Learning to cooperate against ensembles of diverse opponents

Isuri Perera
Monash University
Melbourne, Australia
isuri.perera@monash.edu

Frits de Nijs
Monash University
Melbourne, Australia
frits.nijs@monash.edu

Julian Garcia
Monash University
Melbourne, Australia
julian.garcia@monash.edu

ABSTRACT

The emergence of cooperation in decentralised multiagent systems is challenging; naive implementations of learning algorithms typically fail to converge or converge to equilibria without cooperation. Opponent modelling techniques, combined with Reinforcement Learning have been successful in promoting cooperation, but face challenges when other agents are plentiful or anonymous. We envision environments in which agents face a sequence of interactions with different and heterogeneous agents. Inspired by models of Evolutionary Game Theory, we introduce RL agents that forgo explicit modelling of others. Instead, they augment their reward signal by considering how to best respond to others assumed to be rational against their own strategy. This technique not only scales well in environments with many agents, but can also outperform opponent modelling techniques across a range of cooperation games. Agents that use the algorithm we propose can successfully maintain and establish cooperation when playing against an ensemble of diverse agents. This finding is robust across different kinds of games, and can also be shown not to disadvantage agents in purely competitive interactions. While cooperation in pairwise settings is foundational, interactions across large groups of diverse agents are likely to be the norm in future applications where cooperation is an emergent property of agent design, rather than a design goal at the system level. The algorithm we propose here is a simple and scalable step in this direction.

KEYWORDS

Cooperation, Iterated Prisoner's Dilemma, Evolutionary Game Theory, Best Response, Population games

1 INTRODUCTION

Cooperation is an essential feature of human social interactions and a desirable feature in multi-agent systems. Cooperation happens when agents pay a cost in order to help others. Not cooperating is typically a dominant strategy, but other equilibria may also exist where agents can mutually benefit from learning to work together. To be successful, artificial intelligent agents need the ability to cooperate with humans [44] and other machines in diverse environments [11, 12, 28]. These environments can encompass interactions that range from fully cooperative – where the incentives of all parties are aligned – to fully competitive zero-sum games. Most games of cooperation sit between these two extremes. Convergence to undesirable equilibria is common [2, 16, 45].

In games of cooperation, self-interested agents need to be competitive by avoiding exploitation, but they also need to have the ability to reap the gains to be had by working together whenever these synergies are available. This tension between individual rewards and group outcomes is the defining feature of cooperation [42].

Reinforcement Learning (RL) has achieved unparalleled success in fully cooperative and fully competitive multi-agent environments. However, the application of RL to general-sum games played by many autonomous agents is not trivial. In Multi-agent Reinforcement Learning (MARL), agents learn simultaneously, making all other agents a part of the focal agent's environment. Non-stationarity makes it difficult for naïve reward maximizing agents to converge to equilibria even with simple reward structures [38].

In this paper we study decentralized cooperation. We aim for environments in which agents face a series of interactions with diverse opponents. Examples include self-driving cars, which would resolve a number of interactions as part of a route. Or sequential transactions on a market floor. While opponent modelling may be successful when an agent learns by playing against another, sequences of interactions with several diverse agents may render this approach infeasible. We thus focus on formulating an intrinsic reward function that scales well in this kind of environment, fostering cooperation when there is an opportunity to do so, while being resilient to play against agents that do not cooperate or do not use the same intrinsic reward function.

MARL has been successful in addressing cooperation through opponent modelling [16, 29] and centralized rewards [33]. Opponent modelling attempts to identify the opponent's strategy and actively respond to it, making it easier to achieve cooperation. Centralising rewards refers to agents attempting to improve the collective outcome instead of pursuing their individual rewards. While centralisation is a simple approach to achieve cooperation, the success of centralised architectures relies on the ability to alter the objective functions of individual agents. This assumption is hard to match in a realistic population setting where exchanging rewards may not be possible and communication is limited.

Opponent modelling on the other hand can work in fully decentralised environments, but its computational cost may have a strong impact in large environments where it is necessary to keep track of all others. Moreover, in settings with a diverse ensemble of agents, individuals may only ever get to interact with a handful of others and agents are often anonymous. This, in principle, can also affect the observability necessary to successfully model opponents.

In this paper we introduce best-response guided agents (BRG); which consider an intrinsic reward that measures their ability to make a reward maximizing opponent exploitable in the future. This is computed on the basis of information that the agent already has about his own policy and the game itself. This intrinsic reward allows them to successfully navigate the reward structure in cooperation problems, even when the environment features large

Proc. of the Adaptive and Learning Agents Workshop (ALA 2023), Cruz, Hayes, Wang, Yates (eds.), May 29-30, 2023, London, UK, https://alaworkshop2023.github.io/. 2023.

ensembles of diverse agents. Our agents learn to adopt reciprocating strategies that reward cooperation and penalize defection, in turn guiding other agents towards desirable equilibria.

Agents can either use their knowledge of the game or use self-play [9, 10] to calculate this intrinsic reward. The proposed method does not require specific knowledge or inferences over the strategies of others, which makes it scalable in population games. In addition, preserving the self-centred nature of individual agents makes them less prone to exploitation while maintaining cooperation.

We test our approach on a range of different cooperation games. We start with a standard Iterated Prisoner's Dilemma (IPD) [3]; a game with infinitely many equilibria [22] where standard agent-centric RL algorithms are known to fail to converge to cooperation [16, 18]. We then test our approach on a stochastic game [25], and a standard board game where cooperation and defection are no longer elementary actions, but properties of policies [32]. We also verify that the approach is robust even if the strategic interactions do not present cooperation opportunities, including fully competitive games.

The rest of this paper is organized as follows. In Section 2 we review the relevant literature on MARL, cooperation and population games. Section 3 provides some basic preliminaries, including notation. Our method is introduced in Section 4, and is empirically tested across different environments in Sections 5 – 8.Section 9 concludes the paper and discusses extensions and limitations.

2 RELATED WORK

Interest in Cooperative AI has lead to diverse areas of research such as communication [13, 20, 31], social preferences [38], and collective decision making [7, 8, 39]. Most of this research focuses on scenarios where agents have fully aligned interests [17, 19, 50]. Solutions arising in these cases are hard to apply when central control of individual agents is not feasible.

Our work is focused on games of cooperation where individuals are competing, but have incentives to reap higher benefits when working together. A sizeable part of the literature here is inspired by models of Evolutionary Game Theory (EGT). In these models, a large group of agents is assumed to learn from their rewards they obtain when playing a game successively with others in the population. EGT focuses on predicting stable outcomes from simple learning processes [43].

