# Improving Convergence Rates in Multiagent Learning Through Experts and Adaptive Consultation*

**Greg Hines** GGDHINES@UWATERLOO.CA
*Cheriton School of Computer Science*
*University of Waterloo*
*Waterloo, Ontario, N2L 3G1*

**Kate Larson** KLARSON@UWATERLOO.CA
*Cheriton School of Computer Science*
*University of Waterloo*
*Waterloo, Ontario, N2L 3G1*

## Abstract

We present a multiagent learning algorithm with guaranteed convergence to Nash equilibria for all games. Our approach is a regret-based learning algorithm which combines a greedy random sampling method with consultation of experts that suggest possible strategy profiles. More importantly, by consulting carefully chosen experts we can greatly improve the convergence rate to Nash equilibria, but in the case where the experts do not return useful advice, we still have guarantees that our algorithm will eventually converge. The goal of our work is to bridge the gap between theoretical and practical learning, and we argue that our approach, FRAME, can serve as a framework for a class of multiagent learning algorithms.

## 1. Introduction

How and what agents should learn in the presence of others is one of the important questions in multiagent systems. The problem has been studied from several different perspectives, and in particular has garnished a lot of interest from both the game-theory community (see, for example, [6, 14, 15, 19]) and the AI community (see, for example, [5, 17, 20, 21, 25, 26]).

One of the challenges in multiagent learning is that there are several different agendas being pursued, ranging from the descriptive agenda which aims to describe how agents learn, to the prescriptive which aims to explain how agents should learn [24]. Nor is it entirely clear what it means for an agent in a multiagent setting to be a successful learner. Success might be defined by having the learning process converge to a Nash equilibrium in self-play, by having agents learn the opponents' strategies, or by ensuring agents are guaranteed payoffs above a certain threshold.

In this paper we investigate the well-founded problem of whether identical agents, who repeatedly play against each other, can learn to play strategies which form a Nash equilibrium in the stage game (see, for example [2, 4, 8]). In particular, we are interested in settings where there are potentially more than two agents, and where agents have potentially more than just two actions to choose from.

*. University of Waterloo Technical Report CS-2007-24.

Our learning procedure, a *Framework for Regret Annealing Methods using Experts* or *FRAME*, is a regret-based learning algorithm which combines a greedy random sampling method with consultation of *experts*, that return strategy profiles. More importantly, by consulting carefully chosen experts we can greatly improve the convergence rate to Nash equilibria in self-play, but in the case where the experts do not return useful advice, then we still have guarantees that our algorithm will eventually converge. Although the expert paradigm has been used in machine learning before (for example, [10]), to the best of our knowledge our work is the first that uses them in multiagent learning algorithms that guarantee convergence to Nash equilibrium in self-play.

We further extend our algorithm to allow agents to rely on multiple experts, and to adapt the likelihood of consulting any particular expert based on the quality of the advice it is providing for the current game. This allows agents to learn effectively in new situations as they can adapt their learning styles to the game at hand.

We organize the rest of the paper as follows. In the next section (Section 2) we describe the learning environment we are interested in. In Section 3 we present our learning algorithm, FRAME. We prove results about its behavior and convergence properties, as well as report on our experimental findings. In the next section (Section 4) we extend FRAME so that it can adapt its learning strategies as it discovers more about the game being played. We introduce our adaptive approach, adaptive-FRAME, as well as a new regret-based experts algorithm, LERRM. We conclude the paper and discuss future research directions in Section 5.

## 2. Background

In this section we describe the learning setting studied in this paper. Repeated games are based on stage games, which are introduced in Section 2.1. Section 2.2 describes the idea of regret which is used throughout this work. In Section 2.3 we introduce repeated games. We also discuss learning in repeated games and some properties with which we measure potential solutions.

### 2.1 Stage Games

A $n$-player *stage game* is a tuple $G = \langle N, A = A_1 \times \ldots \times A_n, u_1, \ldots, u_n \rangle$ where $N = \{1, \ldots n\}$ is the set of agents in the game, $A_i$ is the set of actions available for agent $i$ to play, $A$ is the set of possible joint actions and $u_i : A \rightarrow \mathbb{R}$ is the utility function for agent $i$. We denote the size $A_i$ by $m_i$. We let $a_i$ denote a specific action taken by agent $i$. As is standard in the literature, we will use $A_{-i}$ to denote the joint actions of all agents but agent $i$, i.e. $A_{-i} = \{A_1 \times \ldots \times A_{i-1} \times A_{i+1} \times \ldots \times A_n\}$, and $a_{-i} \in A_{-i}$, $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$ to be a particular joint action. Agents are all *self-interested*, in that their only concern is maximizing their own utility.

Figure 2.1 shows Battle of the Sexes, a classic example of a stage game. The basic idea behind Battle of the Sexes is that agents would like to coordinate on an action but cannot agree on which action to coordinate on.

When a stage game is given in matrix form, the game is said to be in *normal form*. Agent 1 and agent 2 will each pick an action simultaneously. Agent 1's action can be thought of as picking a row in the matrix. Likewise, agent 2's action can be thought of as picking a

column in the matrix. The cell at the intersection of the row and column gives the utility for both agents. The first value in that cell gives the utility for agent 1 and the second value gives the utility for agent 2. All of the concepts are generalized in the obvious manner for games with more than 2 agents.

| | | Agent 2 | |
|---|---|---|---|
| | | $a_{2,1}$ | $a_{2,2}$ |
| Agent 1 | $a_{1,1}$ | $1, 0.5$ | $0, 0$ |
| | $a_{1,2}$ | $0, 0$ | $0.5, 1$ |

Figure 1: An example of a stage game: Battle of the Sexes

We assume that agents play *strategies*.

**Definition 1** *A strategy, $\sigma_i$, for agent $i$, is a probability distribution over its action set $A_i$, stating with what probability the agent will play each action. A pure strategy is one in which the agent plays one action with probability equal to one. All other strategies are called mixed strategies. The set of all possible strategies for agent $i$ is $\Sigma_i$. The profile $\sigma = (\sigma_1, \ldots, \sigma_n)$ is a joint strategy among all agents and $\Sigma = \times_{i=1}^{n} \Sigma_i$ is the set of all possible joint strategies.*

*We define $\Sigma_{-i} = \Sigma_1 \times \ldots \times \Sigma_{i-1} \times \Sigma_{i+1} \times \ldots \times \Sigma_n$ and $\sigma_{-i}$ to be an element of $\Sigma_{-i}$.*

By abuse of notation, we define

$$u_i(\sigma_i, \sigma_{-i}) = \sum_{(a_i, a_{-i}) \in A} u_i(a_i, a_{-i}) \sigma_i(a_i) \sigma_{-i}(a_{-i}). \tag{1}$$

In words, agent $i$'s expected utility with respect to its strategy $\sigma_i$ and its opponents' joint strategy $\sigma_{-i}$ is the sum of the utility over all possible joint actions of the utility for agent $i$ of a joint action multiplied by the probability of that joint action happening due to the strategies $\sigma_i$ and $\sigma_{-i}$. We also define an agent's utility with respect to playing a specific again as

$$u_i(a_i, \sigma_{-i}) = \sum_{a_{-i} \in A_{-i}} u_i(a_i, a_{-i}) \sigma_{-i}(a_{-i}). \tag{2}$$

In words, agent $i$'s expected utility with respect to playing action $a_i$ given its opponents' joint strategy $\sigma_{-i}$ is the sum of the utility over all possible joint actions that include $a_i$ of the utility for agent $i$ of such a joint action multiplied by the probability of the that joint action happening due to the joint strategy $\sigma_{-i}$.

By definition, the agents' strategies are a *Nash equilibrium* if no agent is willing to change its strategy, given that no other agents change theirs [22].

**Definition 2** *A strategy profile $\sigma^* = (\sigma_1^*, \ldots, \sigma_n^*)$ is a Nash equilibrium if for every agent $i$*

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(\sigma_i', \sigma_{-i}^*), \ \forall \sigma_i' \neq \sigma_i^*. \tag{3}$$

*A strategy profile $\sigma^*$ is an $\epsilon$-Nash equilibrium if for every agent $i$*

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(\sigma_i', \sigma_{-i}^*) - \epsilon, \ \forall \sigma_i' \neq \sigma_i^*. \tag{4}$$

We denote the set of all Nash equilibria for some game $G$ by $\mathcal{N}^G$, and the set of all $\epsilon$-Nash equilibria by $\mathcal{N}_\epsilon^G$. When it is clear from the context, we will drop the game index $G$ and use $\mathcal{N}$ and $\mathcal{N}_\epsilon$ respectively. Finally, let $\mathcal{N}_\epsilon^c$ be the set of all joint strategies that are not $\epsilon$-Nash equilibria.

### 2.2 Regret

Another notion that agents may use to evaluate their choice of strategy is that of *regret*.

**Definition 3** *Given a joint strategy $\sigma$, agent $i$'s regret is*

$$r_i(\sigma) = \max_{\sigma_i' \in \Sigma_i} \left[ u_i(\sigma_i', \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}) \right]. \tag{5}$$

Given $\sigma$, the regret of a game is the maximum regret among all agents, i.e. $r(\sigma) = \max_{i \in N}(r_i(\sigma))$. When $\sigma$ is obvious, we shall just use $r$. This may be seen as a measure of how much agent $i$ is hurt by playing strategy $\sigma_i$ as opposed to any other strategy.

Two other types of regret are external and internal regret [18, 11]; external regret measures regret against all pure strategies while internal regret measures regret for having played action $a$ instead of action $b$ for all $a \neq b$.

We can use regret to give another way of defining a Nash equilibrium. If all agents have no-regret about the strategies they are playing, i.e. $r_i = 0, \forall i \in N$, then the strategy profile is a Nash equilibrium. Similarly, if $r_i \leq \epsilon$ for all $i$, then we have an $\epsilon$-Nash equilibrium.

Another important notion related to regret and Nash equilibria is *best response*.

**Definition 4** *The best response for agent $i$, if all other agents are playing $\sigma_{-i}$, is*

$$BR_i(\sigma_{-i}) = \{\sigma_i \in \Sigma_i | r_i(\sigma_i, \sigma_{-i}) = 0\}. \tag{6}$$

.

We can also define the *$\epsilon$-best response* in a similar fashion.

**Definition 5** *The $\epsilon$-best response for agent $i$ is*

$$BR_i^\epsilon(\sigma_{-i}) = \{\sigma_i \in \Sigma_i | r_i(\sigma_i, \sigma_{-i}) \leq \epsilon\}. \tag{7}$$

For Battle of the Sexes, if agent 1's strategy is $(1, 0)$, then agent 2's best response is the set $\{(1, 0)\}$ since any strategy in this set maximizes agent 2's utility with respect to agent 1's strategy. Assuming the same strategy for agent 1, an $\epsilon$-best response for agent 2 would any strategy $\{(1 - \omega, \omega)\}$ for any $\omega \leq \epsilon$. Any such strategy would give agent 2 a regret of at most $\epsilon$.

### 2.3 Repeated Games

A repeated game is one where agents play the same stage repeatedly for an infinite number of times. The idea is that for each iteration of the repeated game, agents are able to try and learn from previous iterations and improve their strategies. A *learning algorithm* is any algorithm an agent uses to help improve its strategy.

We consider repeated games where all agents are using identical learning algorithms (i.e. *self-play*). Furthermore, we assume that although agents know they are playing a repeated game, they only care about the utility from the current round. Under these conditions, we feel that a reasonable goal for a learning algorithm is to achieve convergence to the set of Nash equilibria. This means that for any given $\epsilon > 0$, there exists some time $T$ such that all joint strategies starting at time $T$ are $\epsilon$-Nash equilibria. However, when a game has multiple Nash equilibria, convergence to the set of Nash equilibria does not imply convergence to a single specific Nash equilibrium. In fact, it would be possible for the joint strategy to "jump" infinitely often between different Nash equilibria. This could result in fluxuations in agents' utilities. Therefore, whenever possible, we would like to be able to achieve the even stronger result of convergence to a specific Nash equilibrium.

It should be noted that we are interested in having the *period-by-period behaviours* converge to the set of Nash equilibria. This is opposed to just having the *cumulative empirical frequency of play* converge. Too see why, consider the game in Figure 2. This game has 3 Nash equilibria, $\{(1,0),(0,1)\}$, $\{(0,1),(0,1)\}$ and $\{(\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{1}{2})\}$.

| | | Agent 2 | |
|---|---|---|---|
| | | R | L |
| Agent 1 | T | 1,1 | 0,0 |
| | B | 0,0 | 1,1 |

Figure 2: A simple game

Suppose Agent 1 and Agent 2 play the following repeated sequence of actions,

$$(T,L),(B,R),\ldots \tag{8}$$

Each turn both agents get 0 utility. However, the cumulative empirical frequency of play, or the average number of times each agent played each action, corresponds to the Nash equilibrium $\{(\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{1}{2})\}$. Thus, we can say that the cumulative empirical frequency of play converges to a Nash equilibrium. The problem with this definition is that just because the cumulative empirical frequency of play converges, does not mean that agents get the utility from the corresponding Nash equilibrium.

## 3. FRAME

In this section we introduce our algorithm, FRAME. In Section 3.1, we present FRAME and how it builds upon ALERT. In Section 3.2, we discuss and prove FRAME's properties.

### 3.1 Introduction

Foster and Young have recently proposed a set of algorithms that are able to guarantee convergence to the set of Nash equilibria for all two agent games of self-play [12]. Germano and Lugosi generalized Foster and Young's work to include almost all games[1] [16].

1. Germano and Lugosi's algorithms work for all *generic games* [16]. The definition of generic games is left to Germano and Lugosi's paper. However, this class of games includes almost all games and the exceptions tend to be degenerate games.

Germano and Lugosi's first algorithm, *Experimental Regret Testing* (ERT) does not guarantee convergence. Instead, after a certain period of time ERT can guarantee an $\epsilon$-Nash equilibrium with a probability $1-\epsilon$. (Of course, once the agents find an $\epsilon$-Nash equilibrium, there is a chance of them leaving it.) ERT corresponds to Regret Testing when $\Lambda = 0$ and where agents still occasionally experiment with new strategies even when they have low regret. Although ERT has agents playing actions, if we let $T = \infty$, then we can convert ERT so that agents play strategies.

The basic idea of ERT is that if at some point there are $J < N$ agents who have regret less than $\rho$, there is a positive probability of there being $J-1$ agents having regret less than $\rho$ at the next turn. Since this process repeats indefinitely, at some point in the future all agents will have regret greater than $\rho$. At this point they will choose a new joint strategy at random from $\Sigma$ and there is a positive probability of the new joint strategy being an $\epsilon$-Nash equilibrium [16]. Once the agents find an $\epsilon$-Nash equilibrium, the chances of leaving that joint strategy are low.

The main result for ERT is:

**Theorem 1** *Let $G$ be a generic $N$-agent normal form game. There exists a positive number $\epsilon_0$ such that for all $\epsilon < \epsilon_0$ the following holds: there exists positive constants $c_1, \ldots, c_4$ such that if ERT is used by all agents with parameters*

$$\rho \in (\epsilon, \epsilon + \epsilon^{c_1}), \tag{9}$$

$$\zeta \leq c_2 \epsilon^{c_3}, \tag{10}$$

$$\mathcal{T} \geq -\frac{1}{2(\rho-\epsilon)^2} \log(c_4 \epsilon_3^c), \tag{11}$$

$$\tag{12}$$

*then for all $M \geq \log(\epsilon/2)/\log(1-\zeta^N)$,*

$$P_M(\mathcal{N}_\epsilon^c) = P(\sigma^{M\mathcal{T}} \notin \mathcal{N}_\epsilon) \leq \epsilon. \tag{13}$$

In words this means that at the end of $M \cdot \mathcal{T}$ iterations, the probability of not being at an $\epsilon$-Nash equilibrium is at most $\epsilon$.

Germano and Lugosi were able to take their initial algorithm and convert it into one able to achieve convergence. Their new algorithm is called *Annealed Localized Experimental Regret Testing* (ALERT).

