A general criterion and an algorithmic framework for learning in multi-agent systems

Rob Powers, Yoav Shoham, and Thuc Vu
Stanford University
{powers,shoham,thucvu}@cs.stanford.edu

May 2, 2006

Abstract

We offer a new formal criterion for agent-centric learning in multi-agent systems, that is, learning that maximizes one's rewards in the presence of other agents who might also be learning (using the same or other learning algorithms). This new criterion takes in as a parameter the class of opponents. We then provide a modular approach for achieving effective agent-centric learning; the approach consists of a number of basic algorithmic building blocks, which can be instantiated and composed differently depending on the environment setting (for example, 2- versus n-player games) as well as the target class of opponents. We then provide several specific instances of the approach: an algorithm for stationary opponents, and two algorithms for adaptive opponents with bounded memory, one algorithm for the n-player case and another optimized for the 2-player case. We prove our algorithms correct with respect to the formal criterion, and furthermore show the algorithms to be experimentally effective via comprehensive computer testing.

1 Introduction

The past few years have seen a rapidly growing interest in multi-agent systems, and in particular in learning algorithms for such systems. This interest has driven proposals for a growing body of algorithms, and various arguments about their relative merits and domains of applicability (for example, [29] and [32]). In previous work [30] we surveyed this literature, and defined five different coherent agendas one could adopt when concerned with learning in multi-agent systems. We will not repeat the list in this paper, but instead we offer a contribution to one of the learning agendas which we singled out as particularly relevant from the computer science point of view.

The term 'learning' bears some discussion in the context of multi-agent systems. First, let us be precise about the setting in which we will discuss learning, which is known, fully observable (finitely or infinitely) repeated games. We give

the formal definition in Section 2, but, intuitively speaking, these consist of repeating some known matrix game (the 'stage game'), with each agent getting a reward after each play and observing the actions of the other agent(s). We furthermore assume the agents are attempting to maximize average rewards (or limit average in the case of infinite repetition), meaning that each agent's overall reward is the average of the stage-game rewards. While it would certainly be interesting to relax these assumptions, most of the key issues arise already in the current setting. In the final section we discuss possible extensions to unknown games, partially observable games, stochastic games, and discounted rewards, and the additional challenges posed by each of these settings.

Note that even in the relatively simple environment of repeated games, the agent's strategy space is huge, encompassing all mappings from play histories to actions. Many of these strategies are naturally viewed as incorporating a learning element. For example, in *rational learning* [20] an agent starts with some prior probability distribution over its opponent's repeated-game strategies, plays the (stage-game) best response, observes the play of the opponents, updates the probability distribution, and repeats.

And so learning is inherent in repeated games. But it is also qualitatively more complex than in single-agent learning, not the least because one cannot separate the process of *learning* from the process of *teaching*. Consider what may happen when playing the Stackelberg game of Figure 1 repeatedly. Notice that Up is a strictly dominated strategy, so regardless of what the opponent chooses the row agent would prefer to play $Down$. However, if the opponent is also learning, this would presumably prompt it to learn to play $Left$, resulting in a payoff of 2 for the row agent. If the row agent instead played the seemingly suboptimal action of Up, the opponent may learn to instead play $Right$, giving the row agent the higher payoff of 3. We can see that in this instance, teaching can play as much of a role in achieving a desirable outcome as learning.

	$Left$	$Right$
Up	$1, 0$	$3, 2$
$Down$	$2, 1$	$4, 0$

Figure 1: The Stackelberg Game

So how does one think about learning (or, more precisely, learning and teaching) in this setting? This is where we must be very precise about our goals in such learning. In this paper we focus on what we termed the 'agent-centric' agenda in previous work. The agent-centric learning agenda is a prescriptive one and asks how an agent should act in order to maximize its reward in an environment containing other independent agents, who may also be learning (possibly using a different algorithm). The intuition driving agent-centric learning is the following. One cannot learn effectively against arbitrarily complex and strange opponents (we use the term 'opponent' neutrally; we allow for competitive el-

ements, but also cooperative ones). In order to make any headway, one must make some informed guess against the target class and optimize for it. At the same time, one cannot ignore the possibility that one's guess was wrong about the target class, and should protect oneself against such a error. Finally, one should allow for the fact that other agents may be using the same learning algorithm, and should exploit this fact to coordinate with them when advantageous. The question is how to effectively weave these three elements together.

We do this in two steps. First, in Section 3 we survey previous literature (from both AI and game theory) that has provided formal criteria for agent-centric learning, including its strengthes and limitations. Then in Section 4 we provide our own criterion, which we believe strengthens and extends previous criteria (it also unifies and generalizes criteria we ourselves proposed in the past [28]). Despite a number of subtle technical details, the new criterion is conceptually simple and applies broadly (in particular, to any n-player repeated game). After presenting the criterion we discuss some of its properties, including some potential concerns and its special properties in the 2-player case.

In Section 5 we begin to tackle the algorithmic question of how to meet our criterion. In this section we set out an abstract modular system for agent-centric learning. The system consists of several modules including a teaching module, a learning module, a coordination module, and a security module. These modules can then be specialized and composed differently, depending on the setting (2- versus n-player games) and class of opponents. Next we proceed to give two concrete instantiations of the framework. In Section 6 we target the class of stationary opponents, while Section 7 provides two algorithms for a class of adaptive opponents with known memory bounds. In each case we start by proving that the resulting algorithm is correct against our formal criterion. However, we believe that all formal requirements – including our own – are merely baseline guarantees, and any proposed algorithm must be subjected to empirical tests. We think it is fair to say that our level of empirical validation is unprecedented in the literature. We show the results of tests of our new algorithms with a number of major existing algorithms, using a recently-developed game theoretic test-bed called GAMUT [26] to systematically sample a very large space of games.

We conclude in Section 8 with a summary of our main messages, and a brief discussion of some of the additional research avenues awaiting exploration.

2 The environment

In order to formally define the setting considered within this paper, we start with the standard definition of a finite stage game (aka normal form game):

Definition 1. *A stage game is a tuple* $G = (N, A_1, ..., A_n, u_1, ..., u_2)$*, where*

- N *is a finite set of players, with* $n = |N|$
- A_i *is a finite set of actions available to player* i
- $u_i : A_1 \times A_2 \times ... \times A_n \rightarrow \Re$ *is a utility function for player* i

	Dare	*Yield*
Dare	$0, 0$	$4, 1$
Yield	$1, 4$	$2, 2$

(a) Chicken

	Cooperate	*Defect*
Cooperate	$3, 3$	$0, 4$
Defect	$4, 0$	$1, 1$

(b) Prisoner's Dilemma

Figure 2: Example stage games. The payoff for the row player is given first in each cell, with the payoff for the column player following.

Figure 2 shows two well-known games from the literature, to which we'll refer again later.

In a repeated game the stage game is repeated, finitely or infinitely. After each round, each player is informed of the joint set of actions played by all the players and receives its own reward. Each player is assumed to be interested in maximizing its average reward for finitely repeated games and the limit average for infinite games (we ignore the subtlety that arises when the limit does not exist, but this case does not present an essential problem). We will restrict attention to games in which all of the payoffs in the game are within a finite bounded range, [-b, b]. For our purposes, we assume all players have full knowledge about the structure and payoffs of the game at all times, but are unaware of the strategies employed by the other players.

Throughout this paper we will occasionally make reference to some terms and concepts from game theory. For those readers wishing an introduction or refresher, we will devote the rest of this section to defining those concepts used in the rest of this paper as well as clarifying the notation used in our formal definitions.

In general, a strategy in a repeated game is a mapping from the history of the game to a distribution over actions. In our setting, where the game structure is known and the opponents' actions are observable, the full history can be captured by recording the outcomes of each stage game played by the players. A stage game outcome, o, is denoted as the single action played by each player: $o =< a_1, ..., a_n >$, where $a_i \in A_i$. A repeated game outcome, O, is a sequence (finite or infinite) of stage game outcomes: $O =< o_1, o_2, ... >$. The value of the outcome for player i, $V_i(O)$, is then the average of the rewards the player received from each stage game outcome in O. A history, h, of the game is a sequence of outcomes: $h =< o_1, ..., o_t >$, where t is the number of stage games the players have played so far in the repeated game. A strategy, π_i, for the repeated game is then a function mapping each possible history to a distribution over actions for the given player, i, to play in the next time period: $\pi_i : H \rightarrow \Delta A_i$, where H is the set of possible histories and ΔA_i is the space of probability distributions over the set A_i. If a player chooses its actions according to the same distribution regardless of the history it is said to be using a 'stationary strategy', $\pi_i \in \Delta A_i$.

Using π to indicate the joint strategies for all the players, we can define the expected reward a player, i, would receive for a given set of strategies as $V_i(\pi)$. For simplicity in later definitions we can also introduce π_{-i} to indicate the

strategies for all players except player i and $V_i(\pi_i, \pi_{-i})$ for the expected value to player i for playing strategy π_i if all the other players are playing according to π_{-i}. Π_i will represent the space of possible strategies for player i. One subtlety we need to be aware of is the question of whether the opponents must choose their actions independently or can coordinate to randomize over joint actions. For this paper we will assume the worst and define the opponents' joint strategy space as $\Pi_{-i} : H \rightarrow \Delta(A_1 \times ... \times A_{i-1} \times A_{i+1} \times ... \times A_n)$.

We can now introduce the idea of the best response for a player given the strategies used by the other players, $BR_i(\pi_{-i}) = argmax_{\pi_i \in \Pi_i} V(\pi_i, \pi_{-i})$. Note that the best response is technically a set of strategies since the above equation may have multiple solutions in many games. A Nash equilibrium, π, is then a set of strategies such that they are all a mutual best response to one another, $\forall_i \pi_i \in BR_i(\pi_{-i})$.

Note that calculating a best-response requires that a player know the actual strategies used by all the other players. Often however we're concerned with what to do if the other players' policies are unknown. We then can define a security value (aka minimax value) for our player which is the maximum reward it can guarantee regardless of what policies its opponents are using:

$$SV_i = max_{\pi \in \Pi_i} min_{\pi_{-i} \in \Pi_{-i}} V_i(\pi_i, \pi_{-i}) \tag{1}$$

A policy that is guaranteed to achieve this value on expectation is called a security policy or maxmin policy.

Another situation that will be relevant for our work is the case in which multiple players are attempting to cooperate in selecting a joint vector of payoffs. For our purposes we will define a player to be 'individually rational' if it only considers accepting outcomes in which its payoff is at least its security value. Given a set of possible outcomes, an individual outcome (joint action profile), is considered Pareto efficient(PE) over that set if there is no other outcome in the set that dominates it. In this context, one outcome dominates another if it is at least as high for all players and strictly higher for at least one player. Formally, for a set of players, X, and outcomes, O:

$$PE_X(O) = (o \in O | \neg \exists_{o' \in O} (\exists_{i \in X} V_i(o') > V_i(o) \wedge \forall_{j \in X} V_j(o') \geq V_j(o))) \tag{2}$$

3 Previous criteria for multi-agent learning

To our knowledge, Bowling and Veloso [5] were the first in the AI community to explicitly put forth formal requirements. Specifically they proposed two criteria:

Rationality: *If the other players' policies converge to stationary policies then the learning algorithm will converge to a stationary policy that is a best-response (in the stage game) to the other players' policies.*

Convergence: *The learner will necessarily converge to a stationary policy.*

At first glance these criteria are reasonable, but a deeper look is less satisfying. First, note that the property of convergence cannot be applied unconditionally, since one cannot ensure that a learning procedure converges against all possible opponents in finite time without sacrificing rationality. So implicit in that requirement is some limitation on the class of opponents. And indeed Bowling and Veloso acknowledge this and choose to concentrate on the case of self-play, that is, on opponents that are identical to the agent in question. Note that when combined with the rationality criterion this is equivalent to requiring that an algorithm converge to a Nash equilibrium in self-play. Given this constraint, Bowling and Veloso then proposed an algorithm satisfying their criteria for the class of known repeated games with two players and two actions per player. Later work by Conitzer and Sandholm [12] proposed a new algorithm meeting both criteria for arbitrary known repeated games.