As a cooperation testbed we use the Iterated Prisoner's dilemma game [3]. This game is useful because it has many equilibria, including efficient equilibria that sustain cooperation as well as those where defection is prevalent [21]. This game has also been previously used in other studies of MARL, e.g. [16].

The key to achieving cooperation in this setting is reciprocity. Both, Eccles et al. [15] and Lerer and Peysakhovich [32] use this property to design algorithms that achieve better outcomes in general-sum games with RL. These algorithms use a combination of purely cooperative and purely competitive strategies to enforce reciprocity. In contrast, our BRG agents do not explicitly prescribe reciprocity to achieve cooperation but naturally tend to achieve it following the intrinsic reward.

Another class of algorithms which includes WoLF [5], JAL, AWESOME [10] and Lanctot et al. [30] use best response and self-play in general-sum games. WoLF proposes using varying learning rates to achieve convergence and provide proof for a subset of iterative matrix games. JAL attempts to understand the value of joint actions and requires maintaining beliefs about others strategies. Claus and Boutilier [9] show that JAL does not always converge to the optimal equilibrium in situations with many equilibria. AWESOME uses a pre-calculated Nash equilibrium of the one-shot game to guarantee convergence. Hu and Wellman [29] introduce the Nash-Q algorithm where agents use Nash equilibria calculations with opponent modelling to reach higher payoffs in general-sum games. Nash-Q operates with the assumption that opponents attempt to reach the calculated equilibrium and convergence varies with the type of Nash equilibrium chosen. These algorithms focus on convergence more than on achieving Pareto-optimal equilibria or shaping opponents behaviour for better future rewards.

Numerous other approaches to solving MDPs in multi-agent settings involve accounting for others' beliefs based on cognitive hierarchies such as level-k solution concept [14, 23, 26]. These methods lead to more cognitively advanced agents capable of teaching and learning from other agents [27, 51]. While our agents do not explicitly account for distinct hierarchical levels, the concept of using information from best responses to naïve learners shares some similarities with these approaches. Our agents are not modelling others explicitly, but assuming they are responding reasonably to the strategy of our learning agent.

MARL has also achieved success in zero-sum games [36, 46, 49]. Although our work does not focus on purely competitive settings or include opponent modelling, the proactive approach of BRG agents to exploit opponents, in the long run, relates to existing research on opponent modelling and lookahead type algorithms [6, 35]. We use these as inspiration to identify an intrinsic reward based on best response and lookahead calculations. Most of these methods assume a limited number of agents in a controlled environment. It is therefore difficult to apply them to environments with ensembles of diverse agents.

We assume agents are randomly matched for playing a game in each round, drawn in from a population. This matching procedure is common in EGT [43], and resembles a situation where agents are part of a large ecology and move through a series of interactions encoded as a game, which could itself consist of multiple steps, such as in the case of IPD. The reward is taken to be the average reward over a series of matches. Importantly, in this scenario agents are anonymous, self-interested and there is no central control or institution mediating interactions. The links between MARL and EGT are further discussed by Tuyls and Nowé [48].

3 BACKGROUND

The basis for the IPD is the Prisoner's Dilemma (PD). Here, agents can choose from two actions: Cooperate or Defect; mutual cooperation is rewarded with a payoff R, and mutual defection is punished with a payoff P; a defector exploiting a cooperator will get a temptation payoff T, while a cooperator being exploited gets the sucker payoff S. This can be summarized using the payoff matrix $\left(\begin{smallmatrix} R & S \\ T & P \end{smallmatrix}\right)$, with $T > R > P > S$. Defection is the only dominant strategy and it is not Pareto efficient.

When the PD is repeated for an (uncertain) number of rounds, cooperation can be an equilibrium. In the IPD, the probability of having a next round is the continuation probability, δ. The IPD has infinitely many Nash equilibria if δ is large enough [37].

For example, a `TFT` (tit-for-tat) strategy – cooperate in the first round and then follow the opponent's last action – can form the basis of an equilibrium profile (`TFT`, `TFT`). A `TFT`'s threat of punishing defection encourages the opponent to cooperate. A pair of `ALLD` (always defect) strategies also constitutes an equilibrium of the repeated game. Some other widely discussed strategies include `ALLC` (always cooperate) and `DTFT` (tit-for-tat starting with defection).

When playing IPD in a population, pairs of agents are matched to play the game every round. In this paper, we use full matching where each agent plays with all other agents once every round. In population settings, the agents are usually kept anonymous and they make strategically independent decisions that shape the direction of the population in the long run.

Players' strategies can use their memory of the N past rounds (memory-N) to choose actions. Following Press and Dyson [40], we restrict memory to one round. I.e., a player's action depends only on the actions played in the last round. This is discussed in more detail in Sections 5 and 6.

When evaluating a best-response, we construct a single-agent Markov Decision Process (MDP) from the repeated game. This MDP $G = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$ where $s \in \mathcal{S}$ is the state of the environment defined using memory, $a \in \mathcal{A}$ is the action taken by the focal agent, $\mathcal{R}$ is the reward function giving the immediate rewards as given by the payoff matrix, $\mathcal{T}$ is the state transition matrix, and $\gamma \in (0, 1)$ is the discount factor of future rewards. The probability of taking each action in a given state is specified by the policy of an agent. The state transition matrix uses the opponent policy's action selection probabilities, and therefore assumes a fixed opponent policy for the duration of a game.

Our agents use Policy Gradient [47], and update their policy, π, by performing gradient ascent on the expected discounted reward with respect to the policy parameters, $\theta(s, a)$. We use Softmax for the policy parametrization, meaning,

$$\pi_\theta(s, a) = \frac{\exp(\phi(s, a) \cdot \theta)}{\sum_{k=1}^{|\mathcal{A}|} \exp(\phi(s, a_k) \cdot \theta)}, \tag{1}$$

where $\phi(s, a)$ is the feature vector related to a state, action pair.

To identify a best response against a fixed strategy we use Q-value iteration. The recursive Bellman equation [4] defines the expected value of a state, $V(s)$, and the expected value of taking action a in state s and acting optimally from then onwards, $Q(s, a)$. This quantity is calculated iteratively, starting from $Q_0(s, a) = 0$, as

$$Q_{k+1}(s, a) = \sum_{s' \in \mathcal{S}} \mathcal{T}(s, a, s') \left[\mathcal{R}(s, a, s') + \gamma \max_{a' \in \mathcal{A}} Q_k(s', a') \right] \quad \forall s, a$$

Once the state values converge, the maximum value action in a given state s is followed to identify the optimal action sequence.