The basic idea of ALERT is to slowly anneal the value of $\epsilon$ and repeat ERT for each value of $\epsilon$. Any sequence of $\epsilon_l$ for $l = 1, 2, \ldots$ such that $\sum_{l=1}^{\infty} \epsilon_l < \infty$ will work; however, Germano and Lugosi choose to use $e_l = 2^{-l}$. The set of all periods of play for a particular $\epsilon_l$ is called a *regime*. The set of all regimes is indexed by $l$. The number of periods in the $l^{th}$ regime is given by

$$M_l \equiv 2 \left\lceil \frac{\log \frac{2}{\epsilon_l}}{\log \frac{1}{1-\zeta_l}} \right\rceil. \tag{14}$$

$(T, \rho, \zeta)$ must be generalized to depend on $l$, and so the following values are used

$$\mathcal{T}_l = \left\lceil -\frac{1}{2\epsilon_l^{2l}} \log(\epsilon_l^l) \right\rceil, \tag{15}$$

$$\rho_l = \epsilon_l + \epsilon_l^l, \tag{16}$$

$$\zeta_l = \epsilon_l^l. \tag{17}$$

$\sigma_i^{[l]}$ is $\sigma_i$ at the beginning of the $l^{th}$ regime. $D_\infty^i(\sigma_i, \epsilon)$ is the $L_\infty$-ball of radius $\epsilon$ centered around $\sigma_i$.[2]

The main result for ALERT is the following theorem.

**Theorem 2** *Let $G$ be a generic $N$-agent game and $\{\epsilon_l\}_{l=1}^\infty$ be defined $e_l = 2^{-l}$. If each agent plays according to ALERT and using the parameters in Equations 14 through 17, then*

$$\lim_{r \to \infty} \sigma^t \in \mathcal{N} \tag{18}$$

*almost surely.*

In words this means that the limit of the sequence of the joint strategies is a Nash equilibrium.

Note that as presented, ALERT is an uncoupled algorithm. However, Germano and Lugosi also present a variant that is radically uncoupled.

The one drawback of ALERT is its rate of convergence. Since ALERT is uncoupled, the rate of convergence is independent of the game. This can be seen in the game parameters, where $\mathcal{T}_l$ and $M_l$ are both dependent on only $\epsilon_l$ and $\zeta_l$. The downside is that for just about any game of interest ALERT's rate of convergence is impractical.

Specifically, ALERT's slow rate of convergence is caused by two main factors.

1. Since ERT and ALERT were designed as uncoupled algorithms, agents using ERT cannot know with certainty when they have reached an $\epsilon$-Nash equilibrium. Instead agents are only able to bound the probability of not being at an $\epsilon$-Nash equilibrium. Obtaining the necessary bound can require an impractical amount of time. This is exasperated by ALERT calling ERT repeatedly and needing a non-trivial decrease in the size of $\epsilon$ with each call [16].

2. ERT and ALERT pick new strategies uniformly at random. Using this brute force method to find an $\epsilon$-Nash equilibrium is a major reason why ALERT takes so long to converge.

Our algorithm, a *Framework for Regret Annealing Methods using Experts* or *FRAME*, is inspired by ALERT but explicitly addresses these two issues while still providing the theoretical guarantees of ALERT.

To address the first issue, we start by making a number of assumptions.

We first assume that at any given point in time, agents' strategies are fully observable for all past time periods. This is a common approach taken by many algorithms [2, 5].

2. A $L_\infty$-ball can be thought of as a hyper-cube.

This assumption has no effect on the correctness of our algorithm, instead it removes the need for experimentally determining regret which can be very costly time wise. ALERT could get around this assumption by having agents fix their strategy for a certain period of time. At the end of this period, through simple observation of the actions played by each agent, agents will know all of their opponents' strategies. Thus no privacy is lost by this assumption.

We next assume that the maximum regret of all agents is publicly known. Again, this has no effect on the correctness of our algorithm but removes one of the major performance constraints in ALERT. Although this is not as common an assumption, there are other algorithms that make the stronger assumption that agents can determine a potential equilibrium in advance [3, 8]. Determining a potential equilibrium in advance requires agents to share their utility functions, which are more private than strategies since utility functions cannot necessarily be determined experimentally. As well, determining an equilibrium in advance is a computationally complex problem [7, 9].

Finally we assume that while agents are self-interested, they are willing to cooperate to a certain degree. Specifically, we assume that agents will agree to move to a new joint strategy only if it decreases the maximum regret over all the agents. This means that while an agent's regret is guaranteed to converge to zero, at any specific point in time an agent might have its regret increase (or not decrease as much as it would like). This is a strong assumption but it is not without precedence. While Bowling's algorithm GIGA-WoLF guarantees asymptotically no-regret, at any specific point in time it is possible for an agent to experience some regret [4]. Likewise, Hart and Mas-Colell's algorithm also only achieves asymptotically no-regret since agents are still able to experience regret at any specific point in time[19]. Since FRAME actually achieves no-regret (not just asymptotically), this is a stronger result for agents.

Since the maximum regret is publicly known, agents can now know for certain when a better $\epsilon$-Nash equilibrium has been found. This can potentially be much faster than the ERT and ALERT approach (which requires obtaining a probabilistic bound), and also allows us to use a greedy approach when picking a new $\epsilon$-Nash equilibrium. This approach is different enough that our proof does not follow directly from Germano and Lugosi's work [16].

The second problem with ERT and ALERT is that they choose new strategies naively, i.e. uniformly at random. In contrast, FRAME allows an agent, with some probability, to consult an *expert*, which returns a possible new strategy. Any expert will work, even one who makes only useless suggestions. If the expert is able to find new strategies that lead to better $\epsilon$-Nash equilibria, then the agent can take advantage of this to greatly speed up convergence. However, part of the goal of FRAME is that even with useless experts, convergence is still guaranteed.

The FRAME algorithm for agent $i$ is shown in Algorithm 1. We use the following notation in our algorithm: $\mathcal{U}(X)$ denotes a value picked uniformly at random from the set $X$, $e_i(\cdot)$ is the expert and $B(x, d)$ is a bounded search region centered at $x$ with minimum radius $d > 0$.

The FRAME algorithm, with respect to agent $i$, works as follows. At time $t = 0$, agent $i$ chooses a strategy $\sigma_i^0$ uniformly at random from $\Sigma_i$. At any subsequent time $t > 0$, FRAME can consult the provided expert, $(e_i(\cdot))$, to obtain a new strategy. Each agent independently consults $e_i(\cdot)$ with a provided probability of $p_i$. If consulted, the expert returns a possible

strategy $\beta_i^{t+1}$. To provide protection against poor experts, FRAME checks to see if $\beta_i^{t+1}$ is inside the region $B(\sigma_i^t, d(r^t))$.[3] If $\beta_i^{t+1}$ is not, or if the expert was not consulted, $\beta_i^{t+1}$ is chosen uniformly at random from the bounded search region, $B(\sigma_i^t, d(r^t))$. (This may be thought of as consulting the *Naive Expert*, which is an expert that picks strategies uniformly at random.) Agent $i$ then calculates $r(\beta^{t+1})_i$. If $r(\beta^{t+1}) < r(\sigma^t)$, then $\sigma^{t+1} = \beta^{t+1}$, otherwise, $\sigma^{t+1} = \sigma^t$. To avoid the off chance of getting stuck at a locally optimal joint strategy, each agent chooses an alternative strategy $\tau_i^{t+1}$ uniformly at random from $\Sigma_i$. If the regret at $\tau^{t+1}$ is less than half the current regret, then with a given probability $\eta$, the game *resets* to $\tau^{t+1}$. (Any constant fraction less than one will work; one half was chosen for simplicity.) Resetting the joint strategy to $\tau$ just means that $\tau$ becomes the new joint strategy.[4]

This process repeats until the regret is zero.

### 3.2 Theoretical Properties

In this section, we discuss the theoretical properties of FRAME. In particular, we prove that FRAME is guaranteed to converge to the set of Nash equilibria. To show convergence, we show that the limit of the sequence of regret of the agents, all using FRAME, is 0, since 0 regret is the same thing as a Nash equilibrium. Formally, if agents start off with a joint strategy $\sigma^0$ then for the infinite sequence of regret, $(r^t(\sigma^0))_{t=0}^{\infty}$, we must show that

$$\lim_{t\to\infty} r^t(\sigma^0) = 0. \tag{19}$$

We start by examining the case where $\eta = 0$, i.e. the game never resets, for which case we will derive the more relaxed condition,

$$\lim_{t\to\infty} r^t(\sigma^0) = r^\infty \leq r(\sigma^0). \tag{20}$$

This condition lays the foundation for one of the main propositions regarding the correctness of FRAME.

**Proposition 1** *Let $\sigma^\infty$ be the limit of the joint strategies of agents all using FRAME when $\eta = 0$, i.e.*

$$\lim_{t\to\infty} \sigma^t = \sigma^\infty. \tag{21}$$

*Then one of the following two conditions must hold:*

1. *$r^\infty = 0$, i.e. $\sigma^\infty$ is a Nash equilibrium*
2. *$r^\infty > 0$ and the agent with the highest regret at $\sigma^\infty$ is not unique. In this situation $\sigma^\infty$ is called a critical strategy.*[5]

**Proof:** This is proved in Section 3.3. □

3. Any function $d()$ may be used so long as $d(x) > 0$, for $x > 0$.
4. As will be discussed later, the problem with resetting strategies is that it causes random changes in agents' utilities. Therefore, if possible, resetting should be avoided.
5. Formally, we define a joint strategy $\sigma$ to be a *critical strategy* if $r(\sigma) > 0$ and $\sigma \notin \mathcal{N}$. Although this is not a standard term it is related to the idea of a critical point in multivariate calculus.

**Algorithm 1** $FRAME_i(p_i,\ \eta,\ e_i(\cdot),\ d())$

**Require:** $0 \leq p_i < 1$ , $0 < \eta \leq 1$, $d(\epsilon) > 0, \forall \epsilon > 0$

```
σ_i^0 = U(Σ_i)
//Let β be a temporary strategy.
β_i^0 = σ_i^0
for t = 0, 1, ... do
  x_i = U([0, 1])
  // With probability p, consult the expert
  if x_i < p_i then
    β_i^{t+1} is the strategy returned by e_i(·)
    // If β_i^{t+1} is outside of bounded region then must
    // choose a new strategy at random
    if β_i^{t+1} ∉ B(σ_i^t, d(r(σ^t))) then
      x = p_i
    end if
  end if
  //Otherwise, choose a random strategy
  if x_i ≥ p_i then
    β_i^{t+1} = U(B(σ_i^t, d(r(σ^t))))
  end if
  τ_i = U(Σ_i)
  // If new regret is less than current regret, then
  // update current regret and use new joint strategy
  if r(β^{t+1}) < r(σ^t) then
    σ^{t+1} = β^{t+1}
  else
    σ^{t+1} = σ^t
  end if
  x = U([0, 1])
  //If the regret of τ is less than half the current regret,
  //with probability η, the joint strategy will reset to τ
  if x < η and r(τ) < r(σ^t)/2 then
    σ^{t+1} = τ
  end if
end for
```

To avoid the second condition in Proposition 1, it is necessary to be able to jump to a completely new joint strategy. This can be done by having $\eta > 0$. In this case, we can achieve the following, stronger result:

**Proposition 2** *If $\eta > 0$, then*

$$\lim_{t \to \infty} r^t(\sigma^0) = 0. \tag{22}$$

**Proof:** This is proved in Section 3.4. □

As will be shown, the problem with having $\eta > 0$ is that the joint strategy may repeatedly jump to a completely new joint strategy. This can cause chaotic game play and is why we can only guarantee convergence to the set of Nash equilibria, as opposed to convergence to a specific Nash equilibrium. This can result in continually random changes in the utilities for the agents. Obviously there is no way to tell in advance if $\sigma^\infty$ is a critical strategy, but our experimental results chapter shows that this case is rare. Hence, we were able to let $\eta = 0$ for all our experiments and rely solely on Proposition 1 for our correctness.[6]

### 3.2.1 Example

| | | Agent 2 | |
|---|---|---|---|
| | | $a_{2,1}$ | $a_{2,2}$ |
| Agent 1 | $a_{1,1}$ | $-\epsilon$, $-\epsilon$ | 0, 0 |
| | $a_{1,2}$ | 0, 0 | 1, 1 |

Figure 3: A game with a locally optimal and critical joint strategy.

To understand the two conditions in Proposition 1, consider the game in Figure 3. If we use FRAME with $\eta = 0$, this game has two possible outcomes. The first is that $\sigma^\infty = \{(0,1),(0,1)\}$. In words, this means that the game has converged to the joint strategy $(a_{1,2}, a_{2,2})$ which is the game's only Nash equilibrium. This outcome falls under the first condition of Proposition 1.

The second possible outcome is that $\sigma^\infty = \{(1,0),(1,0)\}$ or the joint strategy $(a_{1,1}, a_{2,1})$. Why is this outcome possible? Consider the initial starting strategy $\sigma_0 = \{(\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{1}{2})\}$. For $\sigma_0$, agent 1's regret as a function of agent 2's strategy is[7]

$$\begin{aligned} r_1((\sigma_1,\sigma_2)) &= (1-\sigma_2(a_{2,1})) - (\frac{1-\sigma_2(a_{2,1})}{2} - \frac{\epsilon\sigma_2(a_{2,1})}{2}), \\ &= \frac{1}{2} - \frac{\sigma_2(a_{2,1})}{2} + \frac{\epsilon\sigma_2(a_{2,1})}{2}. \end{aligned} \tag{23}$$

By symmetry of the game, agent 2 has an equivalent regret function. The importance of this function is that as $\sigma_2(a_{2,1})$ decreases, agent 1's regret will increase. Hence, as $\sigma$ moves from $\{(\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{1}{2})\}$ to $\{(1,0),(1,0)\}$, the agent's regret will actually increase (at least for

6. Although having $\eta > 0$ does not make a difference in runtime asymptotically, in practice, having to randomly select a joint strategy and compare it every turn is costly.
7. Note that agent 1's regret is equal to the maximum utility it could have obtained: in this case $1-\sigma_2(a_{2,1})$) minus the utility it did obtain $\frac{1-\sigma_2(a_{2,1})}{2} - \frac{\epsilon\sigma_2(a_{2,1})}{2}$.

a while). Since FRAME only allows new joint strategies to be adopted if they decrease the overall regret, then if $\sigma_0 = \{(\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{1}{2})\}$, FRAME will actually not be able to achieve convergence to the Nash equilibrium. Instead FRAME will converge to the joint strategy $\{(1,0),(1,0)\}$, since that will decrease the overall regret.

Once in the region of $\{(1,0),(1,0)\}$, FRAME will not be able to escape. Hence, $\{(1,0),(1,0)\}$ is a locally optimal joint strategy. It also happens to be a critical strategy since $r_1(\{(1,0),(1,0)\}) = \epsilon = r_2(\{(1,0),(1,0)\})$. Therefore this outcome is covered by the second condition in Proposition 1. This is obviously not a proof that FRAME will always result in one of the two conditions in Proposition 1. However, it does give an idea of how those conditions can arise. To avoid the second condition, it would be necessary to somehow be able to jump from a joint strategy in the region around $\sigma = \{(1,0),(1,0)\}$ to a joint strategy in the region around $\sigma = \{(0,1),(0,1)\}$. This is why Proposition 2 is required.

### 3.3 Proof of Proposition 1

In order to prove Proposition 1, we start by proving the following two lemmas.

