Combining No-regret and Q-learning*

Ian A. Kash
University of Illinois at Chicago
Chicago, IL
iankash@uic.edu

Michael Sullins
University of Illinois at Chicago
Chicago, IL
sullins2@uic.edu

Katja Hofmann
Microsoft Research
Cambridge, UK
katja.hofmann@microsoft.com

ABSTRACT

Counterfactual Regret Minimization (CFR) has found success in settings like poker which have both terminal states and perfect recall. We seek to understand how to relax these requirements. As a first step, we introduce a simple algorithm, local no-regret learning (LONR), which uses a Q-learning-like update rule to allow learning without terminal states or perfect recall. We prove its convergence for the basic case of MDPs (where Q-learning already suffices), as well as limited extensions of them. With a straightforward modification, we extend the basic premise of LONR to work in multi-agent settings and present empirical results showing that it achieves last iterate convergence in a number of settings. Most notably, we show this for NoSDE games, a class of Markov games specifically designed to be impossible for Q-value-based methods to learn and where no prior algorithm is known to achieve convergence to a stationary equilibrium even on average. Furthermore, by leveraging last iterate converging no-regret algorithms (one of which we introduce), we show empirical last iterate convergence in all domains tested with LONR.

KEYWORDS

No-regret learning; reinforcement learning; CFR

1 INTRODUCTION

Versions of counterfactual regret minimization (CFR) [55] have found success in playing poker at human expert level [12, 42] as well as fully solving non-trivial versions of it [9]. CFR more generally can solve extensive form games of incomplete information. It works by using a no-regret algorithm to select actions. In particular, one copy of such an algorithm is used at each *information set*, which corresponds to the full history of play observed by a single agent. The resulting algorithm satisfies a global no-regret guarantee, so at least in two-player zero-sum games is guaranteed to converge to an optimal strategy through sufficient self-play.

However, CFR does have limitations. It makes two strong assumptions which are natural for games such as poker, but limit applicability to further settings. First, it assumes that the agent has perfect recall, which in a more general context means that the state representation captures the full history of states visited (and so imposes a tree structure). Current RL domains may rarely repeat states due to their large state spaces, but they certainly do not encode the full history of states and actions. Second, it assumes that a terminal state is eventually reached and performs updates only after this occurs. Even in episodic RL settings, which do have terminals, it may take thousands of steps to reach them. Neither of these assumptions is required for traditional planning algorithms like value iteration or reinforcement learning algorithms like Q-learning. Nevertheless, approaches inspired by CFR have shown empirical promise in domains that do not necessarily satisfy these requirements [31].

In this paper, we take a step toward relaxing these assumptions. We develop a new algorithm, which we call local no-regret learning (LONR). In the same spirit as CFR, LONR uses a copy of an arbitrary no-regret algorithm in each state. (For technical reasons we require a slightly stronger property we term no-absolute-regret.) The updates for these algorithms are computed in the style of Q-values, which eliminates the need for perfect recall or terminals. Our main result is that LONR has the same asymptotic convergence guarantee as value iteration for discounted-reward Markov Decision Processes (MDP). Our result also generalizes to settings where, from a single agent's perspective, the transition process is time invariant but rewards are not. Such settings are traditionally interpreted as "online MDPs" [18, 39, 40, 53], but also include normal form games. We view this as a proof-of-concept for achieving CFR-style results without requiring perfect recall or terminal states. Under stylized assumptions, we can extend this to asynchronous value iteration and (with a weaker convergence guarantee) a version of RL.

LONR is not an improvement over traditional RL algorithms for solving MDPs. However, naively applying single-agent RL algorithms in settings with multiple agents, such as Markov games, is known to fail to achieve good performance in many cases [29, 54]. In contrast, we believe the robustness provided by no-regret learning will more naturally extend beyond MDPs.

To demonstrate this, in our experimental results we explore settings beyond the exact reach of our theoretical results. Our main results are on a particular class of Markov games known as NoSDE Markov games, which are specifically designed to be challenging for learning algorithms [54]. These are finite two agent Markov games with no terminal states where No Stationary Deterministic Equilibria exist: all stationary equilibria are randomized. Worse, by construction Q-values do not suffice to determine the correct equilibrium randomization. Thus, prior work has focused on designing multiagent learning algorithms which can converge to non-stationary equilibria [54]. The sorts of cyclic behavior that NoSDE games induce has also been observed in more realistic settings of competition between agents [48].

In contrast, we demonstrate that LONR converges to the stationary equilibrium for specific choices of regret minimizer. Furthmore, for these choices of minimizer we achieve not just convergence of the average policy but also of the current policy, or last iterate. Thus our results are also interesting as they highlight a setting for the study of last iterate convergence, an area of current interest, in between simple normal form games [4, 41] and rich, complex settings such as generative adverarial networks (GANs) [14].

*Part of this work was done while Ian Kash was at Microsoft Research. We gratefully acknowledge support from the National Science Foundation via award CCF-1934915. This represents the full version of the paper from AAMAS 2020.

Most work on CFR uses some version of regret matching as the regret minimzer. However, all prior variants of regret matching are known to not possess last iterate convergence in normal form games such as matching pennies and rock-paper-scissors. As part of our analysis we introduce a novel variant, prove that it is no-regret, and show empirically that is provides last iterate convergence in these normal form games as well as all other settings we have tried. This may be of independent interest, as it is qualitatively different from prior algorithms with last iterate covergence which are optimistic versions of standard algorithms [14, 15].

2 RELATED WORK

CFR algorithms remain an active topic of research; recent work has shown how to combine it with function approximation [10, 31, 37, 42, 50], improve the convergence rate in certain settings [20], and apply it to more complex structures [21]. Most relevant to our work, examples are known where CFR fails to converge to the correct policy without perfect recall [35].

Both CFR and LONR are guaranteed to converge only in terms of their average policy. This is part of a general phenomenon for no-regret learning in games, where the "last iterate," or current policy, not only fails to converge but behaves in an extreme and cyclic way [3, 4, 13, 41]. Recent work has explored cases where it is nonetheless effective to use the last iterate. In some poker settings a variant of CFR known as CFR+ [9, 46, 46] has good last iterates, but it is known to cycle in normal-form games. Motivated by training Generative Adversarial Networks (GANs), recent results have shown that certain no-regret algorithms converge in terms of the last iterate to saddle-points in convex-concave min-max optimization problems [14, 15]. The ability to use the last iterate is particularly important in the context of function approximation [1, 28]. Our experimental results provide examples of LONR achieving last iterate convergence when the underlying regret minimizer is capable of it.

Prior work has developed algorithms which combine no-regret and reinforcement learning, but in ways that are qualitatively different from LONR. A common approach in the literature on multi-agent learning is to use no-regret learning as an outer loop to optimize over the space of policies, with the assumption that the inner loop of evaluating a policy is given to the algorithm. There is a large literature on this approach in normal form games [26], where policy evaluation is trivial, and a smaller one on "online MDPs" [18, 39, 40, 53], where it is less so. Of particular note in this literature, Even-Dar et al. [17] also use the idea of having a copy of a no-regret algorithm for each state. An alternate approach to solving multi-agent MDPs is to use Q-learning as an outer loop with some other algorithm as an inner loop to determine the collective action chosen in the next state [25, 29, 38]. Of particular note, Gondek et al. [24] proposed the use of no-regret algorithms as an inner loop with Q-learning as an outer loop while Even-Dar et al. [19] use multi-armed bandit algorithms as the inner loop with Phased Q-learning [33] as the outer loop. In contrast to these literatures, we combine RL in each step of the learning process rather than having one as an inner loop and the other as an outer loop.

Recent work has drawn new connections between no-regret and RL. Srinivasan et al. [45] show that actor-critic methods can be interpreted as a form of regret minimization, but only analyze their performance in games with perfect recall and terminal states. This is complementary to our approach, which focuses on value-iteration-style algorithms, in that it suggests a way of extending our results to other classes of algorithms. Neu et al. [43] study entropy-regularized RL and interpret it as an approximate version of Mirror Descent, from which no-regret algorithms can be derived as particular instantiations. Kovařík and Lisý [34] study algorithms that instantiate a regret minimizer at each state without the counterfactual weightings from CFR, but explicitly exclude settings without terminals and perfect recall from their analysis. Jin et al. [30] showed that in finite-horizon MDPs, Q-learning with UCB exploration achieves near-optimal regret bounds.

The closest technical approach to that used in our theoretical results is that of Bellemare et al. [6] who introduce new variants of the Q-learning operator. However, our algorithm is not an operator as the policy used to select actions changes from round to round in a history-dependent way, so we instead directly analyze the sequences of Q-values.

