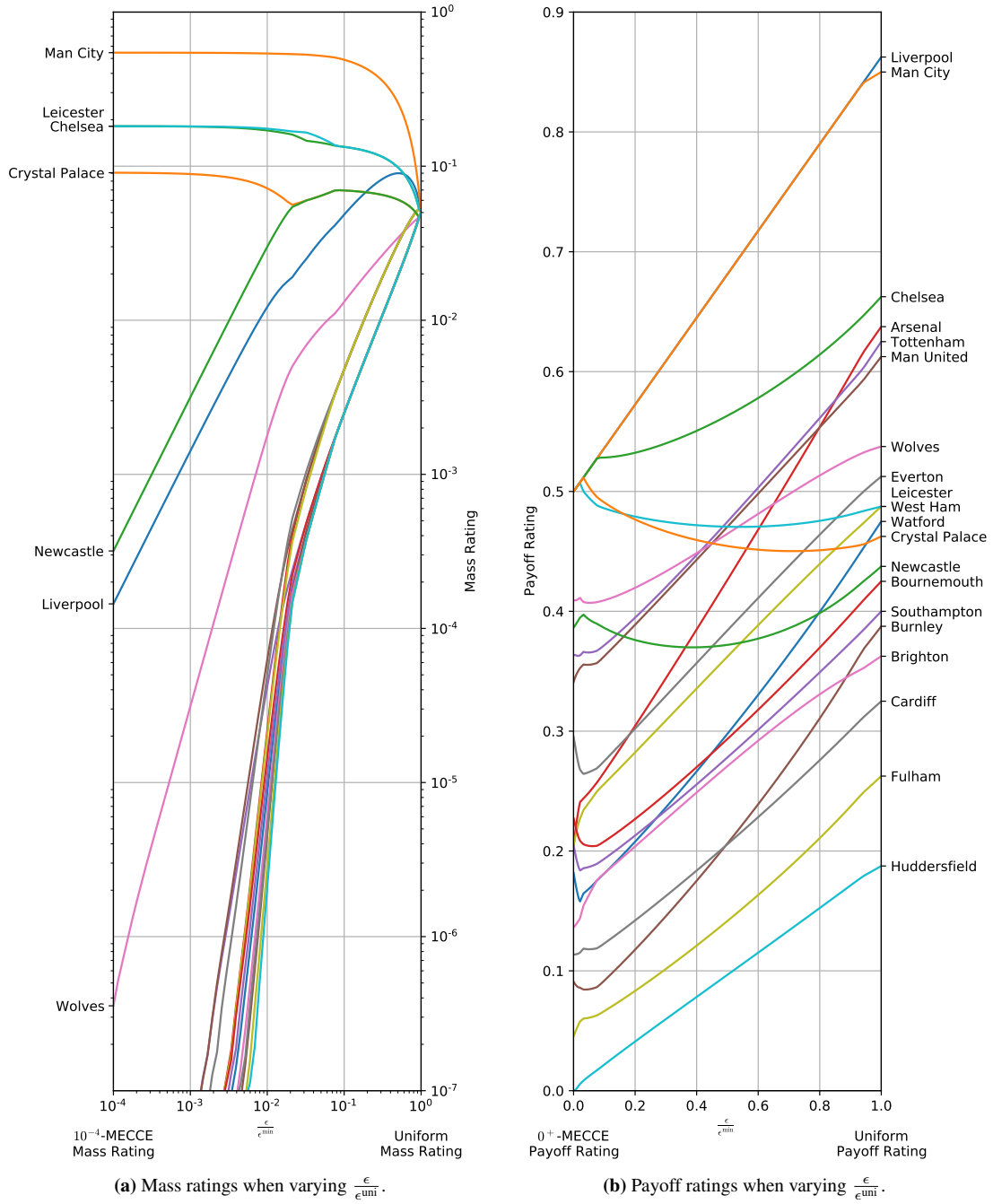


Figure 9.2: Shows how $\frac{\epsilon}{\epsilon_{uni}}$ -MECCE mass (marginals of the joint) and payoff rating varies over normalized ϵ for two-player, zero-sum Premier League ratings. When $\frac{\epsilon}{\epsilon_{uni}} = 0^+$, 0^+ -MECCE payoff rating is recovered, when $\frac{\epsilon}{\epsilon_{uni}} = 1$, uniform payoff rating is recovered. Because this game is two-player, zero-sum, the 0-MECCE is equal to 0-MENE, which is the definition of Nash Average. Lower values of $\frac{\epsilon}{\epsilon_{uni}}$ result in greater attention to cycles in the payoff table. Some clubs see their rankings improved as $\frac{\epsilon}{\epsilon_{uni}}$ is reduced; in particular Leicester, Crystal Palace and Newcastle which draw with Man City.

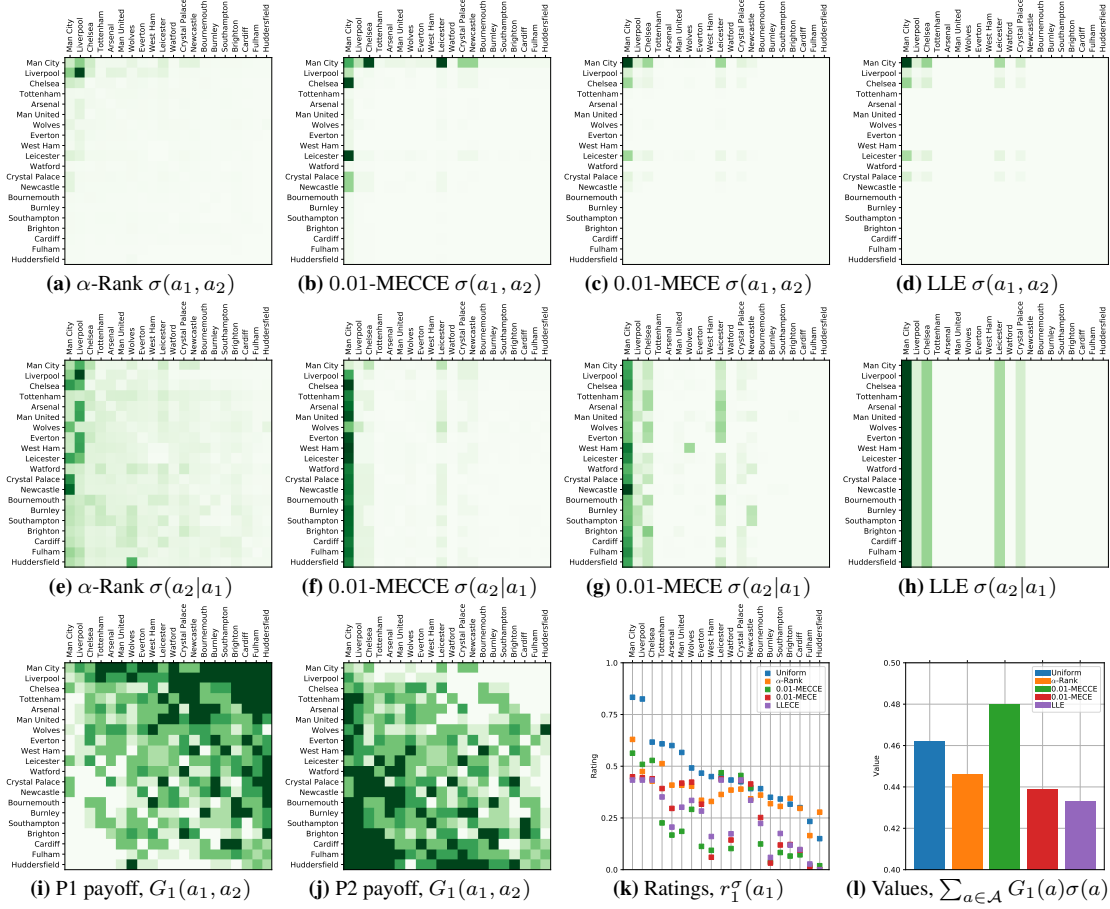


Figure 9.3: Symmetric, two-player, general-sum Premier League game where players pick between clubs as strategies. The clubs are ordered according to their average points (3 points for a win, 1 for a draw, 0 for a loss). The payoff ratings have been scaled by a factor of $\frac{1}{6}$. The joint and conditional distributions for the different ratings are shown for comparison.

face player, which receives the majority of the distribution mass. This means that the players are mainly evaluated according to their performance on grass, of which Djokovic is the strongest (Figure 9.5e).

9.5.6 Multiagent Learning Dynamics

It is well-known that the general multiagent reinforcement learning (MARL) problem is challenging due to nonstationarity (Hernandez-Leal et al., 2017), limited theoretical guarantees (Zhang et al., 2021), computational resource requirements and implementation challenges (Hernandez-Leal et al., 2019). Often there is a lack of “ground truth” to quantify the behavior of the algorithms; hence, the field has developed tools to analyze their dynamics qualitatively (Bloembergen et al., 2015). This subsection demonstrates the use of ratings that change over time as an analysis tool for MARL dynamics. In particular, ratings allow game-theoretic relative performance to be assessed over time. The experiment uses OpenSpiel (Lanctot et al., 2019) agents, with some additional custom agents and experimental setups.

