Strategically Efficient Exploration in Competitive Multi-agent Reinforcement Learning

Robert Loftin[1] **Aadirupa Saha**[2] **Sam Devlin**[1] **Katja Hofmann**[1]

[1] Microsoft Research Cambridge , Cambridge, UK
[2] Microsoft Research NYC , New York, USA

Abstract

High sample complexity remains a barrier to the application of reinforcement learning (RL), particularly in multi-agent systems. A large body of work has demonstrated that exploration mechanisms based on the principle of *optimism under uncertainty* can significantly improve the sample efficiency of RL in single agent tasks. This work seeks to understand the role of optimistic exploration in non-cooperative multi-agent settings. We will show that, in zero-sum games, optimistic exploration can cause the learner to waste time sampling parts of the state space that are irrelevant to strategic play, as they can only be reached through cooperation between both players. To address this issue, we introduce a formal notion of *strategically efficient* exploration in Markov games, and use this to develop two strategically efficient learning algorithms for finite Markov games. We demonstrate that these methods can be significantly more sample efficient than their optimistic counterparts.

1 INTRODUCTION

Despite its success in recent years, the applicability of reinforcement learning is still limited by the enormous amounts of training data required to solve complex tasks, particularly when those tasks involve multiple agents [Vinyals et al., 2019, Berner et al., 2019]. For single-agent problems it has been shown that sample efficiency can be significantly improved with the use of more sophisticated exploration mechanisms that take into account the learner's own uncertainty about the learning task [Pathak et al., 2017, Burda et al., 2018]. Extending these approaches to multi-agent settings, however, remains an open challenge.

In this work, we focus on efficient exploration for reinforcement learning in competitive multi-agent settings. In recent related work, Bai and Jin [2020] have presented algorithms for self-play in finite Markov games with sample complexity bounds that are polynomial in the size of the state and action spaces. These methods are based on the principle of *optimism under uncertainty*, in which each agent acts greedily w.r.t. a statistically plausible model of the learning task that maximizes the agent's expected return. In two-player games, this optimism encourages the players to cooperate to reach states that have not previously been observed (driven by the assumption that both players can receive large positive returns from such unknown states). In zero-sum games, however, such cooperative behavior would never be observed between rational opponents.

In this paper, we show that such cooperative exploration is *strategically inefficient*, and may cause the learner to waste time exploring parts of the state space that provide no additional information about the Nash equilibria of the game. The key question for this work is how a reinforcement learning algorithm can recognize and avoid such strategically irrelevant parts of the state space, while still ensuring that an approximate solution to the game will be found. To address this question, we propose two reinforcement learning algorithms, *Strategic ULCB* and *Strategic Nash-Q*, which are strategically efficient in a suitably well-defined sense. As with the optimistic algorithms of Bai and Jin [2020], these algorithms select exploration policies optimistically w.r.t. a set of statistically plausible games. However, unlike prior work, in our approach each player chooses an optimistic best-response against the strongest known adversary strategy (rather than its opponents' exploration strategy).

In Section 4.1 we will prove that that Strategic ULCB is both strategically efficient and sample efficient in the traditional sense, while in Section 5 we will show that Strategic ULCB and Nash-Q significantly outperform their existing, optimistic counterparts. Our key conclusion is that the direct extension of optimistic exploration to multi-agent RL in competitive settings can be highly inefficient, and that by leveraging the adversarial nature of zero-sum games, it is possible to dramatically improve sample efficiency through the use of strategically efficient exploration mechanisms.

2 PRELIMINARIES

This work focuses on the role of exploration in finite, two-player zero-sum Markov games [Littman, 1994]. We define such a Markov games as a tuple $G = \{S, A, B, P, R, H\}$. Here S is a finite state space, and we let $h \in [1, H]$ be the steps since the start of the current episode. $A_{h,s}$ and $B_{h,s}$ are state and step-dependent action spaces $A_{h,s}$ for the min and max-players respectively, $P_h : S \times A \times B \mapsto \mathbb{P}(S)$ is the step-dependent transition distribution, $R_h : S \times A \times B \mapsto [0, 1]$ is the step-dependent reward function for the max player, and H is the fixed episode length. Let $|S| = \max_h |S_h|$, $|A| = \max_{s,h} |A_{h,s}|$ and $|B| = \max_{s,h} |B_{h,s}|$. We assume that rewards are deterministic. For zero-sum games, we need only specify the reward function for the max-player, with the reward for the min-player defined as $-R_h(s, a, b)$. The restriction to Markov games implies that the state is fully observable to both agents at all times. We also assume that there is a unique initial state s_1.

Training proceeds episodically for K episodes of length H. For the state s^k_h encountered at step h of episode k, the learner samples actions $a^k_h \in A_{h,s^k_h}$ and $b^k_h \in B_{h,s^k_h}$ from the joint exploration policy $\pi^k_h(s, a, b)$. After taking joint action (a^k_h, b^k_h), the learner observes reward $r^k_h = R_h(s^k_h, a^k_h, b^k_h)$, and state $s^k_{h+1} \sim P_h(s^k_h, a^k_h, b^k_h)$ if $h < H$. We assume here that the exploration policy $\pi^k_h : S \mapsto \mathcal{P}(A_{s^k_h} \times B_{s^k_h})$ is computed in advance for all s and h, and fixed throughout episode k. When the exploration policy can be factored into separate policies for the max and min-players, we denote these as μ^k and ν^k respectively, with $\pi^k_h(s, a, b) = \mu^k_h(s, a)\nu^k_h(s, b)$.