3 PRELIMINARIES

Consider a Markov Decision Process $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the (finite) action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition probability kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the (expected) reward function (assumed to be bounded), and $0 < \gamma < 1$ is the discount rate. (Q-)value iteration is an operator $\mathcal{T}$, whose domain is bounded real-valued functions over $\mathcal{S} \times \mathcal{A}$, defined as

$$\mathcal{T}Q(s,a) = r(s,a) + \gamma \mathbb{E}_P[\max_{a' \in \mathcal{A}} Q(s', a')] \tag{1}$$

Due to the presence of γ, this operator is a contraction map in $||\cdot||_\infty$, and so converges to a unique fixed point Q^*, where $Q^*(s,a)$ gives the expected value of the MDP starting from state s, taking action a, and thereafter following the optimal policy $\pi^*(s) = \arg\max_{a \in \mathcal{A}} Q^*(s,a)$ [8].

Our algorithm makes use of a no-regret learning algorithm.[1] Consider the following (adversarial full-information) setting. There are n actions $a_1, \ldots a_n$. At each timestep k an online algorithm chooses a probability distribution π_k over the n actions. Then an adversary chooses a reward $x_{k,i}$ for each action i from some closed interval, e.g. $[0, 1]$, which the algorithm then observes. The (external) regret of the algorithm at time k is

$$\frac{1}{k+1} \max_i \sum_{t=0}^{k} x_{t,i} - \pi_t \cdot x_t \tag{2}$$

An algorithm is *no-regret* if there exists a sequence of constants ρ_k such that regardless of the adversary the regret at time k is at most ρ_k and $\lim_{k\to\infty} \rho_k = 0$. A common bound is that ρ_k is $O(1/\sqrt{k})$.

For our results, we make use of a stronger property, that the *absolute value* of the regret is bounded by ρ_k. We call such an algorithm a *no-absolute-regret* algorithm. Algorithms exist that satisfy the even stronger property that the regret is at most ρ_k and at least

[1] It may seem strange to use an algorithm designed for non-stationary environments in a stationary one. We do so with the goal of designing an algorithm that generalizes to non-stationary settings such as "online" MDPs and Markov games.

0. Such *non-negative-regret* algorithms include all linear cost Regularized Follow the Leader algorithms, which includes Randomized Weighted Majority and linear cost Online Gradient Descent [23].

4 LOCAL NO-REGRET LEARNING (LONR)

The idea of LONR is to fuse the essence of value iteration / Q-learning and CFR. A standard analysis of value iteration proceeds by analyzing the sequence of matrices $Q, \mathcal{T}Q, \mathcal{T}^2Q, \mathcal{T}^3Q, \ldots$. The essence of CFR is to choose the policy for each state locally using a no-regret algorithm. While doing so does not yield an operator, as the policy changes each round in a history-dependent way, this process still yields a sequence of Q matrices as follows.

Fix a matrix Q_0. Initialize $|\mathcal{S}|$ copies of a no-absolute-regret algorithm (one for each state) with $n = |\mathcal{A}|$ and find the initial policy $\pi_0(s)$ for each state s. Then we iteratively reveal rewards to the copy of the algorithm for state s as $x^s_{k,i} = Q_k(s, a_i)$,[2] and update the policy π_{k+1} according to the no-absolute-regret algorithm and $Q_{k+1}(s, a) = r(s, a) + \gamma \mathbb{E}_{P,\pi_k}[Q_k(s', a')]$.

Call this process local no-regret learning (LONR). It can be viewed as a synchronous version of Expected SARSA [49] where instead of using an ϵ-greedy policy with decaying ϵ, a no-absolute-regret policy is used instead. In the rest of this section we work up to our main result, that LONR converges to Q^*. Like many prior results using no-regret learning (e.g. [55]), the convergence is of the average of the Q_k matrices.

We work up to this result through a series of lemmas. To begin, we derive a bound on the average of Q values using the no-absolute-regret property. We use two slightly different averages to be able to relate them using the $\mathcal{T}$ operator.

Lemma 4.1. *Let* $\overline{Q}_k = 1/k \sum_{t=1}^{k} Q_t$ *and* $\underline{Q}_k = 1/k \sum_{t=0}^{k-1} Q_t$. *Then*

$$-\gamma\rho_{k-1} + \mathcal{T}\underline{Q}_k(s, a) \le \overline{Q}_k(s, a) \le \gamma\rho_{k-1} + \mathcal{T}\underline{Q}_k(s, a). \quad (3)$$

Proof. By the definitions of LONR and no-regret,

$$\begin{aligned}
\overline{Q}_k(s, a) &= \frac{1}{k}\sum_{t=1}^{k} Q_t(s, a) \\
&= \frac{1}{k}\sum_{t=0}^{k-1} r(s, a) + \gamma \mathbb{E}_{P,\pi_t}[Q_t(s', a')] \\
&= r(s, a) + \gamma \mathbb{E}_P[\frac{1}{k}\sum_{t=0}^{k-1} \mathbb{E}_{\pi_t}[Q_t(s', a')]] \\
&\ge r(s, a) + \gamma \mathbb{E}_P[\max_i \frac{1}{k}\sum_{t=0}^{k-1} Q_t(s', a_i) - \rho_{k-1}] \\
&= -\gamma\rho_{k-1} + r(s, a) + \gamma \mathbb{E}_P[\max_i \frac{1}{k}\sum_{t=0}^{k-1} Q_t(s', a_i)] \\
&= -\gamma\rho_{k-1} + r(s, a) + \gamma \mathbb{E}_P[\max_i \underline{Q}_k(s', a_i)] \\
&= -\gamma\rho_{k-1} + \mathcal{T}\underline{Q}_k(s, a)
\end{aligned}$$

The key step is the inequality in the fourth line, where we use the fact that the policy for state s' is being determined by a no-regret algorithm, so we can use Equation (2) to bound the expected value of the policy by the value of the hindsight-optimal action and the regret bound of the algorithm. Similarly, by the stronger no-absolute-regret property, we can reverse the inequality to get $\overline{Q}_k(s, a) \le \gamma\rho_{k-1} + \mathcal{T}\underline{Q}_k(s, a)$. This proves Equation (3). □

Next, we show that the range that the Q values take on is bounded. This lemma is similar in spirit to Lemma 2 of Bellemare et al. [6]. The full proof is in Appendix B.

Lemma 4.2. *Let* $||r||_\infty = \max_{s,a} |r(s, a)|$. *Then* $||Q_k - Q_0||_\infty \le 1/(1-\gamma)||r||_\infty + 2||Q_0||_\infty$

Combining these two lemmas, we can show that $\underline{Q}_k$ is an approximate fixed-point of $\mathcal{T}$, and that the approximation is converging to 0 as $k \to \infty$.

Lemma 4.3. $||\underline{Q}_k - \mathcal{T}\underline{Q}_k||_\infty \le \frac{1}{k}(1/(1-\gamma)||r||_\infty + 2||Q_0||_\infty) + \gamma\rho_{k-1}$

It remains to show that a converging sequence of approximate fixed points converges to Q^*, the fixed point of $\mathcal{T}$.

Lemma 4.4. *Let* $Q_0, Q_1, \ldots$ *be a sequence such that* $\lim_{k\to\infty} ||Q_k - \mathcal{T}Q_k||_\infty = 0$. *Then* $\lim_{k\to\infty} Q_k = Q^*$.

Combining Lemmas 4.3 and 4.4 shows the convergence of LONR learning.

Theorem 4.5. $\lim_{k\to\infty} \underline{Q}_k = Q^*$.

4.1 Beyond MDPs

While our results do not rely on perfect recall or terminal states the way CFR does, so far they are limited to the case of MDPs while CFR permits multiple agents and imperfect information. We can straightforwardly extend our results to some settings beyond MDPs. In Appendix A we show that a version of Lemma 4.1 holds in MDP-like settings where the transition probability kernel does not change from round to round but the rewards do. Examples of such settings include "online MDPs" and normal-form games. This last result is not particularly surprising as with a single state LONR reduces to standard no-regret learning, whose convergence guarantees in normal-form games are well understood. In Section 6 we present empirical results that, despite a lack of supporting theory, demonstrate convergence in the richer multi-agent setting of Markov games.

5 EXTENSIONS

In this section we consider two extensions to LONR, one allowing it to be updated asynchronously (i.e. not updating every state in every iteration) and the other allowing it to learn from asynchronous updates with bandit feedback (i.e. the standard off-policy RL setting). These are important as a step toward applying LONR beyond settings small enough for tabular approaches. This introduces novel technical issues around the performance of no-regret algorithms when their performance is assessed on a random sample of their rounds (rather than all of them). Therefore, we analyze convergence only in the simplified case where the state to update at each iteration is chosen uniformly at random. We emphasize that this is an

[2] Note that we are revealing the rewards of *all* actions, so we are in the planning setting rather than the standard RL one. We address settings with limited feedback in Section 5.2.

unreasonably strong assumption in practice, and view our results in this section as providing intuition about why sufficiently "nice" processes should converge. We demonstrate empirical convergence in a more standard on-policy setting in Section 6 and leave a more general theoretical analysis to future work.