Additionally, while it is fine to consider opponents playing stationary policies, there are other classes of opponents that might be as relevant or even more relevant; this should be a degree of freedom in the definition of the problem. For instance, one might be interested in the classes of opponents that can be modelled by finite automata with at most k states; these include both stationary and non-stationary strategies. Also these first proposals only apply in very limited scenarios. The rationality criterion is only required when *all* the opponents converge to stationary policies and the convergence criterion is only applicable when *all* the agents are using the same algorithm. No guarantee at all is required if even one opponent is using a non-stationary strategy or if there are a mix of agents with some using the proposed algorithm and others using stationary strategies. The danger of lacking such a guarantee is that it leaves the algorithm vulnerable to potential exploitation by a clever opponent, such as one using the approach shown by Chang and Kaelbling [9] which capitalizes on algorithms designed around policy hill-climbing.

We also find the property of requiring convergence to a stationary strategy particularly hard to justify. Consider the Prisoner's Dilemma game in Figure 2. Prisoner's Dilemma has been extensively studied [1] and numerous algorithms proposed that allow two agents to cooperate on the advantageous cooperation outcome without being exploited. The simplest but perhaps most effective of these is the Tit-for-Tat algorithm. Tit-for-Tat starts by cooperating and thereafter repeats whatever action the opponent played last. Note that any approach that considers only stationary opponents must always play $Defect$, since this is the unique best response to any stationary opponent. Against Tit-for-Tat this results in a payoff of 1, but the strategy of always playing $Cooperate$ would yield a payoff of 3. Similarly, in the game of Chicken, also shown in Figure 2, strategies that alternate daring while its opponent yields and yielding while its opponent dares achieve higher expected payoffs in self-play than any stationary policy could guarantee. This limitation of using stage game equilibria was directly addressed by Brafman and Tennenholtz [6] and a counter-proposal made for how to consider equilibria in repeated games.

Our final point regarding these two criteria is that they express properties that hold in the limit, with no requirements on the algorithm's performance in

any finite period.

While relatively new to the AI community, these issues have been addressed numerous times in game theory, under the names of universal consistency, no-regret learning, and the Bayes envelope, dating back to at least the work of Hannan [17] (see a paper by Foster and Vohra [14] for an overview of this history). There is a fundamental similarity in approach throughout, and we will take the two criteria proposed by Fudenberg and Levine [15] as being representative.

Safety: *The learning rule must guarantee at least the minimax payoff of the game.*

Consistency: *The learning rule must guarantee that it does at least as well as the best response (in the stage game) to the empirical distribution of play when playing against an opponent whose play is governed by independent draws from any fixed distribution.*

Fudenberg and Levine then define *universal consistency* as the requirement that a learning rule do at least as well as the best response to the empirical distribution of play regardless of the actual strategy the opponent is employing (this implies both safety and consistency) and show that a modification of the fictitious play algorithm [7] achieves this requirement. Fudenberg and Levine later strengthened their requirement by requiring that the learning rule also adapt to simple patterns in the play of its opponent [16].

An equivalent requirement used by other researchers is that an algorithm should achieve no regret in the limit against any opponent. The *regret*, $r_i^t(a_j, s_i)$, of agent i for playing the sequence of actions s_i instead of playing action a_j, given that the opponents played the sequence s_{-i} is defined as follows.

$$r_i^t(a_j, s_i|s_{-i}) = \sum_{k=1}^{t} R(a_j, s_{-i}^k) - R(s_i^k, s_{-i}^k)$$

The total regret for the agent is then the maximum regret for any action. Hart and Mas-Colell proposed a regret matching algorithm [18] that provably achieves at most zero regret in the limit (note that an algorithm could have negative regret against some opponents).

Recently, these ideas have also been adopted by researchers in the artificial intelligence community (e.g., [19] and [34]). In recent work [3], Bowling attempted to combine these criteria by proposing that an agent should both guarantee a no-regret payoff and achieve convergence in self-play. He then put forth GIGA-WoLF, a no-regret algorithm that provably achieves convergence in self-play for games with two players and two actions per player.

In recent work, Banerjee and Peng [2] have addressed our concern about only requiring guarantees about the behavior in the limit. Their algorithm is guaranteed to achieve ϵ-no-regret payoff guarantees with small polynomial bounds on initial exploration time and uses only the agent's ability to observe what payoff it receives for each action.

A limitation common to all these approaches is that the game theoretic basis they're derived from was initially focused on large-population games and therefore ignores the effect of the agent's play on the future play of the opponent. This can pose problems in smaller games. Let us again consider the game of Prisoner's Dilemma with a Tit-for-Tat opponent. The only universally consistent strategy would be to defect at every time step, ruling out the higher payoff achievable by cooperating. Clearly, a universally consistent (or no-regret) policy is not the best response in this richer strategy space.

In principle it would be possible to derive a more powerful notion of regret in which one calculates the regret against a richer strategy space than the set of pure stage-game strategies. By including non-stationary strategies it would become possible to allow strategies that would respond appropriately in a situation like that described above when facing a Tit-for-Tat opponent. While there are a number of challenges involved in making this transition, the first steps towards this stronger notion have been taken recently in both game theory and artificial intelligence [22, 13, 8].

4 A New Criterion

One thing to notice about most of the previous proposals is that they tend to enforce a constraint on how the agent should play. This constraint can either be direct, such as requiring convergence to an equilibrium, or more subtle, such as the requirement in universal consistency to never play an action that is dominated in the stage game. Going back to our original statement of the problem for learning in multi-agent systems, we are really most concerned with creating agents that receive a high payoff in their environment. The question then becomes how high a payoff we can reasonably require. Notice that the payoff that can be achieved varies with the strategies of the other agents. Intuitively, the criterion we are after is rather straightforward: Given a target set of opponents, we would like all agents using our algorithm to achieve at least the value of a joint best response against any opponents in the target set, assuming the other opponents are colluding to lower their payoffs.

It turns out that capturing this condition precisely raises a number of subtleties. For instance, note that while we would ideally like to require that an agent achieve the highest possible value given the actual strategies of the opponents in a given game, this is clearly impossible if we allow arbitrarily complex opponent agents. If the opponents can choose different actions for every possible past history of the game, we may never be able to learn how to optimize our agent's action to account for the opponent's strategy since past observations about the opponent may have no correlation with its future play. We propose to instead require that the agents achieve a jointly optimal best response against a predefined "target" set of possible opponent strategies while still maintaining a security value guarantee against any possible opponent.

Besides the problem of opponents with arbitrary complexity, we can have an additional problem when requiring best-response against any possible opponent

from a set. If we assume that the agent may only play a single repeated game against an opponent, the agent may be forced to play actions with irrecoverable consequences before it has any chance to learn the true best response. As an example, consider the prisoner's dilemma game again, only this time the agent is faced with one of two possible opponents. One opponent plays cooperate until the agent defects even once and then plays defect forever (the so-called 'grim trigger' strategy), while the other plays cooperate until the agent cooperates even once and then plays defect forever. No agent could achieve the best possible payoff against both opponents. We propose to address this by only requiring that an agent achieve the value of the best-response that is possible after an initial period of exploration at the beginning of the game.

We also need to consider the issues of coordination between the agents when selecting random actions. Although other researchers may find different assumptions appropriate for particular settings, we have chosen to focus on the most pessimistic/conservative assumptions:

- All agents using the algorithms under consideration select their action independently from one another.
- There may exist opponents that are capable of selecting actions according to a distribution over joint actions.

To assemble these requirements and intuitions into a formal criterion we need the following definitions in which the set of players is partitioned into three sets:

- The set of "designed players", denoted by X, who adopt the learning algorithm under consideration.
- The set of opponents in the target set, denoted by Y.
- The set of opponents playing in an unconstrained fashion, denoted by Z.

Definition 2. *Given:*

- *an n-player repeated game G*
- *a history H*
- *a 3-way partition (X,Y,Z) of the n players*
- *a specification C of repeated-game strategy for each player in Y*

The set of payoff profiles enforceable by X given C and H *consist of all* $P \in \mathbb{R}^{|X|}$, *such that:*

- $\forall_{i \in X} P_i \geq SV_i$, *where* SV_i *is the security value for the* i^{th} *player in X given history H and the assumption that players in Y play according to C.*
- *There exist a set of strategies for the players in X that have an expected payoff, over all outcomes with an initial history of H, of at least* P_i *for all players, i, in X regardless of what strategy players in Z use as long as players in Y play according to C.*

The set of such payoff profiles is denoted by ENF(X,C,H)

Definition 3. *Given G, H, (X,Y,Z), and C as indicated above, an outcome O of the repeated game is said to be* ϵ-Pareto-efficient enforceable for X given C and H *if there is no profile p* $\in$*ENF(X,C,H) such that p minus* ϵ *dominates* $V_X(O)$*, where* $V_X(O)$ *is the vector of payoffs to players in X for outcome O.*

We combine these definitions to specify a property for a given learning algorithm and set of target opponent strategies.

Definition 4. *Given an n-player repeated game G and a set of target opponent strategies S, an algorithm A is said to be* (ϵ, δ)-guardedly optimal for G given S *if there exists a t such that for any partition (X,Y,Z) of the n players, any specification C:Y*$\rightarrow$*S, and any set of strategies for Z, if players in X play according to A, after any initial history H of length t (the* "initial experimentation period"*), the outcome of the game is* ϵ-Pareto-efficient enforceable for X given C and H *with probability at least* $1 - \delta$*.*

With these definitions we can specify our formal criterion for a learning algorithm A(S), where S is the target set of opponent strategies:

Definition 5 (Guarded Optimality). *Given a class S of possible opponent strategies, an algorithm is* guardedly optimal *if for any choice of* $\epsilon > 0$*,* $\delta > 0$*, and any n-player repeated game G, the algorithm is* (ϵ, δ)*-guardedly optimal for G given S.*

As mentioned earlier, this criterion is somewhat complex because of the various subtleties involved. And so it is instructive to look at it in the special case of two player games. In this case it simplifies to the set of criteria shown below. Note that these are similar to criteria we've previously proposed for two-player games [28], although the new Auto-Compatibility criteria is stronger since it now applies to the *joint* payoffs of the two agents.

Definition 6 (Targeted Optimality). *When the opponent is a member of the target set, the average payoff is at least* $V_{BR} - \epsilon$*, where* V_{BR} *is the expected value of the best response in terms of average payoff against the actual opponent.*

Definition 7 (Auto-Compatibility). *During self-play, the average payoff is Pareto efficient over the set of outcomes in the game.*

Definition 8 (Safety). *Against any opponent, the average payoff is at least* $SV - \epsilon$*.*

Remark 1. *For any two-player repeated game, an algorithm is guardedly optimal if and only if it satisfies targeted optimality, auto-compatibility, and safety.*

Finally, let us step back and see how this proposal compares with the past criteria discussed in Section 3. Considering universal consistency, we can see that our criterion implies the safety condition and the consistency condition for any target class that includes all stationary opponents, but is incomparable with the general concept of universal consistency (or, equivalently, no-regret). Note

that while no-regret is a strictly stronger requirement than security value for a single player, it can be incompatible with other desirable requirements (such as best-response to adaptive opponents or Pareto-efficient self-play) as described in the previous section. While it would be possible to address these conflicts as special cases in a combined criterion, there may exist additional incompatible properties one would want to require for particular applications. Another possible way of reconciling these properties would be to consider stronger notions of regret as discussed at the end of the last section. Requiring that a player instead attain a payoff at least as high as any strategy in a broader set of adaptive strategies would require additional constraints on the play of the agent, but could resolve the inconsistencies between no-regret guarantees and some of the desirable properties referenced earlier. We leave this for future work.