**Lemma 1** *Consider the joint strategy $\sigma$. Let $\sigma_i^* \in BR_i(\sigma_i)$. Assuming that $\sigma_i \notin BR_i(\sigma_i)$, consider the line segment $l$ from $\sigma_i$ to $\sigma_i^*$ such that*

$$l(\Delta) = \sigma_i + \Delta\rho,\ 0 \leq \Delta \leq 1, \tag{24}$$

*where $\rho = \sigma_i^* - \sigma_i$.*

*Then for $0 < x \leq 1$, $u_i(l(x), \sigma_{-i}) > u_i(\sigma_i, \sigma_{-i})$ and $r_i(l(x), \sigma_{-i}) < r_i(\sigma_i, \sigma_{-i})$.*

**Proof:** We first write out $u_i$ as

$$u_i(\sigma_i, \sigma_{-i}) = \sum_{a_i} \sum_{a_{-i}} P(a_i|\sigma_i) P(a_{-i}|\sigma_{-i}) u_i(a_i, a_{-i}). \tag{25}$$

The total differential, $du_i$, of equation 25 is

$$du_i = \frac{\partial u_i}{\partial \sigma_i(a_{i,1})} d\sigma_i(a_{i,1}) + \ldots + \frac{\partial u_i}{\partial \sigma_i(a_{i,|A_i|})} d\sigma_i(a_{i,|A_i|}), \tag{26}$$

$$= \bigtriangledown u_i \cdot < d\sigma_i(a_{i,1}), \ldots, d\sigma_i(a_{i,|A_i|}) > . \tag{27}$$

If we are only interested in the total differential along $l$ then we can simplify equation 27 to

$$du_i = \bigtriangledown u_i \cdot \rho d\Delta. \tag{28}$$

Since Equation 25 is just a summation of linear terms, each of the partial derivatives is constant, and therefore $\bigtriangledown u_i$ is also a constant. Therefore, the rate of change is constant along $l$ and must be increasing. Since the utility is increasing the regret must be decreasing. □

**Lemma 2** *For a given $\epsilon$-Nash equilibrium $\sigma$, let $f_\sigma(\sigma^\infty) : \mathbb{R}^{N|A|} \rightarrow \mathbb{R}$ be the change in regret from moving from the strategy $\sigma$ to the new strategy $\sigma^\infty$, i.e.,*

$$f_\sigma(\sigma^\infty) = r(\sigma) - r(\sigma^\infty). \tag{29}$$

*If there is some strategy $\sigma'$ such that $f_\sigma(\sigma') > 0$ and $|\sigma' - \sigma| < d(\epsilon)$, then there exists some region $Y \subseteq \Sigma$ such that*

$$P(\mathcal{U}(B(\sigma, d(\epsilon))) \in Y) > 0, \tag{30}$$

*and furthermore, for all $\sigma'' \in Y$, $f_\sigma(\sigma'') > 0$. In words, if there is at least one strategy, $\sigma'$, within a bounded region around $\sigma$ which has less regret than $\sigma$, then there is a positive probability of picking a strategy uniformly at random from that bounded region that has regret less than $\sigma$. Furthermore, this region includes $\sigma'$.*

**Proof:** This proof is left for Appendix A. □

Using these two lemmas, we can prove the following proposition.

**Proposition 3** *For any non-critical $\epsilon$-Nash equilibrium $\sigma$, the region $B(\sigma, d(\epsilon))$ contains a region $S$, such that $P(\mathcal{U}(B(\sigma, d(\epsilon))) \in S) > 0$ and for all $\sigma' \in S$, $r(\sigma') < \epsilon$.*

**Proof:** Since $\sigma$ is a non-critical strategy, there exists an unique agent $i$ such that $r_i(\sigma) = r(\sigma) = \epsilon$. Consider agent $i$'s strategy $\sigma_i$ versus its opponents' joint strategy $\sigma_{-i}$. For $\sigma_{-i}$, agent $i$ has a best response strategy $\sigma_i^* \in BR_i(\sigma_{-i})$ such that $u_i(\sigma_i^*, \sigma_{-i}) > u_i(\sigma_i, \sigma_{-i})$. Let $l$ be a line segment from $\sigma_i$ to $\sigma_i^*$. Note $|l| > 0$ since $\sigma_i^* \neq \sigma_i$. When we adjust agent $i$'s policy, by some amount $\Delta(\sigma_i)$, along $l$ towards $\sigma_i^*$, we decrease $i$'s regret using Lemma 1.

However, at the same time we may increase another agent's regret. In the worst case, suppose that agent $j$ has the second largest regret, $r_j$, and its increase, $\gamma$, with respect to $\Delta(\sigma_i)$, is the largest among all agents. Since we want to decrease the overall amount of regret, we want to choose some $\Delta(\sigma_i)$ such that $r_j + \gamma\Delta(\sigma_i) < r$. Set

$$\Delta(\sigma_i) = \min[d(\epsilon), \frac{r - r_j}{2\gamma}], \tag{31}$$

since this guarantees that $\Delta(\sigma_i) < d(\epsilon)$. Now $r(\sigma_i + \Delta(\sigma_i), \sigma_{-i}) < \epsilon$, that is we have found a better $\epsilon$-Nash equilibrium. By Lemma 2, $P(r(\mathcal{U}(B(\sigma, d(\epsilon)))) < r(\sigma)) > 0$. In words, this means that a joint strategy picked uniformly at random from the region $B(\sigma, d(\epsilon))$ has a positive probability of having less regret than $\sigma$. □

We are now ready to prove Proposition 1.

**Proof:** We start by showing that for all non-critical joint strategies, there is always a new joint strategy close by which is closer to being an equilibrium. By close by, we mean within some bounded region centered on the current joint strategy.[8] Proposition 3 shows that such joint strategies do exist and that FRAME has a positive probability of finding them.

Now suppose that agents play a repeated game for an infinite number of turns using FRAME. Agents will move to a new joint strategy if it decreases the overall regret. Therefore, if for some subsequence, $Q = \{q_1, \ldots\}$, of all turns, the sequence $r_t^q$ converges to a specific value, say $r^\infty$, then the sequence of regret for all turns must be at most $r^\infty$.

8. In our code, this bounded region is denoted by $B(\sigma_i^t, d(r(\sigma^t)))$. In our implementation we used the bounded region of a $L\infty$-ball $D_\infty(\sigma_i^t, d(r(\sigma^t)))$, which can be thought of as a hyper-cube centered around $\sigma_i^t$ with width $2d(r(\sigma^t))$.

Every turn there is a $(1-p_e)^n > 0$ chance of all agents picking a new strategy at random. Therefore, let $Q$ be the infinite subsequence of turns where all agents update their strategies at random.

We now prove Proposition 1 by contradiction. Suppose that $r^\infty > 0$ and $\sigma^\infty$ is not a critical strategy. Now consider some finite point in time, $t-1$, where agent $i$ has the largest regret with respect to $\sigma^{t-1}$. Let us assume the worst case, where agent $j$ has both the second largest regret and $r_j$'s rate of increase with respect to $\sigma_i^{t-1}$, $\gamma$, is the largest among all agents. (If the agent is not unique then $j$ may be any of them.) Define

$$D_p(\sigma^{t-1}) = r_i(\sigma^{t-1}) - r_i(\sigma_i^{t-1} + \Delta^t(\sigma_i), \sigma_{-i}^{t-1}) - \xi, \tag{32}$$

for some small $\xi > 0$, where

$$\Delta(\sigma_i) = \min\left[d(r(\sigma^{t-1})), \frac{r(\sigma^{t-1}) - r_j(\sigma^{t-1})}{2\gamma}\right]. \tag{33}$$

By Proposition 3, at time $t$, there is a positive probability of FRAME being able to reduce the overall regret by at least $D_p(\sigma^{t-1})$ versus the regret at time $t-1$. We would like to be able to say something about the behaviour of $D_p(\sigma^t)$ as $t$ approaches infinity. While in general, we would expect $D_p(\sigma^t)$ to be decreasing, unfortunately it is not necessarily a monotonically decreasing function. Furthermore, even if $\sigma^\infty$ is not a critical strategy, it is possible that at some finite time $t$, $\sigma^t$ might be one. (This is possible since critical strategies may include non-locally optimal strategies or locally optimal strategies from FRAME can still escape from.) Hence, $D_p(\sigma^t)$ may at times even be 0. However, there must exist some time $T^c$ after which no critical strategy is encountered (since the game is approaching a non-critical strategy). We thus define

$$D_{\inf}(\sigma) = \inf\{D_p(\sigma^t) | t \in Q, t \geq T^c\}, \tag{34}$$

where inf or infimum is the greatest lower bound. Note that $\delta_{inf}(r) > 0$.

Now consider the actual decreases in regret given by

$$D_a(\sigma^{t-1}, \sigma^t) = r(\sigma^t) - r(\sigma^{t-1}). \tag{35}$$

We know that $\lim_{t\to\infty} D_a(\sigma^{t-1}, \sigma^t) = 0$, and therefore there exists a point in time $T \in Q$ greater than or equal to $T^c$ such that

$$\forall t' \geq T,\ D_a(\sigma^{t'}, \sigma^{t'+1}) < \delta_{inf}(\sigma). \tag{36}$$

By Proposition 3, for all $t' \geq T$ there exists a positive probability of finding a new joint strategy that reduces the overall regret by at least $\delta_{\inf}(\sigma)$. Therefore this must happen once which is a contradiction of Equation 36. Therefore $\sigma^\infty$ cannot be a critical strategy and Proposition 1 is proven. □

### 3.4 Proof of Proposition 2

If agents get stuck in a locally optimal region, FRAME will have to jump to a completely different region of the strategy space. A key part of Proposition 2 is Lemma 3, which says that if FRAME does pick a joint strategy uniformly at random from all possible joint strategies, there is a positive probability of finding a strategy closer to equilibrium.

**Lemma 3** *Given $\sigma$ such that $r(\sigma) > 0$, there is a positive probability of picking a joint strategy $\sigma' \in \Sigma$ uniformly at random such that $r(\sigma') \leq r(\sigma)/2$.*

**Proof:** This is proved in Appendix A. □

Thus the proof for Proposition 2 will require showing that by picking a joint strategy uniformly at random from all possible strategies enough times, FRAME will never get stuck in a locally optimal region.

**Proof:** We now consider the case where $\eta > 0$. It should be noted that with $\eta > 0$, new joint strategies can now come from $\Sigma$. However, it is still the case that these joint strategies will be picked only if they decrease the overall regret. Hence, cases where convergence was achieved when $\eta = 0$ will still achieve convergence when $\eta > 0$. The difference is that we can now deal with cases where the limiting strategy is a critical strategy.

Suppose that when $\eta = 0$, the limiting strategy is indeed a critical one. (In this case FRAME would be unable to achieve convergence.) Let this strategy be $\sigma^\infty$ and the corresponding regret $r^\infty > 0$. Now set $\eta$ to any value such that $\eta > 0$. In this case, FRAME now choose a new joint strategy from all possible joint strategies. The trick is picking a new strategy such that FRAME is no longer stuck (i.e. the limiting strategy is still $\sigma^\infty$). As previously mentioned, there is no way to know in advance if $\sigma^\infty$ is a critical strategy or what $\sigma^\infty$ will be; hence we must assume the worst case that, $\sigma^\infty$ is indeed a critical strategy. However, since we do not have know what $\sigma^\infty$ will be, there is no way of picking a single new joint strategy such that we guarantee FRAME will not be stuck at $\sigma^\infty$.

Instead we will use an infinite sequence $\sigma^{T'} = \{\sigma^{t'_1}, \sigma^{t'_2}, \ldots\}$ such that, no matter what $\sigma^\infty$ actually is, we can guarantee that FRAME will not get stuck. One possibility is a series of such that $r(\sigma^{t'_{i+1}}) \leq r(\sigma^{t'_i})/2$. That way, no matter what $\sigma^\infty$ is, there exists some time $j$ such that for all $i \geq j$, $r(\sigma^{t'_i}) < r(\sigma^\infty)$. Hence $\sigma^{T'}$ will not get stuck at $\sigma^\infty$.

To prove that $\sigma^{T'}$ can exist, we must show that given any $\sigma$ there is a positive probability of finding $\sigma' \in \Sigma$ such that $r(\sigma') \leq r(\sigma)/2$. This is done using Lemma 3. By setting $\eta > 0$, we guarantee that FRAME is able to make $\sigma^{T'}$ a subsequence of $\sigma^t$.

Therefore, if $\eta > 0$, $\sigma^\infty$, if it exists, cannot be a critical strategy and we have convergence to the set of Nash equilibria. □

It is important to note that throughout all of these proofs, any iteration of a game where agents consult an expert were explicitly ignored. Thus, suggestions made by experts have no impact on the correctness of FRAME.

### 3.5 FRAME Experimental Results

In this section we discuss our findings from a series of experiments using FRAME. We first describe our experimental setup, including which experts were chosen and why, as well as which games were used in the experiments. We then report our findings, and illustrate that FRAME is a practical learning algorithm.

### 3.5.1 Experts

While any expert will work in theory, ones that make gradual adjustments to the strategies of the agents are considered to be better, since it is easier to observe their effect. In our experiments we used three such experts; the Hart and Mas-Colell algorithm, logistic fictitious play and Win or Learn Fast. These experts were chosen because all of them work by making gradual adjustments in strategies. Furthermore, they represent the three basic approaches to multiagent learning. Given the fundamental difference between these experts, it is not surprising that each of them has its own area of expertise, or types of games it is best suited for. By experimenting using these different areas of expertise we are able to clearly contrast these experts.

**Hart and Mas-Colell Algorithm:**

The Hart and Mas-Colell algorithm (HMC) is able to achieve an empirical frequency convergence to the set of correlated equilibria for all games [19]. To calculate an agent's strategy for time $t+1$, HMC examines the time segment $T = 0, \ldots, 1$. Supposing that at time $t$ agent $i$ had played action $a_i^t$, HMC calculates the average reward for playing action $a_i$ throughout the time segment $T$. This average utility is compared against the average utility that agent $i$ could have got if every time agent $i$ played action $a_i$, it had instead played a specific different action, say $a_i'$. (This assumes that agent i's opponents' strategies remained unchanged.) The difference between these two average utilities is called $R^t(a_i, a_i')$. (If the difference is negative, $R^t(a_i, a_i')$ is set to 0.) Thus, at time $t+1$,

$$\sigma_i^{t+1}[a_i'] = \frac{1}{\mu} R_i^t(a_i^t, a_i) \tag{37}$$

and

$$\sigma_i^{t+1}[a_i^t] = 1 - \sum_{a_i' \neq a_i^t} \sigma_i^{t+1}(a_i'). \tag{38}$$

We let $\mu = |A_i|$. If agents are playing strategies, $a_i^t$ is selected according to $\sigma_i^t$.

**Logistic Fictitious Play** Logistic fictitious play (LFP) is a variant on fictitious play [14]. LFP is able to achieve period-by-period convergence in 2x2 games as well as a few other games. An action is played with a probability proportional to the exponential estimate of its expected utility. Specifically,

$$\sigma_i^{t+1}(\sigma_{-i}^t)[a_i] = \frac{e^{(1/\lambda)u_i(a_i, \sigma_{-i}^t)}}{\sum_{a_i' \in A_i} e^{(1/\lambda)u_i(a_i', \sigma_{-i}^t)}}, \tag{39}$$

The parameter $\lambda$ is a smoothness factor.

