Learning in Nonzero-Sum Stochastic Games with Potentials

David Mguni [1] Yutong Wu [2] Yali Du [3] Yaodong Yang [1 3] Ziyi Wang [2] Minne Li [3] Ying Wen [4] Joel Jennings [1]
Jun Wang [3]

Abstract

Multi-agent reinforcement learning (MARL) has become effective in tackling discrete cooperative game scenarios. However, MARL has yet to penetrate settings beyond those modelled by team and zero-sum games, confining it to a small subset of multi-agent systems. In this paper, we introduce a new generation of MARL learners that can handle *nonzero-sum* payoff structures and continuous settings. In particular, we study the MARL problem in a class of games known as stochastic potential games (SPGs) with continuous state-action spaces. Unlike cooperative games, in which all agents share a common reward, SPGs are capable of modelling real-world scenarios where agents seek to fulfil their individual goals. We prove theoretically our learning method, **SP**ot-**AC**, enables independent agents to learn Nash equilibrium strategies in *polynomial time*. We demonstrate our framework tackles previously unsolvable tasks such as *Coordination Navigation* and *large selfish routing games* and that it outperforms the state of the art MARL baselines such as MADDPG and COMIX in such scenarios.

1. Introduction

Many real-world systems give rise to multi-agent systems (MAS); traffic network systems with autonomous vehicles (Ye et al., 2015; Zhou et al., 2020), network packet routing systems (Wiering, 2000) and financial trading (Mariano et al., 2001) are some examples. In these systems, self-interested agents act in a shared environment, each seeking to perform some pre-specified task. Each agent's actions affect the performance of other agents and may even prevent them from completing their tasks altogether. For example, autonomous vehicles seeking to arrive at their individual destinations must avoid colliding with other vehicles. Therefore to perform their task the agents must account for other agents' behaviours.

There is therefore a great need for reinforcement learning (RL) agents with their own goals to learn to perform in MAS. In these scenarios, agents are not required to behave as a team nor as perfect adversaries. These settings are modelled by *nonzero-sum* stochastic games (SGs) whose solution concept is a fixed point known as Nash equilibrium (NE). An NE describes the stable point in which all agents respond optimally to the actions of other agent. Computing the NE is therefore central to solving MAS.

Despite its fundamental importance for solving many MAS, computing the NE of *any* SG with a general payoff structure remains an open challenge (Yang & Wang, 2020). Presently, methods to compute NE in SGs that are neither zero-sum nor team settings are extremely scarce and impose limiting assumptions. As such, the application of these methods is generally unsuitable for real world MAS (Shoham & Leyton-Brown, 2008). Moreover, finding NE even in the simple case of normal form games (where agents take only a single action) is generally intractable when the game is nonzero-sum (Chen et al., 2009).

Among multi-agent reinforcement learning (MARL) methods are a class of algorithms known as independent learners e.g. independent Q learning (Tan, 1993). These algorithms ignore actions of other agents and are ill-suited to tackle MAS and often fail to learn (Hernandez-Leal et al., 2017). In contrast, algorithms such as MADDPG (Lowe et al., 2017), COMA (Foerster et al., 2018) and QDPP (Yang et al., 2020) include a centralised critic that accounts for the actions of all agents. To date none of these algorithms have been proven to converge in SGs that are neither team nor zero-sum (adversarial). Additionally, these methods suffer from combinatorial growth in complexity with the number of agents (Yang et al., 2019) leading to prohibitively expensive computations in some systems. There is also a noticeable lack of MARL methods that can handle continuous spaces which is required for tasks such as physical control (Bloembergen et al., 2015). This has left MARL largely unable to solve various practical tasks such as multi-

[1] Huawei R&D UK [2] Institute of Automation, Chinese Academy of Sciences [3] University College London, UK [4] Shanghai Jiao Tong University. Correspondence to: David Mguni <davidmguni@hotmail.com>, Yaodong Yang <yaodong.yang@outlook.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

agent Mujoco (de Witt et al., 2020) which remains an open challenge. This is in contrast to discrete counterpart settings e.g. Starcraft micro-management in which MARL has had notable success (Peng et al., 2017).

In this paper, we address the challenge of solving MAS with payoff structures beyond zero-sum and team game settings in continuous systems. In particular, we develop a MARL solver that computes the NE within a new subclass of continuous nonzero-sum SGs, namely continuous stochastic potential games (c-SPGs) in which the agents' interaction at each stage has a potential game property. Lastly, our solver avoids combinatorial complexity with the number of agents.

Our framework is developed through theoretical results that enable the NE of some SGs to be found tractably. First, we formalise a construction of continuous SGs in which the interaction between agents at each stage can be described by a PG. Thereafter, we show that the NE of the SG can be computed by solving a *dual* Markov decision process (MDP) whose solution *exactly coincides* with the NE of the original SG. This converts the problem of finding a fixed point NE of an (a priori unknown) nonzero-sum SG to solving an (unknown) MDP whose solution as we show, can be found tractably using a new distributed variant of actor-critic methods, which we call **SP**ot-**AC**.

The paper is organised as follows: after the related work next, we present our construction of c-SPGs in Sec. 3. We continue in Sec. 4 to present a simple planner when the environment model is given and prove that c-SPGs have dual representations as MDPs. A polynomial-time fitted Q-learning solver (**SP**ot**Q**) is then given to find the NE in this setting. In Sec. 5, we extend the learning method and propose an actor-critic variant (**SP**ot-**AC**) that solves c-SPGs in unknown environments. A fully distributed variant is also provided that scales with the number of agents. Robustness analysis is followed and we show that the method closely approximates the NE solution when the construction of the potential function has small estimation errors. Lastly in Sec. 6, we conduct detailed ablation studies and performance tests on various tasks and conclude the paper.

2. Related Work

MARL has been successful in zero-sum scenarios (Grau-Moya et al., 2018) and settings of homogeneous agents with population sizes that approach infinity (Mguni et al., 2018; Yang et al., 2018) and team game scenarios (Peng et al., 2017). However, the restrictions on the payoffs therein means that these models are usually far away from many real-world scenarios, prohibiting the deployment of MARL therein. There have been few attempts at computing NE in settings outside of team and zero-sum SGs. Most notably is Nash Q-learning (Hu & Wellman, 2003); it however imposes stringent assumptions that force the SG to resemble a team game. For example, in (Hu & Wellman, 2003) at each iteration a unique Pareto dominant NE must exist and be computed which is generally unachievable. 'Friend or foe' learning (Littman, 2001) establishes convergence to NE in two-player *coordination games* but requires known reward functions and solving a linear program at each time step. Zhang et al. (2020) adopts the *stackelberg equilibrium* as the learning target. More recently, Lowe et al. (2017) suggests an actor-critic method (MADDPG) with centralised training on the critic. Nevertheless Lowe et al. (2017) do not tackle SGs outside of the zero-sum or cooperative cases in either theoretical results or experiments. In particular, the experiments in (Lowe et al., 2017) are all aimed at either adversarial (zero-sum) or the fully cooperative settings.

Very recently (Zhang et al., 2021) consider an SG setting in which all agents' value functions are assumed to satisfy a global PG condition, that is, the incentive of all agents to change their *policies* can now be expressed using a single global function. As noted in their discussion, without further qualification, this assumption is rather strong and difficult to verify except in the case in which all agents share the same objective. In a later work, (Leonardos et al., 2021) consider an SG setting with a PG game property while imposing conditions that either i) reduce the SG to a linear combination of normal form games and removes all planning aspects or ii) limit the agents' interaction to a term in their reward that does not depend on either the state or the agent's own actions. The latter condition (ii) results in an SG which is a restrictive case of our SG, in particular, our SG captures a richer, more general set of strategic interactions between agents (see Sec. 3.1 for a more detailed discussion).