5.1 Asynchronous updates

In Section 4 we analyzed an algorithm, LONR, which is similar to value iteration in that each state is updated synchronously at each iteration. However, an alternative is to update them asynchronously, where an arbitrary single state is updated at each iteration. Subject to suitable conditions on the frequency with which each state is updated, asynchronous value iteration also converges [7].

A line of work has shown that CFR will also converge when sampling trajectories [22, 32, 36].

In this section, we show that LONR also converges with such asynchonous updates. However, this introduces a new complexity to our analysis. In particular, with synchronous updates there is a guarantee that $\overline{Q}_k(s,a)$ sees exactly the first k values of each action of each of its successor states. This allows us to immediately apply the no-regret property (2). With asynchronous updates, even if we update all actions in a state at the same time, $\overline{Q}_k(s,a)$'s successors may have been updated more or fewer than k times, and $\overline{Q}_k(s,a)$ may have missed some of these updates and observed others more than once, meaning we cannot directly apply (2). We prove the following Lemma to show that a particular sampling process converges to a correct estimate of the average regret, but believe that similar characterizations should hold for other "nice" processes. We demonstrate empirical convergence of asynchronous LONR when states are selected in an on-policy manner in Section 6.

Lemma 5.1. *Let $t_1, \ldots, t_k$ be the first k iterations at which s is updated, s' be a successor of s, $\tau_1, \ldots, \tau_{k'}$ be the iterations before t_k at which s' was updated, and $\xi_{ss'}(k) = 1/k \sum_{i=1}^{k} \mathbb{E}_{\pi_{t_i}} Q_{t_i}(s',a) - 1/k' \sum_{i=1}^{k'} \mathbb{E}_{\pi_{\tau_i}} Q_{\tau_i}(s',a)$. If the state to be updated at each iteration is chosen uniformly at random then $\lim_{k\to\infty} \xi_{ss'}(k) = 0$ with probability 1.*

The proof has two main steps: (1) showing that as time grows large the average of the number of times each update to s' is sampled by an update to s goes to 1 and (2) applying a prior result to conclude that this means the average of the samples converges to the true average.

Proof. Let X_i be the number of times s is updated using τ_i. The X_i are i.i.d. random variables whose law is the geometric distribution with probability 0.5. Thus, $\mathbb{E}[X_i] = 1$ and by the strong law of large numbers the sample average of the X_i converges to 1 with probability 1. Let $c_i = \mathbb{E}_{\pi_{\tau_i}} Q_{\tau_i}(s',a)$ and $C_i = \sum_{i=1}^{k'} c_i$. Then by [16, Theorem 3], $\sum_{i=1}^{k'} c_i X_i / C_i$ also converges to 1 with probability 1. Equivalently, $\lim_{k'\to\infty} \sum_{i=1}^{k'} c_i X_i - C_i = 0$ with probability 1. □

With this in hand, we can now prove a result similar to Lemma 4.1 for asynchronous updates. The primary difference is that now have an additional error term in the bounds, but like the term from the regret it goes to zero per Lemma 5.1. The full proof is in Appendix B.

Lemma 5.2. *Let s be the state selected uniformly at random and updated in iteration $t+1$, for which this is the k-th update and let $\overline{Q}_{t+1}(s,a) = 1/k \sum_{i=1}^{k} Q_{t_i}(s,a)$ and $\overline{Q}_{t+1}(s',a) = \overline{Q}_t(s',a)$ for $s' \neq s$. Then*

$$\min_{s'} \gamma(-\xi_{ss'}(k) - \rho_{k'}) + \mathcal{T}\overline{Q}_t(s,a) \leq \overline{Q}_{t+1}(s,a) \leq \max_{s'} \gamma(-\xi_{ss'}(k) + \rho_{k'}) + \mathcal{T}\overline{Q}_t(s,a). \quad (4)$$

It immediately follows that $\overline{Q}_t$ is an approximate fixed-point of $\mathcal{T}$, and that the approximation is converging to 0 as $k \to \infty$.

Lemma 5.3. *Let k be the minimum number of times a state has been chosen uniformly at random for update by time t. Then $||\overline{Q}_t - \mathcal{T}\overline{Q}_t||_\infty \leq \gamma\rho_{k-1} + ||\xi(k)||_\infty$*

Combining Lemmas 5.3 and 4.4 (the latter of which applies without change) shows the convergence of asynchronous LONR learning.

Theorem 5.4. *If states are chosen for update uniformly at random $\lim_{k\to\infty} \overline{Q}_t = Q^*$ with prob. 1.*

5.2 Asynchronous updates with bandit feedback

In RL, algorithms like Q-learning are usually assumed not to know P and so only have access to feedback corresponding to the action actually taken in the current iteration. In such settings, ordinary no-regret algorithms are not applicable because they require the counterfactual results from actions not chosen. However, multi-armed bandit algorithms, such as Exp3 [2], are designed to achieve no-regret guarantees *in expectation* despite only receiving feedback about the outcomes chosen. It would be natural to adapt LONR to the on-policy RL setting by replacing the no-regret algorithm with a multi-armed bandit one. This type of result has previously been obtained for normal-form games [5], where agents can learn to play optimally even if they only learn their payoff at each stage and not what action the other agents took.

To adapt LONR to make use of multi-armed bandit algorithms, we can use the Q update rule $Q_{t+1}(s,a) = 1/\pi_t(s,a)(r(s,a)+\gamma\mathbb{E}_{\pi_t}[Q_k(s',a')]$ if a is the action chosen for state s and $Q_{t+1}(s,a') = 0$ for $a' \neq a$.[3] The no-absolute-regret algorithm for bandit feedback at s can then be updated as $x_t^s = r(s,a)+\gamma\mathbb{E}_{\pi_t}[Q_k(s',a')]$. (We use the raw rather than importance sampling estimate here because, e.g. Exp3 already includes importance weighting.) Unlike in Q-learning, we do not need to average over Q-values to account for the stochasticity in choice of s' because our convergence results are already for the averages of our Q-values.

With these definitions, Lemma 5.2 can be immediately adapted to this setting with the caveat that now the guarantees only hold in expectation over the choice of action at each iteration and the resulting state. Furthermore, since we require the state be chosen uniformly at random, the resulting algorithm is on-policy in the

[3] The use of importance sampling here is to maintain the structure that successor states are evaluated as $\mathbb{E}_{\pi_t}[Q_k(s',a')]$. Alternatively we could use the SARSA-style update $Q_{t+1}(s,a) = r(s,a) + \gamma Q_t(s',a')$ where a' is the action that was chosen the last time s' was updated and leave all other Q-values unchanged (this also requires appropriately adjusting the way the average is computed).

sense that the algorithm is choosing which action to receive feedback about, but does not control the sequence of states in which it acts.

Lemma 5.5. *Let s be the state selected uniformly at random and updated in iteration $t+1$, for which this is the k-th update and let $\overline{Q}_{t+1}(s,a) = 1/k \sum_{i=1}^{k} Q_{t_i}(s,a)$ and $\overline{Q}_{t+1}(s',a) = \overline{Q}_t(s',a)$ for $s' \neq s$. Then*

$$\min_{s'} \gamma(-\xi_{ss'}(k) - \rho_{k'}) + \mathcal{T}\overline{Q}_t(s,a) \tag{5}$$
$$\leq \mathbb{E}[\overline{Q}_{t+1}(s,a)] \leq \max_{s'} \gamma(-\xi_{ss'}(k) + \rho_{k'})\mathcal{T}\overline{Q}_t(s,a).$$

The same analysis from the asynchronous full information case then yields the following theorem.

Theorem 5.6. *If states are chosen for update uniformly at random, then $\lim_{k\to\infty} E[\overline{Q}_t] = Q^*$.*

This convergence of expectation implies that the $\overline{Q}_t$ converge in probability to Q^*, a weaker guarantee than the almost sure convergence of algorithms like Q-learning. We leave deriving a stronger convergence guarantee with more natural assumptions about state selection to future work.

6 EXPERIMENTS

Our theoretical results in Sections 4 and 5 are restricted to (online) MDPs and normal form games and require a number of technical assumptions. The primary goal of this section is to provide evidence that relaxation of these restrictions may be possible.

