Hindawi
Complexity
Volume 2022, Article ID 4830491, 20 pages
https://doi.org/10.1155/2022/4830491

Research Article

Limiting Dynamics for Q-Learning with Memory One in Symmetric Two-Player, Two-Action Games

J. M. Meylahn [1,2] **and L. Janssen** [3]

[1] *Department of Applied Mathematics, University of Twente, Enschede, Netherlands*
[2] *Dutch Institute of Emergent Phenomena, University of Amsterdam, Amsterdam, Netherlands*
[3] *Faculty of Science, University of Amsterdam, Amsterdam, Netherlands*

Correspondence should be addressed to J. M. Meylahn; j.m.meylahn@utwente.nl

Received 1 September 2022; Revised 17 October 2022; Accepted 25 October 2022; Published 8 November 2022

Academic Editor: Hassan Zargarzadeh

We develop a method based on computer algebra systems to represent the mutual pure strategy best-response dynamics of symmetric two-player, two-action repeated games played by players with a one-period memory. We apply this method to the iterated prisoner's dilemma, stag hunt, and hawk-dove games and identify all possible equilibrium strategy pairs and the conditions for their existence. The only equilibrium strategy pair that is possible in all three games is the win-stay, lose-shift strategy. Lastly, we show that the mutual best-response dynamics are realized by a sample batch Q-learning algorithm in the infinite batch size limit.

1. Introduction

As algorithms are increasingly employed in a variety of settings, more situations will arise in which algorithms interact. In the case of reinforcement learning (RL) algorithms, what this interaction will lead to is not easily predictable. This is due to most algorithms being designed for single-agent settings (or stationary environments). The theoretical foundations for multiagent reinforcement learning (MARL) are thin, with few convergence results [1] and dealing with nonstationarity has been dubbed one of the grand challenges of MARL [2]. For an overview of the learning pathologies faced by multiagent learning (MAL) algorithms see [3] and for surveys of MAL in general see [4–7]. The most challenging setting to overcome nonstationarity in is decentralized MARL, in which the algorithms have no means of communicating and no information regarding the opponent's payoffs. One of the advantages of decentralized MARL is that such algorithms scale better in environments with more agents. This is the setting we will focus on here.

There are many application areas in which multiple algorithms can be learned in the same environment. For example, in wireless communications and networking [8], vehicle routing [9], or algorithmic pricing [10]. Interestingly, ideas from game theory and multiagent learning are also applicable in optimal control theory, where control problems may be transformed into games such that the Nash equilibrium provides the optimal control policy [11–13].

One approach to modelling the dynamics of decentralized MARL algorithms is based on continuous time approximations of the learning dynamics (with learning dynamics, we mean the evolution of the Q-values under the Q-value updates) [14–16]. This approach shows that the dynamics of the Q-learning algorithm [17] with softmax exploration and variants of it are well approximated by the replicator dynamics of evolutionary game theory [18], and the insights gained from these models have led to the development of new algorithms such as frequency-adjusted Q-learning [19]. A similar approach is taken in [20] with the difference being that the authors consider Q-learning with ϵ -greedy exploration. In both cases, the resulting ordinary differential equations (ODEs) encode the deterministic dynamics of how one expects the Q-values to evolve in time (these ODEs can be viewed as a mean-field model).

An alternative approach is developed in [21], where the authors define deterministic learning for temporal difference learning algorithms. Deterministic learning encodes the learning dynamics under the assumption that the learners have perfect information; i.e., they have a perfect picture of the environment (including their opponent) for each learning update. The dynamics can be interpreted as the dynamics of sample batch temporal difference learning algorithms in the infinite batch size limit.

Both of the approaches describe the dynamics in the value space, i.e., on the level of the Q-values. With ϵ-greedy exploration, however, the strategy space is finite given a fixed exploration rate ϵ. This allows us to develop a representation of the deterministic learning dynamics, interpreted as in [21], for the case of pure strategies ($\epsilon = 0$) using directed networks. The advantage of our approach, over the previously mentioned approaches, is that the dynamics can easily be represented in cases where the environment can be in multiple states (i.e. Markov games), and it allows for the exact calculation of the possible absorbing states, the conditions for their existence, and their basin of attraction (since the dynamics here are deterministic, the basin of attraction of an absorbing state is the fraction of initial conditions that lead to it). Knowing the possible absorbing states is necessary for defining appropriate learning goals for the algorithm, and their basin of attraction quantify the likelihood with which the learning goals can be achieved.

We will demonstrate our approach in the case of symmetric two-player, (we will use the terms "player" and "agent" interchangeably) two-action games, in which both players can condition their actions on the actions played in the previous round of the game. This is economically relevant since many of the classical strategies (such as the Tit-for-Tat strategy) for two-player, two-action games require a one-period memory [22]. More specifically, we will study the resulting dynamics in the cases of the prisoner's dilemma (PD) [23], the stag hunt (SH) [24], and the hawk-dove (HD) [25] (also known as the chicken or snowdrift) games.

The choice of these games is motivated by a particularly relevant application domain in which understanding the dynamics of multiagent learning is important: algorithmic pricing [26]. Here, multiple algorithms learn optimal prices to maximize profits for their respective firms in a market. If these algorithms converge to higher than competitive prices (and thus learn to cooperate or collude), this has detrimental effects for consumers. Recent work has shown that Q-learning with a one-period memory can learn collusive strategies [27–29], although this takes place on a prohibitively long timescale [30].

The PD is a simpler setting (in relation to pricing environments) in which cooperation (or collusion) can be observed [30]. This is one of the paradigmatic models for studying the emergence of cooperation in social dilemmas [31], pricing duopolies [32], and other economics games [33]. The model's simplicity allows for rigorous analysis, giving insights that can be heuristically extended to more realistic settings. In addition to the PD, we will study the SH game and the HD game. Both of these games retain the interpretation of the two actions as being cooperation and defection and have been studied extensively in the literature (see [34–37] for recent contributions in the context of reinforcement learning). This choice of games is also representative of the different possible combinations of dilemma strength parameters (we exclude the trivial Harmony game in which no dilemma arises). The dilemma strength parameters quantify how advantageous it is to defect when playing against a cooperator and how disadvantageous it is to cooperate when playing against a defector. They are particularly useful for determining outcomes in evolutionary game theory [38–42]. The representation of the game dynamics we develop is, however, applicable to all twelve symmetric two-player, two-action games and can trivially be extended to all games in the periodic table of 2×2 games [43]. For Q-learning without memory, such games have been classified in [44].

The mean-field approach has been employed successfully to study the learning dynamics of decentralized MARL algorithms in normal form games, for example, Q-learning in zero-sum and partnership normal form games [45] and, exponential RL in zero-sum games with an interior Nash equilibrium [46]. More recently, similar techniques have been applied in the stochastic game setting, showing convergence in zero-sum stochastic games [47, 48]. To achieve this, the authors make use of two-timescale learning dynamics, which ensure that the environment is quasi-stationary for all agents.

Best-response dynamics have a long history as a tool for analysing learning in games [49, 50]. The asymptotic properties of the perturbed best-response dynamics in the low noise limit correspond to evolutionary game theory dynamics [51]. Best-response dynamics can be implemented using fictitious play [52], but this requires full information of the agent's own payoffs, which we assume not to be available here.

Our approach lends itself to extensions with which it is possible to approximate the dynamics of sample batch algorithms with finite batch sizes in such a way that the effects of the various components of the algorithms (the batch size, exploration rate, and learning rate) can be isolated. Understanding the interplay of these components is necessary for explaining the learning dynamics of such algorithms in multiagent settings. Such insights can in turn be used to design algorithms that learn more successfully in multiagent settings. Parallel work in this direction [53] shows how the representation developed here complements the deterministic learning dynamics developed in [21].

1.1. Contributions. This article aims at contributing to the modelling of the dynamics of MARL by developing a method using computer algebra systems for representing the (deterministic) mutual best-response dynamics in symmetric two-player, two-action iterated games with one-period memory. More specifically,

(i) This method allows us to study the possible absorbing states, the conditions for their existence, and their basin of attraction.

(ii) We show that the absorbing states identified in [54] for the PD are the only absorbing states and find that the number of absorbing states is significantly higher for both the SH game and the HD games.

(iii) The only strategy pair that is an absorbing state in all three games is when both players use the win-stay, lose-shift (WSLS) strategy. This suggests that the WSLS strategy has a structure which makes it ideal for dealing with dilemmas.

(iv) Finally, we show that the mutual best-response dynamics are realized by a simple sample batch Q-learning algorithm in the infinite bath size limit.

1.2. Structure. In Section 2, we identify notation, define the model, and introduce Q-learning. In Section 3, we describe our representation of deterministic learning. In Section 4, we present the results for the three games we consider, and in Section 5, we show that a simple sample batch Q-learning algorithm converges to the mutual best-response dynamics in the infinite batch size limit. Finally, in Section 6, we conclude and discuss possible extensions and future research.

2. Setting

2.1. Notation. For ease of reference, we provide an overview of the notation used in the article. We denote the actions available to the players by D, for defect and C, for cooperate. For the payoffs of the game, we use T, R, P, and S. The strategy of player a is denoted by π_a and the expectation $\mathbb{E}$ is taken with respect to the strategies of the players. The discount factor is denoted by δ, the learning rate by α, and the exploration rate by ϵ. Q-values are denoted by $Q_{a,\sigma,i}$, where the first index refers to the player, the second index refers to the state, and the third index refers to the action. The related value function is denoted by $V_a(\sigma)$. The indicator function is denoted by $\{A\}$ and evaluates to one when A is true and to zero when A is false. We use $\vee$ to denote the logical "or" and $\wedge$ to denote the logical "and."

2.2. Environment. We consider a setting similar to that in [54], and we will employ broadly similar notation for ease of reference. The setting is that of a two-player, two-action game played by two Q-learners, labelled $a \in \{1, 2\}$ and using the convention $-a = \{1, 2\}/a$, with one-period memory. Each Q-learner has a choice each period between two actions: defect (D) or cooperate (C). We denote the actions by $\sigma_a \in \{C, D\}$. Due to the one-period memory, the game becomes a multistate game, with the possible states being $\sigma = (\sigma_1, \sigma_2) \in \{(D, D), (D, C), (C, D), (C, C)\}$, where the first component of the vector denotes the action of player one and the second component denotes the action of player two. The payoff (or reward) for the player a when the game is in state σ is denoted by $r_a(\sigma)$ and takes the following form:

$$\begin{pmatrix} r_1(1), r_2(1) & r_1(2), r_2(2) \\ r_1(3), r_2(3) & r_1(4), r_2(4) \end{pmatrix} = \begin{pmatrix} P, P & T, S \\ S, T & R, R \end{pmatrix}, \tag{1}$$

where we have enumerated the states as $(D, D) \longrightarrow 1$, $(D, C) \longrightarrow 2$, $(C, D) \longrightarrow 3$, and $(C, C) \longrightarrow 4$. In addition, we make the standard assumptions that $T > R > P > S$ for the PD, $R > T > P > S$ for the SH game and $T > R > S > P$ for the HD game.

2.3. Learning. Each player chooses their actions according to a strategy π_a, which in the Q-learning with period one memory setting is the conditional probability of playing action σ_a given that the game is in state σ', i.e. $\pi_a(\sigma_a|\sigma')$.

The strategies of the players are updated through a reinforcement learning process. We consider the Q-learning algorithm, which is designed to learn the actions that maximize the discounted future rewards, i.e.,

$$\sum_{n=1}^{\infty} \delta^n r_a(\boldsymbol{\sigma}(n)), \tag{2}$$

with $0 < \delta < 1$ (in the case without discounting; i.e., $\delta = 1$, it is necessary either to consider a finite time horizon T or to consider the player's objective to be maximizing their average reward. Our method is not directly applicable in this case, although a similar procedure may be possible in the latter case by using the stationary distribution over the states given fixed strategies) being the discount factor and n denoting the time.

It has been shown that maximizing (2) can be achieved in a stationary environment by solving

$$\begin{aligned} Q_{a,\boldsymbol{\sigma}(t),\sigma_a(t)} &= \mathbb{E}\left[r_a(\boldsymbol{\sigma}(t+1))|\boldsymbol{\sigma}(t), \sigma_a(t)\right] \\ &\quad + \delta\mathbb{E}\left[\max_{i\in\{C,D\}}\left\{Q_{a,\boldsymbol{\sigma}(t+1),i}\right\}|\boldsymbol{\sigma}(t), \sigma_a(t)\right]. \end{aligned} \tag{3}$$

Note that, since the actions taken in the previous round determine the current state, we have used the notation that $\sigma(t) = (\sigma_1(t-1), \sigma_2(t-1))$. The first term on the right-hand side is the expected reward from taking action $\sigma_a(t)$ while in state $\sigma(t)$ and the second term is the discounted expected maximum reward of behaving optimally (according to the current estimate) in the state $\sigma(t+1)$ reached by taking action $\sigma_a(t)$ in state $\sigma(t)$. The Q-function defined by (3) is related to Bellman's value function by $V_a(\sigma) = \max_{i\in\{C,D\}}\{Q_{a,\sigma,i}\}$.