A possible complaint about our approach would be that by specifying our target set of opponents, we leave ourselves open to exploitation by other algorithms outside the target set. While it is true that by knowing the details of our approach it might be possible to craft algorithms that do well in response, this is not necessarily disadvantageous. In many games where cooperation is possible this could encourage the hypothetical algorithm to coordinate in order to achieve a desirable joint outcome. At the same time, in more adversarial games, we still have the default guarantee of the security value for the game to avoid getting taken advantage of arbitrarily.

Another issue that has been raised about the guarded-optimality criterion involves our focus on requiring that multiple agents using the same algorithm collude with one another to achieve a PE outcome. In particular, if an agent knows that other agents are using an algorithm meeting this criterion, would it also wish to adopt such an algorithm? A tempting solution to this question would be to add an additional criterion requiring that the proposed learning algorithms form a learning equilibrium [6] with one another in self-play. Unfortunately this requirement is incompatible with having a security value guarantee against any opponent. To see this, let's once again consider an agent playing a repeated game of Chicken from Figure 2. If the agent knows its opponent must secure at least its security value against any player, then the agent's optimal strategy is to always play "Dare", guaranteeing the agent the highest possible payoff. Therefore two algorithms satisfying the safety criterion could never form a learning equilibrium with one another in this repeated game.

5 Algorithmic Framework

Besides proposing a novel criterion, we also want to provide algorithms that can provably achieve the criterion for particular target sets and perform well in practice against other opponents. One of the main challenges we face is that the algorithm needs to behave differently depending on the types of opponents it is dealing with. In order to deal with this in a general fashion, we propose a modular design based on general building blocks. The key is determining the types of the opponent players (members of the target set, other players using the

same algorithm, or unconstrained players) and then selecting the appropriate algorithm to use.

We propose five main building blocks for various settings:

- **Learn Best Response**: Using observations about the opponents' play, estimate the actual strategies of the opponents and play a best-response strategy.
- **Coordinate**: Select a single, common joint strategy for all the self-play players from among a set of Pareto-efficient possibilities.
- **Secure Value**: Play a strategy that ensures that the player receives at least the security value against any possible set of opponents.
- **Signal**: In some settings it may be necessary to play such as to explicitly signal that the player is a member of a certain class or not a member of another possible class.
- **Teach**: Play so as to produce a particular desired behavior in the actions of an adaptive player.

Additionally, the algorithm will need to observe the opponents' play in order to choose an appropriate component. Depending on the exact implementation of each of these blocks, this may require an additional component for explicit observation and sampling.

In order to create an algorithm for a given target class, we need to follow a number of steps:

1. Choose an appropriate set of building blocks for the setting.
2. Decide on an instantiation for each building block (e.g., determine how to calculate the best-response against a member of the target class).
3. Design the flow of control for calling each building block at the appropriate time given the observations in the game.

In the next section we use this framework to build an algorithm that provably meets the guarded optimality criterion for the target class of stationary opponents. The following section then extends and alters the algorithm to meet the criterion for a class of adaptive strategies with bounded recall.

6 First Instantiation: Stationary Opponents

Even though stationary opponents constitute one of the simplest target classes, there are still many subtleties and complexities involved. While stationary opponents have been dealt with frequently in the literature, relatively little work has addressed situations in which there is a mixture of stationary opponents and other non-stationary players. In this section, we address these cases by discussing the construction of a new algorithm that satisfies guarded optimality

by instantiating the building blocks put forth in the previous section. We will call this algorithm PCM(S) (Partition, Coordinate, and Monitor for the target class of stationary opponents (S)). All the players conforming to the designed algorithm will be called cooperating players (coop players), while other players that do not belong to the target set will be called non-cooperating players (non-coop players). Note that a coop player does not have to use the designed algorithm but only needs to follow the protocol for cooperating.

6.1 Algorithm Description

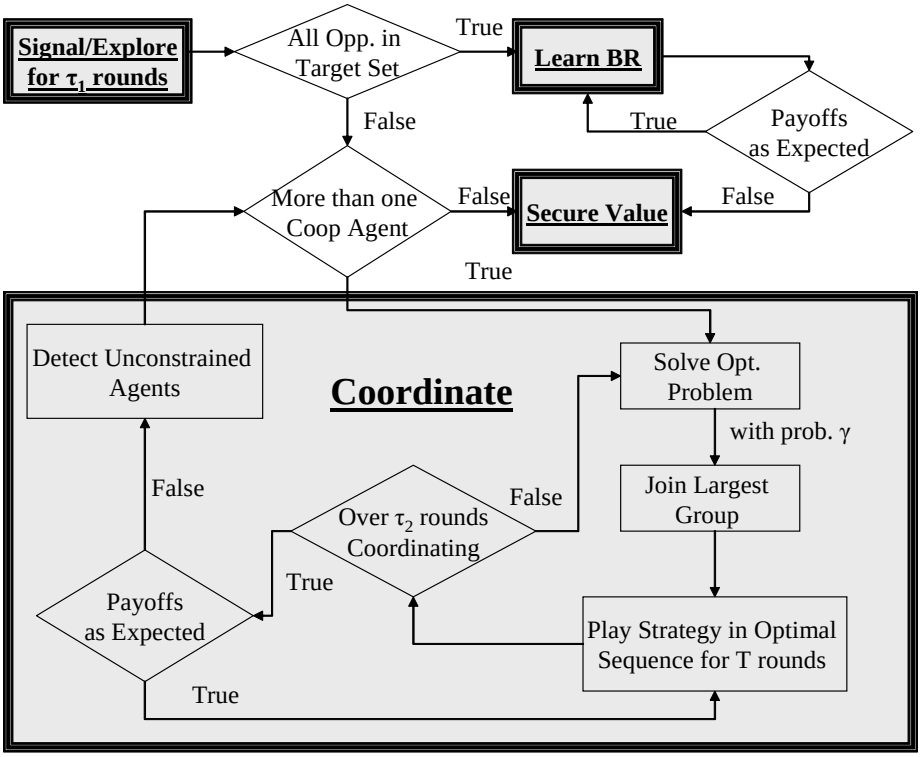

Figure 3: Flow of control for PCM(S) algorithm.

Our main goal is to design as simple an algorithm as possible that can achieve the guarded optimality criterion for the target class of stationary opponents. For our purpose, the four modules "Signal", "Learn Best Response", "Coordinate", and "Secure Value" will suffice to achieve this. In Figure 3, we show how these four blocks can be put together. The four bolded rectangles represent these four blocks. To preserve clarity, the figure only shows a detailed view of the most complex block, "Coordinate". The full pseudocode for the algorithm is provided in Appendix A.

Within the "Signal" block, the coop players will play a pure strategy for τ_1 rounds and then switch to a different pure strategy for another τ_1 rounds. By the end of this block, the coop players will be able to correctly partition all coop players and stationary opponents into two different sets with high probability (the probability that a stationary opponent, i, would generate the two sequences of pure strategies goes to zero at the rate $(\frac{1}{|A_i|})^{\tau_1}$).

Each coop player can now essentially reduce the current game to a smaller game by removing all stationary opponents and using the expected payoffs for each of the remaining outcomes instead. If there is only one remaining player in the reduced game, it can make the transition to the "Learn Best Response" block to find the best response to the stationary opponents. Using the reduced game, finding a BR strategy against stationary opponents is straightforward, since the player can simply choose the action that gives the highest expected payoff.

If there are multiple non-stationary players left in the reduced game, the PCM(S) players will tentatively mark other players as coop and switch to the "Coordinate" block to synchronize with each other on a joint profile that they should adopt to achieve the guarded optimality criterion. This joint profile maximizes the sum of the rewards for the coop players among enforceable outcomes while still guaranteeing individual rationality for each of them. This Pareto-efficient enforceable outcome is the solution of the optimization problem:

$$max_{\pi_X \in \Pi_X}(min_{\pi_Z \in \Pi_Z} \Sigma_{i \in X} V_i(\pi_X, \pi_Z)) \tag{3}$$

In the above equation X is the set of players using PCM(S), and $\Pi_X = \Pi_{X_1} \times ... \times \Pi_{X_m}$, for all $X_i \in X$, subject to the constraint that $\forall_i(\forall_{\pi_Z \in \Pi_Z} V_i(\pi_X, \pi_Z) \geq SV_i)$. Z are the other players, with $\Pi_Z = H \rightarrow \Delta(A_{Z_1} \times ... \times A_{Z_p})$, with $Z = \{Z_1, ..., Z_p\}$. SV_i is the maxmin value of coop player i in the reduced game as defined in Section 2.

The coop players can approximate this outcome by limiting the Π_X in the equation above to be sequences of mixed strategies of length L. The solution of the corresponding optimization problem is then a cycle of length L specifying a mixed strategy for each player in X to follow at each step. For any given $\epsilon' > 0$ and feasible payoff profile in the infinitely repeated game, we can find a sequence of length L polynomial in n and $\frac{1}{\epsilon'}$ such that this sequence approximates the target payoff profile within ϵ'. We give the proof for this in a later section.

Since there are possibly many different correlated sequences that can satisfy the guarded optimality criterion, the coop players need to agree on the same sequence. The "Coordinate" block consists of several coordinating processes between the coop players. Each process will last for τ_2 rounds in which the coop players try to converge to the same sequence. Each coop player will pick one such sequence at the beginning of the process and then with probability γ switch to a different sequence that is being used by more coop players. At the end of each process, they will either succeed by achieving a payoff profile at most ϵ away from the expected payoffs for all the coop players or they will be able to detect at least one non-coop player. In the later case, they will mark the detected player as non-coop and restart the coordinating process.

If all other players are marked as non-coop, the remaining coop player can now make the transition to the "Secure Value" block. Within this block, the player will calculate the maxmin strategy as defined in Section 2 by solving the corresponding linear program for the reduced game with the strategies of the stationary opponents fixed. This is the best payoff the player can guarantee for itself since the non-coop players could force its payoff arbitrarily close to its security value for the reduced game.

For any given $\epsilon > 0$ and $\delta > 0$, we can choose appropriate values for τ_1, τ_2, γ, and ϵ' that allow the player to guarantee it achieves the guarded optimality criterion with probability at least $1 - \delta$. We give a formal proof of this fact in the following section.

6.2 Formal Properties

Theorem 1. *For any given $\epsilon > 0$ and $\delta > 0$, PCM(S) is (ϵ, δ)-guardedly optimal for the class of stationary opponents given an initial experimentation period with length polynomial in $M, n, \frac{1}{\epsilon}$, and $\frac{1}{\delta}$.*

The above theorem holds for repeated games where n is the number of players and each player has at most M actions. Since we are only considering games with bounded payoffs, we can assume, without loss of generality, that all the payoffs are normalized to lie between 0 and 1.

Proof. The proof can be constructed naturally from the following lemmas which are proved in appendix B:

Lemma 1. *For any given $\delta_1 > 0$, $0.5 > \epsilon_1 > 0$, there exists a τ_1 polynomial in $n, M, \frac{1}{\epsilon_1}$, and $\frac{1}{\delta_1}$ such that if a player uses a full action history of length at least $2\tau_1$, and a recent action history of length τ_1, the probability for all coop players to correctly partition stationary and coop players into two different sets is at least $1 - \delta_1$.*

Lemma 2. *Within the "Coordinate" block, for any given $\epsilon_2 > 0$ and $\delta_2 > 0$ there exists a τ_2 polynomial in $n, M, \frac{1}{\epsilon_2}$, and $\frac{1}{\delta_2}$ such that with probability at least $1 - \delta_2$, after at most τ_2 rounds, either all cooperating players will converge to an ϵ_2-Pareto-efficient enforceable outcome or a new non-cooperating player will be identified.*

From Lemma 1, by the end of the "Signal" block, after $2\tau_1$ rounds, the coop players have correctly partitioned stationary players and coop players into two different sets with probability at least $1-\delta_1$. From Lemma 2, after each coordinating process of τ_2 rounds, the players will either achieve an ϵ_2-Pareto-efficient enforceable outcome or identify a new non-coop player with probability at least $1-\delta_2$. Thus they only need to repeat the coordinating process at most n times. The probability that they will correctly partition all coop players and non-coop players into two different sets and converge to an ϵ_2-Pareto-efficient enforceable outcome after at most $n * \tau_2$ rounds is at least $1-n\delta_2$. Therefore the agents will converge to an ϵ_2-Pareto-efficient enforceable outcome with probability at least

$1 - \delta_1 - n\delta_2$. Setting $\epsilon_1 = \epsilon_2 = \epsilon, \delta_1 = \frac{\delta}{2}$, $\delta_2 = \frac{\delta}{2n}$ we can guarantee that the players will achieve the guarded optimality criterion with probability at least $1 - \delta$ after a learning period, τ, that is polynomial in $M, n, \frac{1}{\epsilon}$, and $\frac{1}{\delta}$.