**Win or Learn Fast:** Win or Learn Fast (WoLF) is a type of gradient ascent algorithm [5]. Gradient ascent algorithms use the gradient in an agent's utility space to help find the best direction to update an agent's strategy. At the same time, WoLF compares the current expected utility for an agent's strategy versus the expected utility from the average strategy. If the expected utility for the current strategy is higher, the agent is "winning" otherwise it is "losing". WoLF trick is to adjust an agent's strategy faster when the agent is losing.

| | | Agent 2 | |
|---|---|---|---|
| | | $a_{2,1}$ | $a_{2,2}$ |
| Agent 1 | $a_{1,1}$ | 1, 0.5 | 0, 0 |
| | $a_{1,2}$ | 0, 0 | 0.5, 1 |

Figure 4: Battle of the Sexes

| | | Agent 2 | | |
|---|---|---|---|---|
| | | $a_{2,1}$ | $a_{2,2}$ | $a_{2,3}$ |
| Agent 1 | $a_{1,1}$ | 0, 0 | 1, 0 | 0, 1 |
| | $a_{1,2}$ | 0, 1 | 0, 0 | 1, 0 |
| | $a_{1,3}$ | 1, 0 | 0, 1 | 0, 0 |

Figure 5: Shapley's Game

| | | Agent 2 | | Agent 2 | |
|---|---|---|---|---|---|
| | | $a_{2,1}$ | $a_{2,2}$ | $a_{2,1}$ | $a_{2,2}$ |
| Agent 1 | $a_{1,1}$ | 1, 1, 0 | 0, 0, 0 | 1, 0, 1 | 0, 1, 1 |
| | $a_{1,2}$ | 0, 1, 1 | 1, 0, 1 | 0, 0, 0 | 1, 1, 0 |
| | | Agent 3 - $a_{3,1}$ | | Agent 3 - $a_{3,2}$ | |

Figure 6: 3-Player Matching Pennies: agent 1 chooses the row, agent 2 chooses the column, and agent 3 chooses the matrix

### 3.6 Games

We ran experiments on the games shown in Figures 4 through 6. For each of these games, we ran 1000 trials. While the starting strategies have a definite impact on the convergence rates and possibly on the relative performance of each of the experts, to avoid an overload in information, we examined only one starting strategy for each game. Since only a small value for $1-p$ was needed to obtain a high degree of randomization, results are only shown for $p = 0.75, 0.95$ and $0.98$. Where ever possible, the parameters for each expert were based on the existing literature. Convergence was measured to 2 decimal places.

All results are shown in histogram format. For each run, our data was divided up into 20 intervals. For example, if for some run, the fastest convergence time was 10 iterations and the slowest was 110, then the interval size for that run's histogram would be 5. Thus, the x-axis in each of our graphs is the convergence time divided up into intervals and the y-axis the percentage of trials that fell into each interval.

For each of these games, given the starting strategies and the experts used, there is no risk of running into a locally optimal joint strategy or a plateau. Thus, for these games, we were able to set $\eta = 0$ and have FRAME's correctness rest solely on Proposition 1.

We examine each of the games in turn.

#### 3.6.1 Battle of the Sexes

*Battle of the Sexes* (BoS) has 3 Nash equilibria;

$$\{((1,0),(0,1)),((0,1),(0,1)),((\frac{2}{3},\frac{1}{3}),(\frac{1}{3},\frac{2}{3}))\}.$$

Different learning algorithms can have a bias towards one or two of the Nash equilibria (usually either the pure or mixed equilibria). Thus, we chose this game because it balances

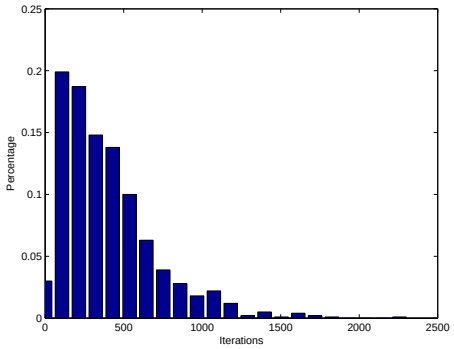


Figure 7: Convergence rates for BoS using a purely random learning algorithm.

a simple joint action set with a complex set of Nash equilibria. We used a starting strategy of $\sigma^0 = \{(\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{1}{2})\}$.

As a reference point, we first had both agents only consulting the Naive Expert. The results are shown in Figure 7. The convergence rates for ALERT on BoS would be off the charts compared to these results (ALERT has a fixed runtime independent of any parameters of the game). Thus, we have already shown that FRAME can be an effective method for learning. However, with the proper use of experts, we can do even better.

The first expert we examined was LFP with a parameter of $\lambda = 0.5$. The results are shown in Figure 8. If $p_e = 1$, it would take around $160,000$ iterations for convergence. However, when $p_e = 0.98$ the convergence rate improves significantly. This shows that occasionally consulting the Naive Expert not only provides theoretical guarantees, it can also be practical.

The next expert we used was WoLF, with parameters $\delta_w = \frac{1}{20000+t}$ and $\delta_l = 2\delta_w$. The results are shown in Figure 9. As shown, WoLF converges very quickly for BoS. In general, however, for such a small game, there is not much difference between a randomized approach and WoLF. The exception is when FRAME forces WoLF to converge to an equilibrium it would not normally converge to. In the case of BoS, by itself, WoLF would never converge to the mixed equilibrium. Hence, when FRAME forces WoLF to do so, convergence takes much longer. This explains why the convergence rate decreases so much for $p_e = 0.95$. When $p_e = 0.98$, there are too few random jumps for FRAME to force WoLF to converge to the mixed equilibrium.

Finally, we used HMC with a parameter of $\mu = 2$. The results are shown in Figure 10. Like WoLF, HMC is able to achieve convergence quickly. Thus, as expected, as $p_e \to 1$, the convergence rate improves.

### 3.6.2 Shapley's Game

Shapley's Game, shown in Figure 5, is a classic game because it was the first game in which fictitious play was shown to not converge in any sense. It is still regarded as a hard game for learning algorithms, with more recent algorithms such as WoLF still unable to achieve

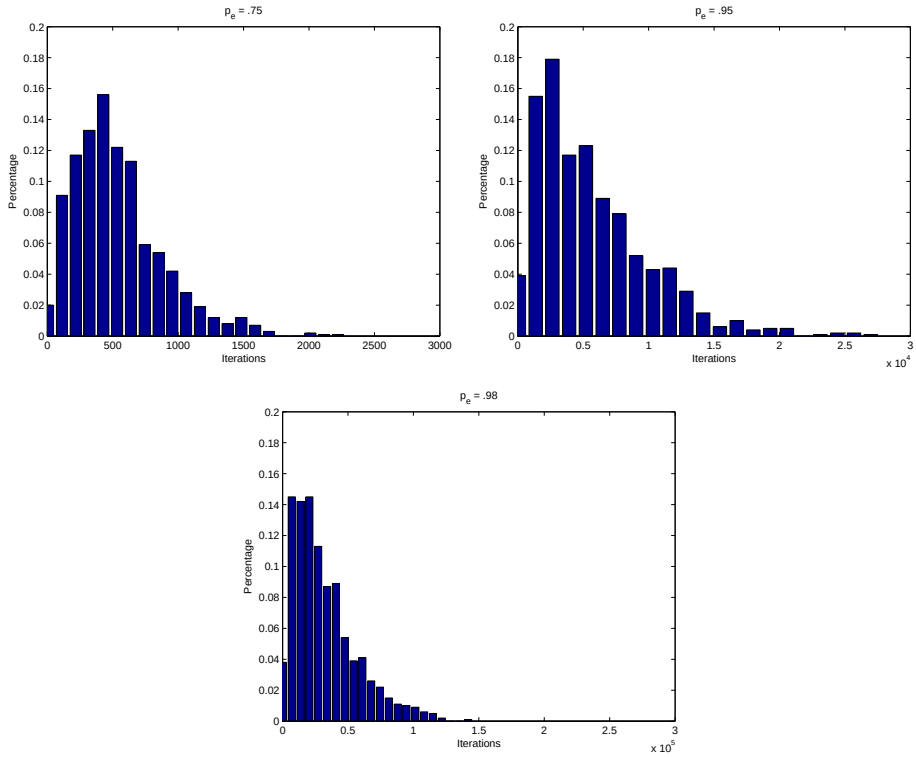


Figure 8: Convergence rates for BoS using FRAME with LFP. Note the difference in scale.

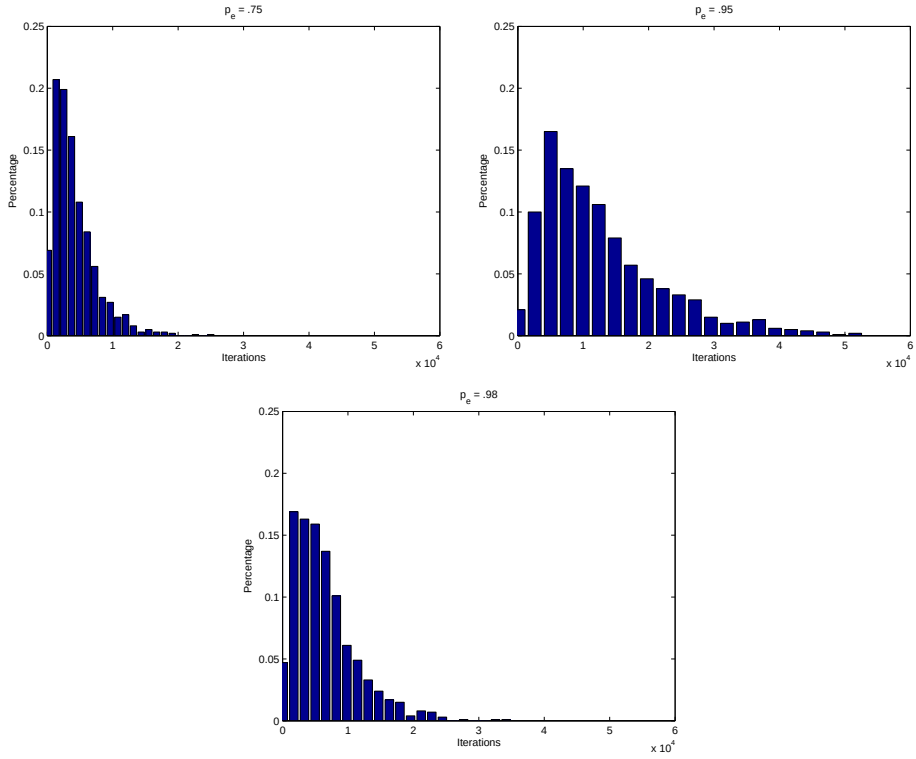


Figure 9: Convergence rates for BoS using FRAME with WoLF.

convergence. However, LFP, given the right value for $\lambda$, converges very quickly. Shapley's game has an unique Nash equilibrium of $\{(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}), (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})\}$.

Again, for reference, we first show, in Figure 11, the results for learning using a purely random learning algorithm.

The results for LFP using a value of $\lambda = 0.5$ are shown in Figure 12. Since LFP converges very quickly for Shapley's game, as $p_e$ increases we could expect to see faster convergence times, which is exactly what happens.

The results for WoLF, using $\delta_w = 1/(100+t)$ and $\delta_l = 3\delta_w$, are shown in Figure 13. Note the difference in scale and that data is presented up to the $98^{th}$ percentile. Since WoLF does not converge for Shapley's Game, as $p_e$ increases, we would expect to see slower convergence rates. This is indeed what happens. More importantly, though, is that as $p_e$ approaches 1, while the convergence rates may increase, we are still achieving convergence. This is an example of FRAME being able to deal with an expert poorly suited for a particular game.

Finally, in Figure 14, we present the results for Shapley's Game using HMC as the expert. Although HMC does not achieve convergence by itself, it is able to achieve convergence to the set of correlated equilibria. As $p_e$ increases, there is no major change in the rate of convergence. This suggests that HMC is no better but also no worse than a purely randomized approach to Shapley's Game.

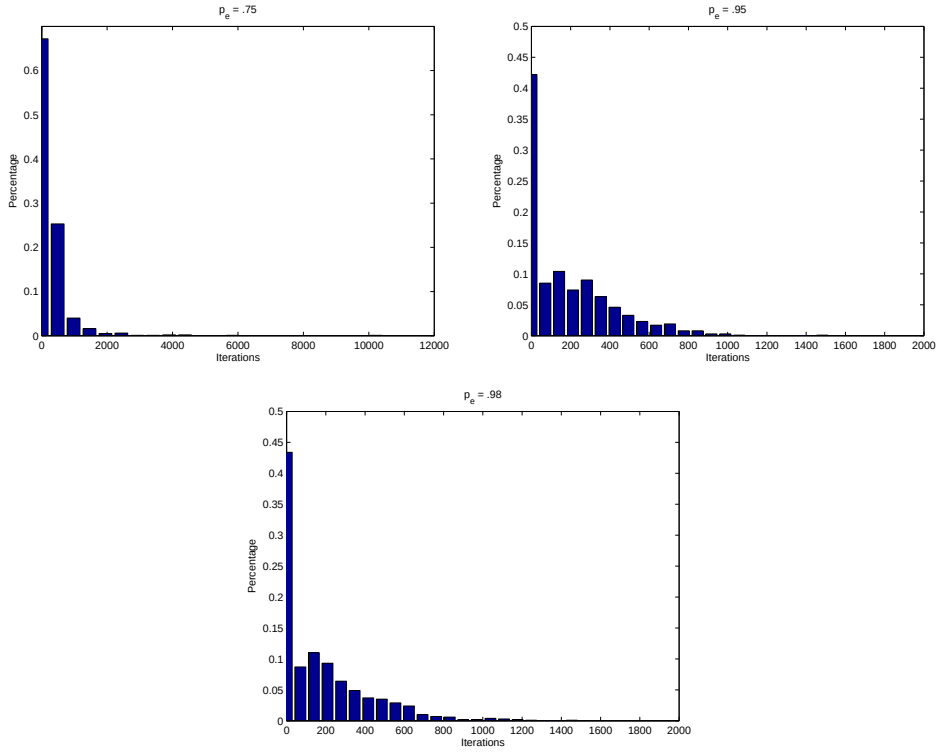


Figure 10: Convergence rates for BoS using FRAME with HMC. Note the difference in scale.

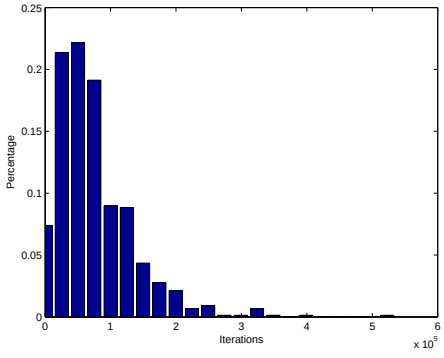


Figure 11: Convergence rates for Shapley's Game using a purely random learning algorithm.

### 3.6.3 3-Player Matching Pennies

3-player Matching Pennies, as shown in Figure 6, is another game which can be very difficult to achieve convergence in. However, unlike Shapley's game, WoLF is able to achieve convergence while LFP is not.

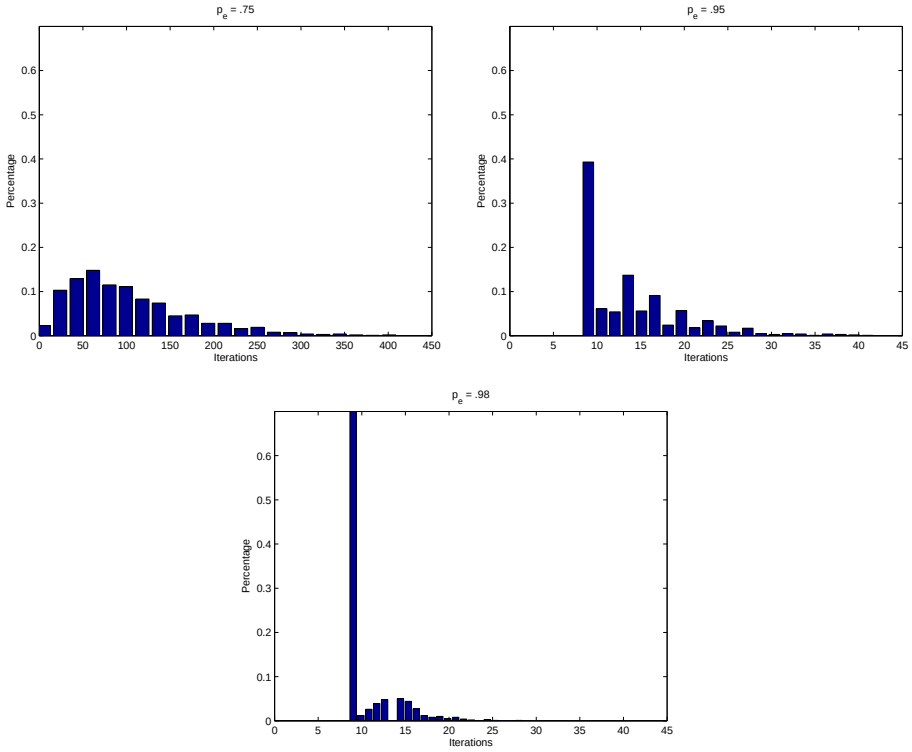


Figure 12: Convergence rates for Shapley's Game using FRAME with LFP. Note the difference in scale.