The Q-learner a solves (3) by initializing a Q-matrix

$$\begin{pmatrix} Q_{a,1,D} & Q_{a,1,C} \\ Q_{a,2,D} & Q_{a,2,C} \\ Q_{a,3,D} & Q_{a,3,C} \\ Q_{a,4,D} & Q_{a,4,C} \end{pmatrix}, \tag{4}$$

and updating the entries according to

$$\begin{aligned} Q_{a,\boldsymbol{\sigma},\sigma_a}(t+1) &= (1-\alpha(t))Q_{a,\boldsymbol{\sigma},\sigma_a}(t) \\ &\quad + \alpha(t)\left[r_a(\boldsymbol{\sigma}(t+1)) + \delta\max_{i\in\{C,D\}}\left\{Q_{a,\boldsymbol{\sigma}(t+1),i}\right\}\right], \end{aligned} \tag{5}$$

where

$$\alpha(t) = \begin{cases} \alpha, & \text{if } (\boldsymbol{\sigma}, \sigma_a) = (\boldsymbol{\sigma}(t), \sigma_a(t)), \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

is the learning rate. Note that $r_a(\sigma(t+1))$ is the reward obtained in period t (i.e., the reward of taking action $\sigma_a(t)$ while in state $\sigma(t)$, which leads to state $\sigma(t+1)$). In stationary environments, the iterative process defined by (5) has been shown to converge under some general conditions, for example, in [55]. At any point in time, the Q-learner's estimate of the optimal strategy is given by the strategy in which the learner selects the action with the largest Q-value in every state.

In contrast to the general multistate setting, the transitions between states here happen deterministically given the actions of the players. This means that the only stochasticity comes from the action-selection mechanism, i.e. the strategies. Such an action-selection mechanism must take into account the exploration-exploitation trade-off, where exploratory actions are conducive to learning and exploratory actions maximize the reward given the agent's current state of knowledge. The methods used in Q-learning for this balancing act can be split into two main categories: the ϵ-greedy mechanisms and the softmax or Boltzman mechanisms. We will focus on mechanisms of the first type and leave the latter mechanisms for future work. With an ϵ -greedy action-selection mechanism, the player chooses the action which it considers to be maximizing its reward with probability $1-\epsilon(t)$ and chooses an action uniformly at random with probability $\epsilon(t)$, where the exploration rate may depend on time.

Q-learning converges to a solution of the Bellman equations in stationary environments. A Q-learner in a multiagent setting, on the other hand, is facing a nonstationary environment since the strategy of the opponent changes over time. One way to mitigate this nonstationarity is by learning in batches, during which both players keep their strategy fixed [21]. At the end of each batch, the players update their strategies simultaneously. The postbatch update can be implemented with a larger learning rate (the learning rate determines how much weight is given to the information obtained during the batch) when the batch is large, as a larger batch leads to a better estimate of the (now stationary) environment. In the infinite batch size limit, the postbatch update may be performed with a learning rate of one, which leads to both players simultaneously playing the best response to the strategy of the opponent after each batch (see Section 5). The absorbing states of this algorithm are the same as the absorbing states of the algorithm in [54] (if the players do not change strategy when mutually best-responding, they will also not change strategy when sequentially best-responding and vice versa).

3. Method

3.1. Self-Consistent Solutions to the Bellman Equations. If we restrict ourselves to pure strategies, each player can pick one of two actions for each of the four states, leading to a total of 2^4 strategies. As a result, the total number of pure strategy pairs is $2^8 = 256$. Using computer algebra software, such as Mathematica, we can automate the calculations in [54], perform them for the 16 symmetric strategy pairs, and repeat them for all possible strategy pairs. This allows us to identify the best response for each strategy.

To this end, we identify each strategy with a 4-dimensional vector. The four entries encode the strategy. A zero in the first entry means that $Q_{1,1,D} > Q_{1,1,C}$, while a one means that $Q_{1,1,D} < Q_{1,1,C}$. The vector $(0,0,0,0)$, for example, represents the all defect (All-D) strategy (the index of the entry of the vector corresponds to the state in which the action is taken). The vector $(1,0,0,1)$ in contrast represents the win-stay-lose-shift (WSLS) strategy. Similarly, we define a strategy pair as an 8-dimensional vector in which the first four entries encode the strategy of player 1, and the last four entries encode the strategy of player 2 (to ensure that the strategies look the same for both players, we associate the 6th entry in the vector with state 3rd and the 7th entry with state 2). Each strategy pair vector leads to a system of sixteen linear equations that can be solved simultaneously for the sixteen Q-values. The Q-values are then expressed in terms of the model parameters T, R, P, and S. The vector, however, also encodes an assumption on the inequalities obtaining between the Q-values. We then perform a self-consistency check to see if the resulting Q-values satisfy the assumed inequalities given the assumptions on the model parameters. In this way, we can determine under which conditions there is a valid solution to the Bellman equations given the strategy pair.

To illustrate the computation, we will consider the PD where both players are playing the all defect strategy, i.e., the vector $(0,0,0,0,0,0,0,0)$. The Bellman (3) for player 1 can be written in terms of indicator functions when considering pure strategies. The equations for player 1 become

$$Q_{1,1,D} = 1_{\{Q_{2,1,D} > Q_{2,1,C}\}}\left(p + \delta Q_{1,1,D} 1_{\{Q_{1,1,D} > Q_{1,1,C}\}} + \delta Q_{1,1,C} 1_{\{Q_{1,1,D} < Q_{1,1,C}\}}\right) + 1_{\{Q_{2,1,D} < Q_{2,1,C}\}}\left(t + \delta Q_{1,2,D} 1_{\{Q_{1,2,D} > Q_{1,2,C}\}} + \delta Q_{1,2,C} 1_{\{Q_{1,2,D} < Q_{1,2,C}\}}\right), \tag{7}$$

$$Q_{1,1,C} = 1_{\{Q_{2,1,D} > Q_{2,1,C}\}}\left(s + \delta Q_{1,3,D} 1_{\{Q_{1,3,D} > Q_{1,3,C}\}} + \delta Q_{1,3,C} 1_{\{Q_{1,3,D} < Q_{1,3,C}\}}\right) + 1_{\{Q_{2,1,D} < Q_{2,1,C}\}}\left(r + \delta Q_{1,4,D} 1_{\{Q_{1,4,D} > Q_{1,4,C}\}} + \delta Q_{1,4,C} 1_{\{Q_{1,4,D} < Q_{1,4,C}\}}\right), \tag{8}$$

$$Q_{1,2,D} = 1_{\{Q_{2,2,D} > Q_{2,2,C}\}}\left(p + \delta Q_{1,1,D} 1_{\{Q_{1,1,D} > Q_{1,1,C}\}} + \delta Q_{1,1,C} 1_{\{Q_{1,1,D} < Q_{1,1,C}\}}\right) + 1_{\{Q_{2,2,D} < Q_{2,2,C}\}}\left(t + \delta Q_{1,2,D} 1_{\{Q_{1,2,D} > Q_{1,2,C}\}} + \delta Q_{1,2,C} 1_{\{Q_{1,2,D} < Q_{1,2,C}\}}\right), \tag{9}$$

$$Q_{1,2,C} = 1_{\{Q_{2,2,D} > Q_{2,2,C}\}}\left(s + \delta Q_{1,3,D} 1_{\{Q_{1,3,D} > Q_{1,3,C}\}} + \delta Q_{1,3,C} 1_{\{Q_{1,3,D} < Q_{1,3,C}\}}\right) + 1_{\{Q_{2,2,D} < Q_{2,2,C}\}}\left(r + \delta Q_{1,4,D} 1_{\{Q_{1,4,D} > Q_{1,4,C}\}} + \delta Q_{1,4,C} 1_{\{Q_{1,4,D} < Q_{1,4,C}\}}\right), \tag{10}$$

$$Q_{1,3,D} = 1_{\{Q_{2,3,D} > Q_{2,3,C}\}}\left(p + \delta Q_{1,1,D} 1_{\{Q_{1,1,D} > Q_{1,1,C}\}} + \delta Q_{1,1,C} 1_{\{Q_{1,1,D} < Q_{1,1,C}\}}\right) + 1_{\{Q_{2,3,D} < Q_{2,3,C}\}}\left(t + \delta Q_{1,2,D} 1_{\{Q_{1,2,D} > Q_{1,2,C}\}} + \delta Q_{1,2,C} 1_{\{Q_{1,2,D} < Q_{1,2,C}\}}\right), \tag{11}$$

$$Q_{1,3,C} = 1_{\{Q_{2,3,D} > Q_{2,3,C}\}}\left(s + \delta Q_{1,3,D} 1_{\{Q_{1,3,D} > Q_{1,3,C}\}} + \delta Q_{1,3,C} 1_{\{Q_{1,3,D} < Q_{1,3,C}\}}\right) + 1_{\{Q_{2,3,D} < Q_{2,3,C}\}}\left(r + \delta Q_{1,4,D} 1_{\{Q_{1,4,D} > Q_{1,4,C}\}} + \delta Q_{1,4,C} 1_{\{Q_{1,4,D} < Q_{1,4,C}\}}\right), \tag{12}$$

$$Q_{1,4,D} = 1_{\{Q_{2,4,D} > Q_{2,4,C}\}}\left(p + \delta Q_{1,1,D\{Q_{1,1,D} > Q_{1,1,C}\}} + \delta Q_{1,1,C\{Q_{1,1,D} < Q_{1,1,C}\}}\right) + 1_{\{Q_{2,4,D} < Q_{2,4,C}\}}\left(t + \delta Q_{1,2,D\{Q_{1,2,D} > Q_{1,2,C}\}} + \delta Q_{1,2,C\{Q_{1,2,D} < Q_{1,2,C}\}}\right), \tag{13}$$

$$Q_{1,4,C} = 1_{\{Q_{2,4,D} > Q_{2,4,C}\}}\left(s + \delta Q_{1,3,D\{Q_{1,3,D} > Q_{1,3,C}\}} + \delta Q_{1,3,C\{Q_{1,3,D} < Q_{1,3,C}\}}\right) + 1_{\{Q_{2,4,D} < Q_{2,4,C}\}}\left(r + \delta Q_{1,4,D\{Q_{1,4,D} > Q_{1,4,C}\}} + \delta Q_{1,4,C\{Q_{1,4,D} < Q_{1,4,C}\}}\right). \tag{14}$$

Since we assume that both players are using the All-D strategy in this example, we can evaluate the indicator functions in (7) as follows:

$$1_{\{Q_{2,1,D} > Q_{2,1,C}\}} = 1,\ 1_{\{Q_{1,1,D} > Q_{1,1,C}\}} = 1,\ 1_{\{Q_{1,1,D} < Q_{1,1,C}\}} = 0, \tag{15}$$

$$1_{\{Q_{2,1,D} < Q_{2,1,C}\}} = 0,\ 1_{\{Q_{1,2,D} > Q_{1,2,C}\}} = 1,\ 1_{\{Q_{1,2,D} < Q_{1,2,C}\}} = 0. \tag{16}$$

So, (7) reduces to

$$Q_{1,1,D} = p + \delta Q_{1,1,D}. \tag{17}$$

By proceeding in the same way for (8)–(14), we obtain the following system of equations:

$$\begin{aligned}
Q_{1,1,D} &= p + \delta Q_{1,1,D},\\
Q_{1,1,C} &= s + \delta Q_{1,3,D},\\
Q_{1,2,D} &= p + \delta Q_{1,1,D},\\
Q_{1,2,C} &= s + \delta Q_{1,3,D},\\
Q_{1,3,D} &= p + \delta Q_{1,1,D},\\
Q_{1,3,C} &= s + \delta Q_{1,3,D},\\
Q_{1,4,D} &= p + \delta Q_{1,1,D},\\
Q_{1,4,C} &= s + \delta Q_{1,3,D}.
\end{aligned} \tag{18}$$

By solving the following equation, we get

$$Q_{1,1,D} = Q_{1,2,D} = Q_{1,3,D} = Q_{1,4,D} = \frac{p}{1-\delta}, \tag{19}$$

$$Q_{1,1,C} = Q_{1,2,C} = Q_{1,3,C} = Q_{1,4,C} = s + \frac{\delta p}{1-\delta}. \tag{20}$$

Since this is a symmetric strategy pair, the solutions to the Bellman equations for player 2 are the same.