Moreover, note that before converging to an ϵ-Pareto-efficient enforceable outcome, a coop player might suffer loss of payoff during the experimentation period. However, since the payoff is bounded between [0,1], the total loss over this period is at most τ. To take this into account, we can set $\epsilon_2 = \frac{\epsilon}{2}$ and allow an additional $\frac{2\tau}{\epsilon}$ rounds to pass. The actual payoffs of the coop players after this period will be at most $\frac{\tau}{\frac{2\tau}{\epsilon}} = \frac{\epsilon}{2}$ away from an $\frac{\epsilon}{2}$-Pareto-efficient enforceable outcome. □

Theorem 1 provides the bound on the number of iterations required by the algorithm for the initial experimentation period before the desired outcome is achieved. This leaves open the question of how much computation is required by the algorithm at each iteration of the game. The answer is given by the following proposition:

Proposition 2. *For any n-player repeated game G, let T be the complexity of solving the optimization problem defined in equation 3 in Section 6.1 and τ_3 be the length of the test for non-cooperative agents. The computational complexity of PCM(S) for one iteration of G is $O(M^n * \max(T, \tau_3))$ in the worst case.*

Proof. Let M be the maximum number of actions for one player. To find the worst case complexity for one iteration, we can calculate the complexity for each block of the algorithm as presented in Figure 3:

- Within the "Signal" block, each step can be done in constant time.
- Within the "Learn Best Response" block, the worst step can be done in $O(M^n)$.
- Within the "Secure Value" block, the worst step can be done in $O(M^n)$.
- Within the "Coordinate" block, in the worst step the agent has to find the largest group to join and then tries to detect non-coop opponents. To find the largest group, it has to go through all different subsets of coop players and solve the optimization problem in equation 3 for each subset. The complexity of this operation is $O(2^n * T)$.
- The operation to detect non-coop opponents may need to check all subsets for τ_3 steps each taking total time proportional to $O(2^n * \tau_3)$.

Since the agent can only be executing one block at a time the complexity for one iteration in the worst case is $O(M^n * \max(T, \tau_3))$, where τ_3 is polynomial in $M, n, \frac{1}{\epsilon}, \frac{1}{\delta}$ □

Even though PCM(S) has an exponential worst-case complexity, it is efficient in practice since for most of the iterations, PCM(S) requires only computation that is linear in $M * n$ and T is usually relatively small. In our experiments, on a 2.4Ghz machine, PCM(S) takes under 1 second when playing a 4-player game for 200,000 iterations with 3 actions for each player.

6.3 Empirical Validation and Discussion

Even though our algorithm has been theoretically proven to correctly achieve our formal criterion, we want to demonstrate empirically that the algorithm performs well against a variety of opponents, including those outside the target class. We will use the testing environment first described in prior work [28] by testing against a number of existing approaches from the multi-agent learning literature over a wide variety of repeated games from GAMUT [26]. GAMUT is the result of a project to develop a comprehensive collection of game theoretic matrix games that have been described by researchers in either game theory or artificial intelligence. It contains generators for creating random instances of 34 individual base game classes as well as numerous additional variants and specialized parameter settings (more information and downloads are available at `gamut.stanford.edu`). The existing algorithms we tested against include Local Q-learning [33], a stochastic version of IGA [31], WoLF-PHC [4], JointQ-Max [11], GIGA [34], GIGA-WoLF [3], a version of NoRA [2] using GIGA as its base class, and smooth fictitious play [15]. We also tested all the algorithms against random stationary strategies (Random), the security value strategy (MiniMax), and random strategies that condition their actions on the past outcome (CondStrat).

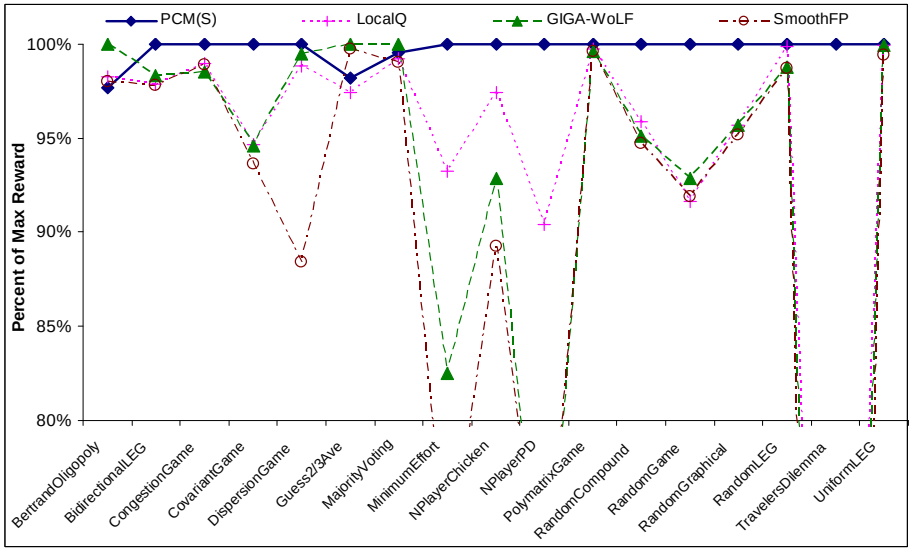

Figure 4: 2 vs. 2: Percent of best reward for last 20K rounds (of 200K) averaged across all opponents for selected games in GAMUT. The rewards were divided by the maximum reward achieved by any player.

We want to focus our attention on settings with more than two players. As the first test we measured the average performance of one pair of players playing against another pair of players. The players in each pair use the same

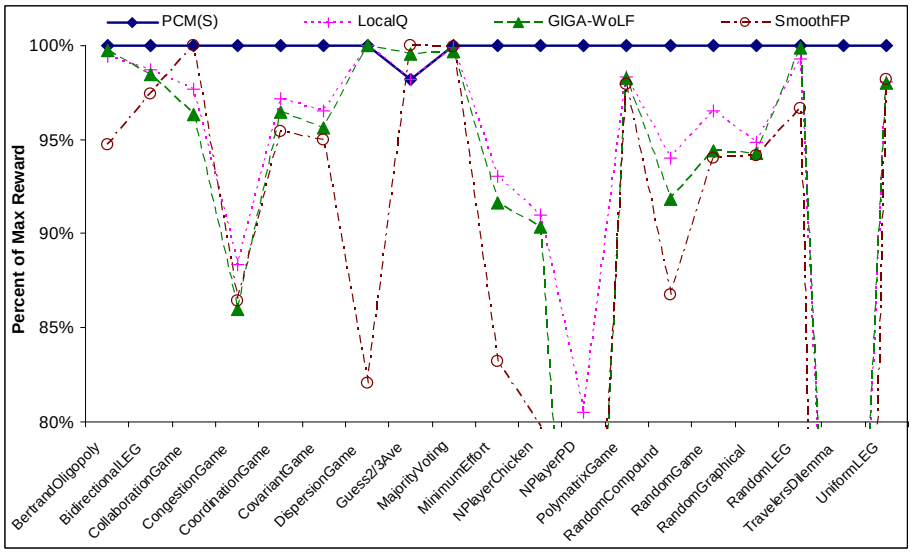

Figure 5: 2 vs. 1: The performance of PCM(S) is even stronger when it outnumbers its opponents.

algorithm though it can be different from the one used by the other pair. In figure 4, we show the average payoffs achieved by each player averaged across the set of possible opponents for a selection of games in GAMUT listed across the x-axis. The y-axis shows the payoff for each algorithm as a percentage of the highest average payoff achieved by any algorithm for the selected game. In order to preserve clarity, we only show the results for four algorithms representative of those with the best performance. PCM(S) achieves the highest or close to the highest payoffs in every game. Unlike other algorithms, PCM(S) has no pitfalls in which its payoffs are significantly worse than those achievable by other approaches in a given game. This is at least partly due to the fact that two players using PCM(S) can cooperate with each other against other players to possibly achieve a higher security value than each individual player could achieve alone.

To further demonstrate this advantage of PCM(S), we slightly adjusted the setting of the experiments to show two players using the same algorithm playing against another algorithm (can be the same or not) and we show the results in figure 5. In this setting, PCM(S) shows an even greater advantage over the other algorithms. The reasons behind the difference in the performance of PCM(S) in the two different settings are the generosity and cooperativeness of PCM(S). PCM(S) will be more likely to cooperate as long as the outcome is PE for all players it considers cooperating. Thus when there are more players using other different algorithms, there are more situations in which PCM(S) will compromise its payoff.

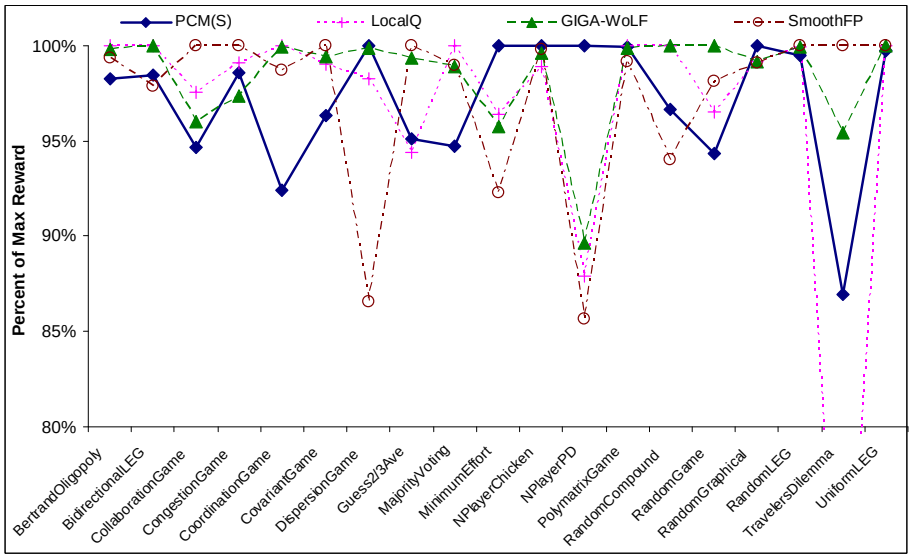

Figure 6: 1 vs. 2: The performance of PCM(S) is weakened by the lack of cooperation opportunities.

We also measure the performance of PCM(S) when playing against two opponents using an identical algorithm. We show the result in figure 6. In this setting, PCM(S) has two disadvantages. The first disadvantage was mentioned above: PCM(S) will be more likely to compromise its payoff if there are more opponents. The second disadvantage is that PCM(S) was designed exclusively to satisfy the criterion for stationary opponents. It does not have the capability to take advantage of adaptive players. When the other two players do not cooperate, PCM(S) will have to resort to the security strategy. In this setting of 1 vs. 2, there is no other player using PCM(S) that it can cooperate with to increase its security values. However, PCM(S) is still able to achieve high payoffs in several games in which there exists a beneficial cooperative outcome due to its flexibility in cooperating with players using other algorithms as long as it still achieves its security guarantee. Note that in this setup, PCM(S) is regularly forced to play its individual security strategy. Given the modular nature of our design, we can easily substitute a different algorithm, such as GIGA, for the security portion that attains the same guarantees for the single PCM(S) agent case. Although we achieve only moderate gains from this augmentation, as seen in figure 7, this simple change rarely hurts the performance. One could extend this approach to add different default behaviors for individual games and then use the methods proposed by McCracken and Bowling [23] to guarantee that the security value is always achieved.