4 METHOD

The naïve form of MARL considers opponents to be a part of the agent's environment and ignores the non-Markovian nature in multi-agent systems [24]. As a result, naïve learners lose the ability to proactively explore the combined action space, tending to converge to undesirable equilibria in most multi-agent settings. If agents can account for how their current policy affects their future rewards they could possibly avoid this mishap.

Algorithm 1 BRG agent i, training on population game G

```
 1: $\pi_i \leftarrow$ initial policy, $ep \leftarrow 0$
 2: while $ep <$ limit do
 3:     $exp \leftarrow$ PLAY_GAMES($G, \pi_i$, batch size)          ▹ $exp = [\langle s_1, a_1, r_1 \rangle, \ldots]$
 4:     $\mathrm{IR}[s, a] \leftarrow 0,\ \forall s, a$
 5:     for $s' \in S, a' \in A$ do
 6:         $\pi_i^{s',a'} \leftarrow$ FIX_POLICY($\pi_i, s', a'$)                   ▹ Equation (2)
 7:         $Q^{\mathrm{BR}}, \bar{\pi}_i \leftarrow$ BEST_RESPONSE($\pi_i^{s',a'}$)
 8:         $Q^{\mathrm{BR}^2}, \bar{\bar{\pi}}_i \leftarrow$ BEST_RESPONSE($\bar{\pi}_i$)
 9:         $\mathrm{IR}[s', a'] \leftarrow \max_a(Q^{\mathrm{BR}^2}[s', a])$
10:     end for
11:     $exp_{\mathrm{IR}} \leftarrow \eta r_j + (1 - \eta) \frac{\mathrm{IR}[s_j, a_j]}{n}$,          $\forall \langle s_j, a_j, r_j \rangle \in exp$
12:     $\pi_i \leftarrow$ POLICY_GRADIENT($\pi_i, exp_{\mathrm{IR}}$)
13:     $ep \leftarrow ep + 1$
14: end while
```

BRG agents achieve this assuming a best responding naïve opponent with perfect information. A measure of the impact of the current policy on future rewards is then given by the reward of the best counter-response to the best responding opponent. This becomes the intrinsic reward of a BRG agent. Instead of only maximizing the current reward, BRG agents attempt to maximize the current reward plus intrinsic reward. We discuss this in detail in the following section.

4.1 Intrinsic reward calculation

The intrinsic reward calculation for the state-action pair, (s', a'), under policy, π, aims to measure how beneficial taking action a' in state s' at this point, will be in future episodes (see Algorithm:1, lines 4-11). To compute the intrinsic reward for (s', a'), we assume the focal agent deterministically picks action a' in state s', while following the current policy π_i in all other states. Thus, FIX_POLICY on line 6 computes a modified policy $\pi_i^{s',a'}$ for which

$$\pi_i^{s',a'}(s, a) = \begin{cases} 1, & \text{if } s = s', a = a', \\ 0, & \text{if } s = s', a \neq a', \\ \pi_i(s, a), & \text{otherwise.} \end{cases} \tag{2}$$

As the next step to identifying future rewards, we assume the opponent will be best responding and calculate the best response, $\bar{\pi}_i$, against $\pi_i^{s',a'}$ using value iteration, Algorithm:1, line 7.

$$Q^{\mathrm{BR}}, \bar{\pi}_i \leftarrow \text{BEST\_RESPONSE}(\pi_i^{s',a'}) \tag{3}$$

Here, Q^{BR} refers to the Q-values; the expected reward of state-action pairs at convergence in value iteration.

Assuming that the opponent will eventually learn to follow this best-response policy $\bar{\pi}_i$, we preempt their best-response by computing the focal agent's best response to this future policy on line 8 of Algorithm 1,

$$Q^{\mathrm{BR}^2}, \bar{\bar{\pi}}_i \leftarrow \text{BEST\_RESPONSE}(\bar{\pi}_i) \tag{4}$$

The value of each action when best responding is given by Q^{BR^2}. The optimal *preemptive* action to follow in state s' according to this best-response could be different from the *intended* action a', i.e., $a' \neq \arg\max_a(Q^{\mathrm{BR}^2}[s', a])$. Therefore, our intrinsic reward of the state-action pair (s', a') is given by the value of the optimal action to take according to Q^{BR^2},

$$\mathrm{IR}[s', a'] \leftarrow \max_a\big(Q^{\mathrm{BR}^2}[s', a]\big) \tag{5}$$

It is important to note here that generally, $\mathrm{IR}[s', a_j] \neq \mathrm{IR}[s', a_k]$ for $j \neq k$, because the input policy $\pi_i^{s',a'}$ depends on a'.

The Q-values capture the expected reward of taking an action in a given state for an entire episode and are not on the same scale as environmental rewards r. Therefore we scale the Q-value by dividing it by the expected number of steps per episode, n. For repeated games with continuation probability, δ, n is expected to be $1/(1-\delta)$. Putting it all together, we obtain the new utility function,

$$\mathcal{U} = \eta r + (1-\eta)\frac{\mathrm{IR}[s_j, a_j]}{n}, \tag{6}$$

where $\eta \in [0, 1]$ specifies the extent to which the agent should focus current and future rewards. See Algorithm:1, line 11. To achieve best performance η needs to be sufficiently low at the start of learning to allow for adequate exploration.

To compute the intrinsic reward for all state-action pairs, we need to compute $2|\mathcal{S}||\mathcal{A}|$ best response value iteration calls, which are themselves polynomial in the size of the single-player MDP folding the target policy into transition function $\mathcal{T}$.

4.2 Intrinsic reward example

To demonstrate the effect of the intrinsic reward on the behaviour of the agents, we consider four example scenarios of BRG agents playing IPD against naïve learners, shown in Table 1. Because the BRG agent observes both the regular episode return r and the intrinsic reward IR, it potentially has a gradient towards improvement in either metric. We observe that, depending on the current policies in the scenario, even when one of the signals is maximized we can often still update the policy to improve on the other dimension.

In situations where the environment reward is already maximized (Scenarios 1 and 2), the intrinsic reward allows the agent to make their policy more cooperative or robust; for example, in scenario 1, an ALLD BRG agent facing an ALLD opponent cannot improve their current reward, but improving the intrinsic reward leads to a DTFT-like strategy, which makes cooperation possible in the future. Similarly, in scenario 2 an ALLC BRG agent facing a TFT opponent will discover its own vulnerability to defection through best-responding to ALLC with ALLD. It can make itself less susceptible to future entrants by maximizing the intrinsic reward to reach a TFT-like policy.