To see if this is a valid solution or not, we check that the inequalities of the assumed strategy pair are satisfiable by the model parameters. In this case, the inequalities require that

$$Q_{a,i,D} > Q_{a,i,C} \text{ for all } a \in \{1,2\} \text{and } i \in \{1,2,3,4\}, \tag{21}$$

so that we must have

$$\frac{p}{1-\delta} > s + \frac{\delta p}{1-\delta}, \tag{22}$$

or equivalently

$$p(1-\delta) > s(1-\delta). \tag{23}$$

This is always satisfied since $p > s$ for the PD. We conclude that the All-D strategy is the best response against an opponent playing the All-D strategy. The best response, given an opponent's strategy, as calculated by solving a linear system of equations such as (18), will be unique unless the payoff parameters lead to a solution with $Q_{a,i,D} = Q_{a,i,C}$ for some a and i. We will neglect such cases, as these occur in a vanishingly small part of the parameter space.

3.2. Best-Response Networks.

We can now construct two types of direct networks for each choice of the parameters t, r, p, s, and δ from the pure strategy best-responses. The first type we call best-response networks (BRNs) and they are constructed as follows: We convert each of the 16 strategies into a number: $(0,0,0,0) \longrightarrow 0$, $(0,0,0,1) \longrightarrow 1$, etc (to do this consistently, we use the binary representation of numbers). These become the labels for the nodes of our graph, and we draw a directed edge from each strategy to its best response.

The second type of networks we call mutual best-response networks (MBRNs). For these, we convert the strategy pairs into numbers so that $(0,0,0,0,0,0,0,0) \longrightarrow 0, (0,0,0,0,0,0,0,1) \longrightarrow 1,$ etc. Now we draw a directed edge from a strategy pair to the strategy pair that contains the best response to player one's strategy in the last four entries and the best response to player two's strategy in the first four entries. The MBRN encodes the mutual best-response dynamics, where in each round each player switches their strategy to the strategy that is a best response to their opponent's previous strategy (see [56]).

In this article, we are interested in studying the limiting dynamics of the mutual best-response dynamics. We will see that the restriction to pure strategies means that the limiting dynamics can consist of absorbing strategy pairs (strategy pairs in which both players play the best response to their opponent's strategy) and limiting cycles in which both players cycle through a sequence of strategies.

Absorbing states for the mutual best-response dynamics will appear as nodes with self-loops in the MBRN. These absorbing states correspond to Nash equilibria for the dynamics. Symmetric absorbing states will appear as self-loops in the BRN as well, but nonsymmetric absorbing states will appear as reciprocal edges. In addition to the absorbing states, the MBRN will exhibit limit cycles consisting of strategy pairs in which each player plays a strategy that is part of an absorbing state but not the same absorbing state. In this situation, the mutual best-response dynamics dictate that the players will switch to the opponent's strategy at each round. This is an example of the miscoordination learning pathology discussed in [3]. Our focus in this article will be on absorbing states because once all the absorbing states are known, all the limit cycles can be constructed (each pair of distinct absorbing states gives rise to a limit cycle by combining the strategy of player one of the first absorbing strategy pair with the strategy of player two of the second absorbing strategy pair).

3.3. Classification of Strategy Pairs. In this section, we introduce two classifications that allow us to identify the types of absorbing strategy pairs. The first classification is based on the type of behaviour the strategy pair gives rise to, and the second classification is based on the symmetries or antisymmetries in the strategy pair.

3.3.1. From Strategies to Actions. Knowing which strategy pair is being played is still not necessarily informative as to which actions will be taken by the algorithms. As an example, consider the symmetric strategy pair of both players using the TFT strategy with exploration. If the system starts in the CC or DD state, it will remain there until one of the players explores and picks D or C, respectively. Then the system will oscillate between the DC and CD states until one of the players again explores to synchronize their action with that of the opponent. This means that the state graph of the game has three disconnected components. Transitions between components occur only due to exploration.

The vector for this state is given by $(0, 1, 0, 1, 0, 1, 0, 1)$ (note that states 2 and 3 require opposite responses from players 1 and 2). The transitions between states of the game when both players play a pure strategy are shown in Figure 1.

We can include the transitions due to exploration as follows: Let the probability that no one explores be $(1 - \epsilon)(1 - \epsilon) = A$, the probability that player 1 explores be $\epsilon(1 - \epsilon) = B$ (this is the same as the probability that player 2 explores), and the probability that both explore be $\epsilon^2 = C$. Then the transition (stochastic) matrix for the game with the symmetric TFT strategy pair and ϵ-greedy exploration is given by

$$P = \begin{pmatrix} A + B + \frac{C}{4} & \frac{B}{2} + \frac{C}{4} & \frac{B}{2} + \frac{C}{4} & \frac{C}{4} \\ \frac{B}{2} + \frac{C}{4} & \frac{C}{4} & A + B + \frac{C}{4} & \frac{B}{2} + \frac{C}{4} \\ \frac{B}{2} + \frac{C}{4} & A + B + \frac{C}{4} & \frac{C}{4} & \frac{B}{2} + \frac{C}{4} \\ \frac{C}{4} & \frac{B}{2} + \frac{C}{4} & \frac{B}{2} + \frac{C}{4} & A + B + \frac{C}{4} \end{pmatrix}. \tag{24}$$

The eigenvector of P with eigenvalue 1 will give the stationary distribution on the game states and therefore tell us how much time is spent in each state. For this matrix, the stationary distribution is the uniform distribution on the states, i.e., $(1/4, 1/4, 1/4, 1/4)$. This means that the system of two players playing TFT spends equal amounts of time at each state.

A similar calculation yields the following stationary distribution for the symmetric grim trigger strategy pair:

$$\left(1 - \frac{(5 - 2\epsilon)\epsilon}{4} \quad \frac{(2 - \epsilon)\epsilon}{4} \quad \frac{(2 - \epsilon)\epsilon}{4} \quad \frac{\epsilon}{4}\right). \tag{25}$$

In the limit as ϵ goes to zero, this is $(1, 0, 0, 0)$, which shows that even though cooperation is possible in the GT symmetric strategy pair when exploration is used, it vanishes in the small exploration rate limit.

By calculating the stationary distributions of the pure strategy pairs in the small exploration rate limit, we can identify which states of the game the learners will spend the most time in, giving us an idea of the actions used and the expected payoff. Using this, we can classify the strategy pairs by the resulting stationary distributions. In our case, we identify an absorbing strategy pair as being conducive to cooperation (CC) or not (NCC). A strategy pair is conducive to cooperation when the strategy pair assigns positive probability to being in the (C, C) state in the small exploration rate limit. This provides an avenue for studying the likelihood of cooperation under pure mutual best-response dynamics.

3.3.2. Structural Symmetry. We find that all absorbing states in the three games we consider fall into one of the following four categories:

Symmetric (Sym). The players play the same action in all states. An example is $(0, 1, 1, 0, 0, 1, 1, 0) = 102$.

Complemented Middle (CM). The players play the same action in the states (0,0) and (1,1), but complimentary

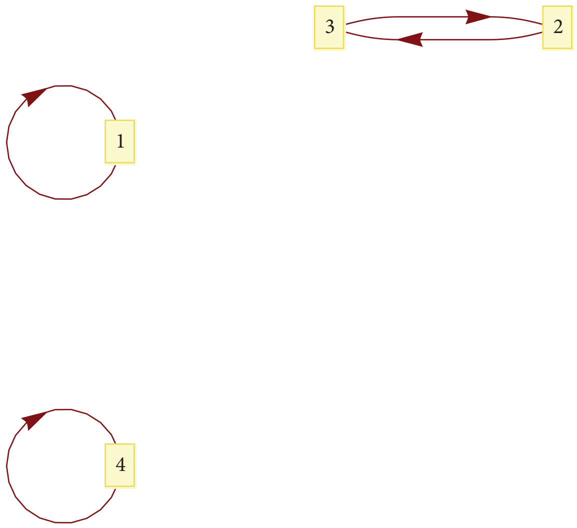


Figure 1: Game state transitions of the symmetric TFT pure strategy pair. Here, the states the game can be in are labelled with 1 corresponding to the DD state, 2 corresponding to the DC state, 3 corresponding to the CD state and 4 corresponding to the CC state.

actions in the states (0,1) and (1,0). An example is $(0,0,1,0,0,1,0,0) = 36$.

Complemented Sides (CS). The players play the same action in the states (0,1) and (1,0), but complimentary actions in the states (0,0) and (1,1). An example is $(0,0,1,1,1,0,1,0) = 58$.

Complemented (Com). The players play opposite actions in all states. An example is $(0,1,1,1,1,0,0,0) = 120$.

4. Results

In this section we collect the critical conditions at which the BRNs change, give examples of the resulting networks, and identify the absorbing state of the mutual best-response dynamics. We do this for the prisoner's dilemma, the stag hunt, and the hawk-dove games, each of which corresponds to a different ordering of the model parameters T, R, P, and S . Given the ordering, we can obtain the sign of the dilemma strength parameters. These are defined as follows:

$$\begin{aligned} D_g &= \frac{T-R}{R-P}, \\ D_r &= \frac{P-S}{R-P}, \end{aligned} \tag{26}$$

where the first is the relative gain from defecting against a cooperator, and the second is the relative risk of cooperating against a defector. For the PD game we thus have that $D_g, D_r > 0$, for the SH game we have $D_g < 0$ and $D_r > 0$, and for the HD game we have $D_g > 0$ and $D_r < 0$. Without loss of generality and for ease of exposition, we will normalize the games by setting $\max\{T, R, P, S\} = 1$ and $\min\{T, R, P, S\} = 0$ when plotting the phase diagrams.

4.1. Prisoner's Dilemma. Using the procedure outlined in Section 3.1, we find that the symmetric absorbing states found by [54] are the only possible absorbing states. This means that there are no nonsymmetric strategy pairs that solve the Bellman equations in this setting.

The only possible pure strategy solutions pairs are the symmetric all defect (All-D) strategy represented by node 0, the grim trigger (GT) strategy represented by node 17, and the win-stay-lose-shift (WSLS) strategy represented by node 153 (the last strategy is also known as the one-period-punishment strategy or the Pavlov strategy). The All-D strategy is always possible, while there are restrictions on the values of the model parameters, determining when the other two are possible. This is summarized in Table 1.

To illustrate what the conditions in Figure 2 the parameters imply, we plot the phase diagram in the normalized PD (i.e., $T = 1$ and $S = 0$) in Figure 2. As expected, we see that increasing R (the parameter controlling how much the players earn when in (C, C)) increases the number of possible solutions. Decreasing P has a similar effect, but there is a critical value for R that dictates whether it is possible to reach a region in which the GT or WSLS strategy pairs exist by decreasing P or not. We show how the regions change as a function of δ in Appendix A.

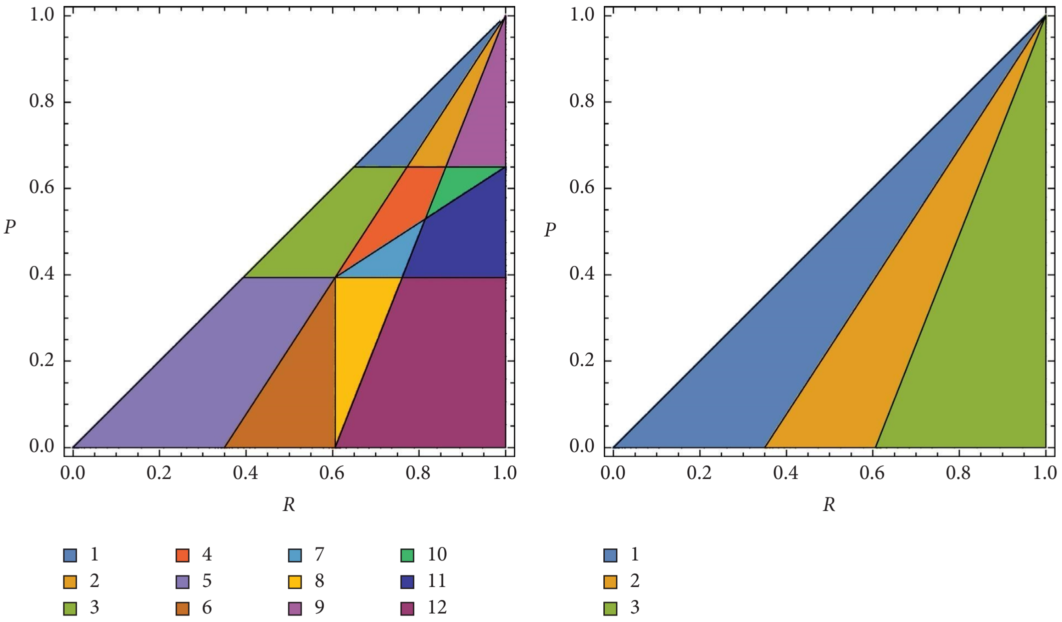
The critical conditions in Table 1 indicate when changes in the BRN lead to the appearance or disappearance of equilibria. In Table 2 we show all critical conditions at which changes in the BRN occur. These results give two extremes in the resolution of the phase diagram: the first is the lowest resolution and the second is the highest. There is an intermediate resolution in which we identify the critical conditions at which changes in the BRN graph lead to changes in the basin of attraction of the equilibria. We leave the study of the basin of attraction of the absorbing states for future work.