We tackle a subclass of SGs which satisfy a PG condition at each stage game. We then show that with this construction, a potentiality property can be naturally extrapolated to the value functions of the SG without imposing restrictive assumptions. With this we prove that the NE of the game can be learned by independent agents without the need to impose restrictive assumptions as in (Hu & Wellman, 2003; Littman, 2001) and in a way that scales with the number of agents; in contrast to centralised critic methods that scale combinatorially.

3. Continuous Stochastic Potential Games

Continuous Stochastic Games

MAS are modelled by SGs (Shoham & Leyton-Brown, 2008; Shapley, 1953). An SG is an augmented MDP involving two or more agents $\{1, 2, \ldots, N\} =: \mathcal{N}$ that simultaneously take actions over many (possibly infinite) rounds. Formally, a continuous SG is a tuple $\mathcal{G} = \langle \mathcal{N}, \mathcal{S}, (\mathcal{A}_i)_{i\in\mathcal{N}}, P, (R_i)_{i\in\mathcal{N}}, \gamma\rangle$ where $\mathcal{S}$ is the set of states, $\mathcal{A}_i \subset \mathbb{R}^q$ is an action set and $R_i : \mathcal{S} \times \boldsymbol{\mathcal{A}} \rightarrow \mathcal{P}(D)$ is the

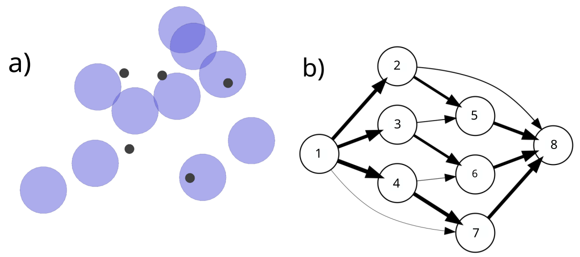


Figure 1. **a)** Coordination Navigation: selfish agents (purple) seek to reach rewards (black) whilst minimising contact with each other. **b)** Routing Networks: agents split their own commodity flow between edges in a network over a sequence of time-steps. Starting at a source (node 1) and arriving at a target node (8), paths that have more commodity incur higher congestion costs.

distribution reward function for agent $i \in \mathcal{N}$ where D is a compact subset of $\mathbb{R}$ and lastly, $P : \mathcal{S} \times \boldsymbol{\mathcal{A}} \times \mathcal{S} \to [0, 1]$ is the probability function $P : \mathcal{S} \times \boldsymbol{\mathcal{A}} \times \mathcal{S} \to [0, 1]$ describing the system dynamics where $\boldsymbol{\mathcal{A}} := \times_{i=1}^{N} \mathcal{A}_i$.

In an SG, at each time $t \in 0, 1, \ldots$, the system is in state $s_t \in \mathcal{S}$ and each agent $i \in \mathcal{N}$ takes an action $a_t^i \in \mathcal{A}_i$. The *joint action* $\boldsymbol{a}_t = (a_t^1, \ldots, a_t^N) \in \boldsymbol{\mathcal{A}}$ produces an immediate reward $R_i(s_t, \boldsymbol{a}_t)$ for agent $i \in \mathcal{N}$ and influences the next-state transition which is chosen according to P. Using a (parameterised) *Markov strategy*[1] $\pi_{i,\boldsymbol{\eta}^i} : \mathcal{S} \times \mathcal{A}_i \to [0, 1]$ to select its actions, each agent i seeks to maximise its *individual* expected returns as measured by its value function: $v_i^{\pi^i, \pi^{j \neq i}}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_i(s_t, \boldsymbol{a}_t) \middle| \boldsymbol{a}_t \sim (\pi^i_{\boldsymbol{\eta}^i}, \pi^{j \neq i}_{\boldsymbol{\eta}^{-i}})\right]$ where $\boldsymbol{\eta}^i \in E_i \subset \mathbb{R}^l$ and Π_i is a compact Markov strategy space. A *pure strategy* (PS) is a map $\pi_i : \mathcal{S} \to \mathcal{A}_i$, for any $i \in \mathcal{N}$ that assigns to any state an action in $\mathcal{A}_i$.

We denote the space of joint policies by $\boldsymbol{\Pi} := \times_{i \in \mathcal{N}} \Pi_i$; where it will not cause confusion (and with a minor abuse of notation) we use the shorthands $\pi_i \equiv \pi^i_{\boldsymbol{\eta}_i}$ and $f^{(\pi^i, \pi^{-i})}(s) = f(s, \pi^i, \pi^{-i}) \equiv \mathbb{E}_{\pi^i, \pi^{-i}}[f(s, a^i, a^{-i})]$.

SGs can be viewed as a sequence of *stage games* $\{\mathcal{M}(s)\}_{s \in \mathcal{S}}$ that take place at each time step where $\mathcal{M}(s) = \langle (\mathcal{A}_i)_{i \in \mathcal{N}}, (R_i(s))_{i \in \mathcal{N}}, \mathcal{N} \rangle$. Therefore, at each time step a stage game is played and then the game transitions to the next stage game which is selected according to P.

Continuous Stochastic Potential Games

We now introduce a new subset of SGs namely c-SPGs which is the framework of our approach.

Definition 1. *An SG is a c-SPG if for all states there exists a function* $\phi : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \mathbb{R}$ *such that the following holds for any* $(a^i, a^{-i}), (a'^i, a^{-i}) \in \boldsymbol{\mathcal{A}}$ *where* $a_t^{-i} := (a^1, \ldots a^{i-1}, a^{i+1}, \ldots, a^N)$*,* $\forall i \in \mathcal{N}, \forall s \in \mathcal{S}$*:*

$$\begin{aligned} &R_i(s, (a^i, a^{-i})) - R_i(s, (a'^i, a^{-i})) \\ &= \phi(s, (a^i, a^{-i})) - \phi(s, (a'^i, a^{-i})). \end{aligned} \tag{1}$$

Condition (1) says that the difference in payoff from a deviation by one of the agents is exactly quantified by a global function ϕ that does not depend on the agent's identity. We call ϕ the *potential function* or potential for short. The condition extends the notion of static *one-shot* potential games (PGs) (Monderer & Shapley, 1996b) to a continuous SG setting that now includes states and transition dynamics.

To complete the construction we introduce a condition which is a natural extension of PGs to state-based settings:

Definition 2. *A stage game* $\mathcal{M}(s)$ *is state transitive if there exists a* $\phi : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \mathbb{R}$ *s.th.* $\forall (a^i, a^{-i}) \in \boldsymbol{\mathcal{A}}, \forall i \in \mathcal{N}, \forall s, s' \in \mathcal{S}$*:*

$$\begin{aligned} &R_i(s, (a^i, a^{-i})) - R_i(s', (a^i, a^{-i})) \\ &= \phi(s, (a^i, a^{-i})) - \phi(s', (a^i, a^{-i})). \end{aligned} \tag{2}$$

The intuition is that the difference in rewards for changing state is the same for each agent. Some classic examples of where state transitivity holds are anonymous games (Daskalakis & Papadimitriou, 2007), symmetric SGs (Jaśkiewicz & Nowak, 2018), team SGs (Cheng et al., 2017).

Our results are built under the assumption[2] that state transitivity assumption holds.