The intrinsic reward is already maximized (scenarios 3 and 4), the current reward may still be improved to make itself less exploitable to opponents. For example in scenario 3, the BRG agent is playing TFT, matched in payoff by ALLC, which in turn leads to ALLD. As such, the intrinsic reward suggests the opponent should become fully exploitable (resulting in reward T). However, the agent can still improve its current reward, by avoiding exploitation in the opening move at s_0.

Table 1: Scenario analysis for BRG agents. Here n is the expected duration of the game.

Scenario	BRG agent	Opponent	Return r	IR
1	ALLD	ALLD	$P \cdot n$	$P \cdot n$
Expected behaviour: BRG agents cannot maximize r but IR can be improved to reach a maximum at DTFT. Maximizing IR makes cooperation possible for new entrants/in the future.				
2	ALLC	TFT	$R \cdot n$	$P \cdot n$
Expected behaviour: BRG agents cannot maximize r but IR can be improved to reach a maximum at TFT. Maximizing IR makes BRG agent less exploitable for a new entrant/in the future.				
3	TFT	ALLD	$S + P \cdot (n-1)$	$T \cdot n$
Expected behaviour: BRG agents cannot maximize IR but r can be improved to reach a maximum at DTFT. Maximizing r makes the BRG agent less exploitable for the current opponent.				
4	TFT	DTFT	$(T + S) \cdot \frac{n}{2}$	$T \cdot n$
Expected behaviour: BRG agent cannot maximize IR but can attempt to improve r by cooperating more at state *CD*. This scenario also highlights the need of assuming a learning opponent.				

5 ITERATED PRISONER'S DILEMMA

Here we describe and perform experiments to evaluate the impact our Intrinsic Reward has on convergence, stability and robustness of efficient equilibria in the IPD. We design a number of experiments to evaluate the following hypotheses about our BRG agents; they

- *consistently* achieve cooperation, even against naïve agents;
- require *less computational resources* than an explicit opponent modelling strategy to do so; and
- are *robust* to exploitation by new entrants.

Each of these hypotheses is tested in a separate experiment, which we describe in the following three subsections. Across all our IPD experiments we use the same game structure, having payoffs $R = 1$, $S = -6$, $T = 0$ and $P = -5$ and a continuation probability $\delta = 0.95$.

Convergence of BRG agents. To evaluate if BRG agents consistently achieve cooperation, we conduct experiments where agents pair off against each other as BRG versus BRG, BRG versus naïve, and naïve versus naïve. Each pair is played to convergence multiple times from 100 different starting policies, to evaluate if convergence is robustly achieved. We sample the starting policies to choose the cooperate action with probabilities sampled uniformly, such that $\forall i, s\colon \pi_i(s) \sim \mathcal{U}(0.1, 0.9)$. We define the achievement of cooperation by the tendency of the pair to converge to an *average* payoff close to R, the payoff of cooperation.

Figure 1(a) presents the average learning trajectories of the three different pairs. As expected, we observe that a pair of BRG agents always achieves the reward for full cooperation, R. Even when playing against a naïve agent, the average reward achieved is close to full cooperation. However, we observe that the standard deviation is larger than the all BRG agents case.

To investigate why, we look at the difference in payoff obtained by BRG and naïve agents, in Figure 1(b). We observe that BRG

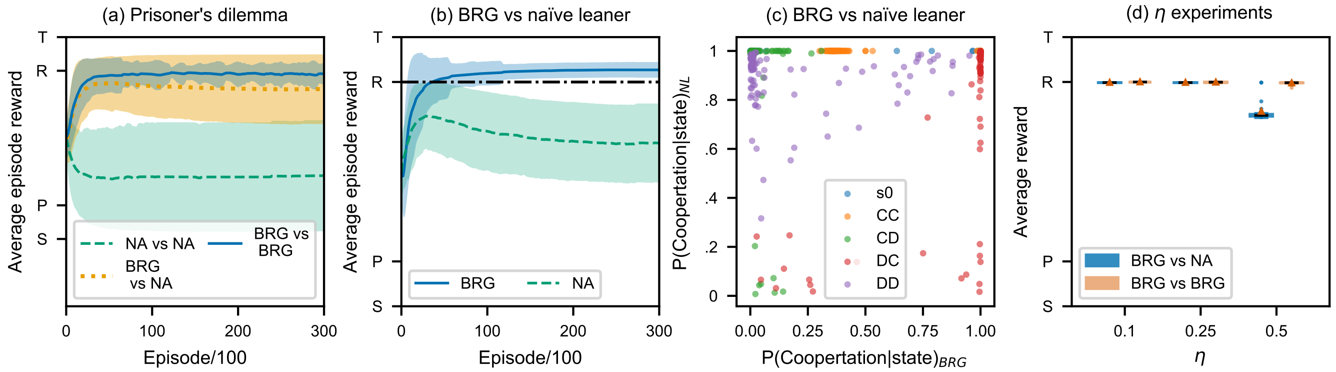

Figure 1: Performance of different types of agents in the prisoner's dilemma. (a) Trajectories of average total episode rewards of both players. (b) Individual reward trajectories of BRGs and naïve learners when playing against each other (yellow in *a*), compared with full cooperation (black dashed line). (c) Scatter plot of the policies at the end of runs from panel *b*. (d) Distributions of final average reward of episodes, as function of η. (NA: naïve agent. BRG: Best-response guided agent. x vs y: setting where agent type x plays type y one-on-one). Shaded area is one standard deviation above and below the average.

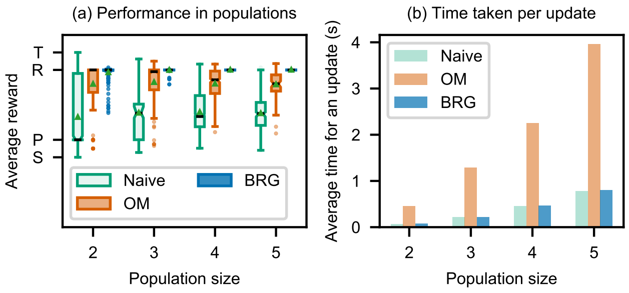

Figure 2: Performance of Naïve, BRG, and OM agents as the (homogeneous) population increases from 2 to 5 agents, showing average of (a) step reward, (b) time per update step.

agents manages to exploit their naïve opponents to gain a reward that's slightly higher than R. BRG agents consistently achieves this stable payoff without a collapse to the low-payoff equilibrium *DD*. This stands in contrast with the experiments with a pair of naïve learners, which *typically* collapse to *DD* except under the occasional fortunate starting conditions.