For any pair of policies μ,ν, we define $V^{\mu,\nu}_h(s)$ to be the expected return of the max player from state s at step h as:

$$V^{\mu,\nu}_h(s) = \mathrm{E}\left[\sum_{i=h}^{H} r_i(s_i, a_i, b_i)|\mu, \nu, s_h = s\right] \quad (1)$$

When training in self-play, we have no way of knowing what adversary the policies we learn will eventually need to play against. We therefore evaluate our learned policies μ and ν in terms of their worst-case return against any adversary policy, which we define as their *exploitability*

$$\mathrm{expl}(\mu) = -\inf_{\nu'} V^{\mu,\nu'}_1(s_1), \quad \mathrm{expl}(\nu) = \sup_{\mu'} V^{\mu',\nu}_1(s_1) \quad (2)$$

For a pair of policies μ and ν, the total exploitability is equal to the NashConv loss [Johanson et al., 2011, Lanctot et al., 2017], defined as

$$\mathrm{NashConv}(\mu, \nu) = \sup_{\mu'} V^{\mu',\nu}_1(s_1) - \inf_{\nu'} V^{\mu,\nu'}_1(s_1), \quad (3)$$

If μ, ν constitute a Nash equilibrium of the game, then $\mathrm{NashConv}(\mu, \nu) = 0$, and if $\mathrm{NashConv}(\mu, \nu) \leq \epsilon$, then μ and ν will constitute an ϵ-Nash equilibrium of the game.

3 RELATED WORK

While we focus on exploration in finite Markov games, this work is motivated by the goal of extending exploration approaches that have proven effective in single-agent deep reinforcement learning to the competitive multi-agent setting. Many successful approaches guide exploration by providing an additional *intrinsic* reward signal that is larger for states and actions for which the learner is less certain, often referred to as *curiosity* [Burda et al., 2019]. These include the Intrinsic Curiosity Module [Pathak et al., 2017], which uses the error of a supervised transition model to estimate uncertainty, and Random Network Distillation [Burda et al., 2018], which uses the error between a fixed, randomly initialized network and a prediction network trained on the states the learner has observed so far. Such intrinsic rewards have also proven effective in cooperative multi-agent RL, where all agents aim to maximize a common reward function [Iqbal and Sha, 2019, Böhmer et al., 2019].

While not subject to the same theoretical guarantees, the use of uncertainty-based intrinsic rewards in deep RL can be motivated by work on finite MDPs, where a number of algorithms based on the principle of optimism under uncertainty have been shown to have good worst-case sample complexity [Jaksch et al., 2010, Strehl and Littman, 2008, Jin et al., 2018]. In particular, Strehl and Littman [2008] and Jin et al. [2018] describe algorithms which incorporate optimism through a count-based exploration bonus of the form $\beta/\sqrt{N(s, a)}$, where $N(s, a)$ is the number of times the state s and action a have been observed previously. We refer to these algorithms as "optimistic" because they select the action that maximizes an upper confidence bound on the expected in the current state, where the upper bound is taken over some set of statistically plausible MDPs.

Recent results have shown that the use of upper confidence bounds can be extended to self-play in two-player zero-sum Markov games. Bai and Jin [2020] present a model-based self-play algorithm, VI-ULCB (which we will refer to as *Optimistic* ULCB in later sections) that finds an ϵ-equilibrium with at most $O(H^4|S|^2|A||B|/\epsilon^2)$ samples. VI-ULCB drives exploration by solving for the Nash equilibrium of an optimistic, *general-sum* corresponding to upper and lower confidence bounds on the max-player returns. Bai et al. [2020] build on this work, presenting a model-free self-play algorithms, Optimistic Nash-Q which finds an ϵ-equilibria in at most $O(H^5|S||A||B|/\epsilon^2)$ samples.

While these bounds are near-optimal in the worst case, they say little about the practical efficiency of these algorithms, or the approaches to exploration that they embody. These results do not rule out the possibility that the learner will need to explore the entire state-action space, even when this is unnecessary for the identification of an ϵ-equilibrium of the game. More specifically, in Section 4 we will show that VI-ULCB can select pairs of output policies such that neither policy can plausibly be an equilibrium of the game. In Section 5, we will empirically compare Optimistic Nash-Q and VI-ULCB, against two novel algorithms that avoid selecting such implausible policies. Through these comparisons we will demonstrate that Optimistic Nash-Q and VI-ULCB can suffer from unnecessarily high sample complexity in environments where large parts of the state space are irrelevant the equilibrium solution.

Finally, we note a connection between the concept of strategically efficient exploration and the Alpha-Beta pruning algorithm from game-tree search [Pearl, 1980]. While Alpha-Beta pruning is limited to deterministic, turn-based games with known transition dynamics (and so is not applicable in most RL settings), it nonetheless exploits the adversarial nature of zero-sum games in much the same way that the algorithms developed in this work will. Like Strategic ULCB and Strategic Nash-Q, Alpha-Beta pruning bounds the value of a state in terms of the strongest adversary strategy it has identified so far, and will not explore states that it knows cannot occur under a minimax optimal strategy for the root player. Unlike Strategic ULCB and Nash-Q however, Alpha-Beta pruning is not optimistic, and will continue evaluating a set of strategies (corresponding to the current sub-game) even when there exist potentially superior alternatives.

4 STRATEGIC EXPLORATION

The worst case sample complexity of optimistic algorithms, such as Optimistic ULCB (Algorithm 1) and Optimistic Nash-Q (Algorithm A.1), correspond to the complexity of learning a *complete* model of the game (even for model free algorithms). In some cases such complete exploration will be necessary, for example, when the game is effectively a single-agent MDP for the max-player, and every possible outcome must be known to ensure the max-player's policy is optimal. For truly competitive games, however, learning a complete model will often be unnecessary to find a Nash equilibrium. When this is the case, an optimistic algorithm may waste time exploring parts of the state space that yield no useful information about the solution to the game.