Fig. 1 illustrates two examples of c-SPGs. For instance, in Coordination Navigation, one can verify that the state transitivity assumption is satisfied: a collection of agents seeks to arrive at some destination $x^\star \in \mathbb{R}^p$. Crucially, the agents must avoid colliding with other agents. Each agent's value function is given by:

$$\begin{aligned} V_i^{\boldsymbol{\pi}}(\boldsymbol{x}) = \frac{1}{2}\mathbb{E}\Bigg[\sum_{t=0}^{\infty} \gamma^t \Big\{ &K_i - \alpha \|x_{i,t} - x^\star\|^2 \\ &- \beta \sum_{j \in \mathcal{N}/\{i\}} (\|x_{i,t} - x_{j,t}\|^2 + \epsilon)^{-1/2} - \|a_{i,t}^T - \rho\|_M^2 \Big\}\Bigg] \end{aligned}$$

where $\|\cdot\|$ and $\|\cdot\|_M$ are Euclidean and Mahalanobis norms respectively, $x_{i,t} \in \mathbb{R}^p$ is the position of agent i at time $t \in \mathbb{N}$ and $\boldsymbol{x}_t = (x_{1,t}, \ldots, x_{N,t}) \in \mathbb{R}^p$; $c, \rho, H, \alpha, \beta\{K_i\}_{i \in \mathbb{N}}$ are constants and $a_{i,t}$ is vector representing the action taken by agent i. It can be readily verified that the game is potential with the following potential function: $\phi^{\boldsymbol{\pi}}(\boldsymbol{x}) = -\alpha\|x - x^\star\|^2 - \beta \sum_{j \in \mathcal{N}} \left(\|x - x_{j,t}\|^2 + \epsilon\right)^{-1/2} + \beta c^{-2} - \|a^T - \rho\|^2$. Similarly, it can be readily verified that the game satisfies the state transitivity assumption.

[1] A Markov strategy requires as input only the current state (and not the game history or other agents' actions or strategies).

[2] Statements of the technical assumptions are in the Appendix.

In our Ablation experiments (see Sec. 6) we show that our method is able to tackle settings in which the potentiality and state transitivity conditions are mildly violated.

C-SPGs also hold in SGs in which the agents have the same reward functions (identical interest games)(Monderer & Shapley, 1996a) such as anonymous games (Daskalakis & Papadimitriou, 2007), team games (Wang & Sandholm, 2003) and mean field games (Mguni et al., 2018). Such SGs are widely used to study distributive systems and coordination problems in MAS such as Starcraft (Samvelyan et al., 2019) and Capture the Flag (Jaderberg et al., 2019). MARL frameworks such as COMA (Foerster et al., 2018), QMIX (Rashid et al., 2018) and QDPP (Yang et al., 2020) are fully cooperative settings and therefore fall within this category.

A key result we prove is that c-SPGs enjoy a dual representation as MDPs therefore enabling their solution to be computed by tackling MDPs. To construct a solution method for c-SPGs, we resolve a number of challenges: **i)** The first involves determining the dual MDP whose solution is to be learned through interaction with the environment. **ii)** The second involves developing a tractable learning procedure that ensures convergence to the game solution. To do this we develop a method that finds the solution of the dual MDP distributively, in doing so we also resolve the problem of combinatorial complexity that afflict MARL methods. **iii)** The method of determining the dual MDP (**i)**) can incur small errors. Our last challenge is to show that small errors in the construction of the dual MDP induce only small errors in the agents' best response actions.

3.1. Link to Potential Games and Discussion

We briefly continue the discussion on related works with a relevant review of PGs. The first systematic treatment of PGs appeared in (Monderer & Shapley, 1996b) in a static setting. PGs constitute a fundamental building block of general-sum games - any general-sum game can be decomposed into two (strategic) parts; PGs and *harmonic games* (Candogan et al., 2011). PGs model many real-world scenarios including traffic network scenarios, network resource allocation (Zazo et al., 2015)) social conflicts (Lã et al., 2016) and consensus problems (Marden et al., 2009). PGs also encompass all team games and some zero-sum games (Balduzzi et al., 2018).

C-SPGs extend PGs to settings with dynamics and future uncertainty. This enables PGs to capture real-world scenarios that involve sequential decision-making and dynamics. Example of these settings traffic networks models, routing and packet delivery problems.

The analysis of dynamic PGs is extremely sparse and does not cover (reinforcement) learning settings in which the system is a priori unknown. In the direction of incorporating potentiality property within an SG, (González-Sánchez & Hernández-Lerma, 2013; Macua et al., 2018) consider an SG in which the potentiality property is *imposed* on the value functions which results in the need for highly restrictive assumptions. In (González-Sánchez & Hernández-Lerma, 2013) the SG is restricted to concave reward functions (in the state variable) and the transition function is required to be invertible (and known). These assumptions are generally incompatible with many MAS settings of interest.[3] Similarly, (Macua et al., 2018) study a discrete Markov game in which the value function is *assumed* to satisfy a PG property. Their construction requires that the agents' policies depend only on disjoint subcomponents of the state which prohibits non-local (strategic) interactions.

Very recently (Zhang et al., 2021) consider an SG setting in which all agents' value functions are assumed to satisfy a global PG property, that is, the incentive of all agents to change their *policies* can now be expressed using a single global function. To construct this relationship using conditions on the stage game, in a later work (Leonardos et al., 2021) consider an SG setting and embed either of two properties into the game structure namely, an *agent-independent transition assumption* (C.1) or an *equality of individual dummy term assumption* (C.2). Using either of these conditions *and* the stage game PG condition (Condition (1)), they show that the PG condition can be extrapolated to a global PG condition on the value functions.

Conditions C.1. and C.2. in (Leonardos et al., 2021) impose heavy restrictions since Condition C.1. reduces the SG to a linear combination of normal form games and removes all planning aspects (hence extrapolating the potentiality of stage games to the agents' value functions is deduced trivially). Condition C.2. restricts the noncooperative (strategic) interaction part of the game to a term that does not depend on the state or the agent's own action. Moreover imposing condition C.2. produces an SG that is a special case of our SG (this can be seen using the equivalence expression in Lemma B (see Sec. H in Appendix) by setting $k(s) \equiv 1$ and restricting h_i to depend only on *other agents'* actions in the reward functions of our SG). Therefore, the generalisation of the PG condition to SGs in (Leonardos et al., 2021) requires strong limitations on the structure of the SG not present in our analysis.

With our new construction which has a PG at each stage game we show that the PG condition can be naturally extrapolated to the value functions of the SG. This provides verifiable assumptions on the game while imposing relatively weak assumptions on the SG in comparison to (Leonardos et al., 2021). With this we prove that the equilibrium of

[3] The result in (González-Sánchez & Hernández-Lerma, 2013) also requires verifying the policy satisfies sufficiency conditions which is generally difficult given the size of the space of functions.

the SG can be found by merely solving an (unknown) MDP without imposing either state disjointness as in (Macua et al., 2018) or concavity as in (González-Sánchez & Hernández-Lerma, 2013).

4. Planning in c-Stochastic Potential Games

We now show that the stable point solution of a c-SPG can be computed tractably by solving a *dual MDP* with reward function ϕ. This leads to a vast reduction in complexity for finding NEs in our c-SPG subclass of nonzero-sum. In what follows, we assume that the environment is known; in Sec. 5 we extend the analysis of this section to unknown environments. We defer the proofs of the results of the following sections to the Appendix.

As SGs are *noncooperative settings*, the solution cannot be described as an optimisation of a single objective. The appropriate solution concept is the following NE variant (Fudenberg & Tirole, 1991):

Definition 3. *A strategy profile* $\hat{\boldsymbol{\pi}} = (\hat{\pi}_i, \hat{\pi}_{-i}) \in \boldsymbol{\Pi}$ *is a Markov perfect equilibrium (MPE) if* $\forall i \in \mathcal{N}$*:*

$$v_i^{(\hat{\pi}_i, \hat{\pi}_{-i})}(s) \geq v_i^{(\pi'_i, \hat{\pi}_{-i})}(s),\ \forall s \in \mathcal{S},\ \forall \pi'_i \in \Pi_i.$$

The condition characterises a fixed point in strategies in which no agent can improve their expected payoff by unilaterally deviating from their current policy. We denote the set of NE of $\mathcal{G}$ by $NE\{\mathcal{G}\}$. Finding NE of nonzero-sum SGs in general involves using fixed point methods which are generally intractable (Chen et al., 2009). Indeed, finding NE in SGs is PPAD complex (Polynomial Parity Arguments on Directed graphs) (Chen et al., 2009) for which brute force methods are intractable. Finding efficient solution methods for nonzero-sum SGs is an open challenge (Shoham & Leyton-Brown, 2008).