In the experiments, agents in a population play against each other. Players play an n -player game; the population has n instantiations of each of 8 agent types (Random, Deep Q-networks (DQN) (Mnih et al., 2015), Neural Fictitious Self-Play (NFSP) (Heinrich and Silver, 2016), Advantage Actor-Critic (A2C) (Mnih et al., 2016), Online MCCFR (Lanctot et al., 2009), Tabular Actor Critic, QPG, and RPG (Srinivasan et al., 2018)), for a total of $8n$ agents. At the start of each episode, an agent type is

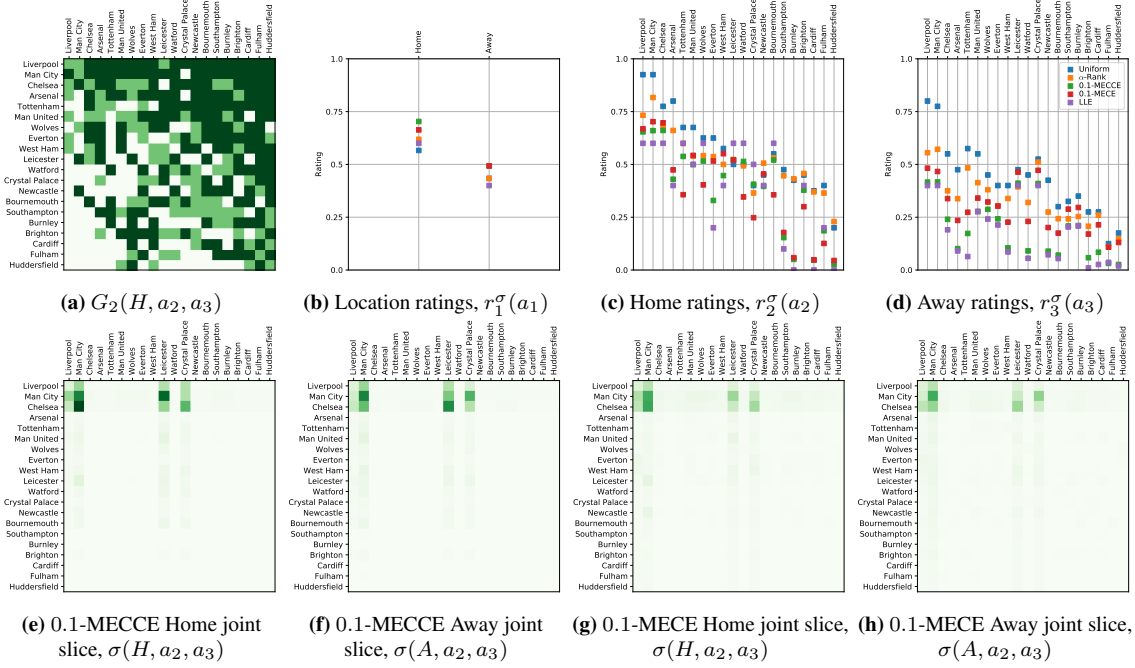


Figure 9.4: Three-Player Premier League game where the players are location vs home club vs away club. The resulting payoff tensor is of shape $3 \times 2 \times 20 \times 20$ and joint strategy distribution of shape $2 \times 20 \times 20$. All slices of the payoff are either arithmetic inverse or transpose of the data shown in Figure 9.4a. The clubs are ordered the same as in Figure 9.1.

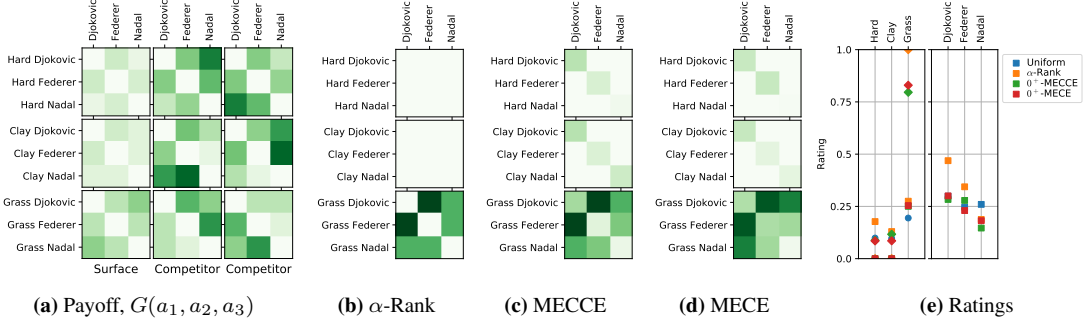
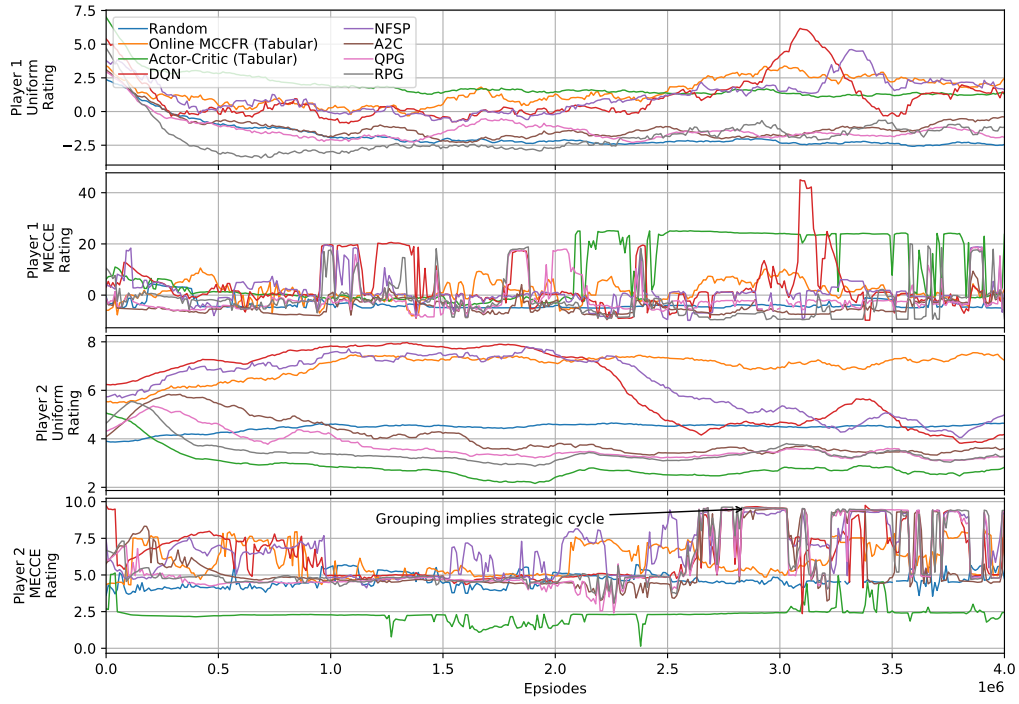


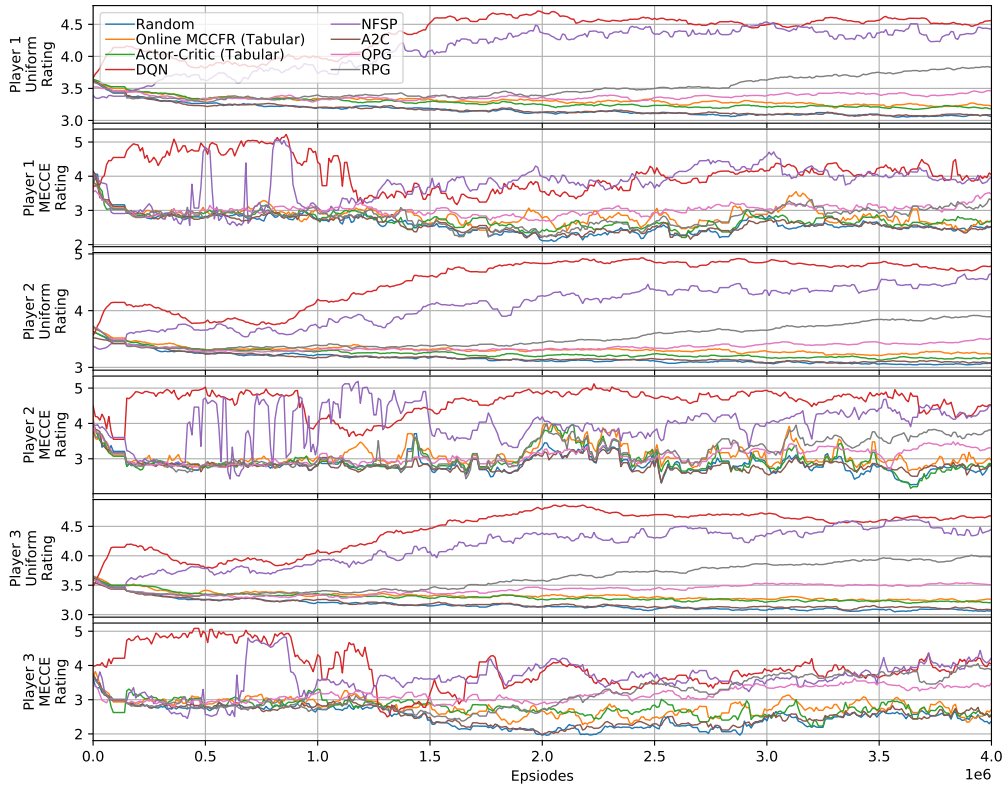
Figure 9.5: Three-player tennis games where the surface player chooses the court surface the game is played on. Grass has the closest games so is favoured most by the surface player. The distributions of all three solution concepts are shown.

uniformly sampled for each player, and observations are extended to include each player's agent type.

Two environments are used: the general-sum game of Sheriff (Farina et al., 2019c) (Figure 9.6a) and the three-player general-sum game of Goofspiel (Farina et al., 2019a) (Figure 9.6b). It is clear to observe the grouping properties of $\epsilon^{\min} +$ -MECCE payoff ratings: when the agents have learned policies that are in strategic cycles with one another their ratings are grouped, allowing for a fairer description of the relative strengths of each policy. This information cannot be learned from studying uniform ratings alone. $\epsilon^{\min} +$ -MECCE therefore provides a remarkable way of summarizing the complex interactions of strategies in n-player general-sum games into an interpretable scalar value for each strategy.



(a) Two-Player General-Sum Sheriff.



(b) Three-Player General-Sum Goofspiel.

Figure 9.6: Multiagent learning dynamics.

9.6 Discussion

Considering the payoff ratings when $\epsilon \rightarrow \epsilon^{\min}$ gives a mathematically sound way of ensuring all joint strategies have positive mass. Furthermore, using a normalised $\frac{\epsilon}{\epsilon_{\text{uni}}}$ allows a smooth parameterized transition from traditional uniform payoff rating to game theoretic payoff rating. Equilibria have a grouping property that ensures strategies that are in strategic cycles with one another have similar ratings. Maximum entropy is a principled way to select amongst equilibria and also gives consistent ratings across symmetric games and repeated strategies.

Formulating the environment as a player (Balduzzi et al., 2018) in an agent vs environment game is an interesting way of ensuring the distribution of tasks (strategies available to the environment player) does not bias the ratings of the agents training on those tasks. Sections 9.5.4 and 9.5.5 examined environment vs agent vs agent games demonstrating that these ideas can be extended to multiagent learning. Indeed a multiagent inspired path to developing increasingly intelligent agents has been proposed (Bansal et al., 2018; Leibo et al., 2019) based on the richness of such dynamics.

As well as evaluating agents (Section 9.5.6), payoff ratings could also be used as a fitness function to evaluate agents within a population to drive an evolutionary algorithm, for example like in population based training (PBT) (Jaderberg et al., 2019). α -PSRO (Muller et al., 2020) can be seen as optimizing agents for the α -Rank mass rating. Similarly, JPSRO (Marris et al., 2021b) can be seen as optimizing agents for the MGCE payoff rating.