The first expert we used was LFP. Unlike with Shapley's Game, by itself LFP cannot achieve convergence in 3-Player Matching Pennies. As a result, the more an agent consults LFP, the slower convergence should be. However, we should still be seeing convergence. The results shown in Figure 15 confirm these expectations.

The convergence rates for WoLF are shown in Figure 17. Since WoLF converges quickly in 3-Player Matching Pennies, we would expect to see faster convergence rates. This is what happens, which shows that a poor expert for one game may actually be an excellent expert in another. This is a strong argument in favour of exploring many different experts.

The results for 3-Player Matching Pennies using HMC are shown in Figure 18. We see that, although HMC is not as well suited for 3-player Matching Pennies as it was for Shapley's game, it is still able to perform decently well. This is reflected in the moderate decrease in convergence rates as $p_e$ increases.

## 4. Adaptive-FRAME

In the previous section, we showed the experimental results for FRAME. These results confirmed that FRAME is able to balance theoretical guarantees and practical concerns.

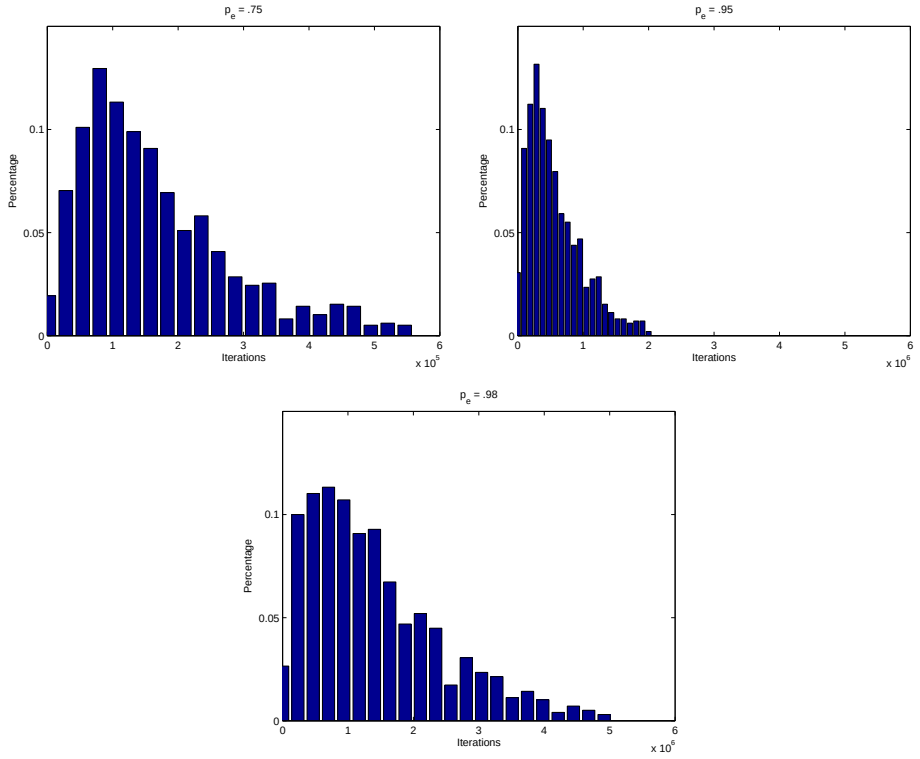


Figure 13: Convergence rates for Shapley's Game using FRAME with WoLF. Note the difference in scale. Each graph is shown up to the $98^{th}$ percentile.

However, FRAME is limited by allowing agents to only consult one expert. Since different experts are better suited for different games, this limits the flexibility of FRAME.

The solution is to allow agents to consult multiple experts. The naive way of doing this is to have agents consult each expert with an equal probability; we call this approach the *Naive Experts Algorithm* (NEA). However, for a given game, if one expert is providing better strategies than another expert, consulting the first expert more would improve convergence rates. Thus, we would like some sort of an adaptive approach to consulting agents.

In order to do so, we have to define some metric for comparing the performance of experts. Based on this metric, we will have to develop some adaptive approach to consulting the experts. It turns out that there are several possible metrics and adaptive approaches. This section is concerned with examining different metrics and approaches; these include existing methods as well as ones designed specifically for FRAME.

### 4.1 An Adaptive Approach

We generalize FRAME in two ways: first, instead of one expert, each agent has a set of experts $E_i = \{e_{i,0}, \ldots, e_{i,|e_i|}\}$ to consult.(For simplicity, $e_{i,0}$ is always the *Naive Expert* which suggests a strategy picked uniformly at random from a bounded region.) Since different

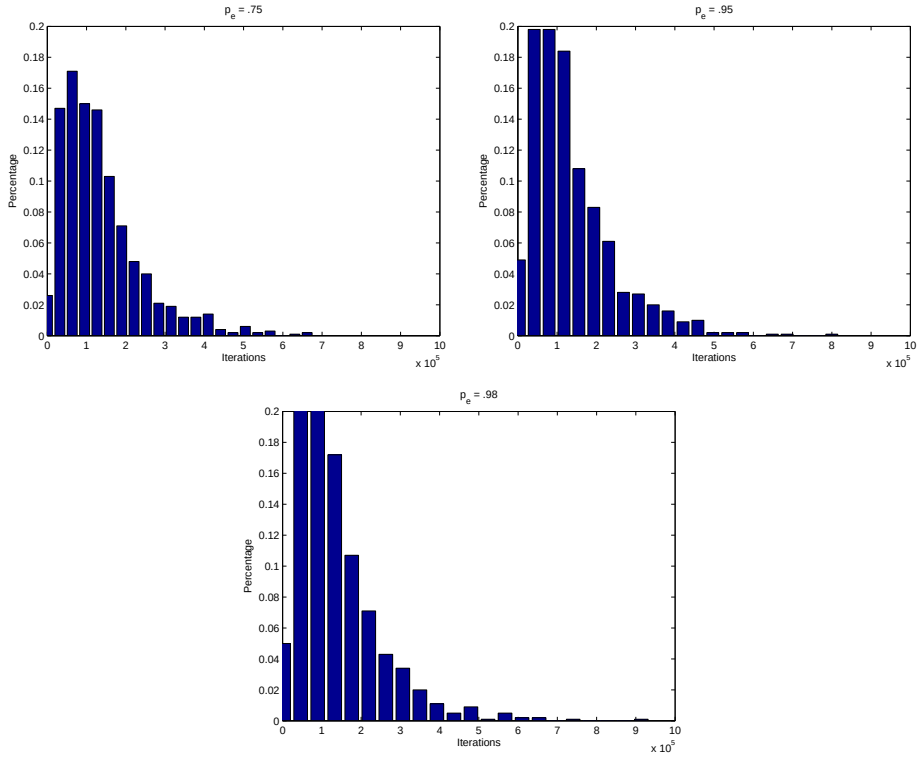


Figure 14: Convergence rates for Shapley's Game using FRAME with HMC.

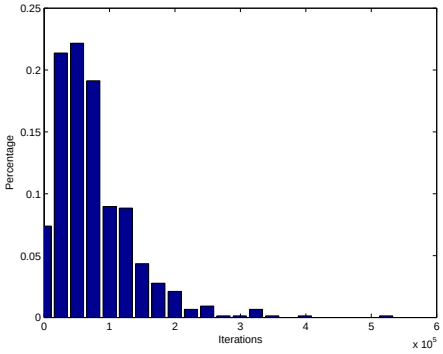


Figure 15: Convergence rates for 3-Player Matching Pennies using a purely random learning algorithm.

experts are better suited for different games, this will allow an agent more flexibility. With slight abuse of notation, we define $e_i$ to be some specific but undefined expert for agent $i$. Expert $e_i$ is consulted with probability $p_{e_i}$ and returns a suggested strategy $\beta_{e_i}$.

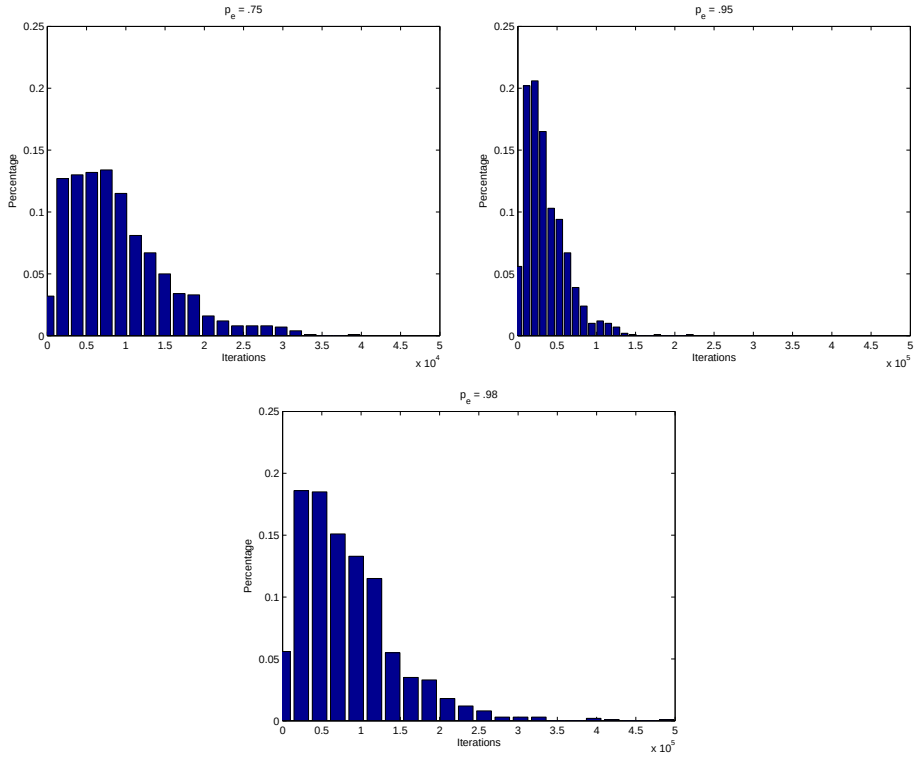


Figure 16: Convergence rates for 3-Player Matching Pennies using FRAME with LFP. Note the difference in scale.

Secondly, we allow the probabilities of consulting each expert to vary over time, that is we generalize $p_{e_i}$ to $p^t_{e_i}$. As a result we are no longer required to decide on and fix the probabilities in advance. We are now able to tune the probabilities to best suit the current game. The most practical way of doing this is to adjust the probabilities while playing the game, since this allows us to deal with new and unknown games. Algorithms that allow us to adjust the probabilities of consulting different experts during game play are called *experts algorithms* [1]. Agent $i$'s experts algorithm is denoted by æ$_i$ and $p_i$ is called æ's policy.

The resulting algorithm, *adaptive-FRAME*, is shown in Algorithm 2. For correctness, we only require that

$$\sum_{t=1}^{\infty} p^t_{e_{i,0}} = \infty. \tag{40}$$

In words this means that the Naive Expert is consulted infinitely often. As long as Equation 40 holds, the correctness for adaptive-FRAME follows directly from Propositions 1 and 2. If Equation 40 does not hold, then by the Borel-Cantelli Lemma, there will only be a finite number of turns where all agents consult the Naive Expert. This would violate the conditions in Proposition 1.

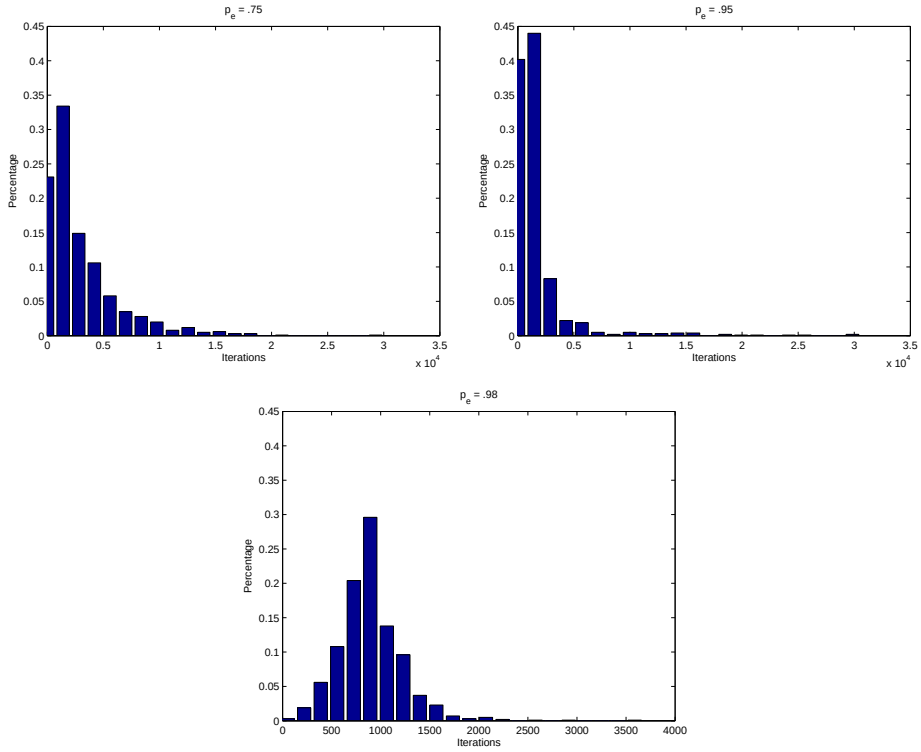


Figure 17: Convergence rates for 3-Player Matching Pennies using FRAME using WoLF. Note the difference in scale.

In practice, this condition does not have to hold. In particular, our experiments were conducted using experts algorithms which did not necessarily satisfy Equation 40. If convergence rates are fast enough then Equation 40 can be relaxed. Hence, it is more of a theoretical condition than a practical one. However, to maintain theoretical correctness of adaptive-FRAME we could set a maximum rate of decay of consulting the Naive Expert, i.e.,

$$\tilde{p}^t_{e_{i,0}} = \max\{g(t), p^t_{e_{i,0}}\} \tag{41}$$

for any $g(t)$ such that $\sum_{t=0}^{\infty} g(t) = \infty$ (for example $g(t) = 1/(t^{0.9})$), and renormalize the other probabilities.

### 4.2 Experts Algorithms

Besides Hedge and SEA, we used two other experts algorithms in our experiments. The *Naive Experts Algorithm*, which chooses each expert with an equal probability, was used as a basis of comparison. We also developed an experts algorithm, *Logistic Expected Regret Reduction Maximization* (LERRM), specifically for use with adaptive-FRAME in the hopes that it could out perform the general purpose experts algorithms.