We now show that c-SPGs exhibit special properties that enable their NE to be computed tractably. In particular, we show that computing the NE of c-SPGs can be achieved by solving an MDP. With this, solving c-SPGs can be approached with stochastic approximation tools. We then present a new Q-learning variant that solves c-SPGs in polynomial time.

To begin, we construct the Bellman operator of $\mathcal{G}$. Let $g : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \mathbb{R}$ and $F : \mathcal{S} \to \mathbb{R}$, for any $s \in \mathcal{S}$ the Bellman operator of the game $\mathcal{G}$ is given by the following:

$$[T_g F](s) := \sup_{\boldsymbol{a} \in \mathcal{A}} \left[g(s, \boldsymbol{a}) + \gamma \int_{s' \in \mathcal{S}} ds' P(s'; \boldsymbol{a}, s) F[s']\right].$$

We now state our first key result which reveals a striking property of the c-SPG class of games:

Theorem 1. *Let* $V : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \mathbb{R}$ *be a test function, then* $\mathcal{G}$ *possesses a fixed point NE in pure (deterministic) Markov strategies characterised by:*

$$\lim_{k \to \infty} T_\phi^k V^{\boldsymbol{\pi}} = \sup_{\hat{\boldsymbol{\pi}} \in \boldsymbol{\Pi}} V^{\hat{\boldsymbol{\pi}}},$$

where ϕ *is the potential of* $\mathcal{G}$*.*

The result states that the MPE of the game exist and in *pure strategies* and correspond to solution of a (dual) MDP $\mathcal{M}^\dagger := \langle \phi, \times_{i \in \mathcal{N}} \mathcal{A}_i, P, \mathcal{S}, \gamma \rangle$. In fact, it is shown that any MPE is a local optimum of the value function associated to $\mathcal{M}$. The value function of $\mathcal{M}$ which we call the *dynamic potential function* (DPF), B, is constructed by $B^{\boldsymbol{\pi}}(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t, \boldsymbol{a}_t) | \boldsymbol{a}_t \sim \boldsymbol{\pi}], \forall s \in \mathcal{S}, \forall \boldsymbol{\pi} \in \boldsymbol{\Pi}$.

The theorem is proven inductively within a dynamic programming argument to extrapolate the potentiality property to the entire SG then showing $\mathcal{G}$ is *continuous at infinity*.

Theorem 1 enables us to compute the MPE by solving an MDP, a task which can be performed in polynomial time.[4] Moreover, Theorem 1 enables a Q-learning approach (Bertsekas, 2012) for finding the MPE of the game. The following fitted Q-learning method computes the approximate B function and the corresponding optimal policy for each agent.

First, let us define by

$$Y_{l_k}(s_{l_k}, \boldsymbol{a}_{l_k}, s'_{l_k}) := \phi_{l_k, \hat{\rho}}(s_{l_k}, \boldsymbol{a}_{l_k}) + \gamma \sup_{\boldsymbol{a'}} \mathbb{E}_{\mathbb{P}} \left[\hat{B}_l\right](s'_{l_k}, \boldsymbol{a'}) \tag{3}$$

At each iteration $k = 0, 1, \ldots$ we solve the minimisation:

$$F_l \in \arg\inf_{\mathcal{F} \in \mathcal{H}} \sum_{l_k=1}^{n_k} \left(Y_{l_k}(s_{l_k}, \boldsymbol{a}_{l_k}, s'_{l_k}) - [\mathcal{F}](s_{l_k}, \boldsymbol{a}_{l_k})\right)^2 \tag{4}$$

Algorithm 1 SPotQ: **S**tochastic **POT**ential **Q**-Learning

```
Input: discount factor $\gamma$ and PF $\phi$.
for $k \in 0, 1, 2, \cdots$ do
    for $i \in \mathcal{N}$ do
        Set the local target $Y_{l_k}$ by (3)
        Update $F$ by minimizing Eq. (4)
    end for
end for
Output: $F, (\pi_i^k)_{i \in \mathcal{N}}$.
```

The minimisation seeks to find the optimal action-value function $Q^\star$. Using this, we can construct our **SPotQ** algorithm that works by mimicking value iteration. By Theorem 1, the algorithm converges to the MPE of the game.

Theorem 1 does not establish uniqueness of B which could lead to ambiguity in the solution. The following result reduces the set of candidates to a single family of functions:

[4] The MDP lies is in a complexity class known as `P-SPACE` which can be solved tractably (Papadimitriou & Tsitsiklis, 1987).

Lemma 1. *If B_1, B_2 are value functions of the dual MDP $\mathscr{M}^\dagger$ then $(B_1^{\boldsymbol{\pi}} - B_2^{\boldsymbol{\pi}})(s) = c, \forall \boldsymbol{\pi} \in \boldsymbol{\Pi}, \forall s \in \mathcal{S}$ where $c \in \mathbb{R}$*

Therefore, the set of candidate functions are limited to a family of functions that differ only by a constant.

Computing the Potential Function ϕ

Theorem 1 requires knowledge of ϕ. Existing methods to find ϕ in PGs e.g. MPD method (Candogan et al., 2013) are *combinatorial* in actions and agents. Indeed, directly applying (1) to compute ϕ requires checking all deviations over pure strategies (deterministic policies) which is expensive since it involves sweeping through the joint action space $\boldsymbol{\mathcal{A}}$. We now demonstrate how to compute ϕ while overcoming these issues by transforming (1) into a differential equation. To employ standard RL methods we require parameterised policies and, in anticipation of tackling an RL setting we extend our coverage to *parameterised* stochastic policies.

Proposition 1. *In any c-SPG the following result holds $\forall s \in \mathcal{S}, \forall \boldsymbol{a} \in \mathcal{A}$, $\forall (\boldsymbol{\eta}^i, \boldsymbol{\eta}^{-i}) \in \boldsymbol{E}^{ps}$:*

$$\mathbb{E}_{\boldsymbol{\pi}(\boldsymbol{\eta})}\left[\frac{\partial \ln \pi_{i,\boldsymbol{\eta}^i}}{\partial \boldsymbol{\eta}^i}\left(\frac{\partial}{\partial a^i} + \frac{\partial}{\partial s}\right)(R_i - \phi)(s, \boldsymbol{a})\right] = \mathbf{0}. \quad (5)$$

The PDE serves as an analogue to the PG condition (1) which now exploits the continuity of the action space and the fact that they agents' actions are sampled from stochastic policies. Therefore Prop. 1 reduces the problem of finding ϕ to solving a PDE.

So far we have considered a planning solution method that solves the game when the agents reward functions are known upfront. In Sec 5, we consider settings in which the reward functions and the transition function are a priori unknown but the agents observe their rewards with noisy feedback.

5. Learning in c-Stochastic Potential Games

In RL, an agent learns to maximise its total rewards by repeated interaction with an unknown environment. The underlying problem is typically formalised as an MDP (Sutton & Barto, 2018). MARL extends RL to a multi-player setting (Yang & Wang, 2020). The underlying problem is modelled as an SG in which the rewards of each agent and transition dynamics are a priori unknown.

We have shown the MPE of a c-SPG can be computed by solving a *Markov team game* $\mathscr{M}^\dagger$, an SG in which all agents share the same reward function ϕ. We now discuss how to solve $\mathscr{M}^\dagger$ from observed data in unknown environments (i.e. if $P, \{R_i\}$ are not known). Additionally, we discuss our approach to enable easy scaling in the number of agents (and avoid combinatorial complexity) using distributive methods.