We can further look into the characteristics of the resulting policies by looking at the distribution of their action probabilities in each state. Figure:1(c) shows BRG agents mostly initiate with defection, but tend to reciprocate cooperation. They control the naive agent to stay cooperative by threatening with a grim trigger-like period of defection (much higher chance of defection in CD or DD states where the opponent chose D action). Despite this, they manage to guide the opponents to full cooperation (under a wide range of η hyper-parameter settings, see Figure 1(d)), by punishing bad deeds from others (when in *CD*) and cooperating when in *DC*. Some level of cooperation is shown in *DD*, arguably to support recovery from accidental defection.

We see this convergence behavior consistently, independent of the particular hyperparameter setting used, in Figure 1(d). For these and following experiments, we therefore keep the hyperparameters constant. The learning rate is set to 0.01, with a batch size of 1000 and η of 0.5.

Computational complexity of BRG agents. To evaluate the computational benefit BRG agents bring to population games, we measure the wall-clock time it takes to perform one batch learning update *for the entire population* as the population size increases, across 100 runs per size. We compare with the baseline of a naïve-agent population, and with a recent successful opponent modelling strategy, Learning with Opponent Learning Awareness (LOLA) with DiCE: The infinitely differentiable Monte Carlo estimator [16, 18].

Figure 2 shows how the performance of the algorithms changes when the population size is increased from pairs of agents to populations containing 5 agents. In every case, the agents play IPD with complete matching, meaning that every agent plays batch-size number of games against every other agent in pairs. The top panel (a) shows that the convergence characteristics of the algorithms are unaffected by the size of the population; as previously observed, BRG and LOLA agent populations consistently achieve cooperation, while naïve agents tend to the low payoff equilibrium.

However, looking at the wall-clock time each batch takes to process (Figure 2b), we see that opponent modelling adds a significant runtime overhead, when compared against BRG agents. Furthermore, this overhead increases with population size, as agents have to train, model and update policies separately for each opponent. In contrast, the overhead of increasing population size on naive or BRG agents is essentially constant, resulting in linear increase in the number of agents. Although we did not use multi-threading in our implementation, BRG agents could potentially update their policies in parallel resulting in minimal computational overhead compared with the baseline of naïve agents.

While it is not possible to characterise the run time of *all possible* OM techniques, in the standard case where an opponent model is kept for every agent the following intuition applies. The learning agent has to keep $n-1$ copies of opponent models, which in the

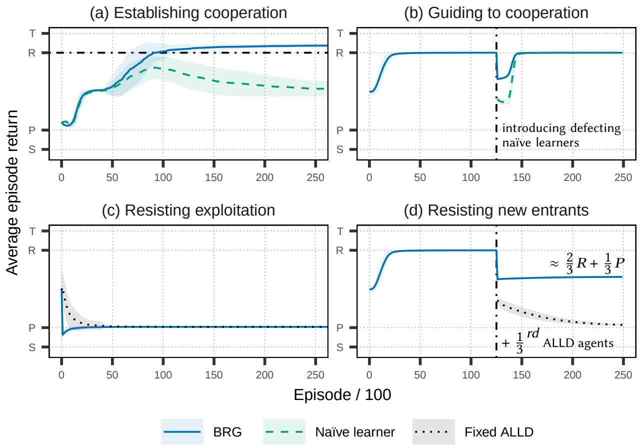

Figure 3: Mean (line) and one standard deviation (shaded area) performance of BRG agents under adverse conditions: (a) starting from 90% defection against naïve learners, (b) defecting naïve learners as new entrants, (c) learning against (fixed) ALLD agents, (d) fixed ALLD as new entrants.

case of neural networks each individually is updated with batch size b/n. BRG keeps a single network, which is updated once with batch size b. Assuming an equal number of parameters in the nets, OM has to update n times the number of variables per gradient step.

In these experiments we use the PyTorch version of LOLA-DiCE [1], for consistency with our own implementation. LOLA agents use their default parameters, except for an increased batch size of 1000, which we found was beneficial for stable convergence to cooperation. In our experiments we only show LOLA with a single lookahead, which not only takes the least amount of time but also showed the best payoffs compared to 2 and 3 lookaheads.

In summary, BRG agents are successful in consistently achieving cooperation for different population sizes when compared with naive and OM approaches, as seen in Figure 2a.

Exploitability and adaptability of BRG agents. Finally, we evaluate the robustness of BRG agents against hostile ALLD (starting) policies, both from a cold start and against an invasion of new entrants. We perform four separate tests,

(a) both BRG and naïve agents starting from ALLD,
(b) new entrant naïve learners starting from ALLD,
(c) BRG agents facing fixed ALLD strategy from $t = 0$, and
(d) new entrant fixed ALLD strategies introduced halfway.

In all cases, the ALLD strategies choose the defect action with probability 0.9 across all states, to allow for limited exploration of cooperation actions at all times. For these experiments, we consider populations of $n = 20$ agents at the start, with 10 additional new entrants introduced halfway for conditions (b) and (d). Here we also conduct each trial 100 times for statistical significance.

The learning trajectories in each of these four cases are shown in their respective panel in Figure 3. We observe that in all situations, self-aware agents are able to defend themselves against the hostile strategies. Starting from ALLD strategies, self-aware agents manage to bring the naïve agents to cooperation (3a), without themselves becoming susceptible to exploitation against fixed policies (3c).

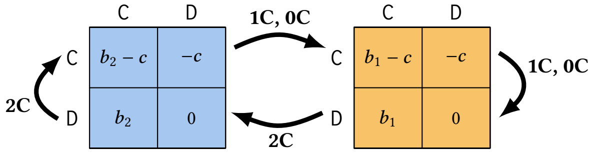

Figure 4: Stochastic iterated prisoner's dilemma [25].

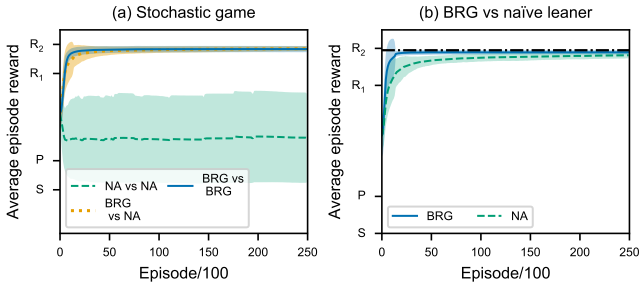

Figure 5: Performance of different types of agents in the stochastic prisoner's dilemma. (a)Trajectories of average total episode rewards of both players. (b) The separate reward trajectories of BRGs and naïve learners when playing against each other.