Another goal of these results is that while the theory behind LONR calls for a regret minimizer with the no-absolute regret property, we seek to understand the performance of various well-known regret minimizers within the LONR framework, which may or may not be no-absolute regret. One popular class of no-regret algorithms is Follow-the-Regularized Leader (FoReL) algorithms, of which Multiplicative Weights Update (MWU) is perhaps the best known. MWU works by determining a probability distribution over actions by normalizing weights assigned to each action, with the weights equal to the exponential sum of past rewards and a learning rate. It satisfies the stronger non-negative regret property and therefore the no-absolute regret property. Another algorithm we consider is Optimistic Multiplicative Weights Update (OMWU), which extends MWU with optimism by making the slight adjustment of counting the last value twice each iteration, a change which guarantees not just that the average policy is no-regret, but that the last one (the *last iterate*) is as well [15]. We also consider Regret Matching [27] (RM) algorithms, which are the most widely used regret minimizers in CFR-based algorithms due to their simplicity and, unlike FoReL, lack of parameters. With RM, the policy distribution for iteration $t+1$ is selected for actions proportional to the accumulated positive regrets over iterations 0 to t. Regret Matching+ (RM+) is a variation that resets negative accumulated regret sums after each iteration to zero, and applies a linear weighing term to the contributions to the average strategy [46]. The current state of the art algorithm, Discounted CFR (DCFR), is a parameterized algorithm generalizing RM+ where the accumulated positive and negative regrets are weighed separately as well the weight assigned to the contribution to the average strategy [11]. The parameters used are $\alpha = 3/2$, $\beta = 0$ and $\gamma = 2$, which are the values recommended by the authors. All of these variants of RM are known to not have last iterate convergence in general and to not satisfy the non-negative regret property. (We do not know whether they satisfy the no-absolute-regret property.)

In addition to these standard no-regret algorithms, we introduce a new variant of RM called Regret Matching++ (RM++), which updates in a similar fashion to Regret Matching but clips the *instantaneous* regrets at 0. That is, if $R^t(a)$ is the regret of action a in round t RM tracks $\sum_t R^t(a)$ while RM++ tracks the upper bound $\sum_t \max(R^t(a), 0)$.[4] In the appendix we prove that RM++ is in fact a no-regret algorithm. The proof is a minor variation of the proof for RM+ [47]. We also demonstrate that RM++ empirically has last iterate convergence in a number of settings. This may be of independent interest as unlike OMWU it is not obviously describable as an optimistic version of another regret minimizer.

Lastly, we present results for the first two versions of LONR we analyzed theoretically: value-iteration style (LONR-V) and with asynchronous updates (LONR-A). For LONR-A, while the theory requires states be chosen for update uniformly at random, we instead run it on policy. (We add a small probability of a random action, 0.1, to ensure adequate exploration.) Our results show that empirically this does not prevent convergence.

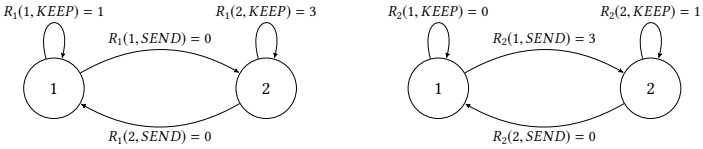
The settings we use for our results are chosen to demonstrate LONR in settings where neither CFR nor standard RL algorithms are applicable. For CFR, this means we choose settings with repeated states and possibly a lack of terminals. For RL, this means considering settings with multiple agents. Since our exposition of LONR is for a single agent setting, we now explain how we apply it in multi-agent settings. We use centralized training, so each agent has access to the current policy of the other agent. This allows the agent to update with the expected rewards and transition probabilities induced by the current policy of the other agent.

6.1 NoSDE Markov Game

Our primary setting is a stateful one with multiple agents. Such settings are naturally modelled as Markov games, a generalization of MDPs to multi-agent settings. A **Markov Game** Γ is a tuple $(S, N, \mathbf{A}, T, R, \gamma)$ where S is the set of states, $N = \{1, ..., n\}$ is the set of players, the set of all state-action pairs $\mathbf{A} = \bigcup_{s\in S}(\{\mathbf{s}\} \times \prod_{n\in N} A_{n,s})$, a transition kernel $T : \mathbf{A} \mapsto \Delta(S)$, and a discount factor γ.

Because Markov Games can model a wide variety of games, algorithms designed for the entirety of this class must be robust to particularly troublesome subclasses. One early negative result found that there exist general-sum Markov Games in which no stationary deterministic equilibria exist, which Zinkevich et al. [54] term NoSDE games. These games have the property that there exists a unique stationary equilibrium with (randomized) policies where the Q-values for each agent are identical in equilibrium but their equilibrium strategies are not. Furthermore, additional complexity exists as the rewards of each player in this NoSDE game can be adjusted within a certain closed interval, where the resulting Q-values remain the same, but the stationary policy changes, thus making Q-value learning even more problematic.

[4] The same idea of clipping instantaneous regrets at 0 has recently been used by actor-critic approaches [45].



(a) Rewards for Player 1 **(b) Rewards for Player 2**

Figure 1: NoSDE Markov Game

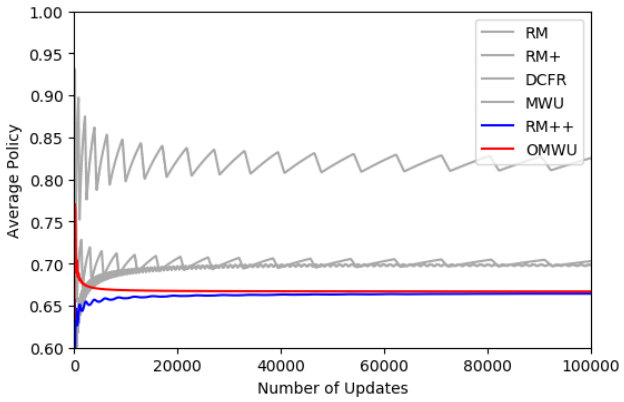


Figure 2: Average Policy for player1. The two lowest lines are the first demonstration of convergence to stationary equilibrium in this setting.

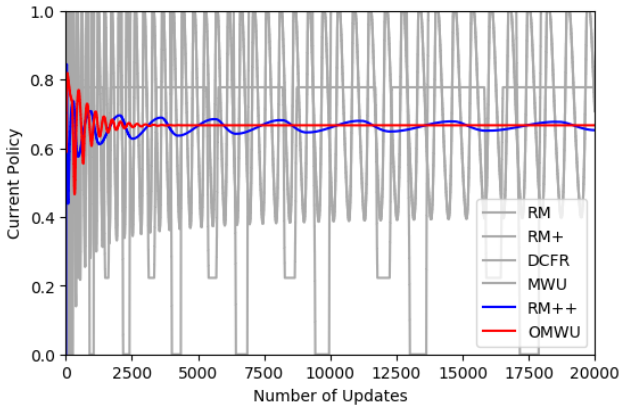


Figure 3: Last iterate for player1. With RM++ or OMWU, LONR converges with the last iterate, not just on average.

The reward structure for the particular NoSDE game we use is shown in Figure 1a for Player 1 and Figure 1b for Player 2. Conceptually, a NoSDE game is a deterministic Markov Game with 2 players, 2 states, and each state has a single player with more than one action. The dynamics of a NoSDE game become cyclic as each player prefers to change actions when the other player does as well, which causes the non-stationarity. In this instance, when player 1 sends, player 2 then prefers to send. This causes player 1 to prefer to keep, which in turn causes player 2 to prefer to keep. Player 1 then prefers to send and the cycle repeats. Due to these negative results, Q-value learning algorithms cannot learn the stationary equilibrium. The state of the art solution is still that of Zinkevich et al. [54] who give a multi-agent value iteration procedure which can approximate a cyclic (non-stationary) equilibrium.

No-regret algorithms are known to converge in self-play, but not necessarily to desirable points, e.g. Nash Equilibrium. This convergence guarantee is in the average policy. Our first results look at the average policies in the NoSDE game with LONR-V. Figure 2 show behavior of the average probability with which player 1 chooses to SEND. The unique stationary equilibrium probability for this action is 2/3. Each algorithm shows convergence, but not to the same value. Not shown but important is that each also is converging to the equilibrium Q^* in the average Q values.

RM and MWU converge to a similar average policy (top two lines). These two algorithms choose based on tracking the sum of regrets and rewards respectively. RM+ and DCFR follow a similar path (next two lines), which makes sense given that RM+ is a special case of DCFR. RM++ and OMWU are the only two which find the stationary equilibrium policy (bottom two lines). These two are also the only two with last iterate convergence properties (OMWU provably and RM++ empirically). Figure 3, which plots the current iterate for each regret minimizer, shows that this holds in our NoSDE game as well. RM++ and OMWU achieve last iterate convergence while for the other four cyclic behavior can be seen.[5] This result highlights NoSDE games as a setting where it would be interesting to theoretically study last iterate convergence in between simple normal form games [4, 41] and rich, complex settings such as GANs [14].