The scheme can be summarised in the following steps:

i) Compute the potential estimate $\hat{\phi}$ by solving the PDE in Prop. 1 using a distributed supervised learning method.

ii) Solve the team game $\hat{\mathscr{M}}^\dagger := \langle \mathcal{N}, \mathcal{S}, \boldsymbol{\mathcal{A}}, P, \hat{\phi}, \gamma\rangle$ with a distributed actor-critic method. The critic is updated with a distributed variant of the fitted Q-learning method in Sec. 4.

5.1. Learning the Potential Function ϕ

Though Prop. 1 reveals that ϕ can be found by solving a PDE, it involves evaluations in pure strategies which can be costly. Moreover, the result cannot be applied directly to estimate ϕ since the agents sample their rewards but not ϕ.

We now show how each agent can construct an approximation of ϕ in a way that generalises across actions and states by sampling its rewards. First, we demonstrate how the potential condition (1) can be closely satisfied using *almost pure* strategies. The usefulness of this will become apparent when we solve the PDE in Prop. 1 to find ϕ.

Lemma 2. *Let F be a bounded and continuous function and let $\Delta F(s_t, (a_t^i, a_t^{\prime i}), a_t^{-i}) := F(s_t, (a_t^{\prime i}, a_t^{-i})) - F(s_t, (a_t^i, a_t^{-i}))$ then there exists $c > 0$ such that*

$$\left|\Delta F(s_t, (a_t^i, a_t^{\prime i}), a_t^{-i}) - \Delta F(s_t, (\pi_i, \pi_i^\prime), \pi_{-i})\right| \leq c\|F\|_\infty \hat{\sigma}_\epsilon^2$$

where the policy $\pi_{i,\epsilon}$ is a nascent delta function[5] *and $\hat{\sigma}_\epsilon^2 := \max\{\mathrm{Var}(\pi_\mathrm{i}), \mathrm{Var}(\pi_\mathrm{i}^\prime)\}$.*

Since the bound approaches 0 in the limit as policies become pure strategies, the potential condition (5) is closely satisfied in nascent stochastic policies.

We now put Lemma 2 to use with a method to compute $\hat{\phi}$ that inexpensively solves the PDE condition (5) over the policy parameter space $\boldsymbol{E}$. Indeed, thanks to Lemma 2, we can learn ϕ through an optimisation over $\boldsymbol{E}$. The method uses a PDE solver over a set of randomly sampled points across $\boldsymbol{E} \times \mathcal{S}$ using the observed data $\{(s_k, \boldsymbol{a}_t, (r_{1,k}, \ldots, r_{N,k}))\}_{k \geq 0}$ where $r_{i,k} \sim R_i(s_k, \boldsymbol{a}_k)$.[6] Therefore, define by:

$$g^i(s, \boldsymbol{\eta}, \hat{\phi}) := \nabla_{\boldsymbol{\eta}^i} \ln \pi_{\epsilon,i}(a^i|s;\boldsymbol{\eta})\left(\frac{\partial}{\partial a^i} + \frac{\partial}{\partial s}\right)[R_i - \hat{\phi}](s, \boldsymbol{a})$$

where $\boldsymbol{\pi}_\epsilon(\boldsymbol{a}|s;\boldsymbol{\eta}) := \pi_{\epsilon,i}(a^i|s;\boldsymbol{\eta}^i)\pi_{\epsilon,-i}(a^{-i}|s;\boldsymbol{\eta}^{-i})$.

Following Prop. 1 we consider the following problem to

[5] A nascent delta function g_ϵ has the property $\lim_{\epsilon \downarrow 0} \int_X g_\epsilon f(x)dx = f(0)$ for any function f. They enable pure strategies to be approximated by stochastic policies with small variance. We denote a nascent policy by $\pi_{\epsilon,i}, \forall \epsilon > 0$.

[6] As with methods with sharing networks (e.g. COMIX, FacMADDPG (de Witt et al., 2020)), agents observe other agents' rewards. The method can be performed using only each agent's data $\{(s_k, \boldsymbol{a}_t, r_{i,k})\}$, however this requires more trajectory data.

compute $\hat{\phi}$:

$$\min_{\boldsymbol{\rho}\in\mathbb{R}^k} \|G(s,\boldsymbol{\eta};\hat{\phi}_{\boldsymbol{\rho}})\|^2_{\boldsymbol{E}\times\mathcal{S},\nu}, \tag{6}$$

where $G(s,\boldsymbol{\eta};\hat{\phi}) := [g^1(s,\boldsymbol{\eta};\hat{\phi}),\ldots,g^N(s,\boldsymbol{\eta};\hat{\phi})]^T$ and $\|f(y)\|^2 := \int_Y |f(y)|^2\nu(y)dy$ and $\nu(y)$ is a positive probability density on Y and $\boldsymbol{\rho}\in\mathbb{R}^k$ are parameters. The optimisation performs evaluations in mixed strategies which is computationally inexpensive. Using a weighted exponential sum method (Chang, 2015), the objective reduces to a least squares problem on a single objective $G(s,\boldsymbol{\eta},\boldsymbol{\rho}) := N^{-1}\sum_{i\in\mathcal{N}}\left(g^i(s,\boldsymbol{\eta},\hat{\phi}_{\boldsymbol{\rho}^i})\right)^2$. The optimisation can be solved with a function approximator on $\hat{\phi}_{\boldsymbol{\rho}}$ e.g. a deep neural network (NN). Under mild conditions (Bertsekas & Tsitsiklis, 2000) the method converges to a critical point of G, that is $\lim_{n\to\infty}\nabla_{\boldsymbol{\rho}}G = 0$. We defer the details of the method to the the Appendix.

Actor-Critic Method

We now return to tackling the problem of solving the team game $\mathscr{M}^\dagger := \langle\mathcal{N},\mathcal{S},\boldsymbol{\mathcal{A}},P,\hat{\phi},\gamma\rangle$. To enable the method to scale and handle continuous actions, we adapt the fitted Q-learning method in Sec. 4 to an actor-critic method (Konda & Tsitsiklis, 2000) for which each agent learns its own policy π_{i,η_i} using the estimate of B. The policy parameter η_i of the policy at the k^{th} iteration is updated through sampled deterministic policy gradients (DPGs) (Silver et al., 2014):

$$\begin{aligned}&\nabla\eta_i\hat{B}^{(\pi^k_{i,\eta_i},\pi^k_{-i,\eta_{-i}})}(s_{l_k})\\ &\approx\frac{1}{L}\sum_{l=1}^{L}\nabla_{\eta_i}\pi_{i,\eta_i}(\cdot|s_{l_k})\nabla_{a^i_{l_k}}F_k(s_{l_k},\boldsymbol{a}_{l_k})\Big|_{a^i_{l_k}\sim\pi^k_{i,\eta_i}}\end{aligned} \tag{7}$$

Equation (7) describes the actor update via a DPG. The complete process is described in Algorithm 1. It involves two optimisations in sequence: the agents individually compute the approximation $\hat{\phi}$ which is then used for computing $\hat{B}$, which approximates the optimal value function B by a Q-learning + decentralised DPG method and outputs each agent's MPE policy. Crucially the method avoids optimisations over the joint space $\times_{i\in\mathcal{N}}\mathcal{A}_i$ enabling easy scaling (in the number of agents) in this component of the algorithm.