An emerging line of research called *gamification* (Gemp et al., 2020) seeks to reinterpret existing problems as games, and apply game theory to improve upon the solutions. Problems with multiple objectives, constraints, competitiveness are potentially amenable to gamification. The ranking problem is suitable for this approach because it is defined in terms of a partial ordering (inequality constraints), can have multiple players, and is inherently competitive.

9.7 Conclusion

This work develops methods for generalising game-theoretic rating techniques to n-player, general-sum settings, using the novel payoff rating definition. This builds upon fundamental rating techniques developed in two-player, zero-sum evaluation. Some parameterizations of these algorithms were suggested. Experiments rated real-world games to demonstrate the ratings' flexibility and ability to summarize complex strategic interactions. Finally, this work demonstrates the power of this rating as a MARL evaluation technique.

Table 9.3: Summary of parameterizations of algorithms available under this general scheme. Of course many other combinations are possible.

Rating	Joint	Selection	Approximation	Algorithm
PR	NE	ME	$\epsilon = 0$	Nash Average (Balduzzi et al., 2018)
MR	α -Rank	N/A	α	α -Rank (Omidshafiei et al., 2019)
PR	CCE	ME	$\epsilon = \epsilon^{\min+}$	$\epsilon^{\min+}$ -MECCE
PR	CCE	ME	$\epsilon^{\min+} \leq \epsilon \leq \epsilon^{\text{uni}}$	$\frac{\epsilon}{\epsilon^{\text{uni}}}$ -MECCE
PR	CE	ME	$\epsilon = \epsilon^{\min+}$	$\epsilon^{\min+}$ -MECE
PR	CE	ME	$\epsilon^{\min+} \leq \epsilon \leq \epsilon^{\text{uni}}$	$\frac{\epsilon}{\epsilon^{\text{uni}}}$ -MECE
PR	NE	LLE	$\lambda = \infty$	∞ -LLE
PR	NE	LLE	λ	λ -LLE

9.A Appendices

9.A.1 Algorithms

Table 9.3 summarises parameterizations of prior art algorithms and the new parameterizations suggested in this chapter. α -Rank uses the marginals of the joint distribution to rate strategies, a technique this work refers to as *mass rating* (MR).

9.A.2 Full Support Conditions

Approximate equilibria are useful because they permit full-support solutions. Full-support solutions produce well-defined payoff ratings for all strategies.

Theorem 9.A.1 (Approximate Full-Support Existence). *When $\epsilon > 0$ there will always exist a full-support, ϵ -NE, ϵ -CE and ϵ -CCE.*

Therefore, the notation $\epsilon = 0^+$ is used when finding a full-support solution close to the equilibrium. Some games have full-support solutions even for negative values of approximation parameter, ϵ .

Remark 9.A.2 (Negative Approximate Full-Support Existence). *When $\epsilon > \epsilon^{\min}$ there will always exist a full-support, ϵ -NE, ϵ -CE and ϵ -CCE.*

Furthermore, the maximum Shannon entropy is guaranteed to select such a full-support solution if one exists.

Theorem 9.A.3 (ϵ -ME Full-Support Solution). *Using an $\epsilon > \epsilon^{\min}$, ϵ -MECE and ϵ -MECCE will select full-support approximate equilibria (Ortiz et al., 2007).*

Other selection methods, like linear objectives and maximum Gini do not have this property because they have finite gradient when $\sigma(a) = 0$, and therefore may not leave the boundary of the probability simplex.

9.A.3 Justification and Intuition

A payoff can be arbitrarily mapped to a scalar for each strategy in a game to achieve a rating. For a rating definition to be compelling one must motivate why it is more interesting than other mappings. The main text makes some intuitive arguments about why game theoretic equilibrium methods are appropriate rating algorithms. Namely, that ratings are defined under joint distributions in equilibrium, so no player has incentive to unilaterally deviate from them. This is in contrast to the uniform distribution which is rarely an equilibrium.

This section makes mathematical arguments to justify and build intuition behind the equilibrium concepts and the ratings they define. To do this, several properties are explored for each equilibrium concept. The first such property is *grouping*, a game theoretic property that enforces strategies that are strategic cycle

with one another should get equal or similar ratings. The second property, *dominance*, tests whether strategic dominance implies an ordering in the ratings. Thirdly, *consistency*, is checked in games with repeated strategies or are part of a symmetric game.

9.A.3.1 NE

Grouping

A curious property of the NE payoff rating (Nash Average (Balduzzi et al., 2018)) is that all strategies with positive support have equal rating⁴. This property is called the *grouping property*, where strategies in strategic cycles together are grouped with similar ratings despite perhaps having very different payoffs.

Theorem 9.A.4 (ϵ -NE Grouping). *Strategies for 0-NE payoff ratings with positive support have equal payoff rating. Strategies for ϵ -NE payoff ratings with positive support have ratings bounded by:*

$$|r_p^\sigma(a'_p) - r_p^\sigma(a_p)| \leq \max \left[\frac{\epsilon_p}{\sigma(a_p)}, \frac{\epsilon_p}{\sigma(a'_p)} \right] \quad (9.4)$$

Proof. Consider the ϵ -NE definition (Equation 2.64) between strategies a_p and a'_p . First expand the definition of the deviation gain and observe that the definitions of the NE payoff ratings appear directly in the constraints.

$$\begin{aligned} \sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_p) \sigma(a_{-p}) G_p(a'_p, a_{-p}) &\leq \sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_p) \sigma(a_{-p}) G_p(a_p, a_{-p}) + \epsilon_p \\ \sigma(a_p) \sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_{-p} | a'_p) G_p(a'_p, a_{-p}) &\leq \sigma(a_p) \sum_{a_{-p} \in \mathcal{A}_{-p}} \sigma(a_{-p} | a_p) G_p(a_p, a_{-p}) + \epsilon_p \\ \sigma(a_p) r_p^\sigma(a'_p) &\leq \sigma(a_p) r_p^\sigma(a_p) + \epsilon_p \end{aligned} \quad (9.5)$$

Therefore a strategy, a_p , has an ϵ -approximate better rating than another, a'_p , if there is negative incentive to deviate from strategy a_p to a'_p under the NE distribution (assuming $\sigma(a_p) > 0$).

$$r_p^\sigma(a'_p) \leq r_p^\sigma(a_p) + \frac{\epsilon_p}{\sigma(a_p)} \quad (9.6)$$

The opposite equation also applies, and if there is support $\sigma(a'_p) > 0$.

$$r_p^\sigma(a_p) \leq r_p^\sigma(a'_p) + \frac{\epsilon_p}{\sigma(a'_p)} \quad (9.7)$$

Therefore, if both a_p and a'_p have support, then Equation (9.6) and Equation (9.7) can be combined:

$$\begin{aligned} r_p^\sigma(a_p) - \frac{\epsilon_p}{\sigma(a'_p)} &\leq r_p^\sigma(a'_p) \leq r_p^\sigma(a_p) + \frac{\epsilon_p}{\sigma(a_p)} \\ -\frac{\epsilon_p}{\sigma(a'_p)} &\leq r_p^\sigma(a'_p) - r_p^\sigma(a_p) \leq \frac{\epsilon_p}{\sigma(a_p)} \end{aligned}$$

Therefore bounds can be derived. When $\epsilon = 0$, $\sigma(a_p) > 0$, $\sigma(a'_p) > 0$, the payoff ratings will be equal: $r_p^\sigma(a_p) = r_p^\sigma(a'_p)$.

$$|r_p^\sigma(a'_p) - r_p^\sigma(a_p)| \leq \max \left[\frac{\epsilon_p}{\sigma(a_p)}, \frac{\epsilon_p}{\sigma(a'_p)} \right]$$

⁴See Figure 9.1d for an example.

□

This result is unsurprising for those familiar with NE: if there is a benefit to deviating strategies the opponents will adjust their distribution to compensate.

Dominance

It is also possible to make arguments around strategy dominance and their resulting payoff rating.

Theorem 9.A.5 (NE Weak Dominance). *If strategy a_p weakly dominates a'_p : $G_p(a_p, a_{-p}) \geq G_p(a'_p, a_{-p}) \forall a_{-p} \in \mathcal{A}_{-p}$, then $r_p^\sigma(a_p) \geq r_p^\sigma(a'_p)$. ϵ -NEs are weakly dominated up to a constant $\frac{\epsilon_p}{\sigma(a_p)}$.*

Proof. This property immediately follows from Equation (9.6). □

It is also possible to prove that strategies that have zero support have payoff rating no better than those with support.

Theorem 9.A.6 (NE Zero Support Bound). *If strategy a_p has zero support, it has a NE payoff rating no greater than a strategy a'_p with support. This is true for ϵ -NEs up to a constant $\frac{\epsilon_p}{\sigma(a_p)}$.*

Proof. By examining Equation (9.5), note that if $\sigma(a_p) = 0$, there are no constraints that $r_p^\sigma(a_p)$ is greater than any other strategy's payoff rating. □

Consistency

When using the maximum entropy (ME) as an equilibrium selection criterion one obtains important consistency properties. Note that these properties are not generally true, even for unique or convex selection criteria.

Theorem 9.A.7 (Repeated Strategies). *When using ϵ -MENE, repeated strategies have equal payoff rating (Balduzzi et al., 2018).*

In games that are symmetric across all n players such as the 7-player meta-game explored in Anthony et al. (2020a), it is desired that each player's ratings is the same. Without careful equilibrium selection, it is possible to be left with at least 2 and possibly n distinct ratings. Maximum entropy selection criterion can avoid this.

Theorem 9.A.8 (Symmetric Games). *When using ϵ -MENE, where all players share the same value approximation parameter $\epsilon_p = \epsilon$, players in symmetric games have equal sets of payoff rating, $r_1^\sigma(a_1) = \dots = r_n^\sigma(a_n)$.*

9.A.3.2 CE

Consistency

Similar consistency proofs for CEs follow the same arguments as their NE counterparts.

Theorem 9.A.9 (Repeated Strategies). *When using ϵ -MECE, repeated strategies have equal payoff rating.*

Theorem 9.A.10 (Symmetric Games). *When using ϵ -MECE, where all players share the same value approximation parameter $\epsilon_p = \epsilon$, players in symmetric games have equal sets of payoff rating, $r_1^\sigma(a_1) = \dots = r_n^\sigma(a_n)$.*

9.A.3.3 CCE

Consistency

Similar consistency proofs for CCEs follow the same arguments as their NE counterparts.