In Table 1 we show the payoff for different algorithms in self-play, that is, when all players use the same algorithm. With an explicit mechanism for sig-

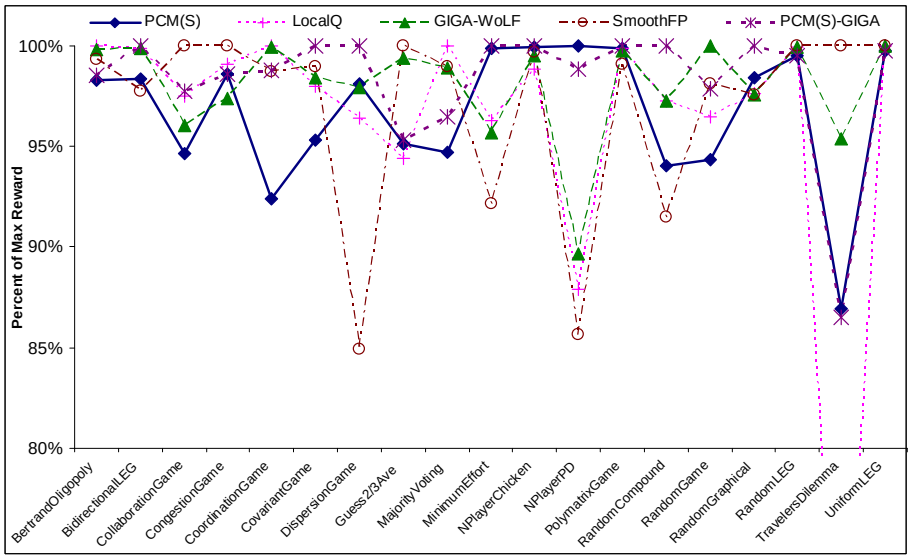

Figure 7: 1 vs. 2: Some empirical gains can be gained by replacing the default security strategy.

naling and coordinating, players using PCM(S) are able to achieve a payoff significantly higher than any other algorithm. As the number of players increases, the gap between the performance of PCM(S) and other algorithms grows larger since those players are much less likely to come across a cooperative beneficial outcome by chance.

	N=2	N=3	N=4
PCM(S)	0.496	0.675	0.559
LocalQ	0.400	0.550	0.340
WoLF-PHC	0.389	0.449	0.292
StochIGA	0.385	0.422	0.257
GIGA-WoLF	0.374	0.411	0.255
SmoothFP	0.118	0.254	0.027
MiniMax	0.103	0.111	0.023

Table 1: Average payoff in self-play by algorithm, as a function of the number of players.

A final analysis we conducted was to address the question of how dependent the empirical performance of the algorithm was on having a long initial training period. While we have formal guarantees that only a polynomial amount of training is necessary, this can still be a long period in practice if we wish small values for ϵ and δ. For instance, selecting an ϵ of 1% will impose a multiple of 10,000 on the amount of training required. In table 2 we compare the

performance given various lengths of training for PCM(S) with GIGA-WoLF. The numbers shown are the average value attained during the last 10% of the rounds. We can see that the performance of PCM(S) degrades gracefully at least down to a range of 10,000 rounds and maintains a significant margin over GIGA-WoLF throughout. Note that the variance is inherently higher for the smallest training periods, so significance becomes harder to estimate.

	5K	10K	25K	50K	100K	200K
PCM(S)	0.259	0.266	0.266	0.268	0.269	0.272
GIGA-WoLF	0.223	0.227	0.228	0.227	0.229	0.230

Table 2: Average payoff over all 4-player environments and opponents as a function of the number of total rounds.

7 Second Instantiation: Adaptive Opponents

Although the PCM(s) algorithm demonstrates desirable formal properties and promising empirical performance, it still fails to address our concerns about the focus in prior work on stationary opponents, since PCM(s) has only weak security-level guarantees for its payoff against opponents whose strategy can depend on the past history of the games. We are aware of very little work to date that deals with adaptive opponents explicitly, although de Farias and Megiddo [13] address it in the design of their experts algorithm and the rational learning approach of Kalai and Lehrer [20] can in principle handle adaptive algorithms of arbitrary complexity as long as they are assigned positive probability in the prior.

One way we could attain better performance against adaptive opponents would be to expand the target set against which we can guarantee a best-response. Note however that we still need to limit the capabilities of the opponents in some way. If we were to consider opponents whose future behavior could depend arbitrarily on the entire history of play, we would lose the ability to learn anything about them in a single repeated game, since we would only ever see a given history once and an opponent's past strategy may have no relation to its future play.

We therefore assume a limit on each opponent's ability to condition on the history. We propose directly limiting the amount of history available, by requiring that each opponent play a conditional strategy (aka bounded recall strategy) where its distribution over actions can only depend on the most recent k periods of past history, $F_i : o_{-1} \times ... \times o_{-k} \rightarrow \Delta A_i$, where o_{-t} is the outcome of the game t periods ago. Additionally, the opponents have a default past history they assume at the start of the game. Note that even this simple model allows us to capture many methods, such as Tit-for-Tat, that most current approaches are unable to properly handle.

7.1 The PCM(A) algorithm

By capitalizing on the modular design of PCM(S), we can design a variant, PCM(A), that achieves our *Guarded Optimality* criterion against this new class of opponents with only minor conceptual modifications. We will be able to use the same flow of control and modules shown in Figure 3 except for the following changes:

- We have replaced the instantiation of the Learn BR module with a new strategy, MemBR, which calculates a best response against conditional strategies. This approach maintains counts of the opponent's actions after each history of length k, which it uses to calculate the optimal set of conditional strategies for each coop player to use.[1] This lets us guarantee that we achieve an ϵ-best response against any members of our target opponent set given that the algorithm observes each length k history a sufficient number of times. This will be satisfied as long as the initial exploration phase continues for a length of time exponential in k. This exponential exploration period is unavoidable since we need to consider the possibility of opponents that only play a desirable action distribution for a single one of the exponentially many possible histories.

- In order to tell if an opponent is a member of the target class we can now calculate the probability that each opponent's play is consistent with our target set by comparing the observed distribution of play for each history at separate times and measuring the deviation in action profiles.

- When there are multiple coop agents and opponents in both the target and unconstrained class, the optimization problem in the coordinate module needs to take into account the distribution over histories generated by the coop and non-coop agents when calculating the achievable payoff profiles.

Theorem 3. *PCM(A) satisfies guarded optimality for the target class of conditional strategies with bounded memory k.*

The outline for the proof of this theorem is included in Appendix C. The initial experimentation period required in satisfying guarded optimality could unfortunately now depend on $(\frac{1}{\lambda})^{(M^{nk})}$, where λ is the minimum probability the opponent assigns to any action ($\lambda = 1$ for opponents that condition only on the coop player's actions). Note that our worst case time complexity also grows similarly as we may now need to solve an optimization problem with up to M^{nk} variables. This has caused us to focus on two-player games in our empirical results, although both of these bounds (computational complexity and amount of training) are based on extremely pessimistic assumptions and are likely to be tractable in practice for larger games with small values of k.

[1] Note that if the opponents condition only on the coop players' actions, we can instead just choose the optimal cycle of player actions with the highest expected reward out of all possible unique player action sequences (those that don't contain a length k repeated subsequence).

7.2 TPCM(A): A teaching algorithm

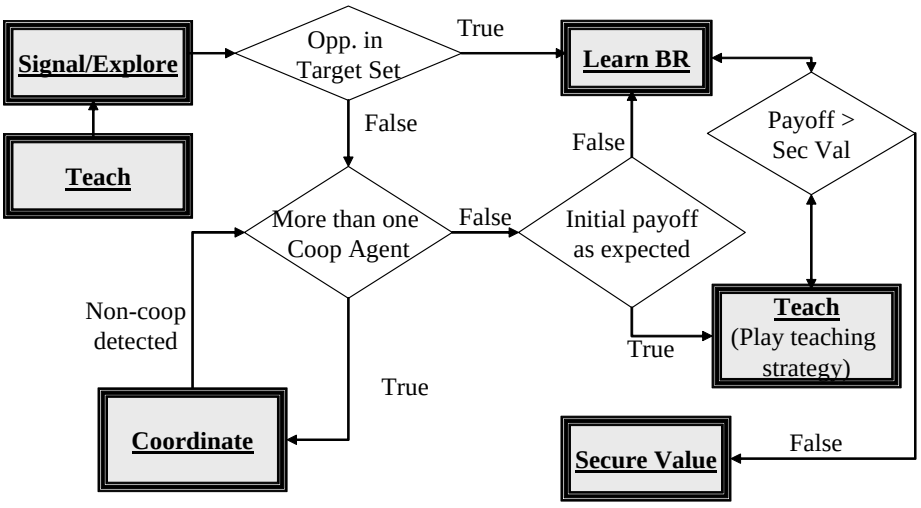

Figure 8: Flow of control for TPCM(A) algorithm.

While the basic modifications described in PCM(A) achieve our formal guarantees against the set of conditional strategies it ignores our intuitions in regards to the importance of teaching as part of an effective strategy against adaptive players. In order to incorporate this idea, notice that we have significant freedom both in how we conduct our initial exploration period and also what default strategy to employ against opponents outside our target class. By restricting our attention to the class of two player games, we can define a new algorithm: Teach, Partition, Coordinate, and Monitor (TPCM(A)), shown in Figure 8. It is based on the PCM(A) algorithm but we have added a new initial block before the signalling step that uses a teaching strategy based on the Godfather algorithm first proposed by Littman and Stone [21]. Godfather selects an outcome in the game matrix which maximizes its own payoff and gives the opponent player at least its security value. Godfather then plays its portion of the target outcome. If the opponent ever plays an action other than the matching action for the target outcome, the player plays a strategy that forces the opponent to achieve no more than its security value until the opponent again plays its target action. TPCM(A) uses a stochastic variation of Godfather that selects a mixed strategy for the player and a target action for the opponent such that the joint strategy gives the opponent a higher expected value than its security value. The stochastic version has two advantages over the deterministic original. First, it can sometimes attain strictly higher payoffs by considering a larger set of outcomes. Secondly, if we additionally require that each action is played with some minimal probability in the player's mixed strategy, we can attain our observation requirements for MemBR while teaching the opponent. After this

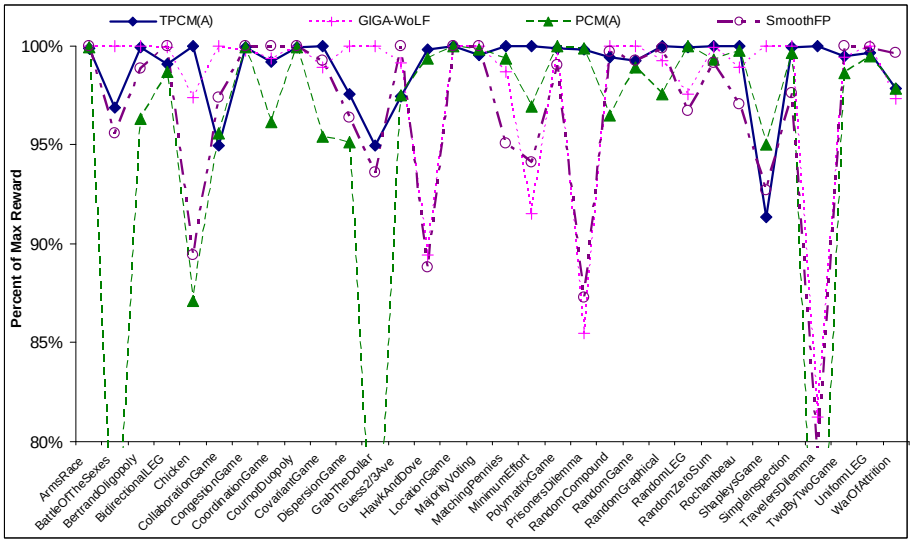

Figure 9: 2 player games: The value of teaching.

initial phase of teaching, TPCM(A) behaves identically to PCM(A) unless it detects that the other agent is not cooperating. When PCM(A) would play the Secure Value module, TPCM(A) instead adopts either MemBR or Godfather as a default strategy depending on its payoff in the initial teaching period. If this payoff was close to the target payoff for the outcome Godfather selected it reverts to Godfather, otherwise it plays MemBR. In either case, it continues to observe its own payoffs and reverts to the security policy if its average payoffs drop below its security value.