From Table 2 we can infer the number of BRNs that are possible. Since some conditions are repeated (e.g., the conditions for nodes 9 and 13), we find that there are 12 distinct BRNs; i.e., there are twelve regions in the phase space. We show the phase diagram for these 12 graphs in the left panel of Figure 2. In Figure 3 we show a graph containing all possible edges in the BRRNs. We see all three of the possible equilibria as self-loops (node 0 for All-D, 1 for GT, and 9 for WSLS).

4.2. Stag Hunt. In the case of the stag hunt game, the number of equilibria increases significantly, which means that a simple assessment of the resulting phase diagram is not feasible. We see in Table 3 that the SH gives rise to a total of 16 possible absorbing states, of which half are symmetric and half are nonsymmetric. The nonsymmetric absorbing states are all of the CM type defined in Section 3.3. We also find that almost 70% of the absorbing states are conducive to cooperation. For the nonsymmetric strategy pairs, each combination of strategies appears twice in Table 3, as the two distinct strategies can be assigned to the two players in two ways. This means that there are 12 different strategy combinations that can give rise to an absorbing strategy pair.

Table 1: All strategy pairs solve the Bellman equations self-consistently for the prisoner's dilemma, their structural and behavioural types, the conditions for their existence, and the regions of Figure 2 in which they exist.

Policy-pair	Type	Conditions	Regions
$(0,0,0,0,0,0,0,0)$	Sym, NCC	Always	$1-12$
$(0,0,0,1,0,0,0,1)$	Sym, NCC	$R+\delta T>T+\delta P$	$2,4,6-12$
$(1,0,0,1,1,0,0,1)$	Sym, CC	$R+\delta R>T+\delta P$	$9-12$



(a)

(b)

Figure 2: (a) Phase diagram for the possible BRNs in the prisoner's dilemma, (b) phase diagram for the possible equilibria in the prisoner's dilemma (only node 0 in region 1, nodes 0 and 1 in region 2, and nodes 0, 1, and 9 in region 3). For both plots, we have $S=0$, $T=1$, and $\delta=0.65$.

Table 2: Critical conditions for the existence of edges in the pure best-response graph of the prisoner's dilemma.

Edge	Conditions
$0\longrightarrow 0$	None
$1\longrightarrow 0$	$R+\delta T<T+\delta P$
$1\longrightarrow 1$	$R+\delta T>T+\delta P$
$2\longrightarrow 0$	None
$3\longrightarrow 0$	None
$4\longrightarrow 0$	$S+\delta T<P(1+\delta)$
$4\longrightarrow 13$	$S+\delta T>P(1+\delta)$
$5\longrightarrow 0$	$(P+R\leq S+T\wedge S+\delta T<P(1+\delta))\vee(P+R>T+S\wedge R+\delta T<T+P\delta)$
$5\longrightarrow 3$	$P+R>T+S\wedge T-R/T-P<\delta<P-S/R-S$
$5\longrightarrow 12$	$P+R<T+S\wedge P-S/T-P<\delta<T-R/R-S$
$5\longrightarrow 15$	$(P+R\leq S+T\wedge R(1+\delta)>T+\delta S)\vee(P+R>T+S\wedge S+\delta R>P+\delta S)$
$6\longrightarrow 0$	$S+\delta T<P+\delta S$
$6\longrightarrow 9$	$S+\delta T>P+\delta S$
$7\longrightarrow 0$	$S+\delta T<P+\delta S$
$7\longrightarrow 8$	$S+\delta T>P+\delta S$
$8\longrightarrow 0$	None
$9\longrightarrow 1$	$(1+\delta)R<T+\delta P$
$9\longrightarrow 9$	$(1+\delta)R>T+\delta P$
$10\longrightarrow 0$	None
$11\longrightarrow 0$	None
$12\longrightarrow 0$	None
$13\longrightarrow 0$	$(1+\delta)R<T+\delta P$
$13\longrightarrow 11$	$(1+\delta)R>T+\delta P$
$14\longrightarrow 0$	None
$15\longrightarrow 0$	None

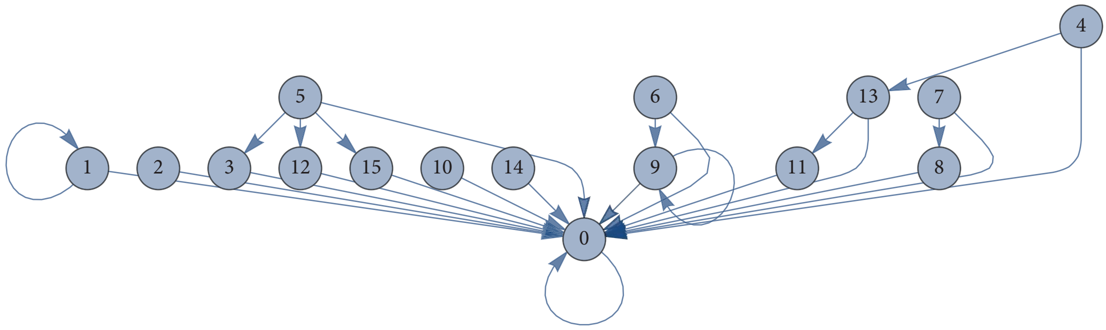


Figure 3: All possible edges in the BRN of the prisoner's dilemma. See Table 2 for conditions for the existence of edges.

Table 3: All strategy pairs solving the Bellman equations self-consistently for the stag hunt game, their structural and behavioural types, the conditions for their existence, and the regions of Figure 4 in which they exist.

Policy-pair	Type	Conditions	Regions
(0, 0, 0, 0, 0, 0, 0, 0)	Sym, NCC	Always	1 – 12
(0, 0, 0, 1, 0, 0, 0, 1)	Sym, NCC	Always	1 – 12
(1, 0, 0, 0, 1, 0, 0, 0)	Sym, CC	Always	1 – 12
(1, 0, 0, 1, 1, 0, 0, 1)	Sym, CC	Always	1 – 12
(1, 1, 1, 1, 1, 1, 1, 1)	Sym, CC	Always	1 – 12
(0, 1, 1, 1, 0, 1, 1, 1)	Sym, CC	$P + \delta S > S + \delta R$	1, 5, 6
(1, 1, 1, 0, 1, 1, 1, 0)	Sym, CC	$R + \delta P > T + \delta T$	5, 7 – 12
(0, 1, 1, 0, 0, 1, 1, 0)	Sym, NCC	$P + \delta P > S + \delta R \land R + \delta P > T + \delta R$	6, 8
(1, 0, 1, 1, 1, 1, 0, 1)	CM, CC	Always	1 – 12
(1, 1, 0, 1, 1, 0, 1, 1)	CM, CC	Always	1 – 12
(0, 0, 1, 1, 0, 1, 0, 1)	CM, CC	$P + \delta S > S + \delta R$	1, 5, 6
(0, 1, 0, 1, 0, 0, 1, 1)	CM, CC	$P + \delta S > S + \delta R$	1, 5, 6
(0, 1, 0, 0, 0, 0, 1, 0)	CM, NCC	$P + \delta P > S + \delta R \land R + \delta P > T + \delta R$	6, 8
(0, 0, 1, 0, 0, 1, 0, 0)	CM, NCC	$P + \delta P > S + \delta R \land R + \delta P > T + \delta R$	6, 8
(1, 0, 1, 0, 1, 1, 0, 0)	CM, CC	$R + \delta P > T + \delta T$	5, 7 – 12
(1, 1, 0, 0, 1, 0, 1, 0)	CM, CC	$R + \delta P > T + \delta T$	5, 7 – 12

Table 4 shows some similarities to Table 2 with many critical conditions being identical when exchanging T and R. This similarity is confirmed by the left panel of Figure 4 where we see the same 12 regions appearing as in the left panel of Figure 2. There is, however, a difference in the regions that are relevant for changes in the set of absorbing states, as seen by comparing the right panels of Figures 2 and 4.

In Figure 5 we see that the network of possible edges in the BRN also differs from the PD case. More specifically, both games have the same number of possible edges (25), but the maximum in-degree of the PD is 16 while the maximum in-degree of the SH is 5. We plot the in-degree distribution in Figure 6.

4.3. Hawk-Dove. For the hawk-dove game, we again find many equilibria, as in the SH game. We see in Table 5 that the HD gives rise to a total of 17 possible absorbing states, of which 16 are not conducive to cooperation. The absorbing state that is CC is the symmetric WSLS, which is also an absorbing state for both of the other games. Half of the remaining absorbing states are C, and the other half are CS. For the nonsymmetric strategy pairs, each combination of strategies appears twice in Table 3. This means that there are nine distinct combinations of strategies that can form an absorbing strategy pair in the Hawk-Dove game.

Table 6 again shows some similarities to Table 2 with many critical conditions being identical when exchanging S and P. Once again, we find the 12 regions seen in the left panel of Figure 2 in the phase diagram in Figure 7. In this case, all regions are relevant for changes in the set of absorbing states.

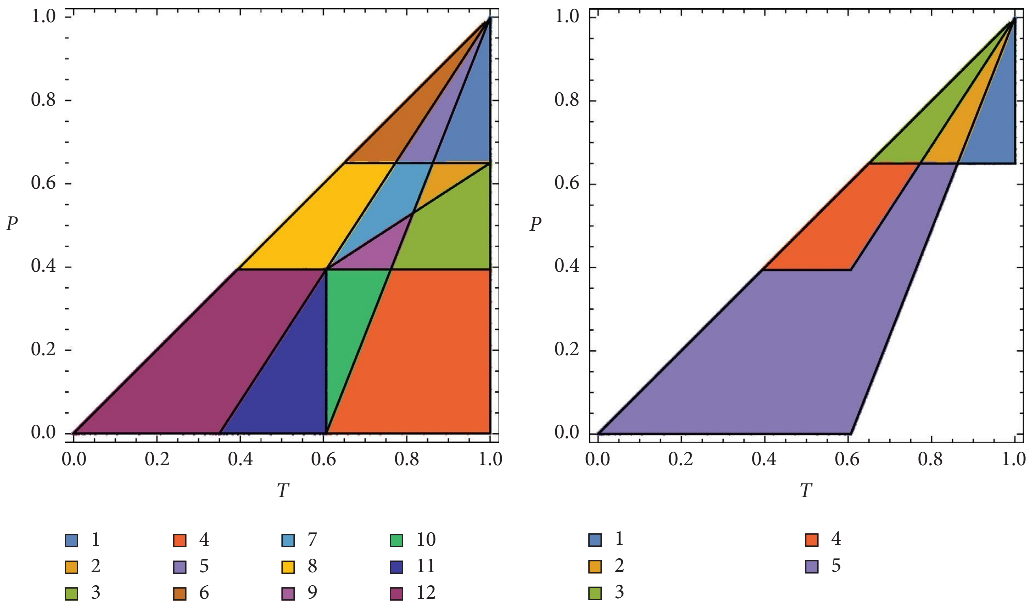
In Figure 8 we see that the network of possible edges in the BRN for the HD game is similar to the network for the SH. More specifically, we see that the maximum in-degree in the HD is also 5. Interestingly, the nodes realizing this maximum are the same for both games (nodes 0 and 15; i.e., the All-D strategy and the strategy pair where both players always cooperate) with the difference that they have a self-loop in the SH but do not have a self-loop in the HD game. In addition, the in-degree distributions for the SH and HD games are identical.

5. Sample Batch Q-Learning

The learning dynamics we have discussed thus far have been framed predominantly in terms of mutual best-response dynamics. In this section, we show that these dynamics are the limiting dynamics of sample batch Q-learning when

Table 4: Critical conditions for the existence of edges in the BRN of the stag hunt.

Edge	Conditions
$0 \longrightarrow 0$	None
$1 \longrightarrow 1$	None
$2 \longrightarrow 0$	$T + \delta R > R + \delta P$
$2 \longrightarrow 4$	$T + \delta R < R + \delta P$
$3 \longrightarrow 5$	None
$4 \longrightarrow 2$	$\delta R + S < \delta (1 + P)$
$4 \longrightarrow 15$	$\delta R + S > \delta (1 + P)$
$5 \longrightarrow 3$	$\delta R + S < P + \delta S$
$5 \longrightarrow 15$	$\delta R + S > P + \delta SP$
$6 \longrightarrow 0$	$P + T > R + S \wedge R - T/R - P < \delta < P - S/T - S$
$6 \longrightarrow 6$	$(P + T < T + S \wedge S + \delta < P(1 + \delta)) \vee P + T \geq R + S \wedge T + \delta R < R + \delta P$
$6 \longrightarrow 9$	$(P + T \leq R + S \wedge R = \delta S < T(1 + \delta)) \vee P + T > R + S \wedge S + \delta T < P + \delta S$
$6 \longrightarrow 15$	$P + T < R + S \wedge P - S/R - P < \delta < R - T/T - S$
$7 \longrightarrow 7$	$\delta R + S < P + \delta S$
$7 \longrightarrow 15$	$\delta R + S > P + \delta S$
$8 \longrightarrow 8$	None
$9 \longrightarrow 9$	None
$10 \longrightarrow 0$	$(1 + \delta)T > R + \delta P$
$10 \longrightarrow 12$	$(1 + \delta)T < R + \delta P$
$11 \longrightarrow 13$	None
$12 \longrightarrow 10$	None
$13 \longrightarrow 11$	None
$14 \longrightarrow 0$	$(1 + \delta)T > R + \delta P$
$14 \longrightarrow 14$	$(1 + \delta)T < R + \delta P$
$15 \longrightarrow 0$	None



(a)

(b)

Figure 4: (a) Phase diagram for the possible BRNs in the stag hunt and (b) phase diagram for the possible equilibria in the stag hunt. For both plots, we have $S = 0, R = 1$, and $\delta = 0.65$.

taking the infinite batch size limit. We will use a simple sample batch algorithm so that it is simple to see that the infinite batch size limit gives rise to the mutual best-response dynamics, but more sophisticated sample batch algorithms exists which make better use of the information gathered during a batch (see [21, 57]).