Theorem 9.A.11 (Repeated Strategies). *When using ϵ -MECCE, repeated strategies have equal payoff rating.*

Theorem 9.A.12 (Symmetric Games). *When using ϵ -MECCE, where all players share the same value approximation parameter $\epsilon_p = \epsilon$, players in symmetric games have equal sets of payoff rating, $r_1^\sigma(a_1) = \dots = r_n^\sigma(a_n)$.*

Chapter 10

General Discussion and Conclusion

The goal of this thesis was to build foundations for calculating equilibria in many-player, mixed-motive, complex games, at scale. Contributions of the work include; defining a metric space over normal-form games, defining equilibrium-invariant embeddings, building intuition of game theory in 2×2 games, visualizing complex extensive-form games, selecting equilibria, efficient equilibrium representations, developing algorithms that compute equilibria in extensive-form games, computing equilibria quickly and approximately using neural networks, and rating strategies in games.

10.1 Introduction

This thesis was motivated by the need to develop multiagent learning algorithms that are both principled and scalable. Principled algorithms are those that are based on sound game-theoretic principles, and that can be proven to converge to equilibria. Scalable algorithms are those that can be applied to games with a large number of players, that are not limited to zero-sum games, and that can be used to solve complex games. To achieve these goals, this thesis makes a deliberate choice to ignore the *metric problem*. The metric problem is the uncertainty about how to measure progress in many-player mixed-motive games, a problem that has plagued and distracted the field for many years. Instead, this thesis focuses on building the fundamental groundwork required for scalable algorithm-building in many-player mixed-motive settings. To this end, this work selects reasonable mediated equilibrium solution concepts (correlated equilibrium and coarse correlated equilibrium), and invents algorithms that compute these equilibria at scales not previously attempted.

10.2 Discussion

From a scholarship perspective, this thesis provides a valuable summary of different equilibrium solution concepts (Section 2.2.2), including Nash equilibrium, correlated equilibrium, coarse correlated equilibrium and their well-supported variants. The connections between the distinct, but related, solution concepts are shown clearly. Additionally, each concept can be defined in several equivalent ways, which this work enumerates. Furthermore, the relations between operators on games (best-response, player subsuming, and strategy mixing) and the definitions of equilibria are made clear. These operators often form components of game-theoretic algorithms. It is hoped that using these operators as building blocks, and clearly demonstrating their connections to equilibria, illuminates the guarantees of many existing multiagent algorithms, justifies the new algorithms introduced in this thesis, and emboldens readers to develop their own.

10.2.1 Equilibrium-Invariant Embeddings

In Chapter 3, this thesis builds on two solution concepts for training in n -player general-sum games: correlated equilibrium and coarse correlated equilibrium. Equilibrium-invariant and equilibrium-symmetric

transforms are used to uncover an equilibrium-invariant embedding in normal-form games. This embedding is special because it has fewer degrees of freedom, but still covers all possible strategically interesting games. This work defines an algorithm for sampling uniformly over this embedding: a useful distribution for testing and training game-theoretic algorithms. Furthermore, it illuminates the dimensions of the payoff definition that influence changes in the equilibria, and similarly illuminates the dimensions where changes do not change the equilibria. Additionally this chapter defines a distance metric between games.

10.2.2 2×2 games

2×2 games are studied extensively by economists and game theorists and are used to predict and explain a wide variety of interactions (Gauthier, 1986; Kelley et al., 2002; Ostrom et al., 1994; Sugden, 1986; Wilkin-son, 1984). This thesis explored two 2×2 game embeddings: the equilibrium-invariant embedding and the best-response-invariant embedding (and their symmetries). The 2×2 equilibrium-invariant-embedding remarkably only requires two variables to parameterize every embedding, which can be visualized easily in two-dimensions. A number of properties can be read from this spatial representation, including zero-sum-invariance, common-payoff-invariance, symmetries, equilibrium support, and best-response dynamics. The 2×2 best-response-invariant embedding contains a set of 15 fundamental games. This set has been proposed before (Borm, 1987) (although it is not established), but this work provides more clarity on its importance and provides distance metrics, names and develops deep intuition for all games in this set. This work could be the clearest explanation of the space of 2×2 games and will be valuable to all game theory practitioners.

10.2.3 Visualizing Large Games

Large many-player mixed-motive normal-form games and complex extensive-form games have historically been considered intractable to visualize. While techniques such as PCA and t-SNE exist to help visualize complex datasets in machine learning, no such tools exist for game theory. In Chapter 5, this thesis proposes techniques for producing such visualizations of complex games. This could be the first work to do so at scale and it is hoped that game theory practitioners will find these visualizations useful in their own analysis.

10.2.4 Equilibrium Selection and Computation

This thesis champions the property that equilibria should be selected consistently and uniquely. In order to use equilibria as building blocks for multiagent algorithms it is important that each game maps to a single solution. Chapter 6 exploits the convexity of (C)CEs to build on previous work around maximum entropy computation and proposes an approximation of this criterion: maximum Gini. Such a criterion can be calculated as a quadratic program and has similar efficient parameterization. This work also highlights the importance of negative approximation parameters, ϵ_p : equilibria that are interior to the equilibrium constraints. Until now, these have not been acknowledged by the literature. Negative approximation parameters are not possible for Nash equilibria in nontrivial games, which may explain why the literature is unfamiliar with them.

10.2.5 JPSRO

Chapter 7 introduces a novel algorithm that converges to normal-form (C)CEs in extensive-form games with many-players and mixed-motives. This algorithm is theoretically capable of finding (C)CEs in any complete information extensive-form game. Extensive-form is the most general game formulation. Complete information is a mild condition which states that all players have known strategies and utilities. This algorithm can be scaled with deep learning and reinforcement learning. It is an extension to the popular algorithm framework, PSRO, used to solve Go, Chess, and StarCraft.

10.2.6 Neural Equilibrium Solver

A component of many multiagent algorithms is finding equilibria in normal-form games. Often this involves deploying a solver that relies on iterative updates to optimize a nonlinear objective function. This could fail or take a non-deterministic time to return. Chapter 8 introduces the Neural Equilibrium Solver, a feedforward neural network that maps any payoff of a specific shape to an equilibrium. It can be trained unsupervised and utilizes a number of techniques explored throughout the thesis including, equilibrium-invariance, symmetries, sensible training distributions, and efficient equilibrium parameterizations. The network can produce equilibria in batches, very quickly. It is hoped this work will enable more scalable principled multiagent algorithms to be developed.

10.2.7 Game-Theoretic Rating

Rating and ranking all types of entities is a common problem within and outside of game theory. Often such problems can be interpreted as a game: the strategies are the entities that require ranking, the payoffs are the scores between entities, and the players can be a variety of perhaps abstract groups such as agents or tasks. Many naive approaches such as averaging over scores are defective in certain properties (Elo, 1978). Chapter 9 explores ideas for leveraging established solution concepts to rate strategies within games. This could help identify cycles within systems and rate the parts more fairly.

10.3 Conclusion

In conclusion, the thesis has achieved its goals. It has built a coherent philosophy for approaching multiagent research: ignore the “metric problem”, and focus on the “algorithm problem” and “complexity problem”. Concretely, solution concepts should be chosen for mathematical convenience (such as convexity), equilibrium selection should be unique, game-theoretic algorithms should converge, and primitives and sub-problems in the algorithms should be amenable to the latest deep learning and reinforcement learning techniques for scale.

Games with more than two players or non-zero-sum incentives are notoriously difficult to study. This thesis has introduced techniques that work for many-player and mixed-motive games, making such games less intimidating and more amenable to future research. Specifically, this work has defined an equilibrium-inspired metric-space and embedding over normal-form games, and introduced a variety of equilibrium computing algorithms for normal-form and extensive-form games. In terms of analysis this work has also thoroughly explained the space of 2×2 games, invented visualizations for large games, and provided an algorithm for ratings strategies.

The experiments in this thesis focused on games, as they are colloquially defined, for convenience. However, the algorithms developed within this thesis are applicable much more broadly. Real-world problems in economics, sociology, politics, defence, and artificial intelligence can be defined as games. And, indeed, these games often have multiple participants and their interactions are non-zero-sum. It is hoped that the work in this thesis will be applicable to the many diverse fields that study interactions between competing entities.

Bibliography

- TPU system architecture. <https://cloud.google.com/tpu/docs/system-architecture-tpu-vm>. Accessed: 2023-04-16.
- M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from tensorflow.org.
- I. Adler, C. Daskalakis, and C. H. Papadimitriou. A note on strictly competitive games. In S. Leonardi, editor, *Internet and Network Economics*, pages 471–474, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-10841-9.
- A. F. Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.
- A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.
- T. Anthony, T. Eccles, A. Tacchetti, J. Kramár, I. Gemp, T. C. Hudson, N. Porcel, M. Lanctot, J. Pérolat, R. Everett, et al. Learning to play no-press Diplomacy with best response policy iteration. *arXiv preprint arXiv:2006.04635*, 2020a.
- T. Anthony, T. Eccles, A. Tacchetti, J. Kramár, I. Gemp, T. C. Hudson, N. Porcel, M. Lanctot, J. Pérolat, R. Everett, R. Werpachowski, S. Singh, T. Graepel, and Y. Bachrach. Learning to play no-press diplomacy with best response policy iteration, 2020b.
- M. ApS. *MOSEK*, 2019. URL <http://docs.mosek.com>.
- R. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67–96, 1974.
- D. Avis, G. D. Rosenberg, R. Savani, and B. Von Stengel. Enumeration of Nash equilibria for two-player games. *Economic theory*, 42(1):9–37, 2010.
- Y. Bai, C. Jin, and T. Yu. Near-optimal reinforcement learning with self-play. *CoRR*, abs/2006.12007, 2020. URL <https://arxiv.org/abs/2006.12007>.
- B. Baker, I. Kanitscheider, T. M. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch. Emergent tool use from multi-agent autocurricula. *CoRR*, abs/1909.07528, 2019. URL <http://arxiv.org/abs/1909.07528>.