We can illustrate this issue with the abstract, turn-based game shown in Figure 1. This game, which we will refer to as the *decoy task game*, is composed of a set of single-player sub-tasks in which only the max-player takes actions. In Section 5 we will show experimental results in this game for a specific choice of sub-task, but for now it is sufficient to assume that each task is "complex" in the sense that a learner will have to attempt the task many times before a successful policy is found. At each episode, the max player

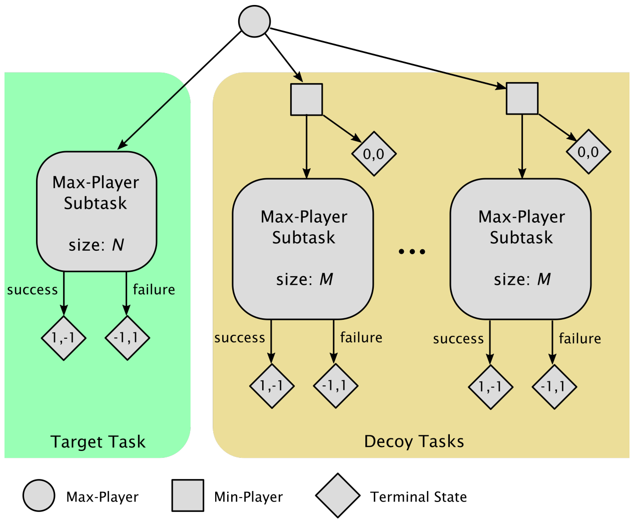


Figure 1: The generic decoy task game. We can choose any single-player game to define the target task and decoy tasks, as long as the max-player can always succeed in these sub-games with the right policy. The $(-1, 1)$ payoffs correspond to a max-player loss, while the $(0, 0)$ is a tie.

first chooses which sub-task they wish to explore. What is important here is that, for all but one sub-task (the *target task*), the min player has the option to end the game immediately, resulting in a tie, or allow the max-player to attempt the sub-task. As a result, the learner gains nothing by solving these alternative, or *decoy tasks*, as with or without a solution the best the learner can hope for is a tie if it chooses one of these sub-tasks during evaluation.

In spite of this, an optimistic algorithm may attempt to solve each of the decoy tasks, because, until a sub-task is solved, it will assume that it is possible for both the min and max-players to simultaneously receive a payoff of 1 when that task is complete. Under an optimistic exploration rule, the learner assumes that the min-player will allow the max-player to complete each of the decoy tasks, when in the underlying game the only reason this would happen is if the max-player gets a payoff ≤ 0 for completing the task (otherwise the min-player with terminate the game early). If there are many decoy tasks, optimistic exploration may be highly inefficient. In this section, we will describe algorithms which, while still sufficiently optimistic to ensure convergence to a solution, will be robust to the existence of such *strategically irrelevant* sub-tasks.

4.1 STRATEGIC EFFICIENCY

To develop strategically efficient algorithms, we will first need to formalize our intuitive notion of strategic efficiency. Here we define strategic efficiency in terms of the marginal exploration strategies μ^k and ν^k the learner follows during training. Loosely speaking, a strategically efficient learning algorithm should not consider strategies that it *believes* do not correspond to equilibria of the game.

Dependence on the agent's own belief is essential, as any strategy could correspond to an equilibrium if we place no restrictions on the set of possible games. Therefore, we need a representation of the "belief state" of a given learning algorithm. It will be sufficient to consider a representation that is independent of the learning algorithm itself, that is, one which only depends on the observable interactions between the learner and the environment. We represent the knowledge state by a sequence of sets $C_{1\leq k} \subset \mathbb{G}(H, S, A, B)$. Here, $\mathbb{G}(S, A, B)$ is the set of games on A, B, and the state space $S \cup \{s^*\}$ (where s^* is a hypothetical absorbing state) and max-player rewards in $[0, H]$. The absorbing state and larger reward range will simplify the task of proving that an algorithm is strategically efficient.

Each set is C_k itself a random variable, that is, a function $C_k(\mathbb{H}_k)$, where $\mathbb{H}_k$ is the history of states, actions, and rewards up to but not including episode k. For $\sigma \in [0, 1]$, we say that $C_{0\leq k}$ are σ-confidence sets if, under any learning algorithm run on a game G

$$\Pr\{\exists k \geq 1 : G \notin C_k\} \leq \sigma \tag{4}$$

It is possible to define the confidence sets with respect to some subset of $X \subset \mathbb{G}(S, A, B, H)$, such as the set $\mathbb{D}(S, A, B, H)$ of games with deterministic state transitions, so long as we can be certain *a priori* that $G \in D$. Our definition of strategic efficiency will be with respect to a given sequence of confidence sets.

Definition 4.1 (Strategic Efficiency). *If $C_{1\leq k}$ are σ-confidence sets w.r.t. $X \subseteq \mathbb{G}(S, A, B)$, then an algorithm is strategically efficient w.r.t. $C_{1\leq k}$ if, for all $k \geq 1$, there exists $\tilde{G} \in C_k$ such that*

$$\exists \tilde{G} \in C_k, \inf_{\nu} V^{\mu^k,\nu}_{\tilde{G},1} \geq \sup_{\mu} \inf_{\nu} V^{\mu,\nu}_{\tilde{G},1} \tag{5}$$

and there exists $\underset{\sim}{G} \in C_k$ such that

$$\exists \underset{\sim}{G} \in C_k, \sup_{\nu} V^{\mu,\nu^k}_{\underset{\sim}{G},1} \leq \inf_{\nu} \sup_{\mu} V^{\mu,\nu}_{\underset{\sim}{G},1} \tag{6}$$

Under this definition, a learning algorithm is strategically efficient if its exploration policies are always a component of a *plausible* Nash equilibrium of the true game G. Note that under the trivial sequence $C_{0\leq k} = \mathbb{G}(S, A, B)$, any learning algorithm would be efficient. To address this, we will require that the confidence sets converge when data is generated by the algorithm under consideration, that is, for any $\epsilon > 0$, $\delta \in (0, 1]$, there exist K, μ, ν s.t. $\text{NashConv}_G(\mu, \nu) \leq \epsilon$ for all $G \in C_K$ with probability at least $1 - \delta$.