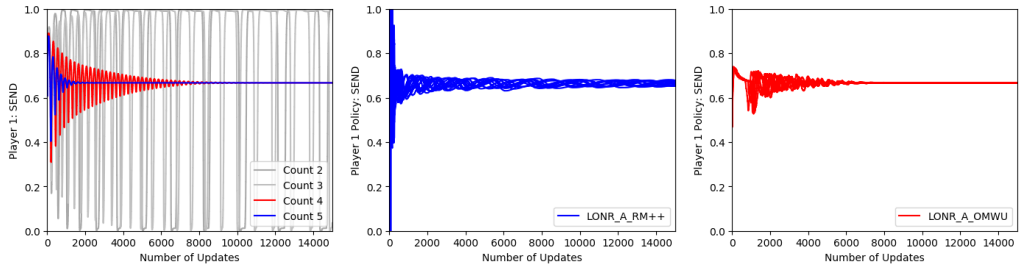
While the theory behind OMWU states that the last value need only be counted twice, our results highlight the difference in the last iterate when more optimism is included (i.e. the last value is counted more than twice.) Specifically, in Figure 4a , we plot the last iterate for increasing counts of the last value. The figure indicates the role increased optimism plays in not only convergence versus divergence, but in how quickly convergence happens. In this case, despite the theory, counting twice does not lead to convergence in the last iterate, but 4 and above does. This simultaneously shows a negative and positive result: increased optimism is not known to work or be required in any other settings. Theoretically exploring this phenomenon is an interesting direction for future work.

Lastly, we analyze LONR-A, the asynchronous version of LONR. We restrict our results to the two which show last iterate convergence, RM++ (Figure 4b) and OMWU (Figure 4c), plotting 100 runs of each. They show that, despite a more natural process for choosing which state to update than our theory permits, we still see convergence.

6.2 Additional Experiments

Additional experiments which bridge the gap from MDPs to NoSDE Markov Games are presented in the Appendix. For a "nicer" Markov game than our deliberately challenging NoSDE game, we use the standard simple 2-player, zero-sum soccer game [38]. With any of our six regret minimizers both LONR-V and LONR-A achieve approximate equilibrium payoffs on average. For a setting to probe the assumptions of our theory in a setting closer to it, we run LONR on the typical benchmark GridWorld environment, an MDP.

[5]The figure shows last iterate convergence of the policy. This also implies convergence of the value estimates. See Appendix C.2.



(a) Impact of level of optimism on convergence **(b) Multiple runs of LONR-A with RM++** **(c) Multiple runs of LONR-A with OMWU**

Figure 4: Additional results for NoSDE Markov Game. (a) Optimism not only leads to convergence of the last iterate, but increasing optimism can affect convergence rates. (b and c) Asynchronous updates still converge using last iterate converging regret minimizers.

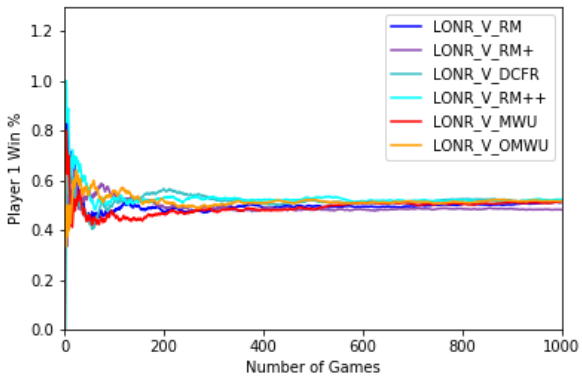


Figure 5: 2-player, zero sum soccer game [38]: All tested no-regret algorithms combined with LONR reach equilibrium between opposing players in self-play.

Specifically we use the standard cliff-walking task which requires the agent to avoid a high-cost cliff to reach the exit terminal state. Again, LONR-V and LONR-A learn the optimal policy (and optimal Q-values) despite regret minimizers that may not satisfy the no-absolute-regret property and, in the case of LONR-A, on policy state selection.

7 CONCLUSION

We have proposed a new learning algorithm, local no-regret learning (LONR). We have shown its convergence for the basic case of MDPs (and limited extensions of them) and presented empirical results showing that it achieves convergence, and in some cases last iterate convergence, in a number of settings, most notably NoSDE games. We view this as a proof-of-concept for achieving CFR-style results without requiring perfect recall or terminal states.

Our results point to a number of interesting directions for future research. First, a natural goal given our empirical results would be to extend our convergence results to Markov games. Second, CFR also works in settings with partial observability by appropriately weighting the different states which correspond to the same observed history. Third, we would like to relax the strong assumptions our results about asynchronous updates require. All three seem to rely on the same fundamental building block of better understanding the behavior of no-regret learners whose rewards are determined by (asynchronous) observations of other no-regret learners. In particular, this leads to challenges due to the resulting non-stationarity of the transition kernels, which leads to hardness results that would need to be circumvented [44, 51, 52]. Some recent progress along these lines has been made [21, 34], but more work is needed.

Orthogonal directions are suggested by our empirical results about last iterate convergence. Can we establish theoretical guarantees for NoSDEs or Markov games more broadly? Is RM++ guaranteed to achieve last iterate convergence? It empirically does in standard games like matching pennies and rock-paper-scissors which trip up most regret minimizers. If so does this represent a new style of algorithm to achieve last iterate convergence or is there a way to interpret its clipping of regrets as optimism?

A BEYOND MDPS

If we move beyond MDPs, P and r are no longer stationary and in general we have a P_k and R_k. This causes problems with the proof of Lemma 4.1. Recall the initial part of that proof, updated to this more general setting:

$$\begin{aligned}\overline{Q}_k(s,a) &= \frac{1}{k}\sum_{t=1}^{k} Q_t(s,a) \\ &= \frac{1}{k}\sum_{t=0}^{k-1} r_t(s,a) + \gamma \mathbb{E}_{P_t,\pi_t}[Q_t(s',a')]\end{aligned}$$

In the original proof, we pulled the expectation over P outside the sum, but now we cannot. In particular, writing the expectation more explicitly gives

$$\frac{1}{k}\sum_{t=0}^{k-1} r_t(s,a) + \gamma \sum_{s'\in\mathcal{S}} P_t(s' \mid s,a)\mathbb{E}_{\pi_t}[Q_t(s',a')] \tag{6}$$

We can still reverse the order of the sums, but the weighting terms now depend on t so they cannot be moved outside. More problematically, they also depend on s and a, so it is not immediately clear how

to generalize our results. For intuition, consider a state s' where there are two actions. At odd k, $r_k(s', a_1) = 1$ and $r_k(s', a_2) = 0$ and vice versa at even k. It is a valid no-regret strategy to randomize uniformly over the actions, but if the P_k are such that you only arrive in s' from s at odd k, then this gives an incorrect estimate. In the remainder of this section, we analyze a special case where we can prove a variant of Lemma 4.1.

A.1 Time-invariant P

If P does not change with k, but r does, we can still prove a version of Lemma 4.1. With a single state, this captures learning in normal-form games, where no-regret learning is indeed known to work. This assumption is also common in the literature on "online MDPs" [18, 39, 40, 53] In this setting, a version of Lemma 4.1 can be proved, but now rather than having a constant operator $\mathcal{T}$ it now changes over time as

$$\mathcal{T}_k Q(s,a) = \underline{r}_k(s,a) + \gamma \mathbb{E}_P[\max_i Q(s', a_i)]. \tag{7}$$

Lemma A.1.

$$-\gamma\rho_{k-1} + \mathcal{T}_k \underline{Q}_k(s,a) \le \overline{Q}_k(s,a) \le \gamma\rho_{k-1} + \mathcal{T}_k \underline{Q}_k(s,a). \tag{8}$$

Proof.

$$\begin{aligned}
&\overline{Q}_k(s,a) \\
&= \frac{1}{k}\sum_{t=0}^{k-1} r_t(s,a) + \gamma \mathbb{E}_P[\frac{1}{k}\sum_{t=0}^{k-1} \mathbb{E}_{\pi_t}[Q_t(s',a')]] \\
&\ge \frac{1}{k}\sum_{t=0}^{k-1} r_t(s,a) + \gamma \mathbb{E}_P[\max_i \frac{1}{k}\sum_{t=0}^{k-1} Q_t(s',a_i) - \rho_{k-1}] \\
&= -\gamma\rho_{k-1} + \underline{r}_k(s,a) + \gamma \mathbb{E}_P[\max_i \underline{Q}_k(s',a_i)] \\
&= -\gamma\rho_{k-1} + \mathcal{T}_k \underline{Q}_k(s,a)
\end{aligned}$$

As before, the key step is applying the no-regret property to obtain the inequality and we apply the same argument with the no-absolute-regret property to obtain the reverse inequality. □

B OMITTED PROOFS

Lemma 4.2. *Let* $||r||_\infty = \max_{s,a} |r(s,a)|$. *Then* $||Q_k - Q_0||_\infty \le 1/(1-\gamma)||r||_\infty + 2||Q_0||_\infty$

Proof. By definition, $Q_k(s,a) = r(s,a) + \gamma \mathbb{E}_{P,\pi_k}[Q_{k-1}(s',a')]$. Thus by the subadditive property of norms, $||Q_k||_\infty \le ||r||_\infty + \gamma||Q_{k-1}||_\infty$. By induction, $||Q_k||_\infty \le (\sum_{t=0}^{k-1}\gamma^k)||r||_\infty + \gamma^k||Q_0||_\infty$. Thus $||Q_k - Q_0||_\infty \le ||Q_k||_\infty + ||Q_0||_\infty \le 1/(1-\gamma)||r||_\infty + 2||Q_0||_\infty$. □