- D. Balduzzi, K. Tuyls, J. Perolat, and T. Graepel. Re-evaluating evaluation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS*, pages 3272–3283, Red Hook, NY, USA, 2018. Curran Associates Inc.
- T. Bansal, J. Pachocki, S. Sidor, I. Sutskever, and I. Mordatch. Emergent complexity via multi-agent competition. In *Proceedings of the Sixth International Conference on Learning Representations*, 2018.
- A. G. Barto, S. J. Bradtke, and S. P. Singh. Learning to act using real-time dynamic programming. *Artif. Intell.*, 72(1–2):81–138, Jan. 1995. ISSN 0004-3702.
- R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957a.
- R. Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, 6(5):679–684, 1957b.
- A. Ben-Tal, L. Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009. ISBN 9781400831050.
- C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. de Oliveira Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang. Dota 2 with large scale deep reinforcement learning. *CoRR*, abs/1912.06680, 2019. URL <http://arxiv.org/abs/1912.06680>.
- A. Bhattacharya. *The Man from the Future: The Visionary Life of John von Neumann*. WW Norton, 2022. ISBN 9781324003991.
- C. Bicchieri. *The Grammar of Society: The Nature and Dynamics of Social Norms*. Cambridge University Press, 2005. doi: 10.1017/CBO9780511616037.
- K. Binmore and E. Binmore. *Game Theory and the Social Contract: Just playing*. Economic Learning and Social Evolution Series. MIT Press, 1994. ISBN 9780262024440.
- C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.
- D. Bloembergen, K. Tuyls, D. Hennes, and M. Kaisers. Evolutionary dynamics of multi-agent learning: A survey. *Journal of Artificial Intelligence Research*, 53:659–697, 2015.
- M. Böörs, T. Wängberg, T. Everitt, and M. Hutter. Classification by decomposition: a novel approach to classification of symmetric 2x2 games. *Theory and Decision*, 93(3):463–508, Oct 2022. ISSN 1573-7187. doi: 10.1007/s11238-021-09850-z. URL <https://doi.org/10.1007/s11238-021-09850-z>.
- P. Borm. A classification of 2x2 bimatrix games. *Cahiers du Centre d’Études de Recherche Opérationnelle*, 29(1-2):69–84, 1987. ISSN 0008-9737. Pagination: 16.
- J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL <http://github.com/google/jax>.
- S. J. Brams. *Introduction*, page 1–18. Cambridge University Press, 1993. doi: 10.1017/CBO9780511558979.001.

- F. Brandt. Fishburn's Maximal Lotteries. *Workshop on Decision Making and Contest Theory*, 1 2017.
- S. Brânzei, N. Devanur, and Y. Rabani. Proportional dynamics in exchange economies. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 180–201, 2021.
- L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.
- M. Breton, G. Zaccour, and M. Zahaf. A game-theoretic formulation of joint implementation of environmental projects. *European Journal of Operational Research*, 168(1):221–239, 2006.
- A. Brock, S. De, S. L. Smith, and K. Simonyan. High-performance large-scale image recognition without normalization. *CoRR*, abs/2102.06171, 2021. URL <https://arxiv.org/abs/2102.06171>.
- G. W. Brown. Iterative solutions of games by fictitious play. 1951.
- N. Brown and T. Sandholm. Superhuman AI for multiplayer poker. *Science*, 365(6456):885–890, 2019. ISSN 0036-8075. doi: 10.1126/science.aay2400.
- B. Bruns and C. Kimmich. Archetypal games generate diverse models of power, conflict, and cooperation. *Ecology and Society*, 26, 2021.
- B. R. Bruns. Names for games: Locating 2×2 games. *Games*, 6(4):495–520, 2015. ISSN 2073-4336. doi: 10.3390/g6040495. URL <https://www.mdpi.com/2073-4336/6/4/495>.
- J. M. Buchanan. *The Samaritan's Dilemma*, pages 71–86. Russell Sage Foundation, 1975. ISBN 9780871546593. URL <http://www.jstor.org/stable/10.7758/9781610446792.10>.
- R. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal of Scientific Computing*, 16:1190–1208, Sept. 1995. ISSN 1064-8275.
- M. Campbell, A. Hoane, and F. hsiung Hsu. Deep blue. *Artificial Intelligence*, 134(1):57–83, 2002. ISSN 0004-3702. doi: [https://doi.org/10.1016/S0004-3702\(01\)00129-1](https://doi.org/10.1016/S0004-3702(01)00129-1). URL <https://www.sciencedirect.com/science/article/pii/S0004370201001291>.
- J. Casti. *Five Golden Rules: Great Theories of 20th-Century Mathematics—and Why They Matter*. Wiley, 1996. ISBN 9780471002611.
- A. Celli, A. Marchesi, T. Bianchi, and N. Gatti. Learning to correlate in multi-player general-sum sequential games, 2019.
- A. Celli, A. Marchesi, G. Farina, and N. Gatti. No-regret learning dynamics for extensive-form correlated equilibrium, 2020.
- N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- T. Chen, X. Chen, W. Chen, H. Heaton, J. Liu, Z. Wang, and W. Yin. Learning to optimize: A primer and a benchmark, 2021. URL <https://arxiv.org/abs/2103.12828>.
- X. Chen and X. Deng. Settling the complexity of two-player Nash equilibrium. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 261–272, 2006. doi: 10.1109/FOCS.2006.69.
- X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM (JACM)*, 56(3):1–57, 2009.

- K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014. URL <http://arxiv.org/abs/1409.1259>.
- C. Cortes and V. Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.
- W. M. Czarnecki, G. Gidel, B. Tracey, K. Tuyls, S. Omidshafiei, D. Balduzzi, and M. Jaderberg. Real world games look like spinning tops. *Advances in Neural Information Processing Systems*, 33, 2020.
- A. Czumaj, M. Fasoulakis, and M. Jurdziński. Approximate well-supported Nash equilibria in symmetric bimatrix games, 2014. URL <https://arxiv.org/abs/1407.3004>.
- C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, page 71–78, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595931341. doi: 10.1145/1132516.1132527. URL <https://doi.org/10.1145/1132516.1132527>.
- C. Daskalakis, P. Goldberg, and C. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM J. Comput.*, 39:195–259, 02 2009.
- A. Deligkas, J. Fearnley, T. P. Igwe, and R. Savani. An empirical study on computing equilibria in polymatrix games. *arXiv preprint arXiv:1602.06865*, 2016.
- A. Deligkas, J. Fearnley, R. Savani, and P. Spirakis. Computing approximate Nash equilibria in polymatrix games. *Algorithmica*, 77(2):487–514, 2017.
- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.
- S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.
- A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference (ECC)*, pages 3071–3076, 2013.
- Y. Du, X. Yan, X. Chen, J. Wang, and H. Zhang. Estimating α -Rank from a few entries with low rank matrix completion. In *International Conference on Machine Learning*, pages 2870–2879. PMLR, 2021.
- Z. Duan, D. Zhang, W. Huang, Y. Du, J. Wang, Y. Yang, and X. Deng. Towards the PAC learnability of Nash equilibrium. *CoRR*, abs/2108.07472, 2021. URL <https://arxiv.org/abs/2108.07472>.
- J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.
- M. Dudik and G. Gordon. A sampling-based approach to computing equilibria in succinct extensive-form games. 05 2012.
- B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. Working Paper 11765, National Bureau of Economic Research, November 2005. URL <http://www.nber.org/papers/w11765>.

- A. E. Elo. *The rating of chess players, past and present*. Arco Pub., New York, 1978. ISBN 0668047216 9780668047210.
- L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, and K. Kavukcuoglu. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures, 2018.
- FAIR, A. Bakhtin, N. Brown, E. Dinan, G. Farina, C. Flaherty, D. Fried, A. Goff, J. Gray, H. Hu, A. P. Jacob, M. Komeili, K. Konath, M. Kwon, A. Lerer, M. Lewis, A. H. Miller, S. Mitts, A. Renduchintala, S. Roller, D. Rowe, W. Shi, J. Spisak, A. Wei, D. Wu, H. Zhang, and M. Zijlstra. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624): 1067–1074, 2022. doi: 10.1126/science.ade9097. URL <https://www.science.org/doi/abs/10.1126/science.ade9097>.
- H. Fargier, P. Jourdan, and R. Sabbadin. A path-following polynomial equations systems approach for computing nash equilibria. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 418–426, 2022.
- G. Farina, T. Bianchi, and T. Sandholm. Coarse correlation in extensive-form games, 2019a.
- G. Farina, T. Bianchi, and T. Sandholm. Coarse correlation in extensive-form games, 2019b.
- G. Farina, C. K. Ling, F. Fang, and T. Sandholm. Correlation in extensive-form games: Saddle-point formulation and benchmarks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019c.
- X. Feng, O. Slumbers, Z. Wan, B. Liu, S. McAleer, Y. Wen, J. Wang, and Y. Yang. Neural auto-curricula in two-player zero-sum games. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 3504–3517. Curran Associates, Inc., 2021. URL <https://proceedings.neurips.cc/paper/2021/file/1cd73be1e256a7405516501e94e892ac-Paper.pdf>.
- P. C. Fishburn. Probabilistic social choice based on simple voting comparisons. *The Review of Economic Studies*, 51(4):683–692, 1984. ISSN 00346527, 1467937X. URL <http://www.jstor.org/stable/2297786>.
- P. C. Fishburn and D. M. Kilgour. Binary 2×2 games. *Theory and Decision*, 29(3):165–182, Nov 1990. ISSN 1573-7187. doi: 10.1007/BF00126800. URL <https://doi.org/10.1007/BF00126800>.
- D. P. Foster and R. V. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21:40–55, 1997.
- N. M. Fraser and D. M. Kilgour. Non-strict ordinal 2×2 games: A comprehensive computer-assisted analysis of the 726 possibilities. *Theory and Decision*, 20(2):99–121, Mar 1986. ISSN 1573-7187. doi: 10.1007/BF00135087. URL <https://doi.org/10.1007/BF00135087>.
- D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991.
- P. Gagniuc. *Markov Chains: From Theory to Implementation and Experimentation*. John Wiley & Sons, 2017. ISBN 9781119387596.
- C. F. Gauss. *Geodäsie. Fortsetzung von Band 4*, volume 9. Göttingen: Königlich-Gesellschaft der Wissenschaften, 1903.