Additionally, we observe that BRG are robust to hostile new entrants. Hostile naïve learners (3b) are quickly converted to full cooperation, while against fixed ALLD new entrants, the BRG agents learn to resist exploitation successfully (3d). We observe that in this setting, the ALLD agents initially manage to exploit the BRG agents some of the time, but as learning proceeds, the BRG agents gradually move to a full TFT strategy. This enables them to continue to cooperate amongst themselves, without losing any payoff against the fixed agents. With one-third of new fixed ALLD entrants, BRG agents still achieve a reasonable level of cooperation. They learn to cooperate only with other BRG agents securing themselves an average reward around $(2/3)R + (1/3)P$.

6 STOCHASTIC PRISONER'S DILEMMA

We now inspect how BRG agents perform in a stochastic game. The Stochastic Iterated Prisoner Dilemma [25, SIPD] models a situation where synergistic cooperation results in increased overall social welfare. Agents start off playing a low-payoff IPD with cooperation benefit b_1 and cooperation cost c, and get the opportunity to play a game with higher benefits $b_2 > b_1$ only if both agents cooperate. This structure is illustrated in Figure 4. A single defection will move the agents back to the state with a less rewarding game.

A SIPD agent with memory restricted to the last round observes 6 states; $s0 - 1$ for $t = 0$ and $CC - 1$, $CC - 2$, $CD - 1$, $DC - 1$ and $DD - 1$ based on the combined actions of the players in the last round and the game to be played next. $s0 - 1$ refers to the initial state where previous actions are unknown and agents start with playing game1 with lower rewards. Thus, the memory-one strategy of an SIPD agent can be represented as $\pi = (\pi(\text{C}|\text{s0-1}), \pi(\text{C}|\text{CC-1}),$

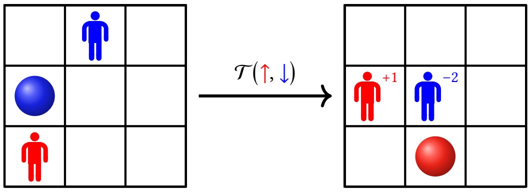

Figure 6: Coin game: socially optimal behaviour entails agents coordinating on the colour to avoid losing points, but picking up any coin is individually dominant.

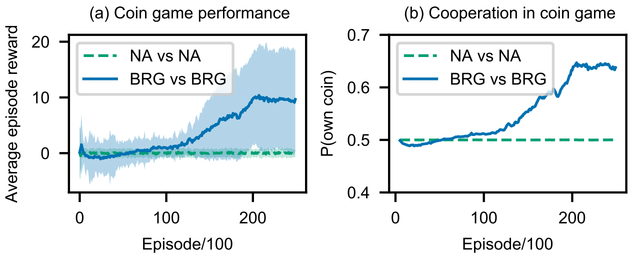

Figure 7: Naïve learners and BRGs playing the coin game. (a) Trajectories of average total episode rewards of both players. (b) Percentage of own colour coins collected.

π(C|CC-2), π(C|CD-1), π(C|DC-1), π(C|DD-1)) where π(C|s0-1) is the probability of cooperation at state s0-1.

We setup additional experiments using SIPD to evaluate,

- the robustness of BRG agents in a stochastic games and,
- the performance improvement compare to naïve learners.

Across all experiments, we use the following payoffs: $R_1 = b_1 - c = 3$, $R_2 = b_2 - c = 4$, $T_1 = b_1 = 4$, $T_2 = b_2 = 5$, $S = -c = -1$ and $P = 0$ with 20 round for an episode, shifting agents to the better game with probability 0.95 upon mutual cooperation. Successful convergence to cooperation should result in a payoff close to $0.95R_2 + 0.05R_1$ ($\approx R_2$) as agents start each episode with the low pay-off game. The experiments here follow the same structure as those discussed before.

Figure 5(a) illustrates the reward trajectories of all three agent mixtures. BRG pairs achieve full cooperation with very low variance. BRG agents playing against naïve learners show a similar path.

As opposed to non-stochastic IPD, the experiments result in significantly lower variance as BRG agents no longer benefit from attempts to exploit naïve learners through occasional defection. Any such attempts are more likely to shift the game to the low pay-off reward structure. Figure 5(b) further verify this with high levels of cooperation.

For our experiments, we used a learning rate of 0.01 and a batch size of 1000. η is set to 0.25 when playing against BRG agents and 0.1 when playing against naïve learners – in a similar pattern to the deterministic game the choice of η does not have dramatic effects.

In summary, BRG agents perform well in the stochastic version of the game: they establish and maintain cooperation.

7 COOPERATION IN COMPLEX POLICIES

In this section, we use the coin game to assess the behaviour of BRG agents in environments where cooperation is not an elementary action, but rather the property of a complex learnt policy [32]. This is more likely to resemble cooperation in the real world.

Two agents (red and blue) are placed on a grid where (red and blue) coins appear uniformly randomly. Agents receive a reward of 1 for picking up any coin, but lose 2 points if the opponent picks a coin of the focal agent's colour. A socially optimal behaviour entails agents coordinating on the colour to avoid losing points, but picking up any coin is always dominant. This results in naïve learners greedily picking up coins making the average reward zero. A standard situation in this game is depicted in Figure 6. In both cases, a socially optimal result entails the agent closest to the coin waiting for a coin of their colour.

The game is challenging, because unlike in normal-form games, cooperation or defection is not a primitive action, but entails a possibly complex policy involving navigation and combinations of actions. It is the behaviour that results from policies that we can qualify as cooperative or not. Due to this property, this game has been used extensively in literature to gauge performance of algorithms in complex cooperative settings [16, 32, 34].

Lerer and Peysakhovich [32] introduce the coin game to test amTFT, an algorithm that allows agents to shift between cooperative and safe policies. `amTFT` agents learn form pixels in a 5x5 grid and develop policies with cooperative properties seen in `TFT`. Foerster et al. [16] use a simpler 3x3 grid to assess `LOLA` agents with and without opponent policy information. They gain little success with modelled opponents but achieve better results when perfect opponent information is provided. Raileanu et al. [41] use a more complex fully collaborative version, with coins of 3 different colours in a 8x8 grid to test another opponent modelling algorithm, SOM (Self-other model). More recent work of Lu et al. [34] evaluate the performance of Model Free Opponent Shaping (MFOS) in a 3x3 grid similar to [16] and report a significant performance improvement.