The sample batch algorithm splits the time horizon into batches of size K and keeps track of two sets of Q-values. The first set, denoted by Q^{act}, will be used to determine the actions that the algorithm takes during a batch. These Q-values are kept constant during the batch. The second set, denoted by Q^{val}, will be used to learn during the batch. At the

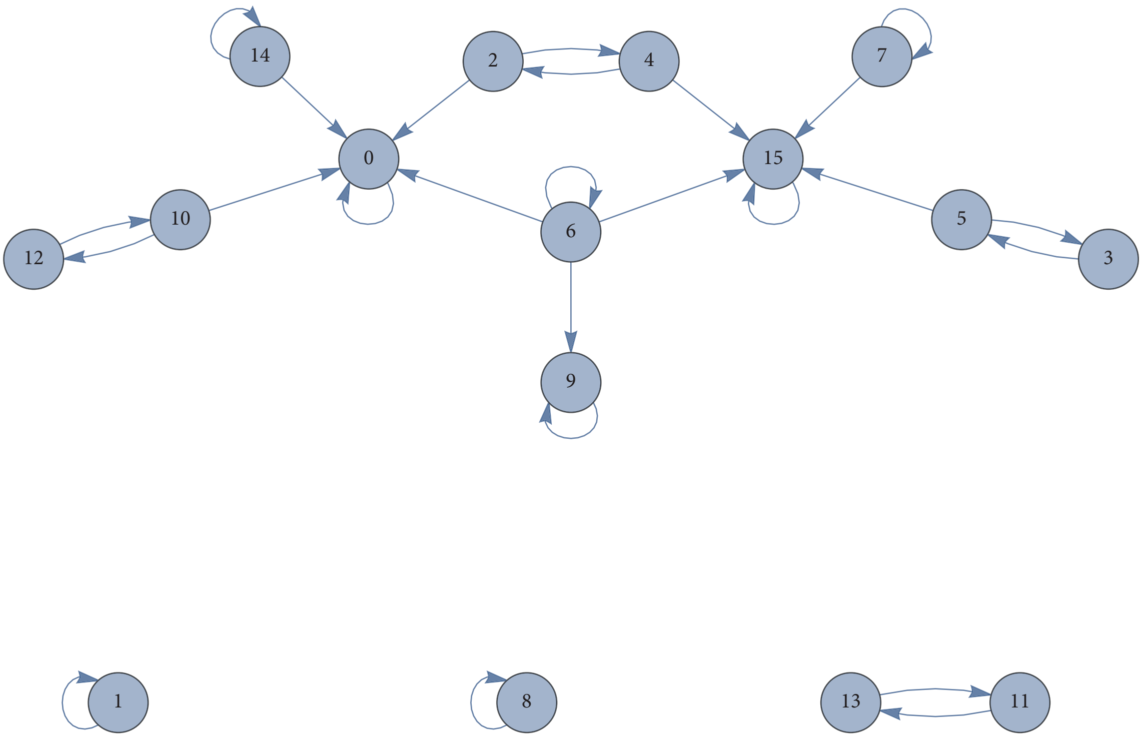


Figure 5: All possible edges in the BRN of the stag hunt. See Table 4 for conditions for the existence of edges.

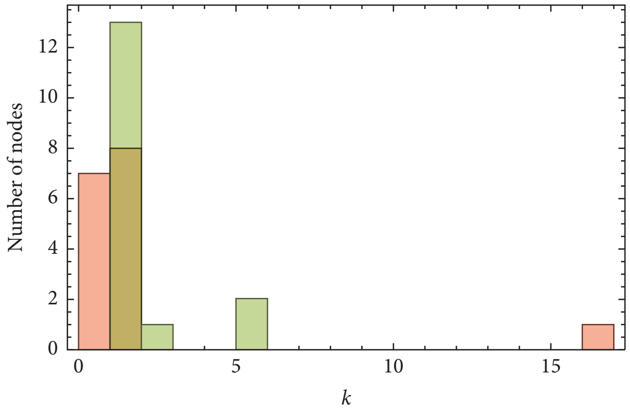


Figure 6: In-degree distribution of the BRN containing all possible edges for the PD (red) and the SH (green).

end of the batch, the first set is replaced by the second and the process is repeated. A diagram illustrating this process is given in Figure 9.

In order to ensure that all states are visited infinitely often in the infinite batch size limit, the algorithm uses an ϵ-greedy action-selection mechanism with a constant exploration rate depending on the batch size. This means that it selects the action with the maximum Q-value with probability $1 - \epsilon$ and selects an action uniformly at random with probability ϵ. Given a batch size K, we define the exploration rate to be

$$\epsilon(K) = \frac{1}{K^{(1-\gamma)/3}}, \tag{27}$$

with $\gamma > 0$ a small constant. The state-action pairs that are visited least are the actions with the smaller Q-value in the states that are only reached when both agents explore, for example, the (4, C) state-action pair when both players use the All-D strategy. Using this, we can calculate a lower bound for the expected number of times any state-action pair is visited in a batch of size K, given an exploration rate $\epsilon(K)$. The probability of reaching such a state-action pair is $\epsilon^3/8$ so that the expected number of visits in a batch of size K is

$$K \times \frac{\epsilon^3(K)}{8} = \frac{K^{\gamma}}{8}. \tag{28}$$

Table 5: All strategy pairs solving the Bellman equations self-consistently for the hawk-dove game, their structural and behavioural types, the conditions for their existence, and the regions of Figure 7 in which they exist.

Policy-pair	Type	Conditions	Regions
$(1,0,0,1,1,0,0,1)$	Sym, CC	$P+\delta R>S+\delta P \wedge R+\delta R>T+\delta P$	$4,5,9,10$
$(0,0,0,0,1,1,1,1)$	Com, NCC	Always	$1-12$
$(1,1,1,1,0,0,0,0)$	Com, NCC	Always	$1-12$
$(0,0,0,1,1,1,1,0)$	Com, NCC	$T+\delta S>R+\delta T$	$1,3,7$
$(1,1,1,0,0,0,0,1)$	Com, NCC	$T+\delta S>R+\delta T$	$1,3,7$
$(0,1,1,0,1,0,0,1)$	Com, NCC	$S+\delta S>P+\delta T \wedge T+\delta S>R+\delta T$	$3,7$
$(1,0,0,1,0,1,1,0)$	Com, NCC	$S+\delta S>P+\delta T \wedge T+\delta S>R+\delta T$	$3,7$
$(0,1,1,1,1,0,0,0)$	Com, NCC	$S+\delta S>P+\delta T$	$3,5-8,10-12$
$(1,0,0,0,0,1,1,1)$	Com, NCC	$S+\delta S>P+\delta T$	$3,5-8,10-12$
$(0,1,0,0,1,1,0,1)$	CS, NCC	$T+\delta S>R+\delta R$	$1-8$
$(1,1,0,1,0,1,0,0)$	CS, NCC	$T+\delta S>R+\delta R$	$1-8$
$(0,1,0,1,1,1,0,0)$	CS, NCC	$T+\delta S>R+\delta R$	$1-8$
$(1,1,0,0,0,1,0,1)$	CS, NCC	$T+\delta S>R+\delta R$	$1-8$
$(0,0,1,0,1,0,1,1)$	CS, NCC	$S+\delta T>P+\delta T$	$7,8,12$
$(1,0,1,1,0,0,1,0)$	CS, NCC	$S+\delta T>P+\delta T$	$7,8,12$
$(0,0,1,1,1,0,1,0)$	CS, NCC	$S+\delta T>P+\delta T$	$7,8,12$
$(1,0,1,0,0,0,1,1)$	CS, NCC	$S+\delta T>P+\delta T$	$7,8,12$

Table 6: Critical conditions for the existence of edges in the BRN of the hawk-dove game.

Edge	Conditions
$0 \longrightarrow 15$	None
$1 \longrightarrow 14$	$R+\delta T<T+\delta S$
$1 \longrightarrow 15$	$R+\delta T>T+\delta S$
$2 \longrightarrow 11$	None
$3 \longrightarrow 10$	None
$4 \longrightarrow 13$	None
$5 \longrightarrow 12$	$R(1+\delta)<T+\delta S$
$5 \longrightarrow 15$	$R(1+\delta)>T+\delta S$
$6 \longrightarrow 9$	None
$7 \longrightarrow 8$	None
$8 \longrightarrow 0$	$P+\delta T>S(1+\delta)$
$8 \longrightarrow 7$	$P+\delta T<S(1+\delta)$
$9 \longrightarrow 0$	$R+S<T+P \wedge S-P/T-S<\delta<T-R/R-P$
$9 \longrightarrow 6$	$(R+S<T+P \wedge P+\delta T<S(1+\delta)) \vee (R+S\geq T+P \wedge R+\delta T<T+\delta S)$
$9 \longrightarrow 9$	$(R+S\leq T+P \wedge T+\delta P<R(1+\delta)) \vee (R+S>T+P \wedge P+\delta R>S+\delta T)$
$9 \longrightarrow 15$	$R+S>T+P \wedge T-R/T-S<\delta<S-P/R-P$
$10 \longrightarrow 0$	$P+\delta T>\delta P+S$
$10 \longrightarrow 3$	$P+\delta T<\delta P+S$
$11 \longrightarrow 0$	$P+\delta T>\delta P+S$
$11 \longrightarrow 2$	$P+\delta T<\delta P+S$
$12 \longrightarrow 5$	None
$13 \longrightarrow 4$	$(1+\delta)R<T+\delta S$
$13 \longrightarrow 15$	$(1+\delta)R>T+\delta S$
$14 \longrightarrow 1$	None
$15 \longrightarrow 0$	None

Since this is a lower bound for the expected number of visits for all state-actions pairs, we see that all state-action pairs are visited infinitely often in the limit $K \longrightarrow \infty$.

During the batch, the second set of Q-values is updated using

$$\begin{aligned} Q^{\text{val}}_{a,\boldsymbol{\sigma},\sigma_a}(k+1) &= (1-\tilde{\alpha}(k))Q^{\text{val}}_{a,\boldsymbol{\sigma},\sigma_a}(k) \\ &\quad + \tilde{\alpha}(k)\left[r_a(\boldsymbol{\sigma}(k+1)) + \delta \max_{i\in C,D}\left\{Q^{\text{val}}_{a,\boldsymbol{\sigma}}(k+1), i\right\}\right], \end{aligned} \tag{29}$$

with

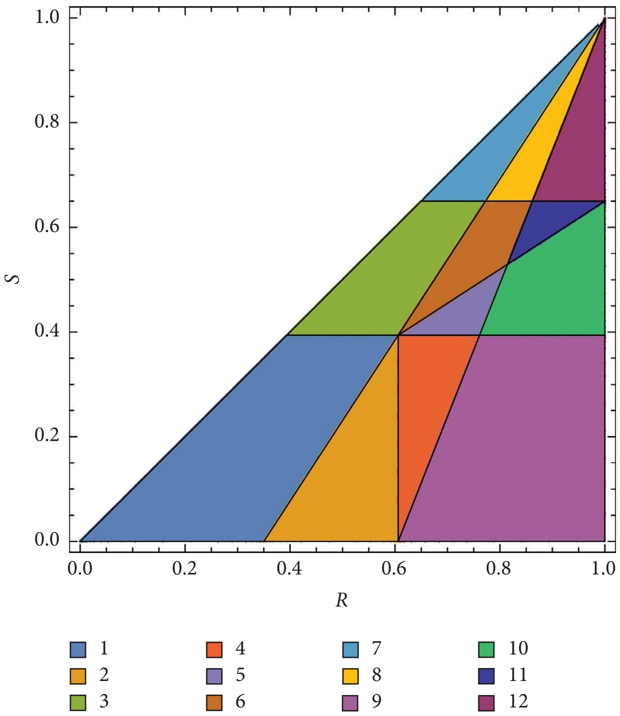


FIGURE 7: Phase diagram for the possible networks in the hawk-dove game with $P = 0, R = 1$, and $\delta = 0.65$.