- D. Gauthier. *Morals by Agreement*. Oxford, GB: Oxford University Press, 1986.
- I. M. Gemp, B. McWilliams, C. Vernade, and T. Graepel. Eigengame: PCA as a Nash equilibrium. *CoRR*, abs/2010.00554, 2020. URL <https://arxiv.org/abs/2010.00554>.
- I. M. Gemp, R. Savani, M. Lanctot, Y. Bachrach, T. W. Anthony, R. Everett, A. Tacchetti, T. Eccles, and J. Kramár. Sample-based approximation of nash in large many-player games via gradient descent. *CoRR*, abs/2106.01285, 2021. URL <https://arxiv.org/abs/2106.01285>.
- F. Germano. On some geometry and equivalence classes of normal form games. *International Journal of Game Theory*, 34(4):561–581, Nov 2006. ISSN 1432-1270. doi: 10.1007/s00182-006-0033-6. URL <https://doi.org/10.1007/s00182-006-0033-6>.
- F. Gers and J. Schmidhuber. Recurrent nets that time and count. volume 3, pages 189 – 194 vol.3, 02 2000. ISBN 0-7695-0619-4. doi: 10.1109/IJCNN.2000.861302.
- S. Gerschgorin. *Über die Abgrenzung der Eigenwerte einer Matrix*. 1931.
- H. Gintis. *The Bounds of Reason: Game Theory and the Unification of the Behavioral Sciences - Revised Edition*. Princeton University Press, 2014. ISBN 9780691160849.
- X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh and M. Titterton, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL <https://proceedings.mlr.press/v9/glorot10a.html>.
- D. Goforth and D. Robinson. *Dynamic Periodic Table of the 2×2 Games: User’s Reference and Manual*. 01 2005.
- P. W. Goldberg and C. H. Papadimitriou. Reducibility among equilibrium problems. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC ’06, page 61–70, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595931341. doi: 10.1145/1132516.1132526. URL <https://doi.org/10.1145/1132516.1132526>.
- P. W. Goldberg, C. H. Papadimitriou, and R. Savani. The complexity of the homotopy method, equilibrium selection, and lemke-howson solutions. *ACM Transactions on Economics and Computation (TEAC)*, 1(2):1–25, 2013.
- I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL <https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf>.
- Google. Cloud tensor processing units (TPUs). <https://cloud.google.com/tpu/docs/tpus>, 2023. Accessed: 2023-04-16.
- T. Graepel. Automatic curricula in deep multi-agent reinforcement learning. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS ’20, page 2, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450375184.

- J. Gray, A. Lerer, A. Bakhtin, and N. Brown. Human-level performance in no-pressure diplomacy via equilibrium search, 2020.
- A. Greenwald and K. Hall. Correlated-Q learning. pages 242–249, 2003.
- J. Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957.
- R. J. Harris. A geometric classification system for 2 x 2 interval-symmetric games. *Behavioral science*, 14(2):138, Mar 01 1969. URL <https://www.proquest.com/scholarly-journals/geometric-classification-system-2-x-interval/docview/1301271952/se-2>. Last updated - 2013-02-24.
- J. Harsanyi and R. Selten. *A General Theory of Equilibrium Selection in Games*, volume 1. The MIT Press, 1 edition, 1988.
- S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.
- J. S. Hartford, J. R. Wright, and K. Leyton-Brown. Deep learning for predicting human strategic behavior. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL <https://proceedings.neurips.cc/paper/2016/file/7eb3c8be3d411e8ebfab08eba5f49632-Paper.pdf>.
- J. Havrda, F. Charvat, and J. Havrda. Quantification method of classification processes: Concept of structural α -entropy. *Kybernetika*, 1967.
- K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL <http://arxiv.org/abs/1512.03385>.
- H. Heaton, D. McKenzie, Q. Li, S. W. Fung, S. J. Osher, and W. Yin. Learn to predict equilibria via fixed point networks. *CoRR*, abs/2106.00906, 2021. URL <https://arxiv.org/abs/2106.00906>.
- F. Heider and M. Simmel. An experimental study of apparent behavior. *The American Journal of Psychology*, pages 243–259, 1944.
- J. Heinrich and D. Silver. Deep reinforcement learning from self-play in imperfect-information games. *CoRR*, abs/1603.01121, 2016.
- J. Heinrich, M. Lanctot, and D. Silver. Fictitious self-play in extensive-form games. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, 2015.
- T. Hennigan, T. Cai, T. Norman, and I. Babuschkin. Haiku: Sonnet for JAX, 2020. URL <http://github.com/deepmind/dm-haiku>.
- R. Herbrich, T. Minka, and T. Graepel. TrueSkill™: A bayesian skill rating system. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 569–576. MIT Press, 2007.
- P. Hernandez-Leal, M. Kaisers, T. Baarslag, and E. M. de Cote. A survey of learning in multiagent environments: Dealing with non-stationarity, 2017.

- P. Hernandez-Leal, B. Kartal, and M. E. Taylor. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems volume*, 33:750–797, 2019.
- M. Hessel, H. Soyer, L. Espeholt, W. Czarnecki, S. Schmitt, and H. van Hasselt. Multi-task deep reinforcement learning with popart, 2018. URL <https://arxiv.org/abs/1809.04474>.
- M. Hessel, D. Budden, F. Viola, M. Rosca, E. Sezener, and T. Hennigan. Optax: composable gradient transformation and optimisation, in JAX!, 2020. URL <http://github.com/deepmind/optax>.
- G. Hinton and S. Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15:833–840, 2003. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.7959&rep=rep1&type=pdf>.
- S. Hochreiter. Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München, 1991.
- S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. *Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies*, pages 237–243. 2001. doi: 10.1109/9780470544037.ch14.
- B. Hoehn, F. Southey, R. C. Holte, and V. Bulitko. Effective short-term opponent exploitation in simplified poker. 2005.
- H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936. ISSN 00063444. URL <http://www.jstor.org/stable/2333955>.
- R. Howard. *Dynamic Probabilistic Systems: Markov models*. Critical Episodes in American Politics. Wiley, 1971. ISBN 9780471416654.
- R. A. Howard. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA, 1960.
- J. Hu and M. Wellman. Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4:1039–1069, 01 2003. doi: 10.1162/1532443041827880.
- J. Hu and M. P. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML ’98, page 242–250, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1558605568.
- A. F. Huertas-Rosero. A cartography for 2x2 symmetric games. *CoRR*, cs.GT/0312005, 2003. URL <http://arxiv.org/abs/cs/0312005>.
- S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL <http://arxiv.org/abs/1502.03167>.
- M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *CoRR*, abs/1611.05397, 2016. URL <http://arxiv.org/abs/1611.05397>.
- M. Jaderberg, W. Czarnecki, I. Dunning, L. Marris, G. Lever, A. Castañeda, C. Beattie, N. Rabinowitz, A. Morcos, A. Ruderman, N. Sonnerat, T. Green, L. Deason, J. Leibo, D. Silver, D. Hassabis, K. Kavukcuoglu, and T. Graepel. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 05 2019.

- E. Janovskaja. Equilibrium points in polymatrix games. *Lithuanian Mathematical Journal*, 8(2):381–384, 1968.
- E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, May 1957.
- A. X. Jiang, K. Leyton-Brown, and N. A. Bhat. Action-graph games. *Games and Economic Behavior*, 71(1):141–173, 2011. ISSN 0899-8256. doi: <https://doi.org/10.1016/j.geb.2010.10.012>. Special Issue In Honor of John Nash.
- C. Jin, Q. Liu, Y. Wang, and T. Yu. V-learning - A simple, efficient, decentralized algorithm for multiagent RL. *CoRR*, abs/2110.14555, 2021. URL <https://arxiv.org/abs/2110.14555>.
- N. Joulami, D. Yoon, G. Kurian, S. Li, N. Patil, J. Laudon, C. Young, and D. Patterson. A domain-specific supercomputer for training deep neural networks. *Communications of the ACM*, 63:67–78, 06 2020. doi: 10.1145/3360307.
- L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *CoRR*, cs.AI/9605103, 1996a. URL <https://arxiv.org/abs/cs/9605103>.
- L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996b.
- M. Kaur and G. Buttar. A brief review on different measures of entropy. 08 2019.
- H. Kelley, J. Holmes, N. Kerr, H. Reis, C. Rusbult, and P. Lange. An atlas of interpersonal situations. 01 2002. doi: 10.1017/CBO9780511499845.
- H. J. Kelley. Gradient theory of optimal flight paths. *Ars Journal*, 30(10):947–954, 1960.
- J. G. Kemeny, J. L. Snell, et al. *Finite Markov Chains*, volume 356. van Nostrand Princeton, NJ, 1960.
- D. M. Kilgour and N. M. Fraser. A taxonomy of all ordinal 2×2 games. *Theory and Decision*, 24:99–117, 1988.
- D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL <https://arxiv.org/abs/1412.6980>.
- V. Konda and J. Tsitsiklis. Actor-critic algorithms. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.
- G. Kreweras. Aggregation of preference orderings. *Mathematics and Social Sciences I: Proceedings of the seminars of Menthon-Saint-Bernard, France (1–27 July 1960) and of Gössing, Austria (3–27 July 1962)*, pages 73–79, 1965.
- H. W. Kuhn. A simplified two-person poker. *Contributions to the Theory of Games*, 1:97–103, 1950.
- H. W. Kuhn and A. Tucker. Extensive games and the problem and information. *Contributions to the Theory of Games, II, Annals of Mathematical Studies*, 28:193–216, 1957.
- M. Lanctot. Further developments of extensive-form replicator dynamics using the sequence-form representation. volume 2, 05 2014.

- M. Lanctot, K. Waugh, M. Zinkevich, and M. Bowling. Monte Carlo sampling for regret minimization in extensive games. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1078–1086, 2009.
- M. Lanctot, V. Zambaldi, A. Gruslys, A. Lazaridou, K. Tuyls, J. Perolat, D. Silver, and T. Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *NIPS*. 2017.
- M. Lanctot, E. Lockhart, J.-B. Lespiau, V. Zambaldi, S. Upadhyay, J. Pérolat, S. Srinivasan, F. Timbers, K. Tuyls, S. Omidshafiei, D. Hennes, D. Morrill, P. Muller, T. Ewalds, R. Faulkner, J. Kramár, B. D. Vyllder, B. Saeta, J. Bradbury, D. Ding, S. Borgeaud, M. Lai, J. Schrittwieser, T. Anthony, E. Hughes, I. Danihelka, and J. Ryan-Davis. OpenSpiel: A framework for reinforcement learning in games. *CoRR*, 2019.
- J. Z. Leibo, E. Hughes, M. Lanctot, and T. Graepel. Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research, 2019.
- C. Lemke and J. Howson, Jr. Equilibrium points of bimatrix games. *Journal of the Society for industrial and Applied Mathematics*, 12(2):413–423, 1964.
- J. W. Leonard J. Savage. The foundations of statistics. 1954.
- C. K. Ling, F. Fang, and J. Z. Kolter. What game are we playing? End-to-end learning in normal and extensive form games. *CoRR*, abs/1805.02777, 2018. URL <http://arxiv.org/abs/1805.02777>.
- M. L. Littman. Friend or foe Q-learning in general-sum games. In *In Proceedings of the 18th Int. Conf. on Machine Learning*, 2001.
- S. Liu, G. Lever, Z. Wang, J. Merel, S. M. A. Eslami, D. Hennes, W. M. Czarnecki, Y. Tassa, S. Omidshafiei, A. Abdolmaleki, N. Y. Siegel, L. Hasenclever, L. Marris, S. Tunyasuvunakool, H. F. Song, M. Wulfmeier, P. Muller, T. Haarnoja, B. D. Tracey, K. Tuyls, T. Graepel, and N. Heess. From motor control to team play in simulated humanoid football. *CoRR*, abs/2105.12196, 2021. URL <https://arxiv.org/abs/2105.12196>.
- S. Liu, M. Lanctot, L. Marris, and N. Heess. Simplex neural population learning: Any-mixture Bayes-optimality in symmetric zero-sum games. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 13793–13806. PMLR, 17–23 Jul 2022a. URL <https://proceedings.mlr.press/v162/liu22h.html>.
- S. Liu, L. Marris, D. Hennes, J. Merel, N. Heess, and T. Graepel. NeuPL: Neural population learning. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=MIX3fJkl_1.
- E. Lockhart, N. Burch, N. Bard, S. Borgeaud, T. Eccles, L. Smaira, and R. Smith. Human-agent cooperation in bridge bidding, 2020.
- L. Marris, P. Muller, M. Lanctot, K. Tuyls, and T. Graepel. Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers. *CoRR*, abs/2106.09435, 2021a. URL <https://arxiv.org/abs/2106.09435>.