4.2 NON-STRATEGIC EXPLORATION

Before discussing the design of strategically efficient learning algorithms, we first demonstrate how the joint-optimism employed by existing approaches can fail to be strategically efficient. Specifically, we show that Optimistic ULCB can fail to be strategically efficient w.r.t. its own *implicit* confidence sets. This is easiest to show this for games with deterministic state transitions (which include matrix games with no transitions). In such games, Optimistic ULCB can be run with an exploration bonus term of $\beta_t = 0$, with efficient exploration being guaranteed by optimistic initialization. The models maintained by Optimistic ULCB will be exact for all observed $(h, s, a, b) \in \mathbb{H}_k$, and its natural confidence sets will be the sets D_k of games that are exactly consistent the the rewards and state transitions observed up to episode k, that is

$$\begin{aligned} D_k = \{G \in \mathcal{D}(S, A, B, H) | P_h(s^k_h, a^k_h, b^k_h, s^k_{h+1}) = 1 \wedge \\ R_h(s^k_h, a^k_h, b^k_h) = r_h, \forall (s^k_h, a^k_h, b^k_h, r^k_h, s^k_{h+1}) \in \mathbb{H}_k\}. \end{aligned} \tag{7}$$

Remark 4.1. *Optimistic ULCB is not guaranteed to be strategically efficient with respect to the confidence sets $D_{1\leq k}$ (Equation 7) for deterministic games.*

We can demonstrate the strategic inefficiency of Optimistic ULCB using a variation on the classical prisoners dilemma. Similar to the prisoner's dilemma, each player has the option to either cooperate (c) with the other player, or defect (d). Unlike the original prisoner's dilemma, however, payoffs in this game are zero-sum

	c	d
c	(x,-x)	(-.5,.5)
d	(.5,-.5)	(0,0)

with $x \in [-1, 1]$. Note that regardless of the value of x, the only equilibrium for this game is the strategy profile in which both players always defect. Assume now that we have run Optimistic ULCB for three episodes, selecting joint strategies such that the only unobserved combination remaining is joint cooperation. To compute the exploration strategy for the next episode, Optimistic ULCB will select a Nash equilibrium of the general-sum game

	c	d
c	(1,1)	(-.5,.5)
d	(.5,-.5)	(0,0)

where joint cooperation is assumed to yield a payoff of 1 for both players because the true payoffs have never been observed. Because we know in advance that the game is zero-sum, however, we know that no matter what the true payoff for joint cooperation is, at least one player will have an incentive to defect. The row player will defect unless its payoff is greater than or equal to .5, but if this is the case. the column player must have a payoff less that or equal to -.5. This implies that joint defection is the only plausible equilibrium, and that neither player will cooperate as part of an equilibrium strategy for any plausible game. Therefore, if Optimistic ULCB chooses joint cooperation as its next strategy, it will fail to satisfy Definition 4.1.

4.3 STRATEGIC ULCB

Algorithm 1 The Strategic (and Optimistic) ULCB algorithms. The function $\mathrm{Nash}(G, G')$ computes a mixed strategy profile (μ, ν) constituting a Nash equilibrium of the two-player game given by the payoff matrices G and G'. Strategic ULCB maintains separate evaluation policies $\tilde{\mu}^k$ and $\tilde{\nu}^k$, while Optimistic ULCB [Bai and Jin, 2020] uses the same policies for exploration and evaluation.

```
 1: Initialize: $\forall h \in [H]$, $s \in S_h$, $a \in A_{h,s}$, $b \in B_{h,s}$, $s' \in S_{h+1}$, $N_h^1(s,a) \leftarrow 0$, $N_h^1(s,a,s') \leftarrow 0$.
 2: for episode $k = 1, \dots, K$ do
 3:     for step $h = H, \dots, 1$ do
 4:         for $s \in S_h, a, b \in A_{h,s} \times B_{h,s}$ do
 5:             $t \leftarrow N_h^k(s,a,b)$
 6:             $\bar{Q}_h^k(s,a,b) \leftarrow \min\{\hat{R}_h^k(s,a,b) + \hat{P}_h^k(s,a,b)^\top \bar{V}_{h+1}^k + \beta_t, H\}$
 7:             $\underline{Q}_h^k(s,ab) \leftarrow \max\{\hat{R}_h^k(s,a,b) + \hat{P}_h^k(s,a,b)^\top \underline{V}_{h+1}^k - \beta_t, 0\}$
 8:         end for
 9:         for $s \in S_h$ do
10:             if Strategic ULCB then
11:                 $\mu_h^k(s), \tilde{\nu}_h^k \leftarrow \mathrm{Nash}(\bar{Q}_h^k(s,\cdot,\cdot), -\bar{Q}_h^k(s,\cdot,\cdot))$
12:                 $\tilde{\mu}_h^k(s), \nu_h^k \leftarrow \mathrm{Nash}(\underline{Q}_h^k(s,\cdot,\cdot), -\underline{Q}_h^k(s,\cdot,\cdot))$
13:             else if Optimistic ULCB then
14:                 $\mu_h^k(s), \nu_h^k \leftarrow \mathrm{Nash}(\bar{Q}_h^k(s,\cdot,\cdot), -\underline{Q}_h^k(s,\cdot,\cdot))$
15:                 $\tilde{\mu}_h^k(s), \tilde{\nu}_h^k \leftarrow \mu_h^k(s), \nu_h^k$
16:             end if
17:             $\bar{V}_h^k(s) \leftarrow \mu_h^k(s)^\top \bar{Q}_h^k(s,\cdot,\cdot)\tilde{\nu}_h^k$
18:             $\underline{V}_h^k(s) \leftarrow \tilde{\mu}_h^k(s)^\top \underline{Q}_h^k(s,\cdot,\cdot)\nu_h^k$
19:         end for
20:     end for
21:     set $s_1^k \leftarrow s_1$
22:     for step $h = 1, \dots, H$ do
23:         Take actions $a_h^k \sim \mu_h^k(s_h^k)$ and $b_h^k \sim \nu_h^k(s_h^k)$
24:         Observe max-player reward $r_h^k$ and next state $s_{h+1}^k$
25:         $N_h^{k+1}(s_h^k, a_h^k, b_h^k) \leftarrow N_h^k(s_h^k, a_h^k, b_h^k) + 1$
26:         $N_h^{k+1}(s_h^k, a_h^k, b_h^k, s_{h+1}^k) \leftarrow N_h^k(s_h^k, a_h^k, b_h^k, s_{h+1}^k) + 1$
27:         $\hat{P}_h^{k+1}(\cdot|s_h^k, a_h^k, b_h^k) \leftarrow \frac{N_h^{k+1}(s_h, a_h^k, b_h^k, \cdot)}{N_h^{k+1}(s_h, a_h^k, b_h^k)}$
28:         $\hat{R}_h^{k+1}(s_h, a_h^k, b_h^k) \leftarrow r_h$
29:     end for
30: end for
```