We simulate the coin game restricting the memory of agents to each player's last action. A state of the game is represented by a vector (d_f, d_o, l, h_f, h_o), where d_f is the distance from the focal agent to the coin, d_o is the distance from the opponent to the coin, l is the coin colour, h_f is the focal agent's cooperation history, and h_o is the opponent's cooperation history. To interpret each other's move, agents use the following heuristic: moving towards a coin of the opponent's colour is perceived as defection and moving away from a coin of different colour is perceived as cooperation. Distance to the coin is defined as the minimum number of steps required to reach the coin, and cooperation history can be −1,1 or 0 , depending on whether the agent has defected, cooperated or unknown (at the beginning). This simplified version facilitates the intrinsic reward calculation for the BRG agents. We use a 3x3 grid where a new coin is always generated 2 steps away from both agents at all times. We expect the game to scale easily provided appropriate training.

For experiments we used policy gradient with learning rate 0.001, batch size 20 and 100-step episodes. To ensure continuous exploration we clip policies to $\pi(s, a) \in (0.05, 0.95)$. Similar to IPD and SIPD we use value iteration for intrinsic reward calculation, but gradually anneal η from $0 \longrightarrow 0.5$. Setting $\eta = 0$ boosts exploration

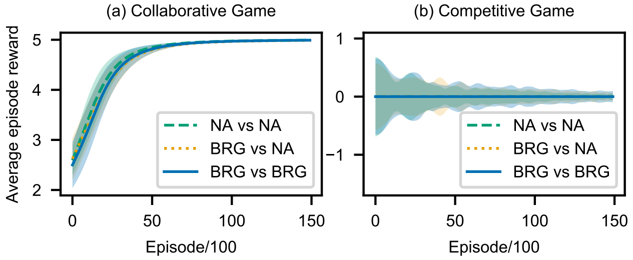

Figure 8: Performance trajectories over average episode returns as both players learn, in: (a) a collaborative game, and (b) matching pennies zero-sum game (NA: naïve agent, BRG: best-response guided agent).

to states with higher future rewards, slowly correcting back to $\eta = 0.5$ to encourage exploitation.

Results are presented for episodes with 100 steps. We performed experiments for scenarios, naïve against naïve, and BRG against BRG, and repeated 20 times till convergence to generate Figure 7(a). As expected, naïve learners converge to the greedy strategy where each agent attempts to pick all coins receiving an average reward of zero. BRG agents achieve a significant level of cooperation reaching an average reward of 10 where the reward for full cooperation is 25. As a better measure of cooperation we use the ratio, collected coins of own colour/total coins picked, Figure 7(b) , similar to [16].

Although a direct comparison of results is not meaningful with prior work we can qualitatively compare the results for the mixed-motive coin game. BRG agents achieve a collected coins of own colour/total coins picked ratio of around 0.65 and an average reward/reward for cooperation ratio of 0.4 which, as for our knowledge, is considerably higher to values in literature .

8 GAMES OTHER THAN COOPERATION

The method proposed does not hamper performance in games other than cooperation. In this section we evaluate the performance of self-aware agents in fully cooperative and competitive settings. For the cooperative setting, we used a symmetric game where agents receive a reward of 5 for mutual cooperation, 1 for mutual defection and 2 otherwise 8(a). For a competitive environment, we used matching pennies 8(b). In both cases we can see SA agents achieving the same equilibrium as naïve learners. This shows that this algorithm can perform even in situations where many different kinds of interactions arise.

9 CONCLUSIONS AND FUTURE WORK

Autonomous agents learning to maximise collective rewards will need to be robust to both the temptation to cheat others, and the risk of other agents attempting to exploit them. While such concerns can be addressed by modelling opponents explicitly when the population size is small and agents are individually identifiable, this strategy becomes intractable when the population is large and interactions are sparse. This situation is important. Consider the case of a self-driving car interacting with others. The agent will face a series of interactions, some of which have a cooperative structure, and some of which are more coordination or pure competition. At the same time, each interaction entails a new opponent. This kind of setup can also arise for example in energy markets, where depending on changes in the environment the structure of incentives can switch from cooperation-like, to fully competitive. All of this across large groups of agents, making opponent modelling impractical.

We present an effective approach to circumvent this problem, by augmenting an agent's internal reward signal with a best-response guided component derived from the best response to the agent's current policy, i.e., forgoing all information from opponents. We demonstrate that this reward signal allows our agents to consistently establish cooperation with naïve learning agents in an iterated prisoner's dilemma and a stochastic game with prisoner's dilemma-like reward structure. The resulting policies are robust to new entrants and shown to be significantly more scalable than opponent modelling approaches as the population grows. These results hold for a simple IPD game, but also for a Stochastic game featuring different states.

In addition, the algorithm performs equally well as naïve learning agents in fully competitive and cooperative environments. In games with higher state spaces like the coin-game, BRG agents achieve a significant level of cooperation compared to naïve learners that converge to the greedy approach.

Future work should focus on applying our reward structure to games and environments with more complex states. Examples include more explicit scenarios, including for example self-driving cars, where every intersection is akin to its own game but actions play out temporally, and smart energy solutions like voluntary load shedding to avoid blackouts and curtailment of solar panels under voltage constraints. Such settings will also motivate methods other than optimal best-response to calculate the intrinsic reward, instead learning the signal in a separate self-play reinforcement loop.

REFERENCES

[1] 2018. Pytorch implementation of LOLA using DiCE. https://github.com/alexis-jacq/LOLA_DiCE. Accessed: 2022-10-25.

[2] Han The Anh, Luís Moniz Pereira, and Francisco C Santos. 2011. Intention recognition promotes the emergence of cooperation. *Adaptive Behavior* 19, 4 (Aug 2011), 264–279. https://doi.org/10.1177/1059712311410896

[3] Robert Axelrod and William Donald Hamilton. 1981. The evolution of cooperation. *science* 211, 4489 (1981), 1390–1396.

[4] Richard Bellman. 1957. A Markovian decision process. *Journal of mathematics and mechanics* 6, 5 (1957), 679–684.

[5] Michael Bowling and Manuela Veloso. 2002. Multiagent learning using a variable learning rate. *Artificial Intelligence* 136, 2 (2002), 215–250.

[6] George W Brown. 1951. Iterative solution of games by fictitious play. *Activity analysis of production and allocation* 13, 1 (1951), 374–376.

[7] Valerio Capraro, Ismael Rodriguez-Lara, and Maria J Ruiz-Martos. 2020. Preferences for efficiency, rather than preferences for morality, drive cooperation in the one-shot Stag-Hunt Game. *Journal of Behavioral and Experimental Economics* 86 (2020), 101535.

[8] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. 2007. A short introduction to computational social choice. In *International Conference on Current Trends in Theory and Practice of Computer Science*. Springer, 51–69.

[9] Caroline Claus and Craig Boutilier. 1998. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI* 1998, 746-752 (1998), 2.

[10] Vincent Conitzer and Tuomas Sandholm. 2007. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning* 67, 1-2 (2007), 23–43.