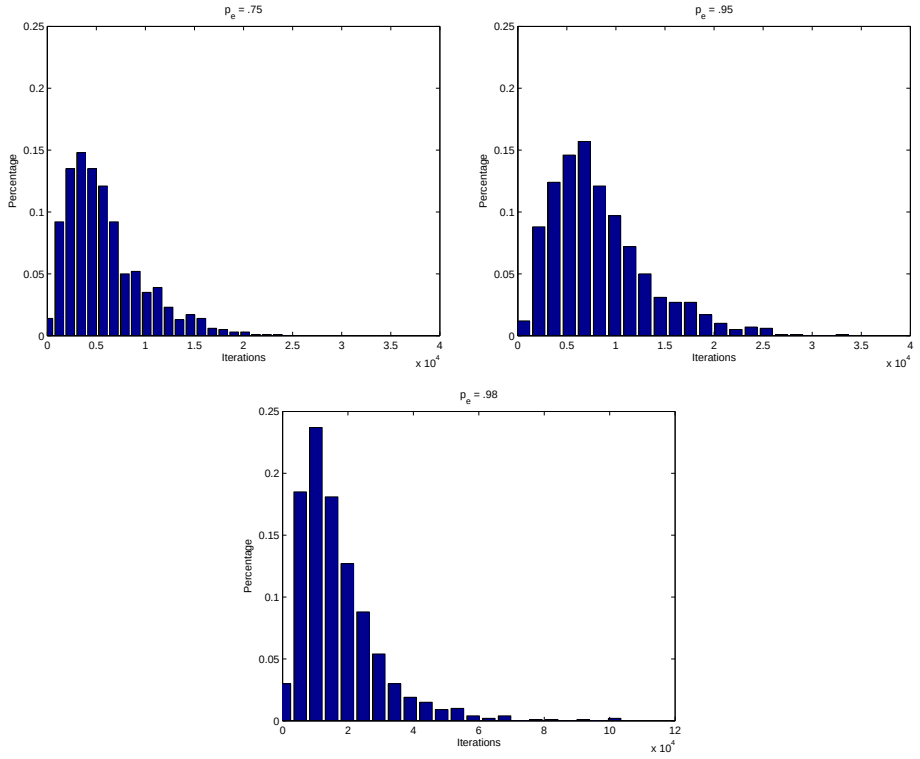


Figure 18: Convergence rates for 3-Player Matching Pennies using HMC. Note the difference in scale.

### 4.2.1 Hedge

The first experts algorithm, Hedge, was created by Auer et al [1]. Here we present the version of Hedge given by Freund and Schapire [13]. Hedge starts by assigning a "weight", $w^1_{e_i}$, to each expert $e_i$ and then consults an expert with a probability equal to that expert's weight proportional to all of the weights. At time $t$, every expert is asked for a suggested strategy even though only one of those strategies is used in the end. Each expert must then calculate the regret its suggested strategy would have obtained had that strategy been used. This regret is denoted by $r^t_{e_i}$. At time $t+1$, each expert's weight is decayed by a factor, $\psi < 1$, raised to $r^t_{e_i}$. Thus, as time proceeds, experts who suggest strategies that would have incurred a high regret are consulted less and less often.

### 4.2.2 Strategic Experts Algorithm

The second experts algorithm we used is by Pucci de Farias and Megiddo [10]. Their algorithm, *Strategic Experts Algorithm* (SEA), differs in two respects. First, once an expert is picked, it is used for a number of consecutive rounds, instead of just one. Secondly, an expert is judged only by how it actually does, as opposed to how it could have done when it was not being consulted.

**Algorithm 2** adaptive-FRAME$_i$ ($æ_i(\cdot)$, $E_i$, $d()$, $\eta$)

$\sigma_i^0 = \mathcal{U}(\Sigma_i)$
**for** $t = 0, 1, \ldots$ **do**
  • $p_i^t = æ_i(\cdot)$ is the probability distribution over the experts $E_i$ for agent $i$ at time $t$
  • $\beta_i^{t+1}$ is the strategy returned by consulting $e_{i,j}(\cdot)$, where $e_{i,j}$ was determined according to $p_i^t$
  **if** $\beta_i^{t+1}$ is not in the bounded region $B(\sigma_i^t, d(r(\sigma^t)))$ **then**
    • $\beta_i^{t+1}$ is the strategy picked uniformly from $B(\sigma_i^t, d(r(\sigma^t))))$
  **end if**
  **if** the regret of $\beta$ is less than the regret of $\sigma^t$ **then**
    $\sigma^{t+1} = \beta^{t+1}$
  **else**
    $\sigma^{t+1} = \sigma^t$
  **end if**
  • $\tau_i$ is strategy picked uniformly at random from $\mathcal{U}(\Sigma_i)$
  **if** the regret of $\tau$ is less than half the regret of $\sigma^{t+1}$ **then**
    • with probability $\eta$, set $\sigma^{t+1} = \tau$
  **end if**
**end for**

The main difference of SEA compared to Hedge is that a new expert to consult is not chosen every turn. Instead, when expert $e_i$ is chosen, it is then consulted for a period of $N_{e_i}$ turns. Initially $N_{e_i} = 1$ but every time expert $e_i$ is consulted $N_{e_i}$ is increased by 1. This means that the more often expert $e_i$ is chosen, the longer it will be consulted for. The other difference between Hedge and SEA is that SEA measures the performance of experts is based on measuring utility, not regret. Specifically, $M_{e_i}$ is used to denote the "average" utility that expert $e_i$'s strategies have obtained. When choosing a new expert to consult at time $t$, with a probability of $1/t$, SEA chooses the expert with the highest $M_{e_i}$ value. Otherwise an expert is chosen at random. Thus, as time goes on, the expert with the highest average utility is consulted more and more often.

#### 4.2.3 Logistic Expected Regret Reduction Maximization

The final experts algorithm we used was one we designed specifically for use with adaptive-FRAME. Logistic Expected Regret Reduction Maximization (LERRM) is inspired by LFP [14]. The metric LERRM uses to measure the performance of an expert is the Expected Regret Reduction (ERR). At time $T$, for agent $i$, expert $e_i$'s ERR is defined as,

$$ERR(e_i)_i^T = \frac{\sum_{t=0}^{T-1}(r_i(\beta^t) - r_i(\beta_{e_i}^{t+1}, \beta_{-i}^{t+1}))}{T}. \tag{42}$$

ERR is a measurement of how much an expert's suggested strategies could have, or did, reduce an agent's regret. Specifically, at time $T$, assuming that all agents other than agent $i$ played the same strategies for $t = 0$ to $t = T$, ERR measures the average reduction in agent $i$'s regret if agent $i$ had always consulted expert $e_i$.

ERR is a better measure of what we are trying to achieve in adaptive-FRAME than an expert's regret or actual utility. Since we are interested in trying to reduce regret, it makes sense to actually measure the ability of each expert to do that. It should be noted that ERR is actually a general purpose measure for multiagent problems and can be used in many situations besides adaptive-FRAME.

**Example:** To demonstrate how ERR is calculated, consider the following example. Suppose that two agents, both using adaptive-FRAME, are playing the repeated game of Battle of the Sexes as shown in Figure 19.

| | | Agent 2 | |
|---|---|---|---|
| | | $a_{2,1}$ | $a_{2,2}$ |
| Agent 1 | $a_{1,1}$ | 1, 0.5 | 0, 0 |
| | $a_{1,2}$ | 0, 0 | 0.5, 1 |

Figure 19: An example for calculating ERR

Suppose that agent 1 has two possible experts to consult each turn, $\{e_{1,1}, e_{1,2}\}$. We wish to calculate the ERR value for both of these experts at time $t = 3$. Table 1 shows the suggested strategies by both experts over the time $t = 0$ to $t = 3$ as well as which expert agent 1 actually consulted each turn. Following adaptive-FRAME, both agents choose their initial strategies uniformly at random instead of consulting an expert. Therefore, $\beta^0_{e_{1,j}}$ is not defined.

| $t$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $\beta^t_{e_{1,1}}$ | - | (1,0) | (1,0) | (1,0) |
| $\beta^t_{e_{1,2}}$ | - | (0.75,0.25) | (0.8,0.2) | (0.7,0.3) |
| Expert consulted | - | $e_{1,1}$ | $e_{1,1}$ | $e_{1,2}$ |
| $\beta^t_1$ | (1,0) | (1,0) | (1,0) | (0.7,0.3) |

Table 1: An example of calculating ERR continued

Suppose that agent 2's strategy over the same period is given by Table 2.

| $t$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\beta^t_2$ | (0,1) | (0.1,0.9) | (0.5,0.5) | (0.6,0.4) |

Table 2: An example of calculating ERR continued

We first calculate expert $e_{1,1}$'s ERR. It does not matter that expert $e_{1,1}$ was not always consulted. What we are interested in is what would have happened if expert $e_{1,1}$ was always consulted, assuming that this would not have changed any of agent 2's strategies. To calculate expert $e_{1,1}$'s ERR, we start by noting that at time $t = 0$, agent 1's regret was 0.5. At time $t = 1$, if agent 1 had gone with the strategy suggested by expert $e_{1,1}$ (which it actually did), agent 1's regret would have become 0.35. Thus by consulting expert $e_{1,1}$

for $\sigma_1^1$, agent 1 would have reduced its regret by 0.15. Similarly, by consulting expert $e_{1,1}$ for $\sigma_1^2$, agent 1 would have reduced its regret by 0.35 since $\beta_{e_{1,1}}^2$ was an optimal strategy. Finally, the strategy $\beta_{e_{1,1}}^3$ would have reduced agent 1's regret by 0 since both $\beta_1^2$ and $\beta_{e_{1,1}}^3$ were optimal strategies. Thus we can calculate expert $e_{1,1}$'s ERR as

$$\begin{aligned} ERR(e_{1,1})_1^3 &= \frac{0.15 + 0.35 + 0}{3}, \\ &= \frac{1}{6}. \end{aligned}$$

Through similar reasoning, we can show that $ERR(e_{1,2})_1^3 = 0.1225$.

If ERR was a perfect measure of an expert's ability to reduce regret, it would make sense to simply consult the agent with the highest ERR. However, at any given time, ERR is only an estimation. Hence, it might be that the agent with the highest ERR is not actually the optimal expert to consult. Furthermore, we want to ensure that there is always a positive probability of consulting the naive expert. Thus, we would like to use ERR to determine some probability of consulting each expert. This is exactly what LERRM does.

$$LERRM(e_i)_i^t = \frac{e^{\frac{1}{\lambda}ERR(e_i)_i^t}}{\sum_{e_i' \in E_i} e^{\frac{1}{\lambda}ERR(e_i')_i^t}}. \tag{43}$$

As in LFP, $\lambda$ is a measure of smoothness. Thus LERRM can serve as a balance between using the expert with the highest ERR and considering other experts.

**Example:** Continuing the example for calculating ERR, we can use these values for LERRM. Supposing that $\lambda = 1$, at $t = 4$, LERRM will consult expert $e_{1,1}$ with a probability of .51104 and expert $e_{1,2}$ with a probability of .48896.

### 4.3 Experimental Setup

The games used in the experiments were Battle of the Sexes (Figure 4), Shapley's Game (Figure 5) and 3-player Chicken (Figure 20). These games were chosen to best illustrate the adaptive aspect of adaptive-FRAME. For each of these games, there are obvious optimal experts. As shown in Table 3, WoLF is the best expert for BoS, LFP is the best expert for Shapley's game and HMC is the best expert for 3-player Chicken (shown in Figure 20). Thus the performance of the different experts algorithms in being able to determine the optimal expert should be easy to measure. As well, Shapley's Game and 3-player Chicken are hard games to learn, so these results will help to reinforce the practicality of adaptive-FRAME.

The same experts were used in testing adaptive-FRAME that were used for testing FRAME. All experts were run with the same parameters used in testing FRAME. For experts algorithms, we used NEA as a basis for comparison. We used all three of the experts algorithms mentioned in the previous section; Hedge, SEA and LERRM. LERRM was run with $\lambda = 0.00005$ and Hedge was run with $\beta = 0.00005$. These values were chosen experimentally.

Since adaptive-FRAME is a random process, there will always be a few exceptionally long runs. These runs are not overly representative of the adaptive-FRAME process. Furthermore, showing these results in graphs often forces a loss of detail in the important regions. Hence, when necessary, results are shown for the $98^{th}$ percentile.

| | | Agent 2 | |
|---|---|---|---|
| | | $a_{2,1}$ | $a_{2,2}$ |
| Agent 1 | $a_{1,1}$ | 0.54, 0.54, 0.54 | 0.54, 1, 0.54 |
| | $a_{1,2}$ | 1, 0.54, 0.54 | 0.46, 0.46,0 |

Agent 3 - $a_{3,1}$

| | Agent 2 | |
|---|---|---|
| | $a_{2,1}$ | $a_{2,2}$ |
| $a_{1,1}$ | 0.54,0.54,1 | 0, 0.46, 0.46 |
| $a_{1,2}$ | 0.46,0,0.46 | 0.46, 0.46, 0 |

Agent 3 - $a_{3,2}$

Figure 20: 3-player Chicken: agent 1 chooses the row, agent 2 chooses the column, and agent 3 chooses the matrix

| Game | Number of Iterations to Convergence | | |
|---|---|---|---|
| | LFP | WoLF | HMC (average) |
| BoS | DNC | 3509 | NT |
| Shapley's Game | 14 | DNC | NT |
| 3-player Chicken | DNC | 64 | $< 10$ |

Table 3: Convergence rates for each expert without the use of FRAME.

For comparison purposes we first tested each expert on its own without the use of FRAME. These convergence rates are presented in Table 3.[9]

### 4.4 Battle of the Sexes

BoS was tested using WoLF and LFP as experts. Both WoLF and the Naive Expert do reasonably well by themselves, as shown in Table 3 and Figure 9, respectively. However, as shown in Table 3, LFP does not achieve convergence at a practical rate. The results in Figure 21 show that all of the experts algorithms are able to outperform the worst expert. These confirm the idea that having multiple experts makes agents more flexible and provides protection against poor experts.

However, Figure 21 also shows that NEA does basically as well as the other experts algorithms. NEA is able to perform that well simply because BoS is such a simple game. Since Hedge, LERRM and SEA still outperform the worst expert, this suggests that for simple games there may not be much benefit to using a more sophisticated approach than NEA but there is also no harm in doing so.

### 4.5 Shapley's Game

Shapley's Game was tested using LFP and WoLF as experts. Figure 22 shows that all of the experts algorithms do much better than WoLF, which is the worst expert for Shapley's Game as shown in Table 3. Hedge and LERRM give, on average, much faster convergence rates compared to the NEA. In particular LERRM performs very well. SEA does only about as well as NEA. However, the range of results is much larger for Hedge and LERRM. One possible explanation is that Hedge and LERRM are both very sensitive to initial conditions;

9. DNC = does not converge. NT = not tested.

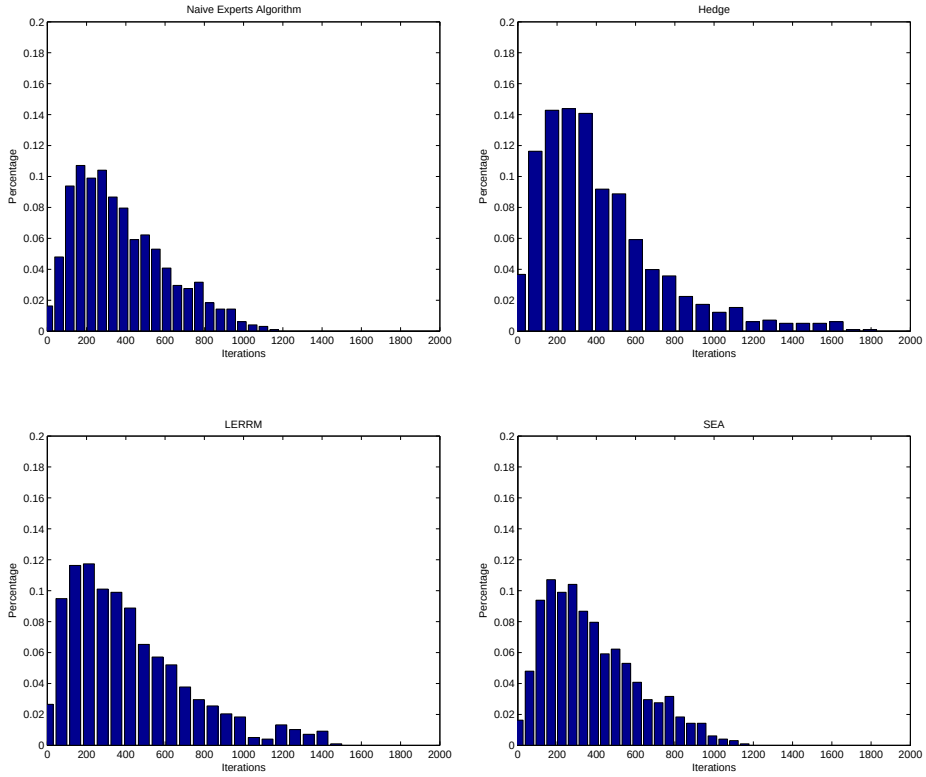


Figure 21: Convergence rates for BoS using adaptive-FRAME. Results are given for the 98th percentile.

having the first few rounds be exceptional cases could throw both of these algorithms off. On the other hand, SEA's performance suggests that it is either poorly suited for use in adaptive-FRAME or convergence happens so quickly that SEA does not have enough time to adapt to consulting the optimal agent.