We now present a model-based learning algorithm that will be provably strategically efficient in some settings. As this new algorithm is similar in structure to Optimistic ULCB, we refer to it as Strategic ULCB (Algorithm 1). Strategic ULCB differs from Optimistic ULCB in three key ways. First, it maintains separate policies $\tilde{\mu}^k$ and $\tilde{\nu}^k$ for evaluation. This is necessary because strategically efficient exploration may converge to a solution before the game has been fully explored, such that the optimistic exploration policies may remain exploitable indefinitely. Second, the max-player exploration policy for each state s is defined as a minimax optimal strategy of the matrix game defined by $\bar{Q}_h^k(s,\cdot,\cdot)$ (the min-player exploration policy is computed w.r.t. $\underline{Q}_h^k(s,\cdot,\cdot)$). This focuses exploration on actions that maximize the return a player can *optimistically guarantee* against an adversary. Finally, the value function updates are

$$\bar{V}_h^k(s) = \mu_h^k(s)^\top \bar{Q}_h^k(s,\cdot,\cdot)\tilde{\nu}_h^k \tag{8}$$

$$\underline{V}_h^k(s) = \tilde{\mu}_h^k(s)^\top \underline{Q}_h^k(s,\cdot,\cdot)\nu_h^k \tag{9}$$

which ensures that $\bar{V}_h^k(s)$ reflects the best return the max-player can expect against a true adversary, rather than the min-player's exploration policy. To demonstrate the correctness of Strategic ULCB, we provide a bound on the total NashConv loss incurred by the evaluation policies $\tilde{\mu}^k$ and $\tilde{\nu}^k$ over K episodes, which we denote as $\mathrm{Regret}(K)$,

$$\mathrm{Regret}(K) = \sum_{k=1}^{K}\left[\sup_{\mu} V_1^{\mu,\tilde{\nu}^k}(s_1) - \inf_{\nu} V_1^{\tilde{\mu}^k,\nu}(s_1)\right] \tag{10}$$

We will show that $\mathrm{Regret}(K) \leq O(\sqrt{K})$, such that the average NashConv loss will decay as $O(1/\sqrt{K})$.

Theorem 4.1. *For any $K \geq 3$ and $\delta \geq 0$, if Strategic ULCB (Algorithm 1) is run with β_t defined as*

$$\beta_t = H\sqrt{\frac{2|S|\ell}{t}} \tag{11}$$

where $\ell = \ln(KH|S||A||B|/\delta)$, then its regret satisfies

$$\mathrm{Regret}(K) \leq 6\sqrt{2KH^4|S|^2|A||B|\ell} \tag{12}$$

with probability at least $1 - \delta$.

The full proof of Theorem 4.1 can be found in Appendix A, and is similar to the proof for the Optimistic ULCB given by Bai and Jin [2020]. We can sketch the main ideas of the proof by assuming $H = 1$ (so we can ignore the state) and that $\underline{Q}^k(a,b) \leq R(a,b) \leq \bar{Q}^k(a,b)$. Because $\mu^k = \max_a \bar{Q}^k(a,\cdot)\tilde{\nu}^k$ and $\nu^k = \min_b(\tilde{\mu}^k)^\top \underline{Q}^k(\cdot,b)$, we have that $\bar{V}^k = \max_a \bar{Q}^k(a,\cdot)\tilde{\nu}^k \geq \max_a R(a,\cdot)\tilde{\nu}^k$, and $\underline{V}^k = \min_b(\tilde{\mu}^k)^\top \underline{Q}^k(\cdot,b) \leq \min_b(\tilde{\mu}^k)^\top R(a,\cdot)$. Therefore, the NashConv loss of the profile $(\tilde{\mu}^k, \tilde{\nu}^k)$ is bounded by $\bar{V}^k - \underline{V}^k$. Note that it is not possible to bound the loss of (μ^k, ν^k) in the same way, and so the need for separate evaluation policies. We then show that $\bar{V}^k$ and $\underline{V}^k$ converge, by showing that they are bounded by the "on policy" confidence bounds $\tilde{V}^k = (\mu^k)^\top \bar{Q}^k \nu^k$ and $\underline{V}^k = (\mu^k)^\top \bar{Q}^k \nu^k$, which do converge under the joint exploration policy. Note that $\tilde{V}^k = (\mu^k)^\top \bar{Q}^k \nu^k \geq (\mu^k)^\top \bar{Q}^k \tilde{\nu}^k$ because $\tilde{\nu}^k$ is also a best-response to μ^k, with the same being true for $\tilde{V}^k$.

We can also show that, for the special case of games with deterministic transitions, Strategic ULCB will be strategically efficient with respect to the confidence sets D_k of games that are exactly consistent the the rewards and state transitions observed up to episode k.

Theorem 4.2. *Strategic-ULCB will be strategically efficient w.r.t. the confidence sets $D_{1\leq k}$ (Equation 7) when run with $\beta_t = 0, \forall t$, on any game with deterministic state transitions.*

The proof of Theorem 4.2 can be found in Appendix B. The restriction to deterministic games is necessary, as without it $\bar{Q}^k$ and $\underline{Q}^k$ may not be exactly realizable for any plausible game, which is essential for the proof. For stochastic games, the bonus terms β_t will be approximations of the true upper and lower bounds over the space of statistically plausible games (to see this, consider the value of $\bar{Q}^k_h(s,a,b)$ when $\bar{V}^k_{h+1} = 0$). Therefore, Strategic ULCB is will only be *approximately* strategically efficient in stochastic games.