- L. Marris, P. Muller, M. Lanctot, K. Tuyls, and T. Graepel. Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7480–7491. PMLR, 18–24 Jul 2021b. URL <http://proceedings.mlr.press/v139/marris21a.html>.
- L. Marris, I. Gemp, T. Anthony, A. Tacchetti, S. Liu, and K. Tuyls. Turbocharging solution concepts: Solving NEs, CEs and CCEs with neural equilibrium solvers. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 5586–5600. Curran Associates, Inc., 2022a. URL https://papers.nips.cc/paper_files/paper/2022/hash/24f420aa4c99642dbb9aae18b166bbbc-Abstract-Conference.html.
- L. Marris, M. Lanctot, I. Gemp, S. Omidshafiei, S. McAleer, J. Connor, K. Tuyls, and T. Graepel. Game theoretic rating in n-player general-sum games with equilibria, 2022b. URL <https://arxiv.org/abs/2210.02205>.
- L. Marris, I. Gemp, and G. Piliouras. Equilibrium-invariant embedding, metric space, and fundamental set of 2x2 normal-form games, 2023. URL <https://arxiv.org/abs/2304.09978>.
- L. Marris, I. Gemp, S. Liu, J. Z. Leibo, and G. Piliouras. Visualizing 2x2 normal-form games: twotwogame latex package, 2024.
- A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
- E. S. Maskin. Mechanism design: How to implement social goals. *American Economic Review*, 98(3): 567–76, 2008.
- J. Matouek and B. Gärtner. *Understanding and Using Linear Programming (Universitext)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 3540306978.
- S. McAleer, J. Lanier, R. Fox, and P. Baldi. Pipeline PSRO: A scalable approach for finding approximate Nash equilibria in large games. In *Neural Information Processing Systems 33*, 2020.
- S. McAleer, J. B. Lanier, P. Baldi, and R. Fox. XDO: A double oracle algorithm for extensive-form games. *CoRR*, abs/2103.06426, 2021. URL <https://arxiv.org/abs/2103.06426>.
- R. D. McKelvey and T. R. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10(1):6–38, 1995. ISSN 0899-8256. doi: <https://doi.org/10.1006/game.1995.1023>. URL <https://www.sciencedirect.com/science/article/pii/S0899825685710238>.
- R. D. McKelvey, A. M. McLennan, and T. L. Turocy. Gambit: Software tools for game theory, version 16.0.1, 2016.
- H. B. McMahan, G. J. Gordon, and A. Blum. Planning in the presence of cost functions controlled by an adversary. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 536–543, 2003.
- T. Mikolov. *STATISTICAL LANGUAGE MODELS BASED ON NEURAL NETWORKS*. Ph.d. thesis, Brno University of Technology, Faculty of Information Technology, 2012. URL <https://www.fit.vut.cz/study/phd-thesis/283/>.

- W. Mischel and E. B. Ebbesen. Attention in delay of gratification. *Journal of Personality and Social Psychology*, 16:329–337, 1970.
- V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, C. B. Stig Petersen, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.
- V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1928–1937, 2016.
- A. Möbius. *Der barycentrische Calcul*. J.A. Barth, 1827.
- D. Monderer and L. S. Shapley. Potential games. *Games and economic behavior*, 14(1):124–143, 1996.
- B. Monnot and G. Piliouras. Limits and limitations of no-regret learning in games. *The Knowledge Engineering Review*, 32:e21, 2017.
- A. W. Moore and C. G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning*, 13(1):103–130, oct 1993.
- G. E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), April 1965.
- H. Moravec. Robots, re-evolving mind. December 2000. URL <https://frc.ri.cmu.edu/~hpm/project.archive/robot.papers/2000/Cerebrum.html>.
- D. Morrill, R. D’Orazio, R. Sarfati, M. Lanctot, J. R. Wright, A. Greenwald, and M. Bowling. Hindsight and sequential rationality of correlated play. In *Proceedings of the The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*, 2021.
- S. Morris and T. Ui. Best response equivalence. *Games and Economic Behavior*, 49(2):260–287, 2004. ISSN 0899-8256. doi: <https://doi.org/10.1016/j.geb.2003.12.004>. URL <https://www.sciencedirect.com/science/article/pii/S0899825604000132>.
- H. Moulin and J.-P. Vial. Strategically zero-sum games: the class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, 7(3-4):201–221, 1978.
- P. Muller, S. Omidshafiei, M. Rowland, K. Tuyls, J. Perolat, S. Liu, D. Hennes, L. Marris, M. Lanctot, E. Hughes, Z. Wang, G. Lever, N. Heess, T. Graepel, and R. Munos. A generalized training approach for multiagent learning. In *International Conference on Learning Representations*, 2020.
- R. Munos, T. Stepleton, A. Harutyunyan, and M. G. Bellemare. Safe and efficient off-policy reinforcement learning, 2016.
- K. Murty. *Linear Programming*. Wiley, 1983. ISBN 9780471097259.
- J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.
- R. Nau, S. G. Canovas, and P. Hansen. On the geometry of Nash equilibria and correlated equilibria. *International Journal of Game Theory*, 32(4):443–453, August 2004.
- H. Nikaidô and K. Isoda. Note on non-cooperative convex games. *Pacific Journal of Mathematics*, 5(S1): 807 – 815, 1955.

- N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007.
- E. Nudelman, J. Wortman, Y. Shoham, and K. Leyton-Brown. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '04, page 880–887, USA, 2004. IEEE Computer Society. ISBN 1581138644.
- J. Nunen. A set of successive approximation methods for discounted markovian decision problems. *Zeitschrift für Operations Research*, 20:203–208, 1976.
- D. P. O’Leary. A generalized conjugate gradient algorithm for solving a class of quadratic programming problems. *Linear Algebra and its Applications*, 34:371–399, 1980/12// 1980.
- S. Omidshafiei, C. Papadimitriou, G. Piliouras, K. Tuyls, M. Rowland, J.-B. Lespiau, W. M. Czarnecki, M. Lanctot, J. Perolat, and R. Munos. α -rank: Multi-agent evaluation by evolution. *Scientific Reports*, 9(1):9937, 2019.
- S. Omidshafiei, K. Tuyls, W. M. Czarnecki, F. C. Santos, M. Rowland, J. Connor, D. Hennes, P. Muller, J. Pérolat, B. D. Vylder, A. Gruslys, and R. Munos. Navigating the landscape of multiplayer games. *Nature Communications*, 11(1):5603, Nov 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-19244-4. URL <https://doi.org/10.1038/s41467-020-19244-4>.
- S. Omidshafiei, A. Kapishnikov, Y. Assogba, L. Dixon, and B. Kim. Beyond rewards: a hierarchical perspective on offline multiagent behavioral analysis, 2022. URL <https://arxiv.org/abs/2206.09046>.
- E. I. Organick. *A FORTRAN IV Primer*. Addison-Wesley, 1966.
- L. E. Ortiz, R. E. Schapire, and S. M. Kakade. Maximum entropy correlated equilibria. In M. Meila and X. Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 347–354, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR.
- E. Ostrom, R. Gardner, J. Walker, and J. Walker. *Rules, Games, and Common-pool Resources*. ACLS Humanities E-Book. University of Michigan Press, 1994. ISBN 9780472065462.
- G. Ostrovski. Topics arising from fictitious play dynamics, October 2013. URL <http://wrap.warwick.ac.uk/58894/>.
- C. H. Papadimitriou and T. Roughgarden. Computing correlated equilibria in multi-player games. *J. ACM*, 55(3), aug 2008. ISSN 0004-5411. doi: 10.1145/1379759.1379762. URL <https://doi.org/10.1145/1379759.1379762>.
- A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.

- K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. doi: 10.1080/14786440109462720. URL <https://doi.org/10.1080/14786440109462720>.
- J. Peng and R. J. Williams. Incremental multi-step q-learning. *Machine Learning*, 22(1):283–290, Mar 1996. ISSN 1573-0565. doi: 10.1007/BF00114731. URL <https://doi.org/10.1007/BF00114731>.
- J. Perolat, B. D. Vyllder, D. Hennes, E. Tarassov, F. Strub, V. de Boer, P. Muller, J. T. Connor, N. Burch, T. Anthony, S. McAleer, R. Elie, S. H. Cen, Z. Wang, A. Gruslys, A. Malysheva, M. Khan, S. Ozair, F. Timbers, T. Pohlen, T. Eccles, M. Rowland, M. Lanctot, J.-B. Lespiaau, B. Piot, S. Omidshafiei, E. Lockhart, L. Sifre, N. Beauguerlange, R. Munos, D. Silver, S. Singh, D. Hassabis, and K. Tuyls. Mastering the game of Stratego with model-free multiagent reinforcement learning. *Science*, 378(6623): 990–996, 2022. doi: 10.1126/science.add4679. URL <https://www.science.org/doi/abs/10.1126/science.add4679>.
- L. Perron and V. Furnon. OR-Tools. URL <https://developers.google.com/optimization/>.
- J. Pineau, G. Gordon, and S. Thrun. Anytime point-based approximations for large pomdps. *J. Artif. Int. Res.*, 27(1):335–380, Nov. 2006. ISSN 1076-9757.
- B. Polyak. The conjugate gradient method in extreme problem. *USSR Computational Mathematics and Mathematical Physics*, 9:94–112, 12 1969.
- R. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a Nash equilibrium. *Games and Economic Behavior*, 63(2):642–662, 2008a. ISSN 0899-8256. doi: <https://doi.org/10.1016/j.geb.2006.03.015>. URL <https://www.sciencedirect.com/science/article/pii/S0899825606000935>. Second World Congress of the Game Theory Society.
- R. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a Nash equilibrium. *Games and Economic Behavior*, 63(2):642–662, 2008b.
- M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
- M. L. Puterman and M. C. Shin. Modified policy iteration algorithms for discounted markov decision problems. *Management Science*, 24(11):1127–1137, 1978.
- A. Rapoport and M. Guyer. A taxonomy of 2x2 games. *General Systems*, 11:203–214, 1966.
- A. Rapoport, M. Guyer, and D. G. Gordon. *The 2 X 2 game*. University of Michigan Press Ann Arbor, 1976. ISBN 0472087428.
- T. Rashid, C. Zhang, and K. Ciosek. Estimating α -rank by maximizing information gain. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5673–5681, 2021.
- L. F. Richardson. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210:307–357, 1911. ISSN 02643952.
- D. Robinson and D. Goforth. *The Topology of the 2x2 games: A New Periodic Table*. 01 2005. doi: 10.4324/9780203340271.