Since the only modification has been to add an initial step of fixed length and replace the Secure Value module with a new module with the same guarantees, it is easy to see that TPCM(A) still satisfies guarded optimality for the class of conditional strategies. Note however that the algorithm is restricted to the set of environments with only two players.

7.3 Experimental Results

In addition to a selection of the opponents that were used for testing PCM(S), we also include random conditional strategies (CondStrat), MemBR, and an implementation of the Godfather algorithm [21]. In Figure 9, we can see that our new algorithm TCPM(A) achieves consistently higher performance than any of the other algorithms in nearly every game. The particular versions of PCM(A) and TCPM(A) shown in the graph take conditional opponents with memory of length 1 as their targets. Results showed a slight improvement when considering opponents with a memory of 2, but training time grows significantly.

In order to understand the source of this performance, let's consider the results against individual opponents. Figure 10 shows results for all three of our

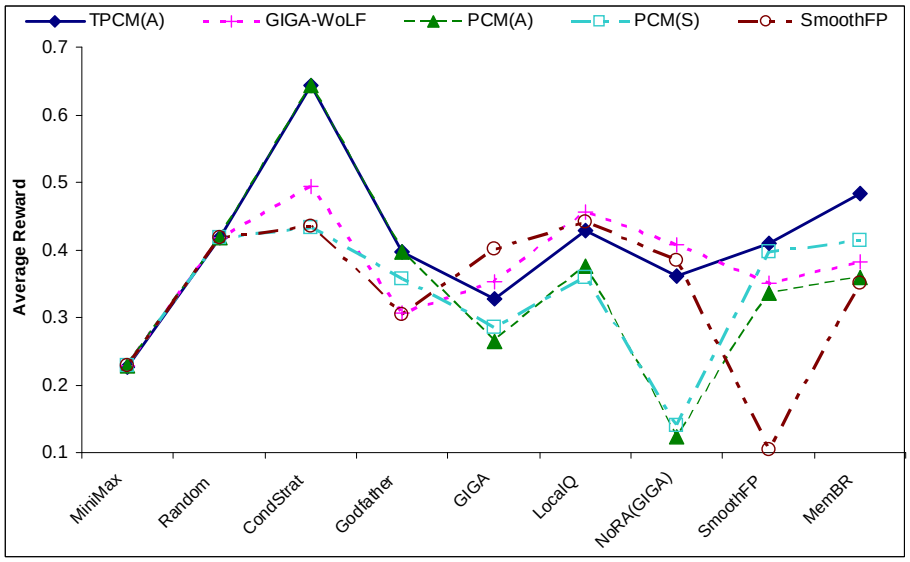

Figure 10: Average value for last 20K rounds (of 200K) by opponent across 2-player games in GAMUT. Game payoffs range from -1 to 1.

new algorithms and two of the most successful previous algorithms for the class of two-player games.

We can see that both PCM(A) and TPCM(A) are able to achieve significant gains in average reward against the non-stationary opponents in their target set, CondStrat and Godfather. Moreover, TPCM(A) shows significantly better performance against the opponents outside its target set. The combination of this improved performance against out of target class opponents and the strong performance in self-play common to all the PCM variants results in the uniformly strong performance we saw in Figure 9.

8 Conclusions and Future Work

We have argued for a new criterion for agent-centric learning in multi-agent systems, one that is more goal oriented than previous proposals, emphasizing high return for the players with fewer constraints on their actual behavior. The criterion also offers the advantage of allowing designers to specify a particular opponent class that an algorithm should perform especially well against.

In order to aid in designing algorithms that meet this criterion, we put forth a general algorithmic framework. The modular nature of our approach allows easy adaptation of successful algorithms to different environments and sets of opponents. Using the framework as a basis we offered three concrete instantiations. The first is optimized to perform well against stationary opponents in n-player repeated games, while the other two focus on opponents whose distribu-

tion of play is a function of the recent history of the game. One of these applies generally in n-player repeated games, while the other is specialized for games with two players. All three of these algorithms meet our formal guarantees and fare well in different environments against a wide variety of opponents.

Going forward, there are many promising areas for future work. One fairly straightforward extension would be to consider other models of adaptive opponents. A common approach used in the literature on bounded rationality [24, 27] is to assume the players can be modeled by finite automata with k states. Note that the automata model is more comprehensive than the set of conditional strategies since any conditional strategy opponent with bounded memory can be modeled by an automata with M^k states if we allow stochastic outputs, but there exist automata that cannot be modeled by any function on a finite fixed history. In the case of automata with deterministic transitions, we can modify our PCM(A) and TPCM(A) algorithms to handle this new class by replacing the best response function. Note that learning a best response to an opponent modeled by an unknown finite automata is equivalent to finding the best policy for an unknown Partially Observable Markov Decision Process, investigated in several papers [10, 25]. While it is a difficult computational problem, we should be able to achieve the same theoretical properties for this alternate set of opponents given similar concerns and caveats to those we encountered in the proof for PCM(A) when ensuring that we get enough observations of the entire state space to calculate an accurate best-response.

Another interesting question to address going forward is whether there is a disciplined way to extend the concept of teaching an opponent to the situation in which there may be multiple opponents. A more general teaching algorithm would allow the TPCM(A) algorithm to be extended to environments with more than two players.

Finally, we are also looking at several ways to expand the set of environments these algorithms can be employed within. Of particular concern is looking for ways to weaken the requirement of full prior knowledge about the payoffs of the game. The major challenge seems to lie in creating the capability to cooperate without knowing or being able to observe the space of payoffs available to the other players. An additional area for further consideration would be the assumption of perfect observability. How could one design an effective algorithm when the players receive only partial information about the past actions of their opponents? Other possible extensions include extending the algorithms to handle stochastic games with multiple states and considering games in which the players care about the discounted sum of the stage game rewards instead of the average.

A PCM(S) Implementation Details

```
∀i oppType[i] ← STATIONARY
for τ1 time steps: Play one action
for τ1 time steps: Play another action
while (∀i oppType[i] = STATIONARY)
  Play BR strategy to all other agents
  For each opponent i, if player i's action distribution for
      the last τ1 rounds deviates by more than ε1 from their
      distribution for the full history
    oppType[i] ← COOP
For all stationary opponents i, stat(i) ← observed distribution
Loop
  Use a plug-in solver to solve equation 3 in Section 4
    seq ← the sequence of mixed strategies in solution
    V_sum ← sum of the payoffs
  for τ2 time steps: \* Coordinating process *\
    Play next mixed strategy in seq
    With probability γ
      For each subset X' of coop agents by decreasing size
        V'_sum ← recalculate the optimal solution with the
          distributions of agents in X' set to the observed
          distributions over the last H * L periods
        seq ← the sequence of mixed strategies in new solution
        If V_sum − V'_sum ≤ ε2 then this is a valid group
          V_sum ← V'_sum; break
  If any coop player i switched to the wrong group during above
    oppType[i] ← NON − COOP; foundNonCoop ← TRUE
  For all coop players i
    V(i) ← expected payoff for i if all coop players follow seq
  For τ3 rounds, play according to seq, recording payoffs in V̂
  While (¬foundNonCoop) \* Monitor for non coop players *\
    If ∃ coop player i such that V̂(i) < V(i) − ε then
      foundNonCoop ← TRUE
      For each subset, Y, of coop and stationary players
          ordered by increasing size
        V' ← Recalculate payoffs for last τ3 periods using the
          target distribution (seq or stat) for players in Y
        If ∃ coop player i such that V̂(i) < V'(i) − ε|Y|/n then
          For all j ∈ Y, oppType(j) ← NON − COOP
    Else play next mixed strategy in seq, update V̂
    While (# coop players = 1) play secure value strategy
```

B Supporting Proofs for PCM(S)

Proof of Lemma 1. Since all coop agents have the same perfect observation of other agents' actions, the probability for all coop agents to correctly partition stationary and coop agents into two sets is equal to the probability for one agent to achieve it. Let d_f be the distribution of the actions of an agent calculated from the full history, d_r be the distribution from the recent history, and d_t be the true distribution of the actions. Let $d(k)$ be the distribution of action k in d. An opponent is assumed to be stationary if $\|d_f, d_r\|_\infty \leq \epsilon_1$, where $\|d_1, d_2\|_\infty = \max_{i=1..M} |d_1(i) - d_2(i)|$. For a stationary opponent we have:

$$\begin{aligned} Prob(\|d_f, d_r\|_\infty \leq \epsilon_1) &\geq Prob\left(\|d_f, d_t\|_\infty \leq \frac{\epsilon_1}{2} \,\&\, \|d_r, d_t\|_\infty \leq \frac{\epsilon_1}{2}\right) \\ &\geq Prob\left(\|d_f, d_t\|_\infty \leq \frac{\epsilon_1}{2}\right) * Prob\left(\|d_r, d_t\|_\infty \leq \frac{\epsilon_1}{2}\right) \end{aligned}$$

Using the Hoeffding inequality we know,

$$\begin{aligned} Prob(|d_t(i) - d_f(i)| > \epsilon_1) &\leq 2\exp(-4(\frac{\epsilon_1}{2})^2\tau_1) \\ Prob(|d_t(i) - d_r(i)| > \epsilon_1) &\leq 2\exp(-2(\frac{\epsilon_1}{2})^2\tau_1) \end{aligned}$$

From the Union Bound Axiom, we get,

$$\begin{aligned} Prob(\forall i, |d_t(i) - d_f(i)| < \epsilon_1) &\geq 1 - 2M\exp(-4(\frac{\epsilon_1}{2})^2\tau_1) \\ Prob(\forall i, |d_t(i) - d_r(i)| < \epsilon_1) &\geq 1 - 2M\exp(-2(\frac{\epsilon_1}{2})^2\tau_1) \end{aligned}$$

And therefore,

$$\begin{aligned} Prob(\|d_f, d_r\|_\infty \leq \epsilon_1) &\geq \left(1 - 2M\exp(-4(\frac{\epsilon_1}{2})^2\tau_1)\right)\left(1 - 2M\exp(-2(\frac{\epsilon_1}{2})^2\tau_1)\right) \\ &\geq 1 - 4M\exp(-2(\frac{\epsilon_1}{2})^2\tau_1) \end{aligned}$$

From the result for one agent, we again apply the union bound to obtain a lower-bound for the probability of checking multiple stationary agents correctly:

$$\begin{aligned} Prob(\exists k : \text{Agent k stationary}, \|d_f(k), d_r(k)\|_\infty \geq \epsilon_1) &\leq n * 4M\exp(-\frac{\epsilon_1^2\tau_1}{2}) \\ Prob(\forall k : \text{Agent k stationary}, \|d_f(k), d_r(k)\|_\infty \leq \epsilon_1) &\geq 1 - 4Mn\exp(-\frac{\epsilon_1^2\tau_1}{2}) \end{aligned}$$

For any $\tau_1 \geq \frac{2}{\epsilon_1^2}\log\frac{4Mn}{\delta_1}$,

$$Prob(\forall k : \text{Agent k stationary}, \|d_f(k), d_r(k)\|_\infty \leq \epsilon_1) \geq 1 - \delta_1$$

For coop agents, $\|d_f, d_r\|_\infty = \frac{1}{2}$ from the algorithm description. So for all $\epsilon_1 < \frac{1}{2}$, no coop agents will be assumed to be stationary and all stationary agents will be correctly identified with probability at least $1 - \delta_3$. □

Before giving the proof for Lemma 2, we need to prove four more lemmas.