Lemma 4.3. $||\underline{Q}_k - \mathcal{T}\underline{Q}_k||_\infty \le \frac{1}{k}(1/(1-\gamma)||r||_\infty + 2||Q_0||_\infty) + \gamma\rho_{k-1}$

Proof.

$$\begin{aligned}
&||\underline{Q}_k - \mathcal{T}\underline{Q}_k||_\infty \\
&\le ||\underline{Q}_k - \overline{Q}_k||_\infty + ||\overline{Q}_k - \mathcal{T}\underline{Q}_k||_\infty \\
&= ||\underline{Q}_k - \overline{Q}_k||_\infty + \max_{s,a} |\overline{Q}_k(s,a) - \mathcal{T}\underline{Q}_k(s,a)| \\
&\le ||\underline{Q}_k - \overline{Q}_k||_\infty + \gamma\rho_{k-1} \\
&= \frac{1}{k}||Q_k - Q_0||_\infty + \gamma\rho_{k-1} \\
&\le \frac{1}{k}(1/(1-\gamma)||r||_\infty + 2||Q_0||_\infty) + \gamma\rho_{k-1}
\end{aligned}$$

The first step follows by the subadditive property of norms, the second by definition, the third by Lemma 4.1, the fourth by definition, and the fifth by Lemma 4.2. □

Lemma 4.4. *Let* $Q_0, Q_1, \ldots$ *be a sequence such that* $\lim_{k\to\infty} ||Q_k - \mathcal{T}Q_k||_\infty = 0$. *Then* $\lim_{k\to\infty} Q_k = Q^*$.

Proof.

$$\begin{aligned}
||Q_k - Q^*||_\infty &\le ||Q_k - \mathcal{T}Q_k||_\infty + ||\mathcal{T}Q_k - Q^*||_\infty \\
&= ||Q_k - \mathcal{T}Q_k||_\infty + ||\mathcal{T}Q_k - \mathcal{T}Q^*||_\infty \\
&\le ||Q_k - \mathcal{T}Q_k||_\infty + \gamma||Q_k - Q^*||_\infty
\end{aligned}$$

The first step follows by the subadditive property of norms, the second by optimality of Q^*, the third because $\mathcal{T}$ is a contraction map. Rewriting yields

$$||Q_k - Q^*||_\infty \le \frac{1}{1-\gamma}||Q_k - \mathcal{T}Q_k||_\infty$$

Thus, by assumption, $\limsup_{k\to\infty} ||Q_k - Q^*||_\infty \le 0$. Since $||Q_k - Q^*||_\infty \ge 0$, $\liminf_{k\to\infty} ||Q_k - Q^*||_\infty \ge 0$. Thus $\lim_{k\to\infty} ||Q_k - Q^*||_\infty = 0$ and the result follows. □

Lemma 5.2. *Let* s *be the state selected uniformly at random and updated in iteration* $t+1$, *for which this is the* k*-th update and let* $\overline{Q}_{t+1}(s,a) = 1/k \sum_{i=1}^{k} Q_{t_i}(s,a)$ *and* $\overline{Q}_{t+1}(s',a) = \overline{Q}_t(s',a)$ *for* $s' \ne s$. *Then*

$$\begin{aligned}
&\min_{s'} \gamma(-\xi_{ss'}(k) - \rho_{k'}) + \mathcal{T}\overline{Q}_t(s,a) \\
&\le \overline{Q}_{t+1}(s,a) \le \max_{s'} \gamma(-\xi_{ss'}(k) + \rho_{k'}) + \mathcal{T}\overline{Q}_t(s,a).
\end{aligned}$$

Proof. By the definitions of LONR and no-regret algorithms,

$$
\begin{aligned}
&\overline{Q}_{t+1}(s,a) \\
&= \frac{1}{k}\sum_{i=1}^{k} Q_{t_i}(s,a) \\
&= \frac{1}{k}\sum_{i=1}^{k} r(s,a) + \gamma \mathbb{E}_{P,\pi_{t_i}}[Q_{t_i}(s',a')] \\
&= r(s,a) + \gamma \mathbb{E}_P[\frac{1}{k}\sum_{i=1}^{k} \mathbb{E}_{\pi_{t_i}}[Q_{t_i}(s',a')]] \\
&= r(s,a) + \gamma \mathbb{E}_P[-\xi_{ss'}(k) + \frac{1}{k'}\sum_{i=1}^{k'} \mathbb{E}_{\pi_{\tau_i}}[Q_{\tau_i}(s',a')]] \\
&\geq r(s,a) + \gamma \mathbb{E}_P[-\xi_{ss'}(k) + \max_{a'} \frac{1}{k'}\sum_{i=1}^{k'} Q_{\tau_i}(s',a') - \rho_{k'}] \\
&\geq \min_{s'} \gamma(-\xi_{ss'}(k) - \rho_{k'}) + r(s,a) + \gamma \mathbb{E}_P[\max_{a'} \frac{1}{k'}\sum_{i=1}^{k'} Q_{\tau_i}(s',a')] \\
&= \min_{s'} \gamma(-\xi_{ss'}(k) - \rho_{k'}) + r(s,a) + \gamma \mathbb{E}_P[\max_{a'} \overline{Q}_t(s',a')] \\
&= \min_{s'} \gamma(-\xi_{ss'}(k) - \rho_{k'}) \mathcal{T}\overline{Q}_t(s,a')]
\end{aligned}
$$

This argument is essentially the same as in the proof of Lemma 4.1, except that in the fourth equality we apply the definition of ξ to yield a form to which we can then apply the no-regret property. As before, the other half of the proof is symmetric and uses the no-absolute-regret property. □

C REGRET MATCHING++

In this section, we prove that RM++ is a no-regret algorithm and then demonstrate that it has empirical last iterate convergence.

Lemma C.1. *Given a sequence of strategies $\sigma^1, ..., \sigma^T$, each defining a probability distribution over a set of actions A, consider any definition for $Q^t(a)$ satisfying the following conditions:*

(1) $Q^0(a) = 0$
(2) $Q^t(a) = Q^{t-1}(a) + (r^t(a))^+$ *where* $(x)^+ = max(0, x)$

The regret-like value $Q^t(a)$ is then an upper bound on the regret $R^t(a) = \sum_{i=1}^{t} r^i(a)$

Proof. The lemma and proof closely resemble those in [46], [11].

For any $t \geq 1$, $Q^{t+1}(a) - Q^t(a) = Q^t(a) + max(r^t(a), 0) - Q^t \geq Q^t(a) + r^t(a) - Q^t(a) = R^{t+1}(a) - R^t(a)$. This gives $Q^t(a) = \sum_{i=1}^{t} Q^i(a) - Q^{i-1}(a) \geq \sum_{i=1}^{t} R^i(a) - R^{i-1}(a) = R^t(a)$. □

Lemma C.2. *Given a set of actions A and any sequence of rewards v^t such that $|v^t(a) - v^t(b)| < \Delta$ for all t and all $a, b \in A$, after playing a sequence of strategies determined by regret matching but using the regret-like value $Q^t(a)$ in place of $R^T(a)$, $Q^t(a) \leq \Delta\sqrt{2|A|T}$*

Proof. Again, the lemma and proof closely follow [46], [11].

$$
\begin{aligned}
&(\max_a Q^t(a))^2 = \max_a Q^t(a)^2 \leq \sum_a Q^t(a)^2 \\
&= \sum_a (Q^{t-1}(a) + (r^t(a))^+)^2 \\
&= \sum_a (Q^{t-1}(a) + (v^t(a) - \sum_b \sigma^t(b) v^t(b))^+)^2 \\
&\leq \sum_a (Q^{t-1}(a) + (v^t(a) \\
&- \sum_b \sigma^t(b) v^t(b)))^2 + \Delta^2 \\
&= \sum_a Q^{t-1}(a)^2 + (v^t(a) - \sum_b \sigma^t(b) v^t(b))^2 + 2Q^{t-1}(a)(v^t(a) \\
&- \sum_b \sigma^t(b) v^t(b)) + \Delta^2 \\
&\leq 2\Delta^2|A| + \sum_a Q^{t-1}(a)^2 + 2(\sum_a Q^{t-1}(a) v^t(a) \\
&- \sum_{a,b} Q^{t-1}(a) v^t(b) \frac{Q^{t-1}(b)}{\sum_c Q^{t-1}(c)}) \\
&= 2\Delta^2|A| + \sum_a Q^{t-1}(a)^2 + 2(\sum_a Q^{t-1}(a) v^t(a) \\
&- \sum_b v^t(b) Q^{t-1}(b) \sum_c \sigma_t(c)) \\
&= 2\Delta^2|A| + \sum_a Q^{t-1}(a)^2
\end{aligned}
$$

$Q^0(a) = 0$ for all a, so by induction $(\max_a Q^t(a))^2 \leq 2T|A|\Delta^2$ which gives $Q^T(a) \leq \Delta\sqrt{2|A|T}$ □

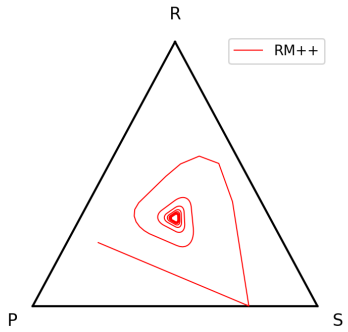


Figure 6: LONR-V policy with RM++ for Rock-Paper-Scissors

C.1 Empirical results for RM++

Figure 6 shows that the last iterate of RM++ converges to the equilibrium of rock-paper scissors. Similar results, not shown, hold for matching pennies. Prior work as shown that both RM and RM+ diverge in these games in terms of the last iterate (although they converge on average). We also tested RM++ in Soccer, and as the no-regret algorithm for CFR in Kuhn poker and for LONR in Grid World. In all cases we achieved last iterate convergence.