- D. Robinson, D. Goforth, and M. Cargill. Toward a topological treatment of the non-strictly ordered 2×2 games. 7 2007.
- R. W. Rosenthal. Correlated equilibria in some classes of two-person games. *International Journal of Game Theory*, 3(3):119–128, Sep 1974. ISSN 1432-1270. doi: 10.1007/BF01763252. URL <https://doi.org/10.1007/BF01763252>.
- M. Rowland, S. Omidshafiei, K. Tuyls, J. Perolat, M. Valko, G. Piliouras, and R. Munos. Multiagent evaluation under incomplete information. *arXiv preprint arXiv:1909.09849*, 2019.
- D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, Oct. 1986.
- G. Rummery and M. Niranjan. On-line q-learning using connectionist systems. *Technical Report CUED/F-INFENG/TR 166*, 11 1994.
- A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell. Policy distillation, 2016.
- R. R. Schaller. Moore’s law: Past, present, and future. *IEEE Spectr.*, 34(6):52–59, jun 1997. ISSN 0018-9235. doi: 10.1109/6.591665. URL <https://doi.org/10.1109/6.591665>.
- T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay, 2016.
- T. Schelling. *Arms and influence*. Yale University Press, 1966.
- D. Schmidtchen. *To Help or Not to Help: The Samaritan’s Dilemma Revisited*, pages 470–484. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002. ISBN 978-3-662-04810-8. doi: 10.1007/978-3-662-04810-8_28. URL https://doi.org/10.1007/978-3-662-04810-8_28.
- J. Schosser. Fairness in the use of limited resources during a pandemic. *Plos one*, 17(6):e0270022, 2022.
- J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. P. Lillicrap, and D. Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *CoRR*, abs/1911.08265, 2019. URL <http://arxiv.org/abs/1911.08265>.
- J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. Trust region policy optimization. *CoRR*, abs/1502.05477, 2015. URL <http://arxiv.org/abs/1502.05477>.
- J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL <http://arxiv.org/abs/1707.06347>.
- C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.
- L. S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953. ISSN 0027-8424.
- L. S. Shapley. Some topics in two-person games. 1963.
- J. Shawe-Taylor. Symmetries and discriminability in feedforward network architectures. *IEEE transactions on neural networks*, 4 5:816–26, 1993.
- J. Shawe-Taylor. Introducing invariance: a principled approach to weight sharing. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN’94)*, volume 1, pages 345–349 vol.1, 1994. doi: 10.1109/ICNN.1994.374187.

- Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.
- Y. Shoham, R. Powers, and T. Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171(7):365–377, 2007. ISSN 0004-3702. Foundations of Multi-Agent Learning.
- D. Silver. Lectures on reinforcement learning, 2015. URL <https://www.davidsilver.uk/teaching/>.
- D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.
- D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018. doi: 10.1126/science.aar6404. URL <https://www.science.org/doi/abs/10.1126/science.aar6404>.
- J. Simpson. *Simulating Strategic Rationality*. PhD thesis, University of Alberta, 2010.
- H.-W. Sinn. A Rehabilitation of the Principle of Insufficient Reason. *The Quarterly Journal of Economics*, 94(3):493–506, 05 1980.
- B. Skyrms. *The Stag Hunt and the Evolution of Social Structure*. Cambridge University Press, 2004. ISBN 9780521533928.
- S. Sokota, E. Lockhart, F. Timbers, E. Davoodi, R. D’Orazio, N. Burch, M. Schmid, M. Bowling, and M. Lanctot. Solving common-payoff games with approximate policy iteration, 2021.
- E. Solan and N. Vieille. Stochastic games. *Proceedings of the National Academy of Sciences*, 112(45): 13743–13746, 2015. ISSN 0027-8424. doi: 10.1073/pnas.1513508112. URL <https://www.pnas.org/content/112/45/13743>.
- F. Southey, M. Bowling, B. Larson, C. Piccione, N. Burch, D. Billings, and C. Rayner. Bayes’ bluff: Opponent modelling in poker. pages 550–558, 2005.
- F. Southey, B. Hoehn, and R. Holte. Effective short-term opponent exploitation in simplified poker. *Machine Learning*, 74:159–189, 02 2009.
- S. Srinivasan, M. Lanctot, V. Zambaldi, J. Pérolat, K. Tuyls, R. Munos, and M. Bowling. Actor-critic policy optimization in partially observable multiagent environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL <http://jmlr.org/papers/v15/srivastava14a.html>.
- B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 2020.
- R. Sugden. *The Economics of Rights, Co-operation and Welfare*. Springer, 1986.

- R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL <http://incompleteideas.net/book/the-book-2nd.html>.
- R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, pages 1057–1063, Cambridge, MA, USA, 1999. MIT Press. URL <http://dl.acm.org/citation.cfm?id=3009657.3009806>.
- G. Tesauro. Temporal Difference Learning and TD-Gammon. *Commun. ACM*, 38(3):58–68, mar 1995. ISSN 0001-0782. doi: 10.1145/203330.203343. URL <https://doi.org/10.1145/203330.203343>.
- E. H. Thiede, T. Hy, and R. Kondor. The general theory of permutation equivariant neural networks and higher order graph variational encoders. *CoRR*, abs/2004.03990, 2020. URL <https://arxiv.org/abs/2004.03990>.
- T. Tieleman, G. Hinton, et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.
- N. T. Trendafilov and R. A. Lippert. The multimode procrustes problem. *Linear algebra and its applications*, 349(1-3):245–264, 2002.
- C. Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52:479–487, 1988.
- K. Tuyls, J. Pérolat, M. Lanctot, E. Hughes, R. Everett, J. Z. Leibo, C. Szepesvári, and T. Graepel. Bounds and dynamics for empirical game theoretic analysis. *Auton. Agents Multi Agent Syst.*, 34(1):7, 2020.
- L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL <http://www.jmlr.org/papers/v9/vandermaaten08a.html>.
- H. van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning, 2015. URL <https://arxiv.org/abs/1509.06461>.
- H. van Seijen, H. van Hasselt, S. Whiteson, and M. Wiering. A theoretical and empirical analysis of expected sarsa. In *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 177–184, 2009. doi: 10.1109/ADPRL.2009.4927542.
- A. Vanderveldt, L. Oliveira, and L. Green. Delay discounting: Pigeon, rat, human-does it matter? *Journal of experimental psychology. Animal learning and cognition*, 42, 02 2016. doi: 10.1037/xan0000097.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.
- O. Vinyals, I. Babuschkin, W. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. Agapiou, M. Jaderberg, and D. Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575, 11 2019.

- J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, 1928. doi: 10.1007/BF01448847.
- J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1947.
- B. von Stengel and F. Forges. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research*, 33(4):1002–1022, 2008.
- A. Wald. Contributions to the theory of statistical estimation and testing hypotheses. *Ann. Math. Statist.*, 10(4):299–326, 12 1939.
- A. Wald. Statistical decision functions which minimize the maximum risk. *Annals of Mathematics*, 46: 265–280, 1945.
- B. Walliser. A simplified taxonomy of 2×2 games. *Theory and Decision*, 25(2):163–191, Sep 1988. ISSN 1573-7187. doi: 10.1007/BF00134158. URL <https://doi.org/10.1007/BF00134158>.
- W. Walsh, R. Das, G. Tesauro, and J. Kephart. Analyzing complex strategic interactions in multi-agent systems. 01 2002.
- Y. Wang and S. Xia. Unifying attribute splitting criteria of decision trees by Tsallis entropy. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2507–2511, 2017.
- Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. de Freitas. Sample efficient actor-critic with experience replay. *CoRR*, abs/1611.01224, 2016.
- C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King’s College, Oxford, 1989.
- M. P. Wellman. Methods for empirical game-theoretic analysis. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 1552–1556. AAAI Press, 2006. URL <http://www.aaai.org/Library/AAAI/2006/aaai06-248.php>.
- G. S. Wilkinson. Reciprocal food sharing in the vampire bat. *Nature*, 308(5955):181–184, Mar 1984. ISSN 1476-4687. doi: 10.1038/308181a0. URL <https://doi.org/10.1038/308181a0>.
- R. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.
- R. Williams, L. C. Baird, and III. Tight performance bounds on greedy policies based on imperfect value functions. Technical report, 1993.
- R. J. Williams and L. C. Baird. Analysis of some incremental variants of policy iteration: First steps toward understanding actor-cr. 1993.
- R. J. Williams and J. Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991. doi: 10.1080/09540099108946587.
- J. Wood and J. Shawe-Taylor. Representation theory and invariant neural networks. *Discrete Applied Mathematics*, 69(1):33–60, 1996. ISSN 0166-218X. doi: [https://doi.org/10.1016/0166-218X\(95\)00075-3](https://doi.org/10.1016/0166-218X(95)00075-3). URL <https://www.sciencedirect.com/science/article/pii/0166218X95000753>.

- M. Woodbury and P. U. D. of Statistics. *Inverting Modified Matrices*. Memorandum Report / Statistical Research Group, Princeton. Department of Statistics, Princeton University, 1950.
- H. P. Young. *Strategic learning and its limits*. OUP Oxford, 2004.
- M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola. Deep sets, 2018.
- W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. *CoRR*, abs/1409.2329, 2014. URL <http://arxiv.org/abs/1409.2329>.
- K. Zhang, Z. Yang, and T. Başar. *Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms*, pages 321–384. Springer International Publishing, Cham, 2021. ISBN 978-3-030-60990-0. doi: 10.1007/978-3-030-60990-0_12. URL https://doi.org/10.1007/978-3-030-60990-0_12.
- M. Zinkevich, A. Greenwald, and M. L. Littman. Cyclic equilibria in markov games. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS’05, pages 1641–1648, Cambridge, MA, USA, 2005. MIT Press.