Scaling in N using Consensus Optimisation

Although the above method represents progress for solving SGs, a scalability issue remains since estimating $\hat{\phi}$ involves a computation over the joint space $\boldsymbol{E}$. This becomes increasingly expensive with large numbers of agents. We now devise a fully distributed version of the method that scales with the number of agents. In this version, each agent i constructs an independent estimate of $\hat{\phi}$ by sampling across $E_i\times\mathcal{S}$ at each step using only its own observed data $\{(s_k,\boldsymbol{a}_t,r_{i,k})\}_{k\geq 0}$. The method includes a consensus step that enables $\hat{\phi}$ (and hence $\hat{B}$) to be accurately computed efficiently in a fully distributed fashion (Tutunov et al., 2019).

Algorithm 2 SPot-AC: **S**tochastic **POT**ential **A**ctor-**C**ritic

```
Input: discount factor $\gamma$, DPF and PF approximation maps
    $F, P_{\boldsymbol{\rho}} \in \mathscr{H}$ (resp.) ($\boldsymbol{\rho}\in\mathbb{R}^k$).
for $k \in 0,1,2,\cdots$ do
    Using $(\pi^k_i)_{i\in\mathcal{N}}$ to rollout, collect the trajectory data
    and save it in the buffer $\mathscr{D}$.
    for $i \in \mathcal{N}$ do
        // Learn the potential function $\hat{\phi}$
        Sample a random minibatch of $L$ samples
        $\{(s_{l_t},(a^i_{l_t})_{i\in\mathcal{N}},s_{l_{t+1}},(r^i_{l_t})_{i\in\mathcal{N}}\}$ from $\mathscr{D}$.
        Compute $\hat{\phi}$ by solving Expression (6)
        // Compute the value function $\hat{B}$
        Set the local target $Y_{l_k}$ by (3)
        Update the shared critic $F$ by minimizing Eq. (4)
        // Learn the individual policy
        Update the policy by minimizing Eq. (7)
    end for
end for
Output: $F, (\pi^k_i)_{i\in\mathcal{N}}$.
```

To enable efficient scaling with the number of agents, we use distributed optimisation (DO) with consensus (Nedic & Ozdaglar, 2009) to find $\hat{\phi}$. Each agent produces its own estimate $\hat{B}$ based on its observed rewards. DO methods efficiently solve large scale optimisation problems (Macua et al., 2010) and yields two major benefits:
i) efficiency: computing ϕ uses feedback from *all* agents' reward samples.
ii) consensus on Q: agents learn ϕ distributively but have identical Q iterates (for computing $\hat{B}$).

The common objective which each agent solves individually, is expressed with a set of local variables $\{\rho_i\}_{i\in\mathcal{N}}$ and a common global variable z:

minimise $\bar{G}(s,\boldsymbol{\rho}) = N^{-1}\sum_{i\in\mathcal{N}}(g^i(s,\boldsymbol{\rho}^i))^2$

s.t. $\boldsymbol{\rho}^i - z = 0, \quad i = 1,\ldots,N,$
where the gradient descent is according to: $\boldsymbol{\rho}^i_n = \boldsymbol{\rho}^i_{n-1} - \alpha\nabla g^i(s,\boldsymbol{\rho}^i)$ for some step size $\alpha > 0$. Note that the constraint prevents convergence to any R_i.

The algorithm works by constructing an estimate $\hat{\phi}$ then solving $\mathscr{M}^\dagger$ in a *distributed fashion* allowing the method to scale with the number of agents.

Algorithm Analysis

Our **SP**ot-**AC** algorithm inherits many useful properties of Q-learning (Antos et al., 2008).[7] Nevertheless, it is necessary to ensure the output of the algorithm still yields good

[7] By Prop. 5 (see Appendix) any MPE is a *local* optimum of B.

performance when the supervised learning approximation of $\mathscr{M}^\dagger$ has small errors. We now analyse the **SP**ot-**AC** algorithm and show that provided errors in approximating $\mathscr{M}^\dagger$ are small the error in the algorithm output is also small.

Our first result bounds the error on the estimate for the DPF from using the approximation method for $\hat{\phi}$.

Proposition 2. *Define by the following* $F_i(s,\boldsymbol{\eta},\rho) := \int_{\boldsymbol{\mathcal{A}}} \boldsymbol{\pi}_\epsilon(\boldsymbol{da},\boldsymbol{\eta},s)\frac{\partial}{\partial\eta_i}\pi_{i,\epsilon}(a^i,\eta_i,s)\nabla F_\rho(s,\boldsymbol{a})$ *and* $U(s,\boldsymbol{\eta},\rho) := \int_{\boldsymbol{\mathcal{A}}} \boldsymbol{\pi}_\epsilon(\boldsymbol{da},\boldsymbol{\eta},s)\frac{\partial}{\partial\eta_i}\pi_{i,\epsilon}(a^i,\eta_i,s)\nabla R_i(s,\boldsymbol{a})$ *then the following bound holds for some* $c > 0$*:*

$$\sum_{i\in\mathcal{N}} \|F_i(s,\boldsymbol{\eta},\rho) - U(s,\boldsymbol{\eta})\| \leq cN^2\epsilon^2,$$

where ϵ *is the approximation error from the SL procedure.*

Our next result ensures that if the estimates of ϕ have only small errors, **SP**ot-**AC** generates policy performances that closely match that of the MPE policies.

Proposition 3. *Define by* $B_\epsilon^{\tilde{\boldsymbol{\pi}}} = \lim_{k\to\infty} T_{\phi_\epsilon}^k B^{\boldsymbol{\pi}}$ *and let the policy* $\hat{\boldsymbol{\pi}}$ *be an MPE strategy i.e.* $\hat{\boldsymbol{\pi}} \in NE\{\mathscr{G}\}$ *(so that* $\lim_{k\to\infty} T_\phi^k B^{\boldsymbol{\pi}} = B^{\hat{\boldsymbol{\pi}}}$*) then for any* $\epsilon > 0$ *the following holds:*

$$\|B^{\hat{\boldsymbol{\pi}}} - B_\epsilon^{\tilde{\boldsymbol{\pi}}}\| \leq (2-\gamma)(1-\gamma)^{-1}\epsilon,$$

whenever $\|\phi^\epsilon - \phi\| < \epsilon$.

The result ensures that given close approximations of ϕ **SP**ot-**AC** in turn yields outputs close to $B^\star$. The result exploits the fact that the dual MDP $\mathscr{M}$ of Theorem 1 exhibits a continuity property so that small errors in the approximation of ϕ and B incur only small changes to the MPE of $\mathscr{G}$.

6. Experiments

We evaluate **SP**ot-**AC** in three popular multi-agent environments: the particle world (Lowe et al., 2017), a network routing game (Roughgarden, 2007) and a Cournot duopoly problem (Agliari et al., 2016). These environments have continuous action and state spaces, and the agents seek to maximise their own interest e.g. reaching target without collisions on particle world and minimising the cost for transporting commodity on routing game. To solve these problems successfully, the agents must learn Markov perfect equilibrium policies in order to respond optimally to the actions of others.

We consider two groups of state-of-the-art MARL baselines that handle continuous actions. The first group use individual rewards for learning: MADDPG (Lowe et al., 2017) and DDPG (Lillicrap et al., 2015). The second group use the collective rewards of all agents: COMIX and COVDN (de Witt et al., 2020). Further details are in the Appendix.

We use two evaluation metrics:
Exploitability (Davis et al., 2014) describes how much additional payoff players can achieve by playing a best-response policy BR (defined in the Appendix). It measures the proximity of the agents' policies to the MPE strategy, defined as $\delta = \frac{1}{N}\sum_i (u_i(\pi^{-i}, \mathrm{BR}(\pi^{-i})) - u_i(\pi))$.
Social welfare is the collective reward of all agents: $r = \sum_{i=1}^N r_i$, this is most relevant in tasks such as team games, where agents seek to maximise total reward.