C.2 Last Iterate Convergence of Value Estimates

Our empirical convergence results show last iterate convergence for policies. However, our theoretical results were about the convergence of the Q-value estimates. At first glance this may appear an oversight, but a simple argument shows that last iterate convergence of the policies implies last iterate convergence of the Q-value estimates. In particular, last iterate converenge means that $|\pi_k \cdot \underline{Q}_k(s) - \max_a \underline{Q}_k(s,a)| \leq \rho_k$, with $\lim_{k\to\infty} \rho_k = 0$. By Theorem 4.5, $\lim_{k\to\infty} \underline{Q}_k = Q^*$. Combining these shows that the π_k are converging to π^*, which implies convergence of the Q-values.

D EXPERIMENTAL DETAILS

D.1 LONR pseudocode

In the following pseudocode, N is the total number of agents, n is current agent, and S and s represent the total states and current state respectively. $A_n(s)$ denotes the set of actions for player n in state

s. $A_{-n}(s)$ denotes the set of actions of all other agents excluding agent n in state s. a refers to action of the current agent n when unspecified. The policy update uses any no-regret algorithm. The update for Regret Matching++ is shown here.

Algorithm 1 LONR and Updates

procedure LONR-V(T, N, S, A_n) ▷ Value iteration
  $\forall\, n \in N, s \in S, a_n \in A_n(s)$:
    $Q_0(n, s, a_n) \leftarrow 0$, $\pi_0(n, s, a_n) \leftarrow 0$
    RegretSums$(n, s, a_n) \leftarrow 0$, PolicySums$(n, s, a_n) \leftarrow 0$

  for t from 0 to T **do**
    $\forall n \in N, s \in S$:
      Q-Update(n, s, t)

    $\forall n \in N, s \in S$:
      Policy-Update(n, s, t)

procedure Q-Update(n, s, t) ▷ Update Q-Values
  for each action $a_n \in A_n(s)$ **do**
    $successors = getSuccessorStatesAndTransitionProbs(n, s, a_n$,
    $ActionValue \leftarrow 0$
    for s', $transProb$, $reward$ in $successors$ **do**
      $nextStateValue \leftarrow \sum_{a'_n} Q_t(n, s', a'_n) \times \pi_t(n, s', a'_n)$
      $ActionValue \leftarrow ActionValue + transProb \cdot (reward + \gamma \cdot nextStateValue)$
    $Q_{t+1}(n, s, a_n) \leftarrow ActionValue$

procedure Policy-Update(n, s, t) ▷ Regret Matching++
  $ExpectedValue = \sum_{a_n} Q_{t+1}(n, s, a_n) \times \pi_t(n, s, a_n)$

  for $a_n \in A_n(s)$ **do** ▷ RM++ Update Rule
    $immediateRegret \leftarrow max(0, Q_{t+1}(n, s, a_n) - ExpectedValue)$
    RegretSums$(n, s, a_n) \leftarrow$ RegretSums$(n, s, a_n) + immediateRegret$

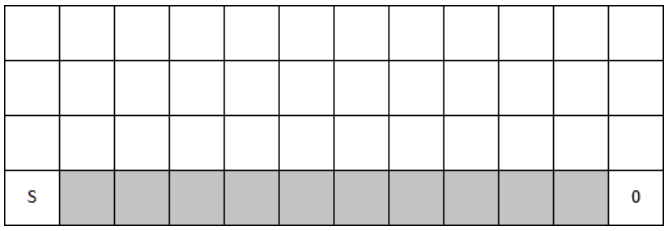
  $totalRegretSum = \sum_i$RegretSums(n, s, i)

  for $a_n \in A_n(s)$ **do** ▷ Update Policy
    if $totalRegretSum > 0$ **then**
      $\pi_{t+1}(n, s, a_n) = \frac{RegretSums(n, s, a_n)}{totalRegretSum}$
    else
      $\pi_{t+1}(n, s, a_n) = \frac{1}{|A_n(s)|}$

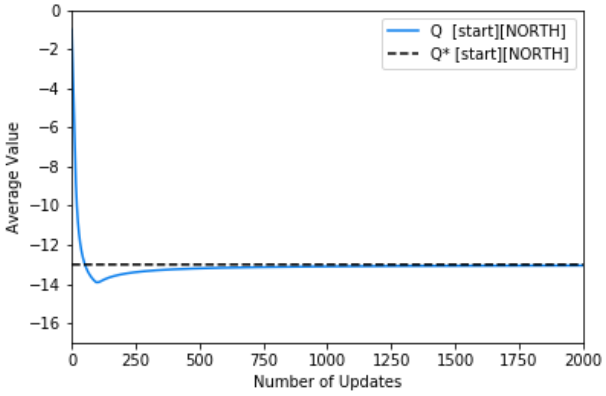
    PolicySums$(n, s, a_n) \leftarrow$ PolicySums$(n, s, a_n) + \pi_{t+1}(n, s, a_n)$

D.2 LONR on GridWorld

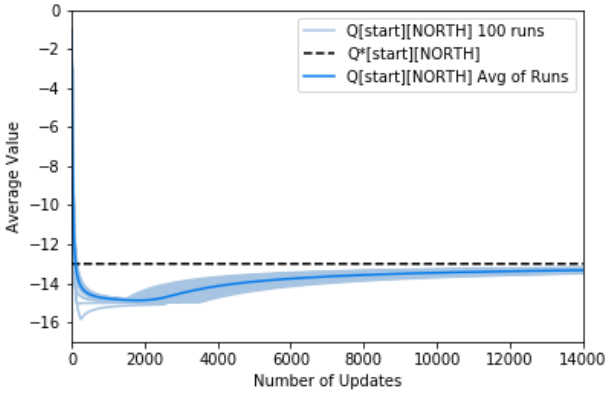
This task is a simple deterministic grid world MDP, in particular the cliff walking task used by [49], illustrated in Figure 7a. As moves have a living cost of 1, we use $\gamma = 1$ (The optimal value from S is therefore -13.) Because of the possibility of revisiting states, and receiving rewards/costs from non-terminals, CFR is not immediately applicable. Figure 7b and Figure 7c shows the results for both LONR-V and LONR-A in terms of the Q-value for the start state's optimal action of North. LONR-V is deterministic with a single agent, so the plot represents a single run. For LONR-A we plot the results of 100 runs and their average. In both cases, we see convergence.



(a) GridWorld: the agent's goal is to move from S to the 0 terminal. Each move has a cost of 1 and the shaded states are terminals with a -100 payoff. This particular grid world is the cliff walking task use by Van Seijen et al. [49] to evaluate Expected SARSA.



(b) LONR-V on GridWorld



(c) LONR-A on GridWorld

Figure 7: Tasks

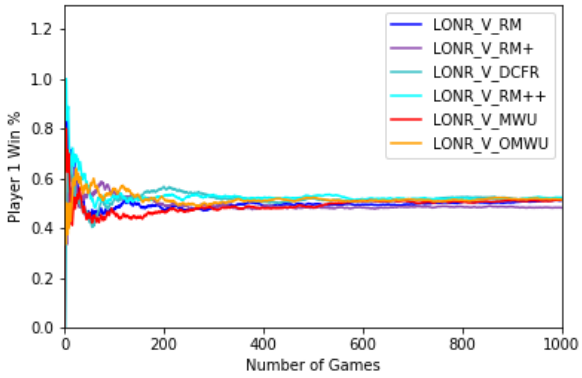


Figure 8: Soccer Game

D.3 Soccer Game

Here we analyze a simplified version of soccer, originally introduced in [38], and subsequently widely used as an early benchmark game for multiagent Markov Games. Our implementation differs slightly in size, but maintains the general rules of the original game.

The soccer game is a two player, grid-style version based on real-life soccer. The size of the grid (field) is 2x4, where the first and last columns are the areas where each player can score. The 2x2 grid between the goal zones are cells in which the players can move. The game begins with each player set to a position on the grid, where one player has control of the ball. At each step of the game, the players each take an action, which are then executed jointly. The defending player is capable of stealing the ball by landing in the same cell as the player with the ball. If the player controlling the ball enters either of the goal cells, they receive 100 points and the other player receives -100 points, thus the game is zero-sum. For additional complexity, the order in which the actions are processed each iteration is randomized.