4.4 MODEL-FREE ALGORITHMS

Bai et al. [2020] present Optimistic Nash-Q as model-free counterpart to Optimistic ULCB. Optimistic Nash-Q maintains tabular estimates of the upper and lower bounds $\bar{Q}_h$ and $\underline{Q}_h$ analogous to those used in Optimistic ULCB, but which are updated online via a Q-learning update, rather than being recomputed at each episode under the current model. We can extend Strategic ULCB to the model-free case in much the same way, defining the current exploration and evaluation policies as

$$\mu^k_h(s), \tilde{\nu}^k_h(s) = \text{Nash}(\bar{Q}_h(s,\cdot,\cdot)) \tag{13}$$

$$\tilde{\mu}^k_h(s), \tilde{\nu}^k_h(s) = \text{Nash}(\underline{Q}_h(s,\cdot,\cdot)) \tag{14}$$

and updating the value function bounds as

$$\bar{V}^k_h(s) = \mu^k_h(s)^\top \bar{Q}_h(s,\cdot,\cdot)\tilde{\nu}^k_h(s) \tag{15}$$

$$\underline{V}^k_h(s) = \tilde{\mu}^k_h(s)^\top \underline{Q}_h(s,\cdot,\cdot)\tilde{\nu}^k_h(s) \tag{16}$$

Like Optimistic Nash-Q, Strategic Nash-Q recomputes the policies and value function bounds for the current state after the $\bar{Q}_h$ and $\underline{Q}_h$ are for the current state and action. We provide the pseudocode for Optimistic Nash-Q in Appendix C, and for Strategic Nash-Q in Appendix D.

5 EXPERIMENTS

In this section, we compare Strategic ULCB and Strategic Nash-Q against Optimistic ULCB and Nash-Q, as well as Independent Q-learning. To highlight the impact of strategically efficient exploration on sample complexity, we first present results in a version of the decoy task game (Figure 1). We will demonstrate that, as the number of strategically irrelevant decoy tasks increases, so too do the advantages of Strategic ULCB and Nash-Q over the alternatives. We also evaluate these algorithms on a set of randomly generated turn-based games, to demonstrate the value of strategic exploration in much more general settings.

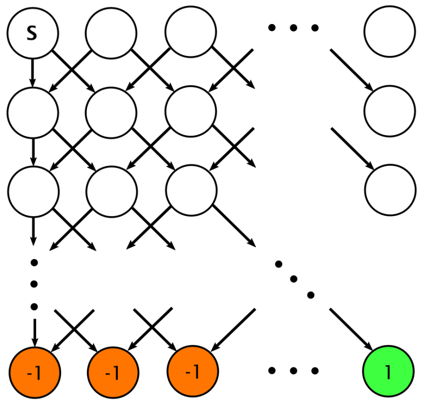


Figure 2: The $k \times k$ deep-sea task. The player always starts in the state marked "S". To reach the goal state, the player must move right for $k-1$ steps.

5.1 ALGORITHMS

The fact that Strategic ULCB and Nash-Q define separate evaluation policies could give them an unfair advantage over the Optimistic baseline algorithms. To provide a fair comparison between these approaches, we therefore modify Optimistic ULCB and Nash-Q to compute *pessimistic* evaluation policies $\tilde{\mu}^k$ and $\tilde{\nu}^k$ a

$$\mu, \tilde{\nu}^k_h(s) = \text{Nash}(\bar{Q}^k_h(s,\cdot,\cdot)) \tag{17}$$

$$\tilde{\mu}^k_h(s), \nu = \text{Nash}(\underline{Q}^k_h(s,\cdot,\cdot)) \tag{18}$$

where $\bar{Q}^k_h$ and $\underline{Q}^k_h$ are the upper and lower confidence bounds maintained by each algorithm. These policies correspond to each player maximizing their expected return in the worst plausible case. Note that existing NashConv regret bounds for Optimistic Nash-Q only apply to a complex, non-stationary mixture of the exploration policies μ^k_h and ν^k_h. In these experiments, however, the NashConv loss is computed for the most recent values of the evaluation policies $\tilde{\mu}^k$ and $\tilde{\nu}^k$. We also note that, as our experiments are conducted in alternating move games, the computation of equilibrium strategies for each state reduces to a simple maximization problem over the actions for the current player.

Independent Q-Learning Additionally, we compare against a learner that trains by running two independent instances of tabular Q-learning against one another. While this approach is not guaranteed to solve a Markov game, the use of independent Q-learning (IQL) has historically proven successful in some multi-agent settings [Tan, 1993, Tesauro, 1994]. In these experiments, we optimistically initialize the Q-function estimates for each learner to their maximum possible return H, which means that both learners engage in optimistic exploration in much the same way that Optimistic Nash-Q does, but without explicit coordination between the learners. Like Optimistic ULCB and Nash-Q, each Q-learner

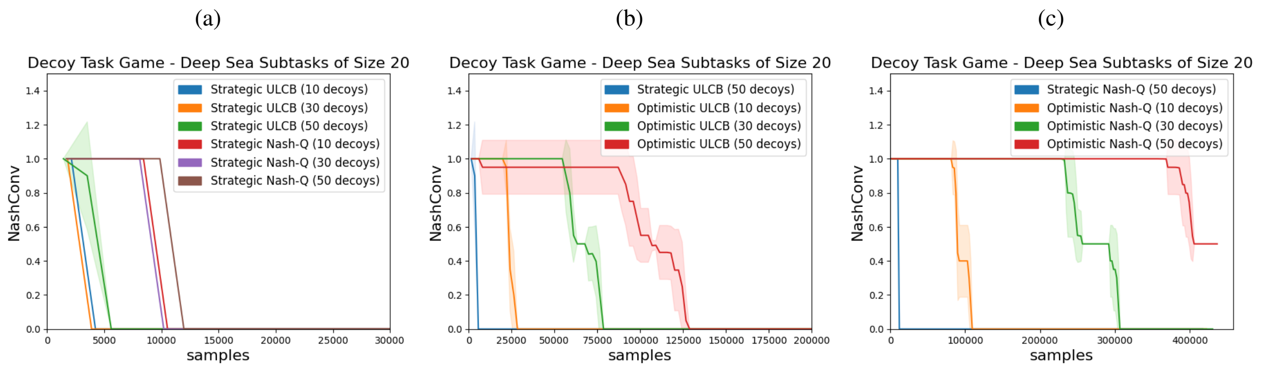


Figure 3: Comparisons between Strategic ULCB, Strategic Nash-Q and their optimistic counterparts on decoy task games with deep-sea sub-tasks of varying sizes. Shows the NashConv loss, with zero corresponding to the point where the game has been solved. Error bars show standard deviations over 10 game instances.

maintains a separate evaluation policy based on a separate, pessimistically initialized Q-function.