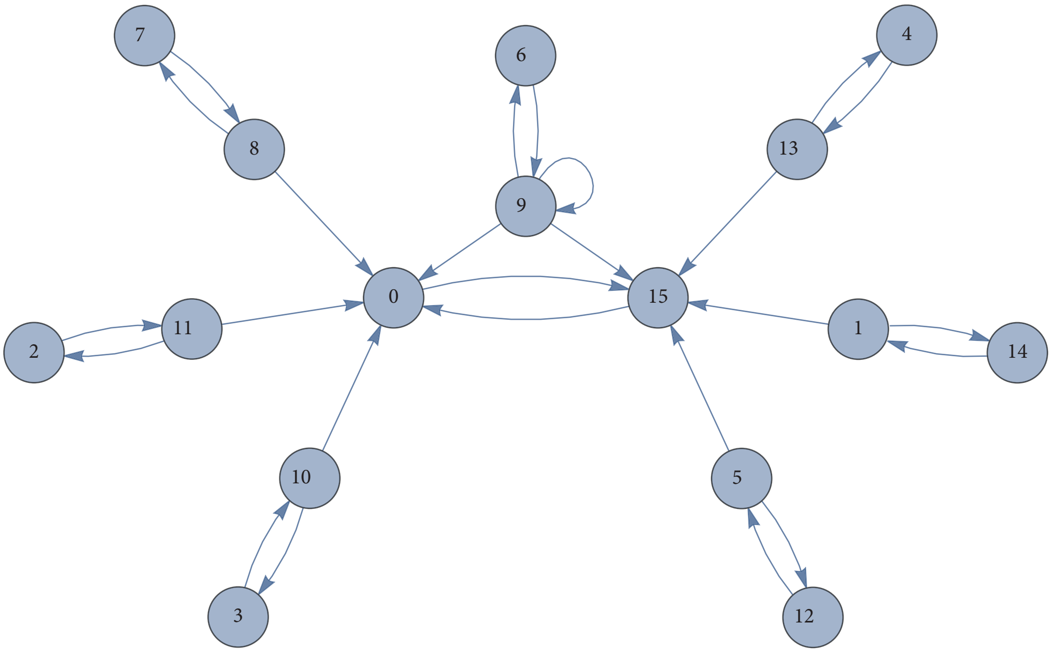


FIGURE 8: All possible edges in the BRNs of the hawk-dove game. See Table 6 for conditions for the existence of edges.

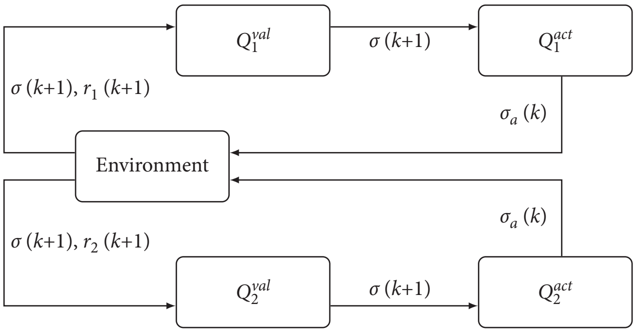
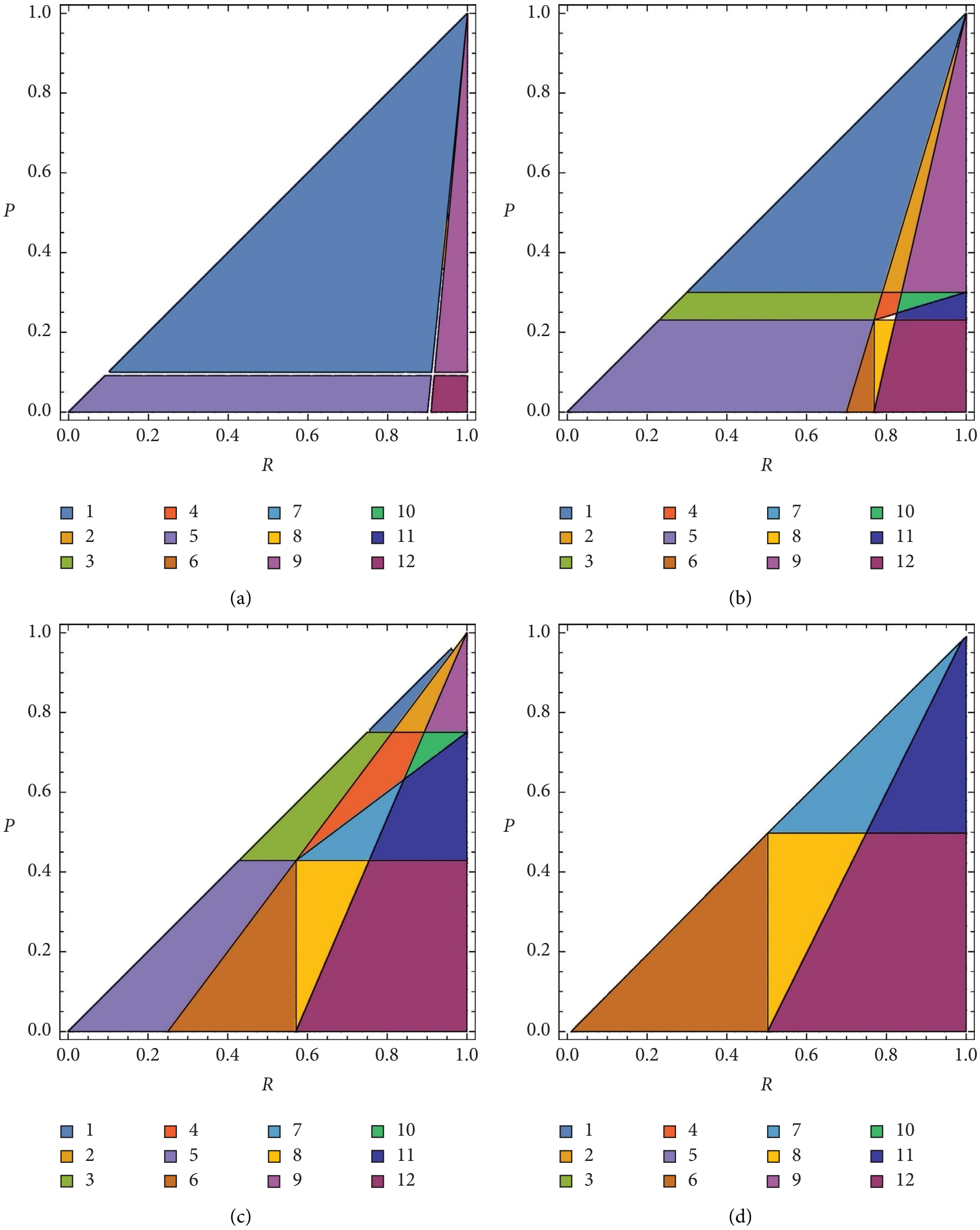


Figure 9: Diagram showing the structure of an interaction between the two players during a batch. The batch size is K so that $k \in \{1, \dots, K\}$ and Q_a^{act} remains fixed during the batch, while Q_a^{val} is updated.



(a)

(b)

(c)

(d)

Figure 10: Phase diagram for the possible BRNs in the prisoner's dilemma for different values of δ: 0.1 (a), 0.3 (b), 0.75 (c), and 0.99 (d).

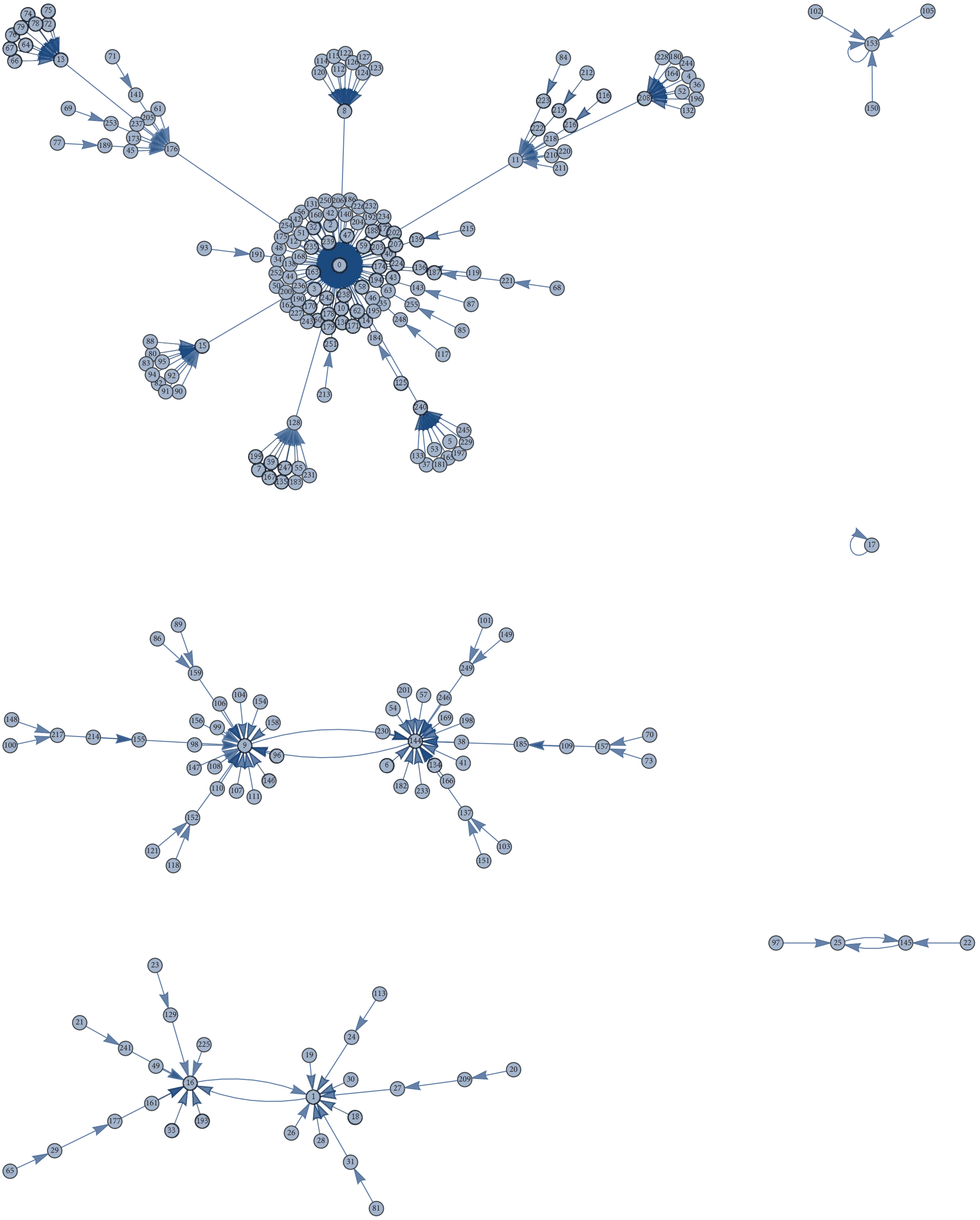


Figure 11: MBRN in the region 12 (in Figure 2) of the PD.