How are Hedge and LERRM able to achieve this performance? Since LFP gives the fastest convergence rate, Hedge and LERRM should consult LFP with a very high probability. The left column in Figure 23 shows the probability Hedge and LERRM have, respectively, of consulting LFP at the time of convergence. These results show that both experts algorithms, on average, do consult LFP with a very high probability. For Shapley's Game, the other experts are, practically speaking, equally inefficient. Therefore, we would expect both experts algorithms to consult the Naive Expert and WoLF with roughly equal probability. The right column of Figure 23 shows the probability of consulting the Naive Expert minus the probability of consulting WoLF at the point of convergence for Hedge and LERRM, respectively. These results show that in fact, on average, there is no major difference in the probability of consulting the two experts. Thus, we are able to see that Hedge and LERRM are able to adapt so they consult the most appropriate expert for the

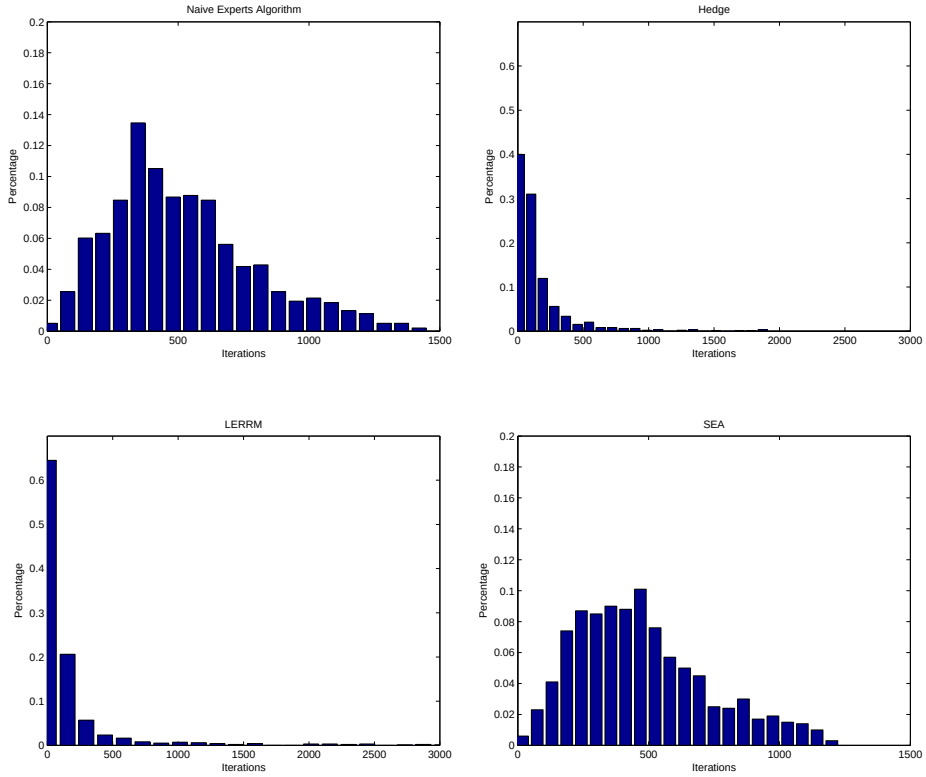


Figure 22: Convergence Rates for Shapley's Game using adaptive-FRAME. Note the difference in scale. Results are given for the 98th percentile.

game. LERRM's superior performance can be attributed to it adapting so that it places most of its weight on LFP.

However, we see that both Hedge and LERRM occasionally perform very poorly. Specifically, while the slowest convergence for SEA was 2021 iterations, Hedge and LERRM's slowest convergence was 12012 and 16848 iterations, respectively. Roughly 3% and 2% of LERRM and Hedge's trials took longer than 2021 iterations respectively. While this is a noticeable number, Hedge and LERRM perform well enough on average that these exceptional cases do not have a noticeable impact. To understand these cases, note that as shown in Figure 23, both Hedge and LERRM will very occasionally wind up consulting LFP with a very low probability. The problem is that both Hedge and LERRM adapt quickly enough so that they are very sensitive to the results from the first few iterations. Given the random nature of adaptive-FRAME, it is not surprising that these iterations are not always representative of the true state of the game. When this is the case, Hedge and LERRM can wind up with an incorrect idea of which experts are optimal to consult. However, even in these exceptional cases adaptive-FRAME is still achieving convergence. On the other hand, since SEA is so slow to adapt, it does not suffer from this problem.

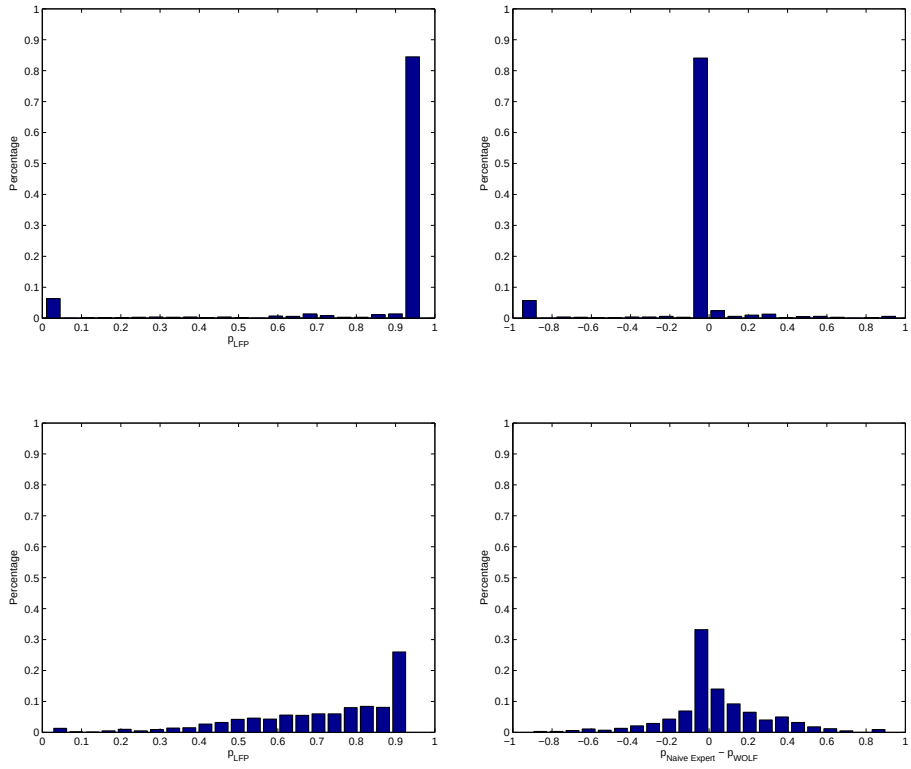


Figure 23: Expert usage statistics for Shapley's Game

### 4.6 3-Player Chicken Game

We present two sets of results for 3-player Chicken. The first set of results, shown in Figure 24, is with just WoLF and LFP as experts. These results show all of the experts algorithms easily outperforming the worst expert for 3-player Chicken, LFP, as shown in Table 3. One of the major differences between 3-player Chicken and Shapley's Game is that SEA can do much better than NEA. This indicates that SEA is able to learn which is the optimal expert to consult; it just takes longer to do so than Hedge or LERRM. The second set of results, shown in Figure 25, is with WoLF, LFP and HMC as experts. HMC is by far the best expert for 3-Player Chicken, hence we would hope to see a noticeable improvement in the convergence rate. However, it might be possible that with an additional expert, it would take longer for the experts algorithms to find the optimal expert.

How are Hedge and LERRM able to outperform NEA? The left column of Figure 26 shows the probability of consulting WoLF in the 3-player Chicken Game. This column shows that both Hedge and LERRM consult WoLF with a very high probability. The right column of Figure 26 shows the difference in probability between consulting the Naive Expert and LFP. Since neither expert is very useful, we do not expect to see much difference in how much they are consulted. This is indeed what we see, therefore, we can conclude that Hedge and LERRM perform well since they both adapt so that they consult the best expert

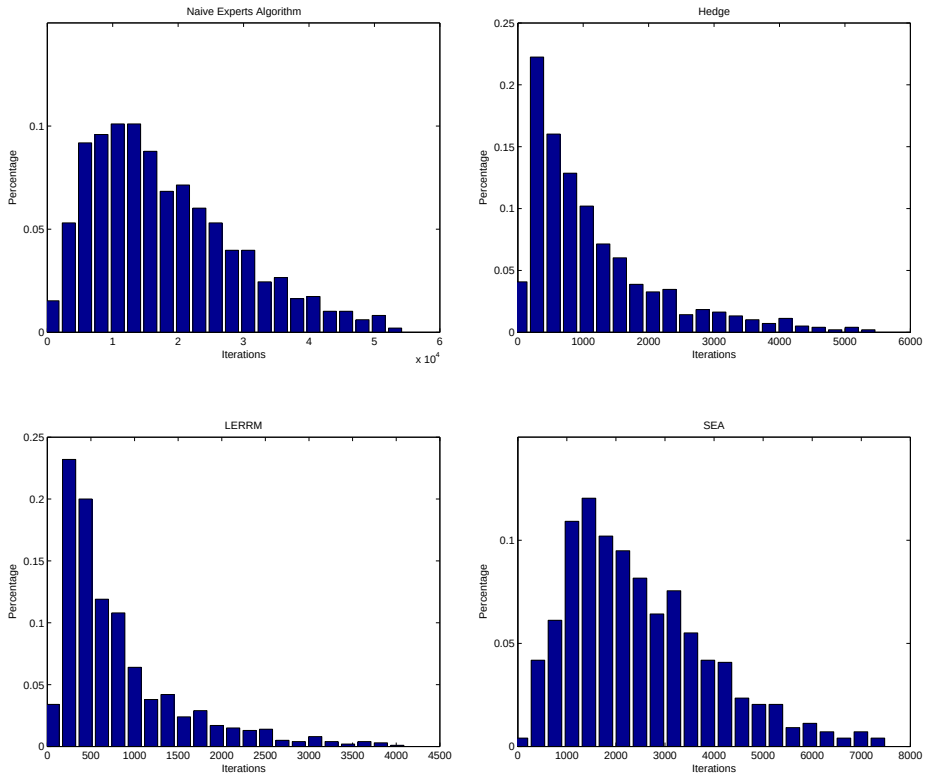


Figure 24: Results for 3-Player Chicken with $E_i = \{NaiveExpert, LFP, WoLF\}$. Note the difference in scale.

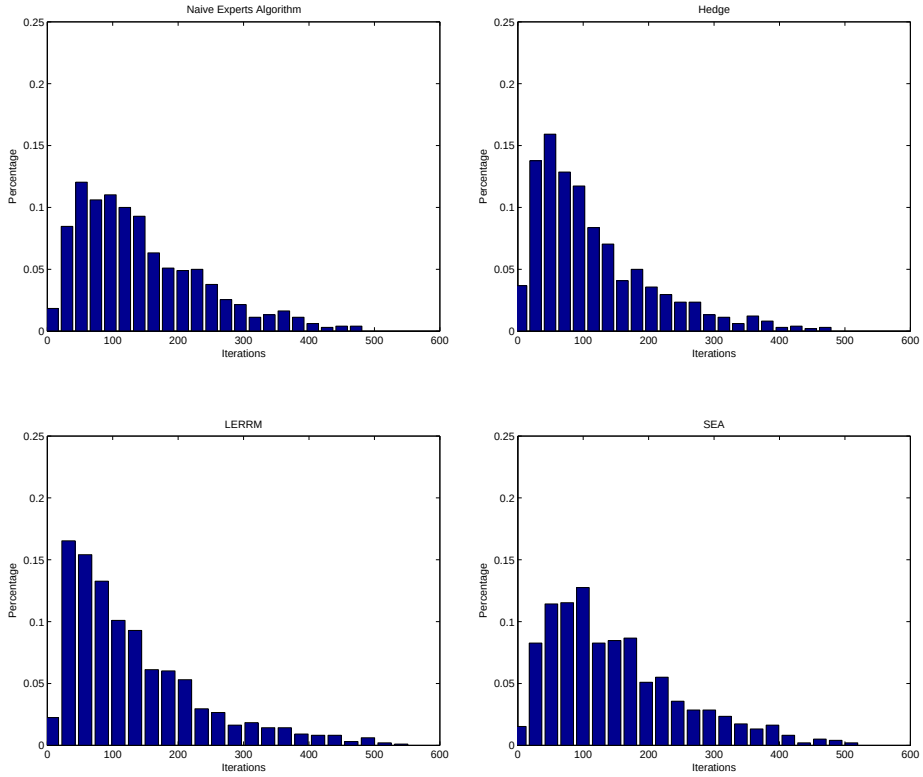


Figure 25: Results for 3-Player Chicken with$E_i = \{NaiveExpert, LFP, WoLF, HMC\}$.

for the game. This time, both Hedge and LERRM consult the best expert with roughly equal probability, which explains their similar results.

As with Shapley's game, the results for 3-player Chicken show that both Hedge and LERRM occasionally have very slow convergence rates. The same analysis applies here.

## 5. Conclusion

Our goal in the research described in this paper was in developing a bridge between theoretical and applied work in multiagent learning. In particular, we were interested in developing learning algorithms for agents playing repeated games, which have strong theoretical guarantees in general, yet can still take advantage, when possible, of the structure within games in order to get desirable real-time performance. To this end, we introduced three key ideas:

- FRAME, a multiagent learning algorithm that balances the theoretical property of guaranteeing convergence to the set of Nash equilibria with the practical requirement of being useable in practice. To do this, FRAME allows agents to use an expert to help provide advice for new strategies. When experts provide good advice, the convergence greatly increases. However, when experts provide poor advice, convergence is still guaranteed.

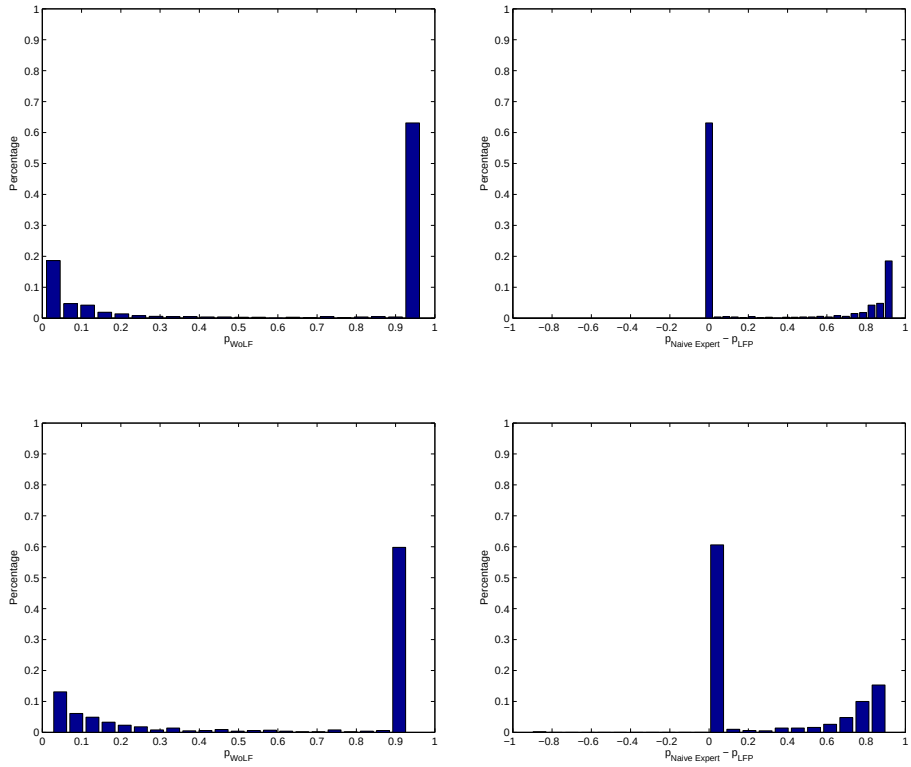


Figure 26: Expert usage statistics for 3-Player Chicken with $E_i = \{NaiveExpert, LFP, WoLF\}$.