Hyper-parameters For Strategic and Optimistic ULCB, the only hyperparameter that needs to be defined is the exploration bonus β_t (Strategic and Optimistic Nash-Q require this parameter as well). For our deterministic environments, however, we can set $\beta_t = 0$ for all t. For Optimistic and Strategic Nash-Q (as well as independent Q-learning), we set the learning rate $\alpha_t = \frac{H+1}{H+t}$, where $t = N_h^k(s, a, b)$ ($t = N_h^k(s, a)$ for IQL), the theoretically justified value which proved reliable in practice [Jin et al., 2018, Bai et al., 2020]. For IQL, we found empirically that using ϵ-greedy exploration (in addition to optimistic exploration) led to better performance, with $\epsilon = 0.05$ being most effective.

5.2 DECOY TASK GAMES

To demonstrate advantage of strategically efficient exploration, we first evaluate Strategic ULCB and Strategic Nash-Q on instances of the decoy task game, illustrated in Figure 1. The challenge for exploration in these games is the tendency of the *decoy* tasks to distract algorithms that explore without regard for the adversarial nature of the game. These games are representative of the broader class of two-player zero sum games in which the bulk of the state space is strategically irrelevant, that is, it does not need to be explored to find an equilibrium solution. To understand the impact of such irrelevant states, we consider games with a single target task, but varying numbers of decoy tasks. To keep rewards normalized in $[0, 1]$, we modify the payoff structure shown in Figure 1 such that a max-player loss corresponds to a max-player reward of 0, and a tie a reward of $1/2$. Each decoy and target task is separate (solving one does not help the learner solve the others), and we also randomize the the index of the action leading to the target task.

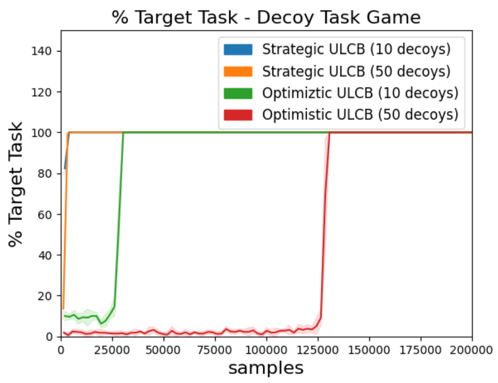
5.2.1 Deep Sea Sub-Task

In our experiments with decoy task games, both the target and decoy tasks are instances of the *deep sea* environment [Osband et al., 2019, 2020]. We chose the deep sea environment as it is specifically designed to be difficult to solve using simple exploration strategies such as ϵ-greedy, while being reliably solved using count-based exploration bonuses or optimistic initialization of the value function. The deep sea environment (Figure 2) is an $n \times n$ grid of states, with the initial state in the top-left corner, and the goal state in the bottom right corner. The player moves down, to the left or right, at each step, and to reach the goal, the player must go right for $n - 1$ steps. For large n, random action selection will have a very small probability of reaching the goal. The use of instances of the deep sea environment as target and decoy tasks in the decoy task game leads to a task for which efficient exploration is essential, but the naive application of single-agent exploration mechanisms perform poorly when there are a large number of decoy tasks. This combination is therefore ideal for evaluating the strategic efficiency of a learning algorithm.

5.2.2 Decoy Task Game Results

Figure 3 shows a set of comparisons between Strategic ULCB and Strategic Nash-Q against their optimistic counterparts, with pessimistic evaluation policies, in several instances of the the decoy task game. We compare these algorithms on instances with 10, 30 and 50 decoy tasks, where both target and decoy sub-tasks are instances of the 20x20 deep sea environment. Figure 3a shows that, in terms of the NashConv loss, the performance of Strategic ULCB and Strategic Nash-Q, is largely insensitive to the number of decoy tasks. In contrast, Figures 3b and 3c show that the number of samples required for Optimistic ULCB and Optimistic Nash-Q to solve the game (where the loss

(a)



(b)

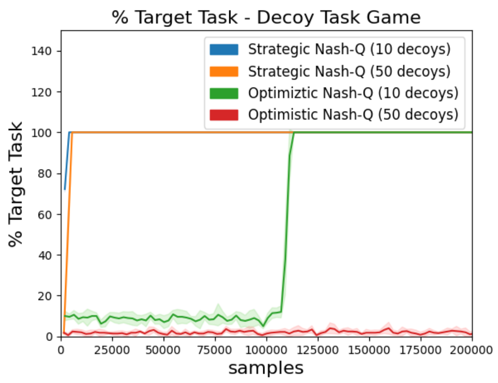


Figure 4: Comparisons between Strategic ULCB, Strategic Nash-Q and their optimistic counterparts on decoy task games with deep-sea sub-tasks. Shows the percentage of episodes per iteration that explore the target task. Error bars show standard deviations over 5 games.

goes to zero) grows roughly proportionately with the number of decoy tasks. For clarity of presentation, Figures 3b and 3c only show the performance of Strategic ULCB and Strategic Nash-Q for the most difficult case with 50 decoys, which is nonetheless significantly better than that of the optimistic algorithms for even the easiest case with 10 decoys. As expected, the model-based ULCB algorithms are more sample-efficient than their model-free counterparts.

These results are consistent with our hypothesis that the strategically efficient algorithms will be able to quickly recognize that for any decoy task, the best return the max-player can expect against a rational opponent would be $^1/_2$, and so will prioritize solving the target instance of the deep-sea sub-task. We can see this behavior in Figure 3a, where Strategic ULCB and Nash-Q take slightly longer to solve games with more decoys, corresponding to the time required to determine that each decoy task is irrelevant. To further support this hypothesis, in Figure 4 we show the percentage of episodes in which each algorithm explored the target task. We can see that Strategic ULCB and Nash-Q concentrate exploration on solving the target task much more quickly than Optimistic ULCB and Nash-Q, which waste time attempting to solve each decoy task.