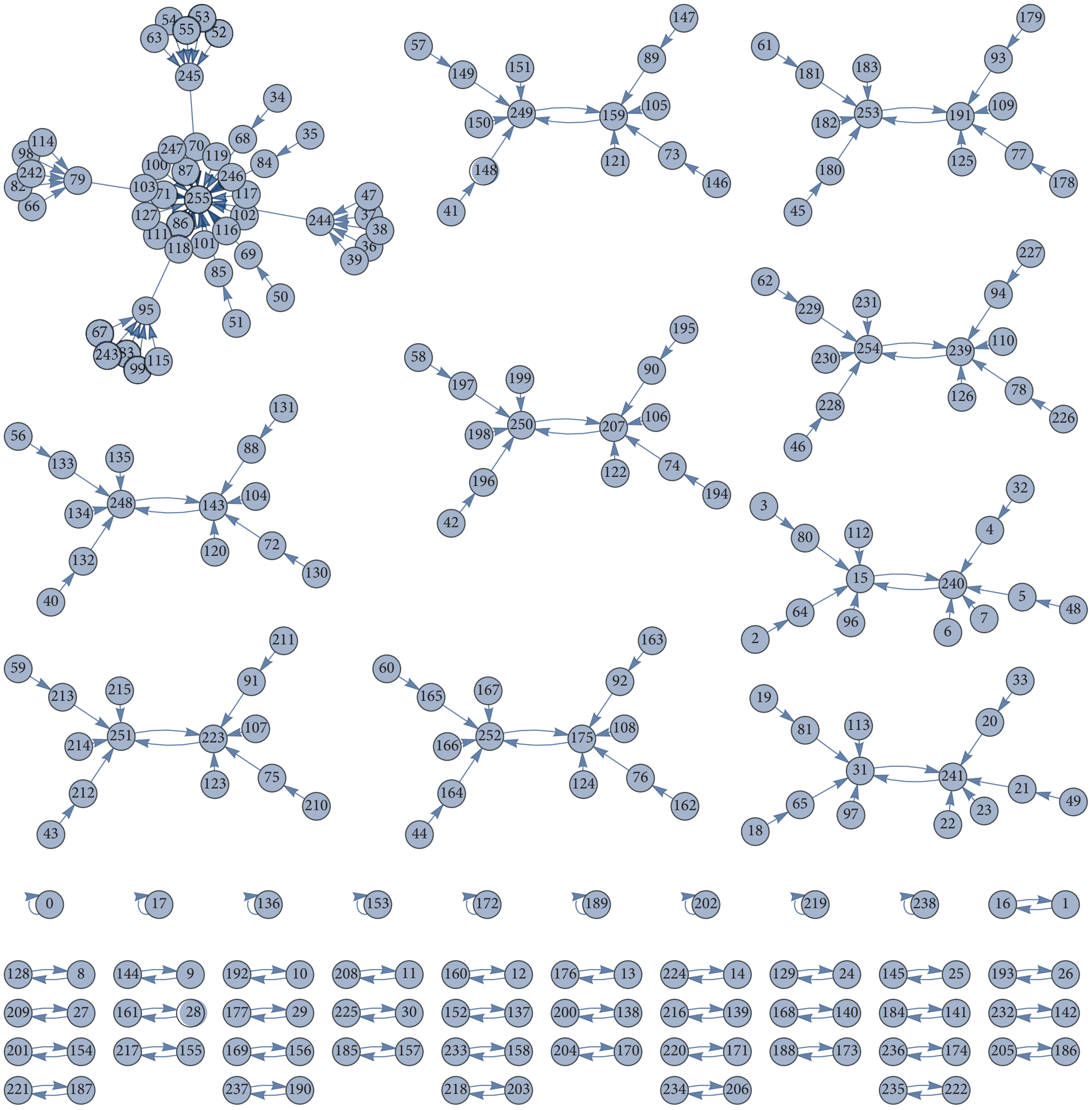


FIGURE 12: MBRN in the region 12 (in Figure 4) of the SH.

$$\tilde{\alpha}(k)=\begin{cases}\dfrac{1}{k_{\boldsymbol{\sigma},\sigma_a}(k)+1}, & \text{if } (\boldsymbol{\sigma},\sigma_a)=(\boldsymbol{\sigma}(k),\sigma_a(k)),\\[2ex] 0, & \text{otherwise},\end{cases} \tag{30}$$

where $k_{\sigma,\sigma_a}(k)$ is a local time, given by the number of times that the state-action pair (σ,σ_a) has been visited up to period k. At the end of a batch, the Q-values used for determining the actions are updated as follows:

$$Q^{\text{act}}_{a,\boldsymbol{\sigma},\sigma_a}=Q^{\text{val}}_{a,\boldsymbol{\sigma},\sigma_a}(K), \tag{31}$$

for all (a,σ,σ_a). This update is equivalent to performing an update using a learning rate of $\alpha=1$ for the postbatch update.

The Q-value update during the batch defined as above satisfies the conditions for convergence given in [17], since the environment in which the updates are made is stationary. In the infinite batch size limit, the Q-values are updated to the solution of the Bellman equations given the opponent's strategy, as calculated in Section 3.1. We conclude that, if the initial Q-values of the algorithm are ordered in each state (so that the optimal strategy based on the Q-values is a pure strategy), it will follow the mutual best-response dynamics represented by the MBRNs.

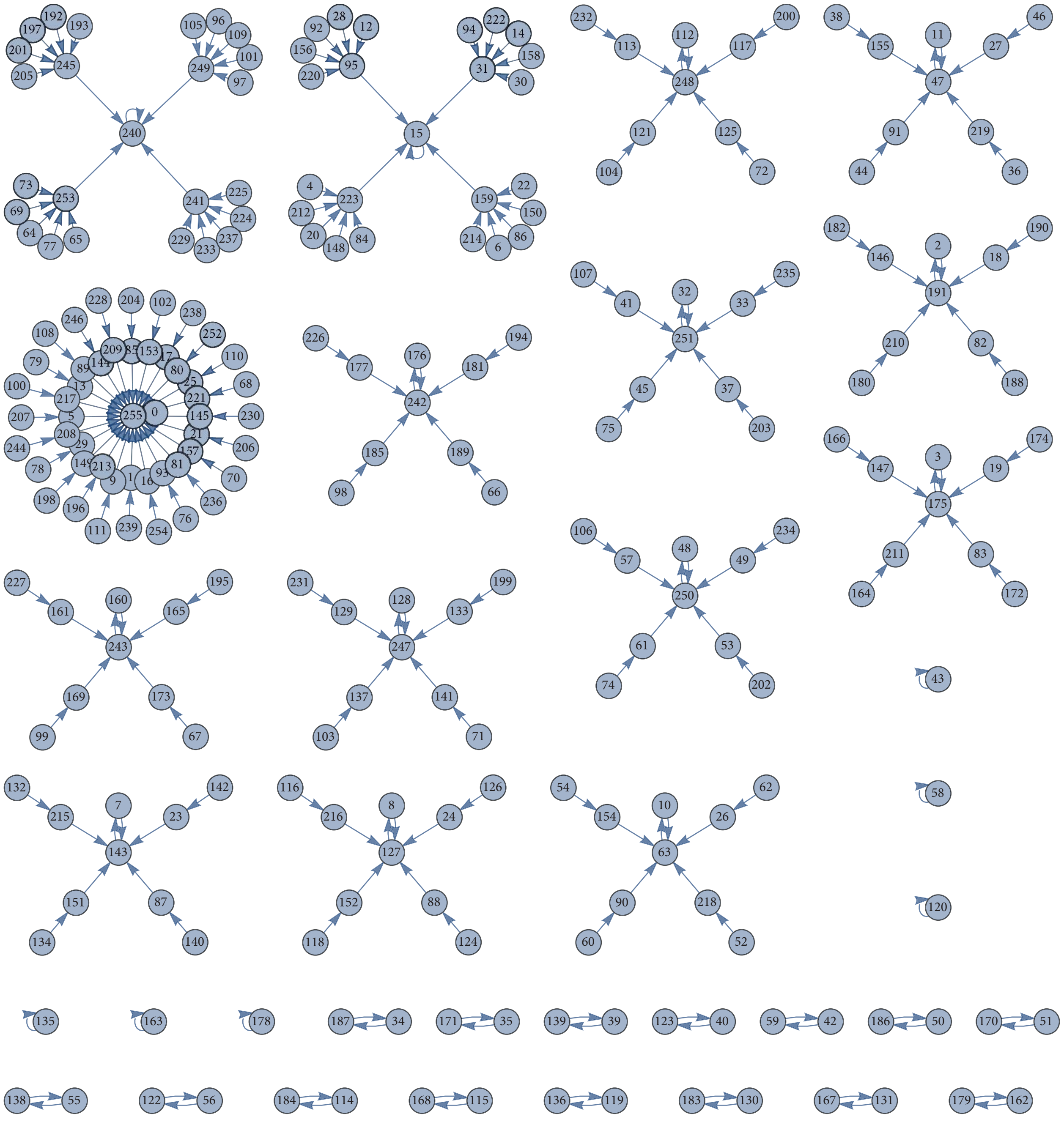


FIGURE 13: MBRN in the region 12 (in Figure 7) of the HD game.

TABLE 7: Conditions for the regions arising in the PD.

Region	Conditions
1	$\delta < P \wedge \delta + R < 1 + \delta P$
2	$\delta < P \wedge \delta + R > 1 + \delta P \wedge R(1+\delta) > 1 + \delta P$
3	$\delta > P \wedge \delta < P(1+\delta) \wedge \delta + R < 1 + \delta P$
4	$\delta > P \wedge 1 + \delta P < \delta + R \wedge \delta R < P \wedge 1 + \delta P > R(1+\delta)$
5	$P(1+\delta) < \delta \wedge 1 + \delta P > R + \delta$
6	$1 + \delta P < R + \delta \wedge R(1+\delta) < 1$
7	$\delta < P(1+\delta) \wedge R(1+\delta) < 1 + \delta P \wedge \delta R > P$
8	$R(1+\delta) < 1 + \delta P \wedge \delta > P(1+\delta) \wedge \delta + R > 1 + \delta P \wedge (P + R > 1 \vee R(1+\delta) > 1)$
9	$\delta < P \wedge 1 + \delta P < R(1+\delta) \wedge \delta R < P$
10	$\delta > P \wedge 1 + \delta P < R(1+\delta) \wedge \delta R < P$
11	$\delta > P \wedge \delta < P(1+\delta) \wedge \delta R > P \wedge R(1+\delta) > 1 + \delta P$
12	$1 + \delta P < R(1+\delta) \wedge P(1+\delta) < \delta$

6. Conclusion

6.1. Contributions. The MBRNs developed in this article are a network representation of the mutual best-response dynamics in symmetric two-player, two-action repeated games played by players with a one-period memory. We identify all the edges that are possible in the BRNs and provide the conditions for their existence. This allows us to plot the phase diagram of the game, where changes in phase are identified with changes in the network structure. We find that the phase diagrams are identical for the three games we consider here, with the only difference being the network structure associated with each region.

The limiting dynamics of the mutual best-response dynamics fall into one of two categories: absorbing states and limit cycles. Our focus is on absorbing states for the reason given in footnote 10. We restrict the phase diagram to only show critical boundaries at which a change in the set of absorbing states occurs. The similarity between the three games observed in the full-phase diagram is lost under this restriction.

For the PD, we have shown that the three symmetric solutions to the Bellman equations identified by [54] (All-D, GT, and WSLS) are the only absorbing strategy pairs possible. There are a total of 16 possible absorbing strategy pairs in the SH game, the majority of which are conducive to cooperation. In the HD game, there are 17 possible absorbing strategy pairs, almost all of which are not conducive to cooperation. The absorbing strategy pairs are all symmetric in the PD. In the SH, half are symmetric, and the other half are complimented in the asymmetric states. The HD game has only one symmetric absorbing strategy pair, with the rest being complimented either completely or in the symmetric states.

The WSLS strategy pair is the only absorbing strategy pair possible in all three games. It is always an absorbing strategy pair for the SH game, and the fraction of the parameter space in which it is possible for the PD and HD games increases as the discount factor increases (see Figure 10). The TFT strategy, on the other hand, does not appear as an absorbing strategy pair in any of the three games. This is in agreement with the result of [58], showing that the TFT strategy is outperformed by the WSLS strategy in the PD. This suggests that the WSLS strategy pair is a good learning goal in social-dilemma type games in which the players care about future outcomes.

The mutual best-response dynamics are not only useful for identifying absorbing strategy pairs but are also the (deterministic) limiting dynamics of sample batch Q-learning with infinite batch size. The MBRNs thus also correspond approximately to the dynamics of sample batch Q-learning with large but finite batch sizes. By studying the basin of attraction for the absorbing states in the MBRNs, we can approximate the probability of observing different learning outcomes (e.g., cooperation) when using such algorithms (assuming that the initial strategy is sampled uniformly at random from the strategy space) as a function of the model parameters. We leave such an analysis for future work.

6.2. Extensions. The representation developed here lends itself to extensions such as considering two period memory algorithms, three action games, nonsymmetric games, more than two players, or placing alternative conditions on the model parameters. Note, however, that the computational complexity of the method we propose here limits its immediate application to realistic environments (the number of equations in the linear system of Bellman equations that must be solved given a strategy grows as $|\mathscr{A}|^{N+1}$, where $|\mathscr{A}|$ is the number of actions available to the players and N is the number of players. This can be solved in $\mathcal{O}(|\mathscr{A}|^{3N+3})$, but this has to be repeated for all strategies. The number of strategies grows as $|\mathscr{A}|^{|\mathscr{A}|^N}$. The computational complexity thus grows exponentially in both the number of actions and the number of players). As the goal here is to understand multiagent reinforcement learning dynamics in simple environments, this should not be regarded as a relevant limitation.

Another direction is the extension of the results to study how the networks change when considering a smaller learning rate, i.e., where the sample batch algorithms move in the direction of the best response instead of playing the best response immediately or including a constant ϵ-greedy exploration rate. It would also be possible to study the effect of noise due to using large but finite batch sizes. As these extensions can be introduced one by one, the method allows us to isolate the effects of these three components. These extensions are being investigated in parallel[53].

The effect of a smaller learning rate, exploration, or noise due to finite batch size will depend on the details of the sample batch learning algorithm being used. Alternatives to the sample batch algorithm considered here, such as SARSA and actor critic learning, are discussed in [21]. In addition, it is possible to modify the postbatch update to make use of techniques such as optimistic learning [59] or leniency [60].