Lemma 3. *For any given $\epsilon_3 > 0$ and $\delta_3 > 0$ there exists an H polynomial in M, $\frac{1}{\epsilon_3}$, and $\frac{1}{\delta_3}$ such that if the opponent plays according to a stationary strategy the observed distribution of a sequence of at least H actions will be within ϵ_3 of the true distribution for each action with probability at least $1-\delta_3$. This implies that the expected payoff for a strategy will be within $M\epsilon_3$ of the actual payoff with probability at least $1-\delta_3$.*

Proof. Let $\hat{d}$ be the observed distribution of actions for an agent observed using a history of length H, and d be the true distribution of actions. Let $d(k)$ be the probability for action k in distribution d.

Using Hoeffding's inequality we obtain the result:

$$\begin{aligned}
\forall i \in [1,M], Prob(|\hat{d}(i) - d(i)| \geq \epsilon_3) &\leq 2\exp(-2(\epsilon_3)^2 H) \\
Prob(\exists i \in [1,M] : |\hat{d}(i) - d(i)| \geq \epsilon_3) &= Prob(\cup_{i=1..M} : |\hat{d}(i) - d(i)| \geq \epsilon_3) \\
&\leq \sum_{i=1}^{M} Prob(|\hat{d}(i) - d(i)| \geq \epsilon_3) \\
&\leq 2M\exp(-2(\epsilon_3)^2 H) \\
Prob(\forall i \in [1,M], |\hat{d}(i) - d(i)| \leq \epsilon_3) &\geq 1 - 2M\exp(-2(\epsilon_3)^2 H)
\end{aligned}$$

Setting $H = \frac{1}{2(\epsilon_3)^2}\ln\frac{2M}{\delta_3}$ we obtain:

$$Prob(\forall i \in [1,M], |\hat{d}(i) - d(i)| \leq \epsilon_3) \geq 1 - \delta_3$$

Since the payoff of the agents is bounded by 0 and 1, the difference between actual and expected payoff is at most $\sum_{i=1}^{M} |p_o(i) - p_t(i)|$. Therefore expected payoffs will be within $M\epsilon_3$ of actual payoffs with probability at least $1-\delta_3$. □

Lemma 4. *For any given $\epsilon_4 > 0$, $\delta_4 > 0$, and subset of players, X, any feasible payoff profile, p, in the infinitely repeated game can be approximated by a sequence S, with length L, of joint mixed strategies such that with probability at least $1-\delta_4$, the difference between p and the actual average payoff achieved by each player using S repeatedly for at least H times is at most ϵ_4. L and H are both polynomial in $M, n, \frac{1}{\epsilon_4}$, and $\frac{1}{\delta_4}$.*

Proof. Any feasible payoff profile, p, in an infinite repeated game can be thought of as the expected payoff profile from a distribution, d, over feasible payoff profiles in the stage game. We will now show the existence of a polynomial-length sequence, S, with expected payoff profile approximating that of d. We generate S by taking L random draws from the distribution d. Let p_i be the expected payoff to player i from playing d and $\hat{p}_i$ be the actual payoff achieved after L draws from d. For any given $\epsilon' > 0$ we can use Hoeffding's inequality to

bound the probability that the difference in payoffs exceeds ϵ':

$$\begin{aligned}
\forall i \in X, Prob(|\hat{p}_i - p_i| \geq \epsilon') &\leq 2\exp(-2(\epsilon')^2 L) \\
Prob(\exists i \in X : |\hat{p}_i - p_i| \geq \epsilon') &= Prob(\cup_{i \in X} |\hat{p}_i - p_i| \geq \epsilon') \\
&\leq \sum_{i \in X} Prob(|\hat{p}_i - p_i| \geq \epsilon') \\
&\leq 2n\exp(-2(\epsilon')^2 L) \\
Prob(\forall i \in X, |\hat{p}_i - p_i| \leq \epsilon') &\geq 1 - 2n\exp(-2(\epsilon')^2 L) \\
Prob(\forall i \in X, |\hat{p}_i - p_i| \leq \epsilon') &\geq 1 - \delta'
\end{aligned}$$

when L is at least $\frac{1}{2(\epsilon')^2}(\ln \frac{2n}{\delta'})$.

If we let $\epsilon' = \frac{\epsilon_4}{2}$ and $\delta' = \frac{\delta_4}{2}$, the expected payoff of S will be within $\frac{\epsilon_4}{4}$ of p with probability at least $1 - \frac{\delta_4}{2}$. Using Lemma 3, with $\epsilon_3 = \frac{\epsilon_4}{2M}$ and $\delta_3 = \frac{\delta_4}{2L}$, we can show that the actual payoff achieved for each step in S after H repetitions is within $\frac{\epsilon_4}{2}$ of the expected payoff with probability at least $\frac{\delta_4}{2L}$. Thus the payoffs achieved from using S for H times will be at most ϵ_4 from p with probability at least $1 - \frac{\delta_4}{2} - L * \delta_3 = 1 - \delta_4$.

Substituting in the new values for $\epsilon', \delta', \epsilon_3$, and δ_3 we have:

$$L = \frac{2}{\epsilon_4^2} \ln \frac{4n}{\delta_4}, H \geq \frac{2M^2}{\epsilon_4^2} \ln \frac{4ML}{\delta_4}$$

Thus L and H are polynomial in $M, n, \frac{1}{\epsilon_4}$, and $\frac{1}{\delta_4}$. □

Lemma 5. *Let K be the number of times the players change their distributions of actions. Within the "Coordinate" block, for any given $\delta_5 > 0$, $T > 0$, and $\gamma \leq 1 - (1 - \delta_5)^{\frac{1}{KnT}}$, if each cooperating player attempts to change its distribution of actions on each round with probability γ, the probability that no two players will make the attempt within T rounds of each other is at least $1 - \delta_5$.*

Proof.

$$Prob(\text{No agent changes within T turns after another}) \geq (1 - \gamma)^{nT}$$

$$Prob(\text{No agent changes within T turns after K changes}) \geq (1 - \gamma)^{KnT}$$

Solving for γ given $(1 - \gamma)^{KnT} \geq 1 - \delta_5$ we obtain: $\gamma \leq 1 - (1 - \delta_5)^{\frac{1}{KnT}}$. □

Lemma 6. *Within the "Coordinate" block, for any given $\delta_6 > 0$, and the same δ_5, T, γ, and K as in Lemma 5, there exists a τ' polynomial in $n, K, T, \frac{1}{\delta_5}$ and $\frac{1}{\delta_6}$ such that if each cooperating player tries to change the distribution of its actions with probability γ, the probability for all players to do so at least once after τ' rounds is at least $1 - \delta_6$.*

Proof.

$$\begin{aligned} Prob(\text{One agent has not tried by } \tau') &= (1-\gamma)^{\tau'} \\ Prob(\exists i \in [1,n] : \text{Agent i has not tried by } \tau') &\leq n(1-\gamma)^{\tau'} \\ &\leq n(1-(1-(1-\delta_5)^{\frac{1}{KnT}}))^{\tau'} \\ &\leq n(1-\delta_5)^{\frac{\tau'}{KnT}} \end{aligned}$$

Setting $\tau' = KnT\log_{1-\delta_5}(\frac{\delta_6}{n})$ we have,

$$\begin{aligned} Prob(\text{All agents have tried by } \tau') &\geq 1 - n(1-\delta_5)^{\frac{\tau'}{KnT}} \\ &\geq 1 - n(1-\delta_5)^{\log_{1-\delta_5}(\frac{\delta_6}{n})} \\ &\geq 1 - \delta_6 \end{aligned}$$

Moreover,

$$\begin{aligned} \tau' &= KnT\log_{1-\delta_5}(\frac{\delta_6}{n}) = KnT\frac{\log(\frac{\delta_6}{n})}{\log(1-\delta_5)} \\ &= KnT\frac{\log(\frac{n}{\delta_6})}{\log(\frac{1}{1-\delta_5})} = KnT\frac{\log(n)+\log(\frac{1}{\delta_6})}{\log(1+\frac{\delta_5}{1-\delta_5})} \end{aligned}$$

We can assume w.l.o.g. that $\delta_5 < \frac{1}{2}$, and that therefore $\frac{\delta_5}{1-\delta_5} < 1$. Performing a power series expansion, with $|x| < 1$, we have:

$$\begin{aligned} \log(1+x) &= x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + ... \\ &\geq x - \frac{x^2}{2} \\ \log(1+\frac{\delta_5}{1-\delta_5}) &\geq \frac{\delta_5}{1-\delta_5} - \frac{\delta_5^2}{2(1-\delta_5)^2} \\ \tau' &\leq KnT(\log(n)+\log(\frac{1}{\delta_6}))\frac{1}{\frac{\delta_5}{1-\delta_5} - \frac{\delta_5^2}{2(1-\delta_5)^2}} \\ \tau' &\leq KnT(\log(n)+\log(\frac{1}{\delta_6}))\frac{2(1-\delta_5)^2}{2\delta_5 - 2\delta_5^2 - \delta_5^2} \\ \tau' &\leq 2KnT(\log(n)+\log(\frac{1}{\delta_6}))\frac{(1-\frac{1}{\delta_5})^2}{\frac{2}{\delta_5}-3} \\ \tau' &\leq 2KnT(\log(n)+\log(\frac{1}{\delta_6}))(1-\frac{1}{\delta_5})^2 \end{aligned}$$

Thus τ' is polynomial in $n, K, T, \frac{1}{\delta_5}$ and $\frac{1}{\delta_6}$. □

With the additional lemmas above, we can now prove Lemma 2.

Proof of Lemma 2. Let's consider first the case in which all agents are partitioned correctly into three sets: target agents, coop agents, and non-coop agents. All coop agents have the same partitions since they all have the same perfect observation of the actions of the agents. From Lemma 6, we can find τ' such that all coop agents will attempt to switch group at least once with probability at least $1-\delta_6$ after τ' rounds. Every time an agent attempts to switch group, it will either join another group of the same or larger size (thereby increasing the size of that group) or remain in the current group if all the others are smaller.

When an agent joins a new group, it has to recalculate the optimal solution to the optimization problem shown in equation 3 in Section 4 given the observed distributions the actions for the other agents in the group. If we choose ϵ_4 in Lemma 4 to be $\frac{\epsilon_2}{2n}$, then $H * L$ periods later, after going through the sequence S for H times, the actual payoff each agent received can only be at most $\frac{\epsilon_2}{2n}$ away from the targeted Pareto-efficient enforceable outcome. This lets us derive two things. First, if a player is thought to be within the largest group at the time another player switches to that group, then setting $T = H * L$ and using Lemma 5 we can show that T periods later no other agents have switched and the payoff will have changed by less than $\frac{\epsilon_2}{n}$ so the player will still be observed to be within the largest group (since the allowable error for determining the largest group is $\frac{\epsilon_2}{n}$ times the number of players in the group). Since once a player switches groups it will remain in the largest group, no player will change groups more than once, and thus the value of K in Lemma 6 will be bounded by n. Second, since an error of at most $\frac{\epsilon_2}{2n}$ is introduced each time an agent switches groups, once all agents have switched the total error from the optimal Pareto-efficient enforceable outcome is less than $\frac{\epsilon_2}{2}$. Finally we need to show that the observed payoffs of all coop agents, X, are within $\frac{\epsilon_2}{2}$ of the expected payoffs for all periods after τ_3.

$$\forall i \in X, Prob(\exists t > \tau_3 : |\hat{V}(i) - V(i)| \geq \frac{\epsilon_2}{2}) \leq \sum_{t=\tau_3}^{\infty} 2\exp(-2(\frac{\epsilon_2}{2})^2 t)$$

$$\forall i \in X, Prob(\exists t > \tau_3 : |\hat{V}(i) - V(i)| \geq \frac{\epsilon_2}{2}) \leq 2\exp(-\frac{\epsilon_2^2}{2}) \sum_{t=\tau_3}^{\infty} (\frac{1}{e})^t$$

$$\forall i \in X, Prob(\exists t > \tau_3 : |\hat{V}(i) - V(i)| \geq \frac{\epsilon_2}{2}) \leq 2\exp(-\frac{\epsilon_2^2}{2}\tau_3)\frac{e}{e-1}$$

$$Prob(\exists i \in X, t > \tau_3 : |\hat{V}(i) - V(i)| \geq \frac{\epsilon_2}{2}) \leq 4n\exp(-\frac{\epsilon_2^2}{2}\tau_3)$$

$$Prob(\forall i \in X, t > \tau_3 : |\hat{V}(i) - V(i)| \leq \frac{\epsilon}{2}) \leq 1 - 4n\exp(-\frac{\epsilon_2^2}{2}\tau_3)$$

$$Prob(\forall i \in X, t > \tau_3 : |\hat{V}(i) - V(i)| \leq \frac{\epsilon}{2}) \geq 1 - \delta'$$

In the equation above τ_3 has been set to $\frac{2}{\epsilon_2^2}\ln(\frac{2n}{\delta'})$.