5.3 TREE-STRUCTURED GAMES

While strategically efficient exploration has a dramatic impact on performance in the decoy task game, we can also show that it can have a significant impact on performance in much more general classes of games. In this section, we evaluate Strategic ULCB and Strategic Nash-Q in a space of tree-structured, alternating move games, where, for each random game instance, the max-player rewards for each terminal state is drawn from the uniform distribution over $[0, 1]$. While there are no states in these games that are designated as being strategically irrelevant, we can nonetheless bound the plausible return each player can guarantee from a given state without knowing the payoffs of all terminal states reachable from that state. Strategic exploration may therefore still be beneficial, if it can prioritize states for which a strong adversary policy has not yet been identified.

We consider alternating-move games of depth 5 and 6, with either 5 or 6 actions available to the active player in each state. Figure 5a compares the average performance of Strategic and Optimistic ULCB in solving games of depth 5 with 6 actions per state, where Strategic ULCB has a clear advantage in how fast its NashConv loss converges to zero. In Figures 5b and 5c, Strategic Nash-Q shows an advantage over Optimistic Nash-Q (as well as independent Q-learning) in games of depth 5 and 6, with Strategic Nash-Q having a large advantage over the alternatives for games of depth 6.

6 CONCLUSION

Reducing sample complexity will be critical if reinforcement learning is to see widespread use in solving real-world problems, particularly for tasks that involve interaction between multiple agents. Here we have considered approaches to exploration in competitive multi-agent tasks, and have shown that the use of *strategically efficient* exploration mechanisms can significantly reduce sample complexity relative to non-strategic, optimistic mechanisms. We have presented novel, strategically efficient reinforcement learning algorithms for finite Markov games, and demonstrated that they can be significantly more sample efficient than their optimistic counterparts in challenging exploration games, while preserving the same sample complexity guarantees as existing approaches across all possible games.

While this work is limited to Markov games with small, finite state and action spaces, the concept of strategically efficient exploration can be applied to games with infinite

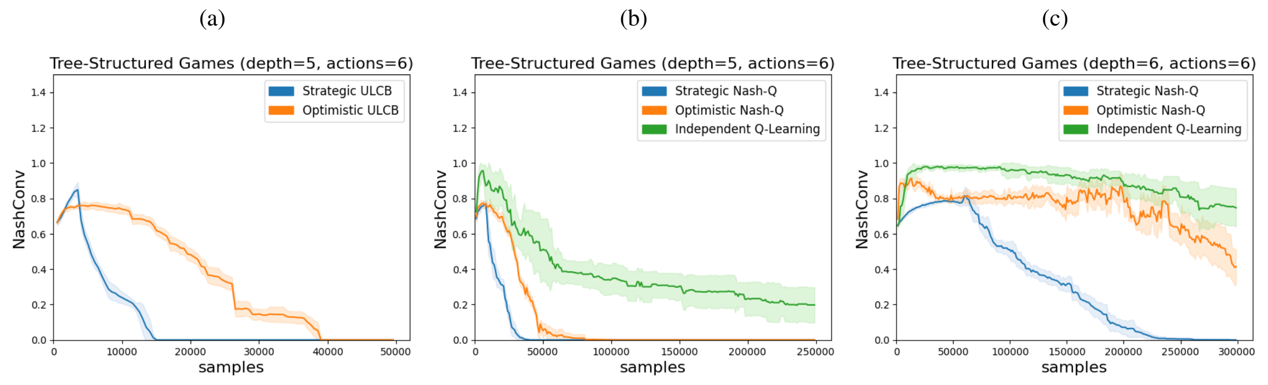


Figure 5: Comparisons of Strategic-ULCB and Strategic-Nash-Q against their optimistic counterparts and IQL, in tree-structured, alternating move games with randomly generated payoffs. Shows the NashConv loss, with zero corresponding to the point where the game has been solved. Error bars show standard deviation over 10 randomly generated game instances.

state and action spaces. Future work would focus on the development of strategically efficient algorithms that are compatible with the use of function approximation. The finite algorithms developed in this work may serve as the basis for strategically efficient alternatives to existing frameworks for deep multi-agent RL. Future theoretical work would seek to extend the notion of strategic efficiency to n-player general-sum games, and games with imperfect information.

Acknowledgements

We would like to thank Akshay Krishnamurthy for his valuable feedback during the development of this work.

References

Yu Bai and Chi Jin. Provable self-play algorithms for competitive reinforcement learning. In *International Conference on Machine Learning*, pages 551–560. PMLR, 2020.

Yu Bai, Chi Jin, and Tiancheng Yu. Near-optimal reinforcement learning with self-play. *Advances in Neural Information Processing Systems*, 33, 2020.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *CoRR*, 2019.

Wendelin Böhmer, Tabish Rashid, and Shimon Whiteson. Exploration with unreliable intrinsic reward in multi-agent reinforcement learning. *arXiv preprint arXiv:1906.02138*, 2019.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2018.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*, 2019.

Shariq Iqbal and Fei Sha. Coordinated exploration via intrinsic rewards for multi-agent reinforcement learning. *arXiv preprint arXiv:1905.12127*, 2019.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pages 4863–4873, 2018.

Michael Johanson, Kevin Waugh, Michael Bowling, and Martin Zinkevich. Accelerating best response calculation in large extensive games. In *IJCAI*, volume 11, pages 258–265, 2011.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in neural information processing systems*, pages 4190–4203, 2017.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

Ian Osband, Benjamin Van Roy, Daniel J Russo, and Zheng Wen. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019.

Ian Osband, Yotam Doron, Matteo Hessel, John Aslanides, Eren Sezener, Andre Saraiva, Katrina McKinney, Tor Lattimore, Csaba Szepesvari, Satinder Singh, Benjamin Van Roy, Richard Sutton, David Silver, and Hado Van Hasselt. Behaviour suite for reinforcement learning. In *International Conference on Learning Representations*, 2020.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017.

Judea Pearl. Asymptotic properties of minimax trees and game-searching procedures. *Artificial Intelligence*, 14(2): 113–138, 1980.

Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74 (8):1309–1331, 2008.

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

Gerald Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2):215–219, 1994.

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