Ablation Studies

To test the robustness of **SP**ot-**AC** and the baselines, we perform a set of ablation studies within routing games.
Ablation 1 analyses **SP**ot-**AC** in SGs that progressively deviate from c-SPGs, showing that **SP**ot-**AC** can handle SGs that mildly violate the c-SPG conditions (i.e. the potentiality requirement).
Ablation 2 analyses **SP**ot-**AC** in SGs that progressively deviate from team games but retain the potential game property. We demonstrate that, unlike other methods, **SP**ot-**AC** is able to converge to the Markov Perfect Equilibrium in non-cooperative SGs. We also report results on the classic Cournot Duopoly and show convergence of **SP**ot-**AC** to NE.

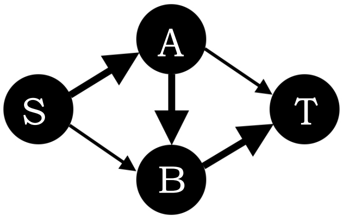
Non-atomic Routing Games involve a set of N selfish agents seeking to transport their commodity from a source node to a goal node in a network. This commodity can be divided arbitrarily and sent between nodes along edges.

At each time step, each agent has a distribution of commodity over the nodes of the graph. It assigns a fraction of its commodity in each node to travel along the edges emerging from those nodes. There are multiple agents (given by $N \in \{2,4,6,8\}$), using the same network (number of nodes $K \in \{20,40\}$) and agents pay a cost related to the total congestion of every edge at each time step. We design the game so that the MPE is socially efficient, i.e. playing an MPE strategy leads to high individual returns. We repeat the experiments for 5 independent runs and report the mean and standard deviation of the rewards. Further details on the settings can be found in Appendix.

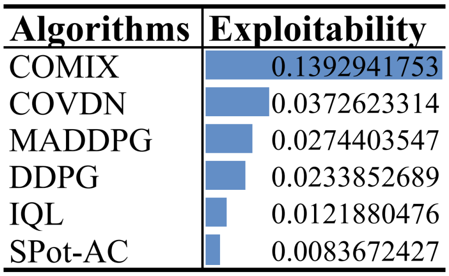
Results

Exploitability: We test **SP**ot-**AC** in a simple Braess' paradox game. The exploitability of **SP**ot-**AC** (Fig. 2) quickly converges to close to 0, indicating it learns NE policies (negative values are due to the fact that we are approximating best-responses). In contrast, the high exploitability values of existing MARL methods indicate that they fail to converge to NE policies. The algorithms that involve reward sharing (COMIX, COVDN) attempt to maximise social welfare, which is incompatible with this non-cooperative setting, so can be exploited by a best-response strategy.

Social welfare: In the cooperative, non-atomic routing game environment, we see in Fig. 3, using **SP**ot-**AC** (or-



(a) **SP**ot-**AC**'s policy on Braess' paradox



Algorithms	Exploitability
COMIX	0.1392941753
COVDN	0.0372623314
MADDPG	0.0274403547
DDPG	0.0233852689
IQL	0.0121880476
SPot-AC	0.0083672427

(b) Exploitability of different methods.

Figure 2. Results of exploitability. (a) a visualization of learned policy flows by **SP**ot-**AC**. (b) Exploitability results of all methods.

ange), each agent learns how to split their commodity G in a way that maximises rewards (minimises costs) and matches the shared reward baselines. Conversely, MADDPG (orange) and DDPG (blue) yield low rewards with high variance.

Coordination Navigation An OpenAI Multi Agent Particle Environment task (Lowe et al., 2017) involves n agents and n landmarks. Each agent must reach the target while avoiding collisions with other agents and fixed landmarks. Agents can observe the relative positions of other agents and landmarks, and have five actions {up, down, left, right, stay}. The reward is calculated as the agent's distance to each landmark with penalties for collisions with other agents.

This is a non-cooperative SG, so we compare **SP**ot-**AC** to DDPG and MADDPG, algorithms that are able to learn policies in which agents can act selfishly. We perform the exploitability analysis as above. Fig. 4 shows **SP**ot-**AC** achieves the best performance in terms of minimum distance to target *and* number of collisions, demonstrating that **SP**ot-**AC** enables agents to learn to coordinate while pursuing their own goals.

7. Conclusion

In this paper, we describe the first MARL framework that tackles MAS with payoff structures beyond zero-sum or team games. In doing so, the results we establish pave the way for a new generation of solvers that are able to tackle classes of SGs beyond cases in which the payoff structures lie at extremes. Therefore, the results of this paper open the door for MARL techniques to address a wider range of multi-agent scenarios. By developing theory that shows a class of SGs, namely c-SPGs have a dual representation as MDPs, we showed that c-SPGs can be solved by MARL agents using a novel distributed method which avoids the combinatorial explosion therefore allowing the solver to scale with the number of agents. We then validated our theory in experiments in previously unsolvable scenarios showing our method successfully learns MPE policies in contrast to existing MARL methods.

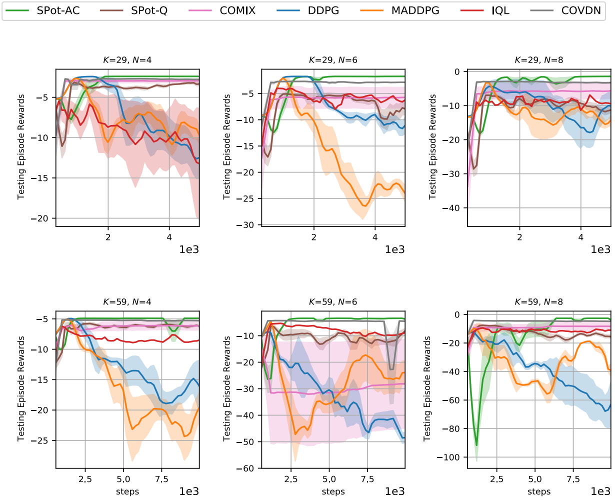


Figure 3. **Top** Average agent returns in Network with $N = 4, 6, 8$ agents, $K = 20$ nodes. **Bottom** Average agent returns in Network with $N = 4, 6, 8$ agents, $K = 59$ nodes.

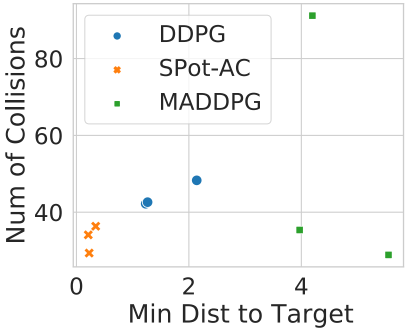


Figure 4. Coordination Navigation problem.

Acknowledgements

YW and ZW are partly supported by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No.XDA27000000. DM is grateful to Mohammed Amin Abdullah for insightful discussions relating to the convergence proofs in this work.

References

Agliari, A., Naimzada, A. K., and Pecora, N. Nonlinear dynamics of a cournot duopoly game with differentiated products. *Applied Mathematics and Computation*, 281: 1–15, 2016.

Antos, A., Szepesvári, C., and Munos, R. Fitted q-iteration in continuous action-space mdps. In *Advances in neural information processing systems*, pp. 9–16, 2008.

Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K., and Graepel, T. The mechanics of n-player differentiable games. *arXiv preprint arXiv:1802.05642*, 2018.

Bertsekas, D. P. Approximate dynamic programming. 2012.

Bertsekas, D. P. and Tsitsiklis, J. N. Gradient convergence in gradient methods with errors. *SIAM Journal on Optimization*, 10(3):627–642, 2000.

Bloembergen, D., Tuyls, K., Hennes, D., and Kaisers, M. Evolutionary dynamics of multi-agent learning: A survey. *Journal of Artificial Intelligence Research*, 53:659–697, 2015.