Finally, the method may be extended to the sequential move setting as in [54] by constructing a directed network with two types of edges: the first dictating how the strategy pair changes when player one acts, and the second dictating how the strategy pair changes when player two acts. The sequential best-response dynamics in this network will follow paths of alternating edge types.

Our results, together with the extensions outlined above, provide an avenue for understanding the dynamics of multiagent reinforcement learning algorithms in simple settings. The insights gained through this understanding may also be applicable to more complex environments and may be used in designing algorithms that learn successfully in such settings or in designing policies to regulate the use of certain techniques in multiagent settings with undesirable outcomes (such as price collusion).

Appendix

A. Phase Diagram as a Function of the Discount Factor

In Figure 10 we plot the 12 regions of the phase diagram for the PD for different values of δ. This shows how the regions change as we vary the discount factor. The plots for the SH and HD games will show the same behaviour, with the difference that the regions represent different graphs. In particular, the regions associated with the existence of specific absorbing states are different.

The plots show that an increase in δ leads to an increase in the fraction of the phase space occupied by the region where all three possible equilibria exist in the PD. For the SH and HD, in contrast, we find that regions in which certain combinations of absorbing states exist disappear as we increase δ.

B. Examples of MBRN

In Figure 11–13, we show the MBRN in the 12$^{\text{th}}$ region of the PD, SH, and HD, respectively. We indeed see that the sizes of the basin of attraction for the absorbing states are more evenly distributed in the SH and HD than in the PD.

C. Region Conditions for the PD

In Table 7, we give the conditions for the regions of the normalized PD. The conditions for the other two (normalized) games can easily be derived from the conditions given here by replacing R with T for the SH and P with S for the HD game.

Data Availability

No data were used to support this study.

Disclosure

This research work is an updated version of the preprint [61].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Dutch Institute for Emergent Phenomena (DIEP) cluster at the University of Amsterdam.

References

[1] L. Buşoniu, R. Babuska, and B. De Schutter, “A comprehensive survey of multiagent reinforcement learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[2] Y. Yang and J. Wang, “An overview of multi-agent reinforcement learning from game theoretical perspective,” 2020, https://arxiv.org/abs/2011.00583.

[3] G. Palmer, *Independent Learning Approaches: Overcoming Multi-Agent Learning Pathologies in Team-Games*, The University of Liverpool, United Kingdom, 2020.

[4] L. Buşoniu, R. Babuška, and B. De Schutter, “Multi-agent reinforcement learning: an overview,” *Innovations in multi-agent systems and applications-1*, pp. 183–221, 2010.

[5] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, “A survey and critique of multiagent deep reinforcement learning,” *Autonomous Agents and Multi-Agent Systems*, vol. 33, no. 6, pp. 750–797, 2019.

[6] L. Panait and S. Luke, “Cooperative multi-agent learning: the state of the art,” *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pp. 387–434, 2005.

[7] K. Zhang, Z. Yang, and T. Başar, “Multi-agent reinforcement learning: a selective overview of theories and algorithms,” *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021.

[8] Y. Zhang and M. Guizani, *Game Theory for Wireless Communications and Networking*, CRC Press, Florida, Fl, USA, 2011.

[9] S. Koh, B. Zhou, H. Fang et al., “Real-time deep reinforcement learning based vehicle navigation,” *Applied Soft Computing*, vol. 96, Article ID 106694, 2020.

[10] S. Assad, R. Clark, D. Ershov, and L. Xu, “Algorithmic pricing and competition: empirical evidence from the German retail gasoline market,” *CESifo Working*, 2020.

[11] Y. Yang, Da-W. Ding, H. Xiong, Y. Yin, and D. C. Wunsch, “Online barrier-actor-critic learning for H_∞ control with full-state constraints and input saturation,” *Journal of the Franklin Institute*, vol. 357, no. 6, pp. 3316–3344, 2020.

[12] Y. Yang, M. Mazouchi, and H. Modares, “Hamiltonian-driven adaptive dynamic programming for mixed H_2/H_∞ performance using sum-of-squares,” *International Journal of Robust and Nonlinear Control*, vol. 31, no. 6, pp. 1941–1963, 2021.

[13] Y. Yang, K. G. Vamvoudakis, and H. Modares, “Safe reinforcement learning for dynamical games,” *International Journal of Robust and Nonlinear Control*, vol. 30, no. 9, pp. 3706–3726, 2020.

[14] M. Benaïm, “Dynamics of stochastic approximation algorithms,” in *Seminaire de probabilites XXXIII*, pp. 1–68, Springer, Berlin, Germany, 1999.

[15] D. Bloembergen, K. Tuyls, D. Hennes, and M. Kaisers, “Evolutionary dynamics of multi-agent learning: a survey,” *Journal of Artificial Intelligence Research*, vol. 53, pp. 659–697, 2015.

[16] P. Mertikopoulos and W. H. Sandholm, “Riemannian game dynamics,” *Journal of Economic Theory*, vol. 177, pp. 315–364, 2018.

[17] C. J. Watkins and P. Dayan, “Q-learning,” *Machine learning*, vol. 8, no. 3/4, pp. 279–292, 1992.

[18] J. Hofbauer and K. Sigmund, “Evolutionary game dynamics,” *Bulletin of the American Mathematical Society*, vol. 40, no. 4, pp. 479–519, 2003.

[19] M. Kaisers and K. Tuyls, “Frequency adjusted multi-agent q-learning,” *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, vol. 1, pp. 309–316, 2010.

[20] E. Rodrigues Gomes and R. Kowalczyk, “Dynamic analysis of multiagent q-learning with ε -greedy exploration,” in *Proceedings of the 26th Annual International Conference on*

Machine Learning, pp. 369–376, Montreal Quebec Canada, June 2009.

[21] W. Barfuss, J. F. Donges, and J . . Kurths, "Deterministic limit of temporal difference reinforcement learning for stochastic games," *Physical Review A*, vol. 99, no. 4, Article ID 043305, 2019.

[22] R. Axelrod and W. D. Hamilton, "The evolution of cooperation," *Science*, vol. 211, no. 4489, pp. 1390–1396, 1981.

[23] A. Rapoport, A. M. Chammah, and J. Carol, *Prisoner's dilemma: A study in conflict and cooperation*, Vol. 165, University of Michigan press, Michigan, MI, USA, 1965.

[24] J.-J. Rousseau, *Discours sur l'origine et les fondements de l'inégalité parmi les hommes*, Bordas, Paris, 1754.

[25] J. M. Smith and G. R. Price, "The logic of animal conflict," *Nature*, vol. 246, no. 5427, pp. 15–18, 1973.

[26] J. M. Meylahn and A. V den Boer, "Learning to Collude in a Pricing Duopoly," *Manufacturing & Service Operations Management*, 2022.

[27] E. Calvano, G. Calzolari, V. Denicoló, J. E. Harrington, and S. Pastorello, "Protecting consumers from collusive prices due to AI," *Science*, vol. 370, no. 6520, pp. 1040–1042, 2020.

[28] E. Calvano, G. Calzolari, V. Denicoló, and S. Pastorello, "Artificial intelligence, algorithmic pricing, and collusion," *The American Economic Review*, vol. 110, no. 10, pp. 3267–3297, 2020.

[29] T. Klein, "Autonomous algorithmic collusion: Q-learning under sequential pricing," *The RAND Journal of Economics*, vol. 52, no. 3, pp. 538–558, 2021.

[30] V. Arnoud, J. M. Meylahn, and Maarten Pieter Schinkel, "Artificial collusion: examining supra-competitive pricing by autonomous q-learning algorithms," *Amsterdam Center for Law & Economics*, 2022.

[31] M. W. Macy and A. Flache, "Learning dynamics in social dilemmas," *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl_3, pp. 7229–7236, 2002.

[32] L. Waltman and U. Kaymak, "-learning agents in a Cournot oligopoly model," *Journal of Economic Dynamics and Control*, vol. 32, no. 10, pp. 3275–3293, 2008.

[33] M. Rabin, *Incorporating Fairness into Game Theory and Economics*, pp. 1281–1302, The American economic review, Tennessee, USA, 1993.

[34] R. Lahkar, "Equilibrium selection in the stag hunt game under generalized reinforcement learning," *Journal of Economic Behavior & Organization*, vol. 138, pp. 63–68, 2017.

[35] Z. Joel, V. Zambaldi, M. Lanctot, J. Marecki, and T. Graepel, "Multi-agent reinforcement learning in sequential social dilemmas," 2017, https://arxiv.org/abs/1702.03037.

[36] W. Mao and T. Başar, "Provably efficient reinforcement learning in decentralized general-sum Markov games," *Dynamic Games and Applications*, pp. 1–22, 2022.

[37] P. Alexander and L. Adam, "Prosocial learning agents solve generalized stag hunts better than selfish ones," arXiv preprint arXiv:1709.02865, 2017.

[38] M. R. Arefin, K. M. A. Kabir, M. Jusup, H. Ito, and J. Tanimoto, "Social efficiency deficit deciphers social dilemmas," *Scientific Reports*, vol. 10, no. 1, pp. 16092–16099, 2020.

[39] H. Ito and J. Tanimoto, "Scaling the phase-planes of social dilemma strengths shows game-class changes in the five rules governing the evolution of cooperation," *Royal Society Open Science*, vol. 5, no. 10, Article ID 181085, 2018.

[40] J. Tanimoto, *Sociophysics approach to epidemics*, Vol. 23, Springer, , Berlin, Germany, 2021.

[41] J. Tanimoto and H. Sagara, "Relationship between dilemma occurrence and the existence of a weakly dominant strategy in a two-player symmetric game," *Biosystems*, vol. 90, no. 1, pp. 105–114, 2007.

[42] Z. Wang, S. Kokubo, M. Jusup, and J. Tanimoto, "Universal scaling for the dilemma strength in evolutionary games," *Physics of Life Reviews*, vol. 14, pp. 1–30, 2015.

[43] B. R. Bruns, "Names for games: locating 2 × 2 games," *Games*, vol. 6, no. 4, pp. 495–520, 2015.

[44] M. Wunder, M. L. Littman, and M. Babes, "Classes of Multiagent Q-Learning Dynamics with Epsilon-Greedy Exploration," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, Haifa, Israel, June 2010.

[45] D. S. Leslie and E. J. Collins, "Individual q-learning in normal form games," *SIAM Journal on Control and Optimization*, vol. 44, no. 2, pp. 495–514, 2005.

[46] P. Mertikopoulos and W. H. Sandholm, "Learning in games via reinforcement and regularization," *Mathematics of Operations Research*, vol. 41, no. 4, pp. 1297–1324, 2016.

[47] A. Ozdaglar, M. O. Sayin, and K. Zhang, "Independent learning in stochastic games," 2021, https://arxiv.org/abs/2111.11743.

[48] M. Sayin, K. Zhang, D. Leslie, T. Basar, and A. Ozdaglar, "Decentralized q-learning in zero-sum Markov games," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18320–18334, 2021.

[49] F. Drew and D. K. Levine, *The Theory of Learning in Games*, MIT press, Massachusetts, MA, USA, 1998.

[50] F. Drew and K. David, *Levine. Learning and Equilibrium*, Annual Review of Economics, California, CL, USA, 2009.

[51] E. Hopkins, "A note on best response dynamics," *Games and Economic Behavior*, vol. 29, no. 1-2, pp. 138–150, 1999.

[52] G. W. Brown, "Iterative solution of games by fictitious play," *Act. Anal. Prod Allocation*, vol. 13, no. 1, p. 374, 1951.

[53] W. Barfuss and J. M. Meylahn, "Intrinsic fluctuations of reinforcement learning promote cooperation," 2022, https://arxiv.org/abs/2209.01013.

[54] Y. Usui and M. Ueda, "Symmetric equilibrium of multi-agent reinforcement learning in repeated prisoner's dilemma," *Applied Mathematics and Computation*, vol. 409, Article ID 126370, 2021.

[55] T. Jaakkola, M. I. Jordan, and S. P. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural Computation*, vol. 6, no. 6, pp. 1185–1201, 1994.

[56] F. Drew and T. Jean, *Game Theory*, MIT press, Massachusetts, MA, USA, 1991.

[57] W. Barfuss, "Reinforcement learning dynamics in the infinite memory limit," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1768–1770, New Zealand, May 2020.

[58] M. Nowak and K. Sigmund, "A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game," *Nature*, vol. 364, no. 6432, pp. 56–58, 1993.

[59] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, "Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 64–69, IEEE, San Diego, CA, USA, October 2007.

[60] E. Wei and S. Luke, "Lenient learning in independent-learner stochastic cooperative games," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2914–2955, 2016.

[61] M. Janusz and J. Lars, "Limiting dynamics for Q-learning with memory one in symmetric two-player, two-action games," 2022, https://arxiv.org/abs/2107.13995.