- Adaptive-FRAME, a generalization of FRAME. Since no expert is ideally suited for every possible game, adaptive-FRAME allows agents to consult multiple experts. The idea of experts algorithms was incorporated to allow agents to measure the performance of each experts and adapt dynamically to consult the best expert for a given game.
- LERRM, an experts algorithm designed specifically for adaptive-FRAME which helped improve adaptive-FRAME's convergence rate even more. While LERRM was well designed for use with adaptive-FRAME, LERRM can also serve as an experts algorithm in a general MAL setting.

This work opens up several directions for future work. First, we believe that the more experts FRAME and adaptive-FRAME have available to them, the more effective learning procedures they will be. Thus, we are interested in collecting and developing more experts for multiagent learning problems. Second, in this work we studied what was achievable using a specific definition of regret. Expanding out work to encompass concepts like internal-regret or external-regret is an interesting future direction. For example, it might be possible to use

the concept of no-internal regret in FRAME in order to converge to the set of correlated equilibria. Finally, the algorithms in this paper only work in repeated games. We are interested in expanding these ideas to work in stochastic games.

## References


[1] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. Gambling in a rigged casino: The adversarial multi-arm bandit problem. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 322–331, 1995.

[2] Bikramjit Banerjee and Jing Peng. Performance bounded reinforcement learning in strategic interactions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 2–7, San Jose, CA, USA, 2004.

[3] Bikramjit Banerjee and Jing Peng. $RV_{\sigma(t)}$: A unifying approach to performance and convergence in online multiagent learning. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems. (AAMAS)*, pages 2–7, Hakodate, Japan, 2006.

[4] Michael Bowling. Convergence and no-regret in multiagent learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 209–216, Vancouver, Canada, 2005.

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

[6] George W. Brown. Iterative solutions of games by fictitious play. In T.C. Koopmans, editor, *Activity Analysis of Production and Allocation*. Wiley, 1951.

[7] Xi Chen and Xiaotie Deng. Settling the complexity of 2-player Nash-equilibrium. In *FOCS*, 2006.

[8] Vincent Conitzer and Tuomas Sandholm. Awesome: a general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1-2):23–43, 2006.

[9] Costas Daskalakis, Paul Goldberg, and Christos H. Papadimtriou. The complexity of computing a Nash equilibrium. In *Proceedings of the 38th ACM Symposium on Theory of Computing (STOC 2006)*, pages 71–78, Seattle, May 2006.

[10] Daniela Pucci de Farias and Nimrod Megiddo. How to combine expert (or novice) advice when actions impact the environment? In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2003.

[11] Dean Foster and Rakesh Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 21:40–55, 1997.

[12] Dean P. Foster and H. Peyton Young. Regret testing: a simple pay-off based procedure for learning Nash equilibrium. *Theoretical Economics*, 1(3):341–367, 2006.

[13] Yoav Freund and Robert Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[14] Drew Fudenberg and David M Kreps. Learning mixed equilibria. *Games and Economic Behavior*, 5(3):320–367, 1993.

[15] Drew Fudenberg and David Levine. *The Theory of Learning in Games*. MIT Press, 1998.

[16] Fabrizio Germano and Gabor Lugosi. Global Nash convergence of Foster and Young's regret testing. *Games and Economic Behavior*, 2007. To appear.

[17] Amy Greenwald and Keith Hall. Correlated Q-learning. In *International Conference on Machine Learning (ICML)*, pages 242–249, Washington, DC, USA, 2003.

[18] James Hannan. Approximation to Bayes risk in repeated plays. In M. Dresher, A. W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, pages 97–139. Princeton University Press, 1957.

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

[20] Junling Hu and Michael P. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *International Conference on Machine Learning (ICML)*, pages 242–250, 1998.

[21] Michael Littman. Markov games as a framework for multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 157–163, 1994.

[22] John Nash. Equilibrium points in n-person games. *Proc. of the National Academy of Sciences*, 36:48–49, 1950.

[23] Jeffrey S. Rosenthal. *A First Look at Rigorous Probability Theory*. World Scientific, 2000.

[24] Yoav Shoham, Rob Powers, and Trond Grenager. If multiagent learning is the answer, what is the question? *Artificial Intelligence (Special Issue on the Foundations of Research in Multiagent Learning)*, 2007. To appear.

[25] Satinder Singh, Michael Kearns, and Yishay Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 541–548, Stanford, CA, 2000.

[26] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, pages 928–936, Washington, DC, USA, 2003.

## Appendix A. Measure Theory

This Appendix provides a background on the measure theory used in this thesis. Specifically, measure theory is required for the main Propositions regarding FRAME.

An essential step in proving FRAME's correctness is to examine what happens when a strategy is selected uniformly at random from $\Sigma$ or some subset of it. We are unable to use basic probability theory since it only deals with with probabilities involving discrete sample spaces or very basic situations involving continuous sample spaces. Instead we must use a generalization of probability theory called measure theory.

Specifically, this Appendix proves Lemmas 2 and 3 from Chapter 4. Lemma 3 is proved first since its proof provides an introduction to measure theory. A more thorough introduction is given by Rosenthal[23].

**Lemma 4** *Given $\sigma$ such that $r(\sigma) > 0$, there is a positive probability of picking a joint strategy $\sigma' \in \Sigma$ uniformly at random such that $r(\sigma') \leq r(\sigma)/2$.*

**Proof:**

We start by defining a *probability measure space* as the triple $(\Sigma, \mathcal{F}, P)$:[23]

- The joint strategy space $\Sigma$ is also our sample space.
- The $\sigma$-algebra $\mathcal{F}$ is a collection of subsets of $\Sigma$ such that:
  - The sets $\Sigma$ and $\emptyset$, the empty set, are both contained in $\mathcal{F}$.
  - $\mathcal{F}$ is closed under complements and countable unions and intersections.
- The measure probability $P$ which is a mapping from $\mathcal{F}$ to $\mathbb{R}$ such that:
  - $0 \leq P(A) \leq 1$ for all $A \in \mathcal{F}$.
  - If $A_1, A_2, \ldots$ are a countably infinite number of subsets of $\Sigma$ then

$$P(A_1 \cup A_2 \cup \ldots) = P(A_1) + P(A_2 \backslash A_1) + P(A_3 \backslash A_2 \backslash A_1) + \ldots \tag{44}$$

  - $P(\emptyset) = 0$ and $P(\Sigma) = 1$.

Note that $P(X)$ is defined if and only if $X \in \mathcal{F}$.

Although we are free to choose any $\sigma$-algebra, for simplicity we chose one based on $L_\infty$-balls. A $L_\infty$-ball, $D_\infty(\gamma, \epsilon)$, is a hypercube with a center at $\gamma \in \Sigma$ and a width of $2\epsilon \geq 0$. Let $\mathcal{J}$ denote the set of all possible $L_\infty$-balls inside of $\Sigma$. Our $\sigma$-algebra will be $\mathcal{B} = \sigma(\mathcal{J})$, also known as the Borel set, which is the smallest $\sigma$-algebra containing all elements of $\mathcal{J}$.

Finally we must define $P(X)$ for all $X \subseteq \mathcal{B}$.To do this, we will rely on another measure, the *Lebesgue measure* or $\mu$. The Lebesgue measure may be thought of as an extension of volume to a higher dimension. Like $P$, $\mu$ is a mapping from $\mathcal{F}$ to $\mathbb{R}$. However, we do not require that $\mu(\Sigma) = 1$.

We are now define $P(X)$ as

$$P(X) = \frac{\mu(X)}{\mu(\Sigma)}, \tag{45}$$

assuming that $X \in \mathcal{B}$. This definition meets all the requirements for a measure probability and leads to the intuitive idea of what a probability should be.

In the case of this lemma, we are interested in finding $P(\mathcal{N}_\epsilon)$ for some $\epsilon > 0$. In order to find $P(\mathcal{N}_\epsilon)$, we must find $\mu(\mathcal{N}_\epsilon)$ and $\mu(\Sigma)$. Since $\Sigma$ is a solid region, $\mu(\Sigma)$ has a positive value. (Finding the exact value is unnecessary, the important part is that it is positive.) However we can not find a minimum value for $\mu(\mathcal{N}_\epsilon)$ since it may not exist [23]. In other words, there are subsets of $\Sigma$ that do not have a measure defined for them. Although these cases are rare in stage games, for completeness we now consider how to deal with them [16].

Since the measure of a set $X$ is defined for all elements in $\mathcal{F}$, if some $X$ does not have a measure defined for it, this means that $X \notin \mathcal{F}$. How can this happen? Returning to the definition of $\mathcal{F}$, we see that if $A_1$ and $A_2$ are both in $\mathcal{F}$ then so is $A_1 \cup A_2$, and furthermore $\mu(A_1 \cup A_2) = \mu(A_1) + \mu(A_2 \backslash A_1)$. In the context of our definition of $\mathcal{F}$ for the joint strategy space, if $A_1$ and $A_2$ are both $L_\infty$-balls (i.e. both $A_1$ and $A_2$ are in $\mathcal{F}$)) then $A_1 \cup A_2 \in \mathcal{F}$ and furthermore, $\mu(A_1 \cup A_2) = \mu(A_1) + \mu(A_2 \backslash A_1)$. We can expand on this inductively by adding in as many $A_i$'s as we want. In fact, as long as we have a countably infinite number of $A_i$'s all of this will still hold.

The problem arises when we have an uncountably infinite number of $A_i$'s. The difference between countably and uncountably infinite is vital. A set of infinite numbers is countably infinite if they can be enumerated. For example, the set of positive integers is countably infinite because you could start listing off all of them and every positive integer would eventually be included in your list. The same goes for rational numbers. More thought has to be put into your list but there is a way of listing off all the rational numbers such that every one is eventually included. The cardinality of these sorts of sets is $\aleph_0$. For sets that are uncountably infinite there is no way of enumerating them. For example the real numbers are uncountably infinite. No matter what method you try to enumerate them with, there will always be numbers that are never included in your list. The cardinality of reals and similar sets is $\aleph_1$.

The importance of all of this is that $\mathcal{F}$ is not closed under an uncountable number of unions. This means that if $X$ is the union of an uncountably infinite number of $L_\infty$-balls, then its measure may not be defined. To examine this in the context of this thesis, we make the following definition. For a Nash equilibrium $\sigma^{\mathcal{N}_i}$, let $\mathcal{N}_\epsilon(\sigma^{\mathcal{N}_i})$ denote the region in $\Sigma$ that is an $\epsilon$-Nash equilibrium with respect to $\sigma^{\mathcal{N}_i}$. Therefore

$$\mathcal{N}_\epsilon = \cup_{\sigma^{\mathcal{N}_i} \in \mathcal{N}} \mathcal{N}_\epsilon(\sigma^{\mathcal{N}_i}). \tag{46}$$

Since each $\mathcal{N}_\epsilon(\sigma^{\mathcal{N}_i})$ is a subregion of $\Sigma$, $\mu(\mathcal{N}_\epsilon(\sigma^{\mathcal{N}_i}))$ is positive for all $i$. Therefore

$$\begin{aligned} \mu(\mathcal{N}_\epsilon) &= \sum_{\sigma^{\mathcal{N}_i} \in \mathcal{N}} \mu(\mathcal{N}_\epsilon(\sigma^{\mathcal{N}_i})), \\ &> 0, \end{aligned} \tag{47}$$

as long as $\mathcal{N}$ is at most countably infinite. Under these circumstances, $P(\mathcal{N}_\epsilon)$ is always defined and positive. In fact, Germano and Lugosi approach this problem by basically assuming that there are only a finite number of Nash equilibria [16].

Thus, the problem only arises when there are an uncountably infinite number of Nash equilibria. An example of this is shown in Figure 27. This game is not actually a problem

| | | Agent 2 | |
|---|---|---|---|
| | | $a_{2,1}$ | $a_{2,2}$ |
| Agent 1 | $a_{1,1}$ | 1,1 | 1,1 |
| | $a_{1,2}$ | 1,1 | 1,1 |

Figure 27: A simple game with an uncountably infinite number of Nash equilibria.

since every joint strategy is a Nash equilibrium, however it is possible to create more complex games which are. To deal with these problem games we simply consider a single Nash equilibrium, $\sigma^{\mathcal{N}_i}$, out of all possible ones. Since $\mathcal{N}_\epsilon(\sigma^{\mathcal{N}_i}) \subseteq \mathcal{N}_\epsilon$, the probability of randomly picking a strategy that is in $\mathcal{N}_\epsilon$ is at least as high as the probability of picking a strategy in $\mathcal{N}_\epsilon(\sigma^{\mathcal{N}_i})$, which is positive. Thus we simply define $P(\mathcal{N}_\epsilon)$ to be positive in this case as well.

To complete this proof we simply pick some $\epsilon < r(\sigma)/2$. □

Next, we prove Lemma 2 from Chapter 4.

**Lemma 3** *For a given $\epsilon$-Nash equilibrium $\sigma$, let $f_\sigma(\tilde{\sigma}) : \mathbb{R}^{N|A|} \to \mathbb{R}$ be the change in regret from moving from the strategy $\sigma$ to the new strategy $\tilde{\sigma}$, i.e.,*

$$f_\sigma(\tilde{\sigma}) = r(\sigma) - r(\tilde{\sigma}). \tag{48}$$

*If there is some strategy $\sigma'$, such that $f_\sigma(\sigma') > 0$, and $||\sigma' - \sigma|| < d(\epsilon)$ then there exists some region $Y \subseteq \Sigma$ such that*

$$P(\mathcal{U}(B(\sigma, d(\epsilon))) \in Y) > 0, \tag{49}$$

*and furthermore, for all $\sigma'' \in Y$, $f_\sigma(\sigma'') > 0$. In words, if there is at least one strategy $\sigma'$ within a bounded region around $\sigma$ which has less regret, then there is a positive probability of picking a strategy uniformly at random from that bounded region that has regret less than $\sigma$. Furthermore, this region includes $\sigma'$.*

**Proof:** Note that $f$ is continuous (since $r$ is also continuous). Thus by definition of continuity, for every $\tilde{\sigma}$ and every $\delta > 0$ there exists an $\epsilon > 0$ such that if the distance from $\tilde{\sigma}$ to $\tilde{\sigma}'$ is less then $\epsilon$ than the distance between $f(\tilde{\sigma})$ and $f(\tilde{\sigma}')$ is less than $\delta$.

Let $\delta = f(\sigma')/3$. By continuity, there exists some $\epsilon$ such that if $\sigma''$ is within an $\epsilon$-ball of $\pi'$,

$$f(\sigma') - \delta < f(\sigma'') < f(\sigma') + \delta. \tag{50}$$

Considering the first half of the inequality 50, and substituting in $f(\sigma')/3$ for $\delta$, we get

$$f(\sigma') - \frac{1}{3}f(\sigma') = \frac{2}{3}f(\sigma') < f(\sigma''). \tag{51}$$

Now since $f(\sigma') > 0$, $f(\sigma'') > 0$. Since the $\epsilon$-ball has a positive measure, we have found a region of positive measure around $\sigma'$ where equation 29 is positive.

Therefore, by definition of positive measure, $P(\mathcal{U}(B(\sigma, d(\epsilon))) \in Y) > 0$. □