Candogan, O., Menache, I., Ozdaglar, A., and Parrilo, P. A. Flows and decompositions of games: Harmonic and potential games. *Mathematics of Operations Research*, 36 (3):474–503, 2011.

Candogan, O., Ozdaglar, A., and Parrilo, P. A. Near-potential games: Geometry and dynamics. *ACM Transactions on Economics and Computation (TEAC)*, 1(2):1–32, 2013.

Chang, K.-H. Chapter 5 - multiobjective optimization and advanced topics. In Chang, K.-H. (ed.), *Design Theory and Methods Using CAD/CAE*, pp. 325 – 406. Academic Press, Boston, 2015.

Chen, X., Deng, X., and Teng, S.-H. Settling the complexity of computing two-player nash equilibria. *Journal of the ACM (JACM)*, 56(3):1–57, 2009.

Cheng, Y., Diakonikolas, I., and Stewart, A. Playing anonymous games using simple strategies. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 616–631. SIAM, 2017.

Daskalakis, C. and Papadimitriou, C. Computing equilibria in anonymous games. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pp. 83–93. IEEE, 2007.

Davis, T., Burch, N., and Bowling, M. Using response functions to measure strategy strength. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

de Witt, C. S., Peng, B., Kamienny, P.-A., Torr, P., Böhmer, W., and Whiteson, S. Deep multi-agent reinforcement learning for decentralized continuous cooperative control. *arXiv preprint arXiv:2003.06709*, 2020.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Fudenberg, D. and Tirole, J. Game theory. *MIT Press*, 726: 764, 1991.

González-Sánchez, D. and Hernández-Lerma, O. *Discrete–time stochastic control and dynamic potential games: the Euler–Equation approach*. Springer Science & Business Media, 2013.

Grau-Moya, J., Leibfried, F., and Bou-Ammar, H. Balancing two-player stochastic games with soft q-learning. *arXiv preprint arXiv:1802.03216*, 2018.

Hernandez-Leal, P., Kaisers, M., Baarslag, T., and de Cote, E. M. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv preprint arXiv:1707.09183*, 2017.

Hu, J. and Wellman, M. P. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

Jaderberg, M., Czarnecki, W. M., Dunning, I., Marris, L., Lever, G., Castaneda, A. G., Beattie, C., Rabinowitz, N. C., Morcos, A. S., Ruderman, A., et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.

Jaśkiewicz, A. and Nowak, A. S. On symmetric stochastic games of resource extraction with weakly continuous transitions. *Top*, 26(2):239–256, 2018.

Konda, V. R. and Tsitsiklis, J. N. Actor-critic algorithms. In *Advances in neural information processing systems*, pp. 1008–1014, 2000.

Lã, Q. D., Chew, Y. H., and Soong, B.-H. *Potential Game Theory*. Springer, 2016.

Leonardos, S., Overman, W., Panageas, I., and Piliouras, G. Global convergence of multi-agent policy gradient in markov potential games. 2021.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Littman, M. L. Friend-or-foe q-learning in general-sum games. In *ICML*, volume 1, pp. 322–328, 2001.

Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Abbeel, O. P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, pp. 6379–6390, 2017.

Macua, S. V., Belanovic, P., and Zazo, S. Consensus-based distributed principal component analysis in wireless sensor networks. In *2010 IEEE 11th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5. IEEE, 2010.

Macua, S. V., Zazo, J., and Zazo, S. Learning parametric closed-loop policies for markov potential games. *arXiv preprint arXiv:1802.00899*, 2018.

Marden, J. R., Arslan, G., and Shamma, J. S. Cooperative control and potential games. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(6): 1393–1407, 2009.

Mariano, P., Pereira, A., Correia, L., Ribeiro, R., Abramov, V., Szirbik, N., Goossenaerts, J., Marwala, T., and De Wilde, P. Simulation of a trading multi-agent system. In *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236)*, volume 5, pp. 3378–3384. IEEE, 2001.

Mguni, D., Jennings, J., and de Cote, E. M. Decentralised learning in systems with many, many strategic agents. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Monderer, D. and Shapley, L. S. Fictitious play property for games with identical interests. *Journal of economic theory*, 68(1):258–265, 1996a.

Monderer, D. and Shapley, L. S. Potential games. *Games and economic behavior*, 14(1):124–143, 1996b.

Nedic, A. and Ozdaglar, A. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

Nicolaescu, L. I. The coarea formula. In *seminar notes*. Citeseer, 2011.

Papadimitriou, C. H. and Tsitsiklis, J. N. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

Peng, P., Wen, Y., Yang, Y., Yuan, Q., Tang, Z., Long, H., and Wang, J. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.

Rashid, T., Samvelyan, M., De Witt, C. S., Farquhar, G., Foerster, J., and Whiteson, S. Qmix: monotonic value function factorisation for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1803.11485*, 2018.

Roughgarden, T. Routing games. *Algorithmic game theory*, 18:459–484, 2007.

Samvelyan, M., Rashid, T., de Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G., Hung, C.-M., Torr, P. H., Foerster, J., and Whiteson, S. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

Shapley, L. S. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

Shoham, Y. and Leyton-Brown, K. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. 2014.

Simon, L. et al. *Lectures on geometric measure theory*. The Australian National University, Mathematical Sciences Institute, Centre . . . , 1983.

Slade, M. E. What does an oligopoly maximize? *The Journal of Industrial Economics*, pp. 45–61, 1994.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Szepesvári, C. and Munos, R. Finite time bounds for sampling based fitted value iteration. In *Proceedings of the 22nd international conference on Machine learning*, pp. 880–887, 2005.

Tan, M. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pp. 330–337, 1993.

Tutunov, R., Bou-Ammar, H., and Jadbabaie, A. Distributed newton method for large-scale consensus optimization. *IEEE Transactions on Automatic Control*, 64(10):3983–3994, 2019.

Ui, T. A shapley value representation of potential games. *Games and Economic Behavior*, 31(1):121–135, 2000.

Wang, X. and Sandholm, T. Reinforcement learning to play an optimal nash equilibrium in team markov games. In *Advances in neural information processing systems*, pp. 1603–1610, 2003.

Wiering, M. Multi-agent reinforcement learning for traffic light control. In *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*, pp. 1151–1158, 2000.

Yang, Y. and Wang, J. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv preprint arXiv:2011.00583*, 2020.

Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., and Wang, J. Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 5571–5580. PMLR, 2018.

Yang, Y., Tutunov, R., Sakulwongtana, P., Ammar, H. B., and Wang, J. α-rank: Scalable multi-agent evaluation through evolution. *arXiv preprint arXiv:1909.11628*, 2019.

Yang, Y., Wen, Y., Wang, J., Chen, L., Shao, K., Mguni, D., and Zhang, W. Multi-agent determinantal q-learning. In *International Conference on Machine Learning*, pp. 10757–10766. PMLR, 2020.

Ye, D., Zhang, M., and Yang, Y. A multi-agent framework for packet routing in wireless sensor networks. *sensors*, 15(5):10026–10047, 2015.

Zazo, S., Macua, S. V., Sánchez-Fernández, M., and Zazo, J. Dynamic potential games in communications: Fundamentals and applications. *arXiv preprint arXiv:1509.01313*, 2015.

Zhang, H., Chen, W., Huang, Z., Li, M., Yang, Y., Zhang, W., and Wang, J. Bi-level actor-critic for multi-agent coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7325–7332, 2020.

Zhang, R., Ren, Z., and Li, N. Gradient play in multi-agent markov stochastic games: Stationary points and convergence. *arXiv preprint arXiv:2106.00198*, 2021.

Zhou, M., Luo, J., Villela, J., Yang, Y., Rusu, D., Miao, J., Zhang, W., Alban, M., Fadakar, I., Chen, Z., et al. Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving. *arXiv preprint arXiv:2010.09776*, 2020.