We have now shown that the agents will converge for all time when the

conditions in Lemmas 4, 5, and 6 hold for all coop agents. Therefore:

$$
\begin{aligned}
&Prob(\text{All coop-agents have converged by } \tau') \geq \\
&Prob((\text{No two agents changed distribution within T}) \ \& \\
&\qquad (\text{All agents attempted to switch groups}) \ \& \\
&\qquad (\forall \text{ players } i : \text{when } i \text{ switches groups, all other agents} \\
&\qquad \text{observed play is within } \frac{\epsilon_2}{2n} \text{ of their actual distributions}) \\
&\qquad (\text{All payoffs stay within } \frac{\epsilon_2}{2} \text{ of expected payoffs for all time t past } \tau_3)) \\
&\geq 1 - \delta_5 - \delta_6 - n * \delta_4 - \delta'
\end{aligned}
$$

We can assign $\delta_4 = \frac{\delta_2}{4n}, \delta_5 = \frac{\delta_2}{4}, \delta_6 = \frac{\delta_2}{4}$, and $\delta' = \frac{\delta_2}{4}$ so that the above hold with probability at least $1 - \delta_2$. Thus τ' is the value for τ_2 that we are looking for and τ' is polynomial in $M, n, \frac{1}{\epsilon_2}$, and $\frac{1}{\delta_2}$ since we know from Lemma 6 that τ' is polynomial in $n, K, T, \frac{1}{\delta_5}, and \frac{1}{\delta_6}$, and we know from Lemma 4 that $T = L * H$ is polynomial in $M, n, \frac{1}{\epsilon_4}, and \frac{1}{\delta_4}$.

Now let us consider the case in which there are non-cooperating agents. In order to alter the payoffs they will need to pretend to be either coop or stationary agents. If they conform to the coordination process in PCM(S), then the payoff constraints will be satisfied and the lemma will hold regardless. PCM(S) checks to make sure all agents it thinks are cooperating followed the switching policy by switched at most once and only to the largest group. So any non-cooperating agents will need to make it appear that all agents have coordinated on a single group in order to avoid giving themselves away. They can still change the other agent's payoffs by either altering their distribution later or correlating in a way that influences the payoffs while leaving each individual distribution the same. However, whenever the payoffs vary by more than ϵ from the expected values, PCM(S) tries recalculating payoffs using the target distributions instead of the actual plays. Clearly if a group, Y, containing all the non-cooperating agents is checked the payoffs must deviate by more than $\frac{\epsilon|Y|}{n}$, since for τ_3 greater than $H * L$, we know that the cooperating and stationary agents contribute no more than $\frac{\epsilon|Y|}{n}$ error each. Similarly we don't need to worry about finding a group with both non-cooperating agents and a coop or stationary one. If the group had a deviation of at least $\frac{\epsilon|Y|}{n}$, the group without the coop or stationary agent must have had an error of at least $\frac{\epsilon|Y|-1}{n}$ at would have been detected first. Therefore whenever the payoff is more than ϵ below the target profile at least one non-cooperating agent will be found and no coop or stationary agents will be misclassified. □

C Proof Outline for PCM(A)

The proof of theorem 3 follows from the proof framework of theorem 1 with only a few modifications. First, we will need to observe the opponents for a longer

period of time. We now need to show that after a period of H, the observed distribution for a conditional strategy player is within ϵ' of the true distribution for all actions and all possible histories. To show this, we can use the proof for Lemma 3, but with the number of different probability distributions set to $M * M^{nk}$ instead of M, giving us:

$$H \geq \frac{1}{2\epsilon'^2} \log \frac{2M^{nk+1}}{\delta_1}$$
$$\epsilon' = \frac{\epsilon}{4nM^{nk+1}}$$
$$H \geq \frac{8n^2 M^{2nk+2}}{\epsilon^2} \log \frac{2M^{nk+1}}{\delta_1}$$

An additional complexity arises if the opponents play can depend on their own past actions. In this case we don't have the ability to take samples at will for the different histories, but may instead need to follow a chain of different histories in order to manipulate the opponent's play so that we can observe a particular outcome. In the worst case the length of this chain of histories may approach the size of the full set of unique histories, M^{nk} and each desired transition may occur with a probability as small as λ, where λ is the minimum non-zero probability any opponent in our target class assigns to any action in some history. Therefore, the average amount of exploration to get even one observation of a particular history could require time proportional to $(\frac{1}{\lambda})^{M^{nk}}$. We can think of this term as the mixing time of agents' exploration policy in the stochastic game defined by letting each k-length history be a state with the opponents' conditional strategies as their policies.

Unfortunately this factor for the time to achieve a desired history can also affect the maximum length, L, of the sequence we need to approximate any possible PE solution of the repeated game. To see this, consider an environment in which the players can only achieve a Pareto-efficient enforceable outcome by mixing over the outcomes of playing a particular strategy for two different starting histories. In order to approximate the mix, the player may need to spend an exponential amount of time moving between the two histories. In particular we can replace the proof of lemma 4 with a symmetric proof approximating the feasible payoff profile by a distribution over conditional strategies with bounded memory k. However, since the short-term payoff of a conditional strategy is dependent on the starting history, in order to get a guarantee that the empirical payoffs are near the expected payoff, we will need to play each strategy for a time proportional to its mixing time in the stochastic game formed by the play of the opponents in the target class. Therefore in the worst-case the T in the proof of lemma 4 will also include a factor of $(\frac{1}{\lambda})^{M^{nk}}$.

Finally, we need to address the issue that unconstrained opponents can potentially prevent the agents from observing particular histories. However, the self-play agents can safely assume the most advantageous member from their payoff point of view out of the set of target opponents consistent with the other

observations. In order to prove this assumption wrong and negatively affect the payoffs, the unconstrained agents would need to allow this history to be played with positive probability, thereby allowing it be observed. Once it has been observed sufficiently often, the agents can replan. This can happen at most once for each history so the agents would need to coordinate at most M^{nk} times.

References

[1] Robert Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.

[2] Bikramjit Banerjee and Jing Peng. Efficient no-regret multiagent learning. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, pages 41–46, 2005.

[3] Michael Bowling. Convergence and no-regret in multiagent learning. In *Advances in Neural Information Processing Systems 17*, pages 209–216, 2005.

[4] Michael Bowling and Manuela Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.

[5] Michael Bowling and Manuela M. Veloso. Existence of multiagent equilibria with limited agents. Technical report CMU-CS-02-104, Computer Science Department, Carnegie Mellon University, 2002.

[6] Ronen Brafman and Moshe Tennenholtz. Efficient learning equilibrium. *Artificial Intelligence*, 159(1-2):27–47, 2004.

[7] George Brown. Iterative solution of games by fictitious play. In *Activity Analysis of Production and Allocation*, pages 374–376. John Wiley and Sons, New York, 1951.

[8] Yu-Han Chang and Leslie Kaelbling. Hedged learning: Regret-minimization with learning experts. In *Proceedings of the 22nd International Machine Learning Conference*, 2005.

[9] Yu-Han Chang and Leslie Pack Kaelbling. Playing is believing: The role of beliefs in multi-agent learning. In *Advances in Neural Information Processing Systems 14*, pages 1483–1490, 2002.

[10] Lonnie Chrisman. Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 183–188, 1992.

[11] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746–752, 1998.

[12] Vincent Conitzer and Tuomas Sandholm. Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 83–90, 2003.

[13] Daniela Pucci de Farias and Nimrod Megiddo. How to combine expert (or novice) advice when actions impact the environment. In *Advances in Neural Information Processing Systems 16*, 2004.

[14] Dean Foster and Rakesh Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 29:7–36, 1999.

[15] Drew Fudenberg and David Levine. Universal consistency and cautious fictitious play. *Journal of Economic Dynamics and Control*, 19:1065–1089, 1995.

[16] Drew Fudenberg and David K. Levine. *The Theory of Learning in Games*. MIT Press, Cambridge, MA, 1998.

[17] J. F. Hannan. Approximation to Bayes risk in repeated plays. *Contributions to the Theory of Games*, 3:97–139, 1957.

[18] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68:1127–1150, 2000.

[19] Amir Jafari, Amy Greenwald, David Gondek, and Gunes Ercal. On no-regret learning, fictitious play, and Nash equilibrium. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 226–223, 2001.

[20] Ehud Kalai and Ehud Lehrer. Rational learning leads to Nash equilibrium. *Econometrica*, 61(5):1019–1045, 1993.

[21] Michael Littman and Peter Stone. Implicit negotiation in repeated games. In *Proceedings of The Eighth International Workshop on Agent Theories, Architectures, and Languages*, pages 393–404, 2001.

[22] Shie Mannor and Nahum Shimkin. Adaptive strategies and regret minimization in arbitrarily varying markov environments. In *Proceedings of the Fourtheenth Annual Conference on Computational Learning Theory*, pages 128–142, 2001.

[23] Peter McCracken and Michael Bowling. Safe strategies for agent modelling in games. In *AAAI 2004 Symposium on Artificial Multi-Agent Learning [FS-04-02]*. AAAI Press, 2004.

[24] Abraham Neyman. Bounded complexity justifies cooperation in finitely repeated prisoner's dilemma. *Economic Letters*, pages 227–229, 1985.

[25] Daniel Nikovski and Illah Nourbakhsh. Learning probabilistic models for decision-theoretic navigation of mobile robots. In *Proceedings of the International Conference on Machine Learning*, pages 266–274, 2000.

[26] Eugene Nudelman, Jenn Wortman, Kevin Leyton-Brown, and Yoav Shoham. Run the GAMUT: A comprehensive approach to evaluating game-theorectic algorithms. In *Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, 2004.

[27] Christos H. Papadimitriou and Mihalis Yannakakis. On complexity as bounded rationality. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pages 726–733, 1994.

[28] Rob Powers and Yoav Shoham. New criteria and a new algorithm for learning in multi-agent systems. In *Advances in Neural Information Processing Systems 17*. MIT Press, 2005.

[29] S. Sen and G. Weiss. Learning in multiagent systems. In *Multiagent systems: A modern introduction to distributed artificial intelligence*, pages 259–298. MIT Press, 1998.

[30] Yoav Shoham, Rob Powers, and Trond Grenager. On the agenda(s) of research on multi-agent learning. In *AAAI 2004 Symposium on Artificial Multi-Agent Learning [FS-04-02]*. AAAI Press, 2004.

[31] S. Singh, M. Kearns, and Y. Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 41–48, 2000.

[32] Peter Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3), 2000.

[33] Chris Watkins and Peter Dayan. Technical note: Q-learning. *Machine Learning*, 8(3/4):279–292, May 1992.

[34] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.