[11] Allan Dafoe, Yoram Bachrach, Gillian Hadfield, Eric Horvitz, Kate Larson, and Thore Graepel. 2021. Cooperative AI: machines must learn to find common ground.

[12] Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R McKee, Joel Z Leibo, Kate Larson, and Thore Graepel. 2020. Open problems in cooperative AI. *arXiv preprint arXiv:2012.08630* (2020).

[13] Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. 2017. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*. 2951–2960.

[14] Prashant Doshi, Piotr Gmytrasiewicz, and Edmund Durfee. 2020. Recursively modeling other agents for decision making: A research perspective. *Artificial Intelligence* 279 (2020), 103202.

[15] Tom Eccles, Edward Hughes, János Kramár, Steven Wheelwright, and Joel Z Leibo. 2019. The Imitation Game: Learned Reciprocity in Markov games.. In *AAMAS*. 1934–1936.

[16] Jakob Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. 2018. Learning with Opponent-Learning Awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 122–130.

[17] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[18] Jakob Foerster, Gregory Farquhar, Maruan Al-Shedivat, Tim Rocktäschel, Eric Xing, and Shimon Whiteson. 2018. Dice: The infinitely differentiable monte carlo estimator. In *International Conference on Machine Learning*. PMLR, 1529–1538.

[19] Jakob Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip HS Torr, Pushmeet Kohli, and Shimon Whiteson. 2017. Stabilising experience replay for deep multi-agent reinforcement learning. In *International conference on machine learning*. PMLR, 1146–1155.

[20] Jakob N Foerster, Yannis M Assael, Nando De Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. *arXiv preprint arXiv:1605.06676* (2016).

[21] Julián García and Arne Traulsen. 2019. Evolution of coordinated punishment to enforce cooperation from an unbiased strategy space. *Journal of the Royal Society Interface* 16, 156 (2019), 20190127.

[22] Julián García and Matthijs van Veelen. 2018. No strategy can win in the repeated prisoner's dilemma: linking game theory and computer simulations. *Frontiers in Robotics and AI* 5 (2018), 102.

[23] Piotr J Gmytrasiewicz and Prashant Doshi. 2005. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research* 24 (2005), 49–79.

[24] Pablo Hernandez-Leal, Michael Kaisers, Tim Baarslag, and Enrique Munoz de Cote. 2017. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv preprint arXiv:1707.09183* (2017).

[25] Christian Hilbe, Štěpán Šimsa, Krishnendu Chatterjee, and Martin A Nowak. 2018. Evolution of cooperation in stochastic games. *Nature* 559, 7713 (2018), 246–249.

[26] Teck-Hua Ho and Xuanming Su. 2013. A dynamic level-k model in sequential games. *Management Science* 59, 2 (2013), 452–469.

[27] Trong Nghia Hoang and Kian Hsiang Low. 2013. Interactive POMDP lite: towards practical planning to predict and exploit intentions for interacting with self-interested agents. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. 2298–2305.

[28] Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. 2020. "Other-Play" for Zero-Shot Coordination. In *International Conference on Machine Learning*. PMLR, 4399–4410.

[29] Junling Hu and Michael P Wellman. 2003. Nash Q-learning for general-sum stochastic games. *Journal of machine learning research* 4, Nov (2003), 1039–1069.

[30] Marc Lanctot, Vinicius Zambaldi, Audrūnas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. 2017. A unified game-theoretic approach to multiagent reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 4193–4206.

[31] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2016. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182* (2016).

[32] Adam Lerer and Alexander Peysakhovich. 2017. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv:1707.01068* (2017).

[33] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 6382–6393.

[34] Chris Lu, Timon Willi, Christian Schroeder de Witt, and Jakob Nicolaus Foerster. 2022. Model-Free Opponent Shaping. In *ICLR 2022 Workshop on Gamification and Multiagent Solutions*.

[35] Richard Mealing and Jonathan L Shapiro. 2015. Opponent modeling by expectation–maximization and sequence prediction in simplified poker. *IEEE Transactions on Computational Intelligence and AI in Games* 9, 1 (2015), 11–24.

[36] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. 2017. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* 356, 6337 (2017), 508–513.

[37] Roger B. Myerson. 1997. *Game theory: analysis of conflict*. Harvard University Press.

[38] Alexander Peysakhovich and Adam Lerer. 2018. Prosocial Learning Agents Solve Generalized Stag Hunts Better than Selfish Ones. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 2043–2044.

[39] Jeremy Pitt, Lloyd Kamara, Marek Sergot, and Alexander Artikis. 2006. Voting in Multi-Agent Systems. *Comput. J.* 49, 2 (2006), 156–170. https://doi.org/10.1093/comjnl/bxh164

[40] William H Press and Freeman J Dyson. 2012. Iterated Prisoner's Dilemma contains strategies that dominate any evolutionary opponent. *Proceedings of the National Academy of Sciences* 109, 26 (2012), 10409–10413.

[41] Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. 2018. Modeling others using oneself in multi-agent reinforcement learning. In *International conference on machine learning*. PMLR, 4257–4266.

[42] David G. Rand and Martin A. Nowak. 2013. Human Cooperation. *Trends in Cognitive Sciences* 17, 8 (Aug. 2013), 413–425. https://doi.org/10.1016/j.tics.2013.06.003

[43] William H. Sandholm. 2010. *Population Games and Evolutionary Dynamics*. MIT Press.

[44] Fernando P. Santos, Jorge M. Pacheco, Ana Paiva, and Francisco C. Santos. 2019. Evolution of Collective Fairness in Hybrid Populations of Humans and Agents. *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (July 2019), 6146–6153. https://doi.org/10.1609/aaai.v33i01.33016146

[45] Fernando P. Santos, Francisco C. Santos, and Jorge M. Pacheco. 2018. Social Norm Complexity and Past Reputations in the Evolution of Cooperation. *Nature* 555, 7695 (March 2018), 242–245. https://doi.org/10.1038/nature25763

[46] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484–489.

[47] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[48] Karl Tuyls and Ann Nowé. 2005. Evolutionary Game Theory and Multi-Agent Reinforcement Learning. *The Knowledge Engineering Review* 20, 1 (March 2005), 63–90. https://doi.org/10.1017/S026988890500041X

[49] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[50] Chao Wen, Xinghu Yao, Yuhui Wang, and Xiaoyang Tan. 2020. Smix (λ): Enhancing centralized value functions for cooperative multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7301–7308.

[51] Mark P. Woodward and Robert J. Wood. 2012. Learning from Humans as an I-POMDP. arXiv:1204.0274 [cs.RO]