We run 2 agents against each other with LONR with each regret minimizer for 1000 games (a game runs until a player scores). We discount with $\gamma = 0.9$ to induce agents to score quickly. Before each new game, the position of the players (restricted to non-goal areas) is randomized, as well as who has initial control of the ball. The players then play 1000 games (with initial conditions again randomized) against each other using their learned policies (in this case, the average policy after training.) Figure 8 shows the results of the trials. Each regret minimizer shows signs of convergence in self-play, as indicated by neither player dominating the other (each ends in the average as ties.)

REFERENCES

[1] Jacob Abernethy, Kevin A Lai, and Andre Wibisono. 2019. Last-iterate convergence rates for min-max optimization. *arXiv preprint arXiv:1906.02027* (2019).

[2] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. 2002. The nonstochastic multiarmed bandit problem. *SIAM journal on computing* 32, 1 (2002), 48–77.

[3] James P Bailey, Gauthier Gidel, and Georgios Piliouras. 2019. Finite Regret and Cycles with Fixed Step-Size via Alternating Gradient Descent-Ascent. *arXiv preprint arXiv:1907.04392* (2019).

[4] James P Bailey and Georgios Piliouras. 2018. Multiplicative Weights Update in Zero-Sum Games. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. ACM, 321–338.

[5] Bikramjit Banerjee and Jing Peng. 2005. Efficient no-regret multiagent learning. In *AAAI*. 41–46.

[6] Marc G Bellemare, Georg Ostrovski, Arthur Guez, Philip S Thomas, and Rémi Munos. 2016. Increasing the Action Gap: New Operators for Reinforcement Learning.. In *AAAI*. 1476–1483.

[7] Dimitri Bertsekas. 1982. Distributed dynamic programming. *IEEE transactions on Automatic Control* 27, 3 (1982), 610–616.

[8] Dimitri P. Bertsekas and John Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena Scientifit.

[9] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. 2015. Heads-up limit hold'em poker is solved. *Science* 347, 6218 (2015), 145–149.

[10] Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. 2018. Deep Counterfactual Regret Minimization. *arXiv preprint arXiv:1811.00164* (2018).

[11] Noam Brown and Tuomas Sandholm. 2019. Solving imperfect-information games via discounted regret minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1829–1836.

[12] Noam Brown and Tuomas Sandholm. 2019. Superhuman AI for multiplayer poker. *Science* 365, 6456 (2019), 885–890.

[13] Yun Kuen Cheung and Georgios Piliouras. 2019. Vortices Instead of Equilibria in MinMax Optimization: Chaos and Butterfly Effects of Online Learning in Zero-Sum Games. *arXiv preprint arXiv:1905.08396* (2019).

[14] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. 2017. Training GANs with Optimism. *arXiv preprint arXiv:1711.00141* (2017).

[15] Constantinos Daskalakis and Ioannis Panageas. 2018. Last-Iterate Convergence: Zero-Sum Games and Constrained Min-Max Optimization. *arXiv preprint arXiv:1807.04252* (2018).

[16] Nasrollah Etemadi. 2006. Convergence of weighted averages of random variables revisited. *Proc. Amer. Math. Soc.* 134, 9 (2006), 2739–2744.

[17] Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. 2005. Experts in a Markov decision process. In *Advances in neural information processing systems*. 401–408.

[18] Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. 2009. Online Markov decision processes. *Mathematics of Operations Research* 34, 3 (2009), 726–736.

[19] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. 2002. PAC bounds for multi-armed bandit and Markov decision processes. In *International Conference on Computational Learning Theory*. Springer, 255–270.

[20] Gabriele Farina, Christian Kroer, Noam Brown, and Tuomas Sandholm. 2019. Stable-Predictive Optimistic Counterfactual Regret Minimization. *arXiv preprint arXiv:1902.04982* (2019).

[21] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. 2018. Composability of Regret Minimizers. *arXiv preprint arXiv:1811.02540* (2018).

[22] Richard G Gibson, Marc Lanctot, Neil Burch, Duane Szafron, and Michael Bowling. 2012. Generalized Sampling and Variance in Counterfactual Regret Minimization.. In *AAAI*.

[23] Eyal Gofer and Yishay Mansour. 2016. Lower bounds on individual sequence regret. *Machine Learning* 103, 1 (2016), 1–26.

[24] David Gondek, Amy Greenwald, and Keith Hall. 2004. QNR-Learning in Markov Games. (2004).

[25] Amy Greenwald, Keith Hall, and Roberto Serrano. 2003. Correlated Q-learning. In *ICML*, Vol. 3. 242–249.

[26] Amy Greenwald and Amir Jafari. 2003. A general class of no-regret learning algorithms and game-theoretic equilibria. In *Learning Theory and Kernel Machines*. Springer, 2–12.

[27] Sergiu Hart and Andreu Mas-Colell. 2000. A simple adaptive procedure leading to correlated equilibrium. *Econometrica* 68, 5 (2000), 1127–1150.

[28] Johannes Heinrich and David Silver. 2016. Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121* (2016).

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

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

[31] Peter H Jin, Sergey Levine, and Kurt Keutzer. 2017. Regret Minimization for Partially Observable Deep Reinforcement Learning. *arXiv preprint arXiv:1710.11424* (2017).

[32] Michael Johanson, Nolan Bard, Marc Lanctot, Richard Gibson, and Michael Bowling. 2012. Efficient Nash equilibrium approximation through Monte Carlo counterfactual regret minimization. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 837–846.

[33] Michael J Kearns and Satinder P Singh. 1999. Finite-sample convergence rates for Q-learning and indirect algorithms. In *Advances in neural information processing systems*. 996–1002.

[34] Vojtěch Kovařík and Viliam Lisỳ. 2018. Analysis of hannan consistent selection for monte carlo tree search in simultaneous move games. *arXiv preprint arXiv:1804.09045* (2018).

[35] Marc Lanctot, Richard Gibson, Neil Burch, Martin Zinkevich, and Michael Bowling. 2012. No-regret learning in extensive-form games with imperfect recall.

arXiv preprint arXiv:1205.0622 (2012).

[36] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. 2009. Monte Carlo sampling for regret minimization in extensive games. In *Advances in neural information processing systems*. 1078–1086.

[37] Hui Li, Kailiang Hu, Zhibang Ge, Tao Jiang, Yuan Qi, and Le Song. 2018. Double Neural Counterfactual Regret Minimization. *arXiv preprint arXiv:1812.10607* (2018).

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

[39] Yao Ma, Hao Zhang, and Masashi Sugiyama. 2015. Online Markov decision processes with policy iteration. *arXiv preprint arXiv:1510.04454* (2015).

[40] Shie Mannor and Nahum Shimkin. 2003. The empirical Bayes envelope and regret minimization in competitive Markov decision processes. *Mathematics of Operations Research* 28, 2 (2003), 327–345.

[41] Panayotis Mertikopoulos, Christos Papadimitriou, and Georgios Piliouras. 2018. Cycles in adversarial regularized learning. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2703–2717.

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

[43] Gergely Neu, Anders Jonsson, and Vicenç Gómez. 2017. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798* (2017).

[44] Goran Radanovic, Rati Devidze, David Parkes, and Adish Singla. 2019. Learning to collaborate in markov decision processes. *arXiv preprint arXiv:1901.08029* (2019).

[45] Sriram Srinivasan, Marc Lanctot, Vinicius Zambaldi, Julien Pérolat, Karl Tuyls, Rémi Munos, and Michael Bowling. 2018. Actor-critic policy optimization in partially observable multiagent environments. In *Advances in Neural Information Processing Systems*. 3422–3435.

[46] Oskari Tammelin. 2014. Solving large imperfect information games using CFR+. *arXiv preprint arXiv:1407.5042* (2014).

[47] Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. 2015. Solving Heads-Up Limit Texas Hold'em.. In *IJCAI*. 645–652.

[48] Gerald Tesauro and Jeffrey O Kephart. 2002. Pricing in agent economies using multi-agent Q-learning. *Autonomous Agents and Multi-Agent Systems* 5, 3 (2002), 289–304.

[49] Harm Van Seijen, Hado Van Hasselt, Shimon Whiteson, and Marco Wiering. 2009. A theoretical and empirical analysis of Expected Sarsa. In *Adaptive Dynamic Programming and Reinforcement Learning, 2009. ADPRL'09. IEEE Symposium on*. IEEE, 177–184.

[50] Kevin Waugh, Dustin Morrill, James Andrew Bagnell, and Michael Bowling. 2015. Solving Games with Functional Regret Estimation.. In *AAAI*, Vol. 15. 2138–2144.

[51] Yasin Abbasi Yadkori, Peter L Bartlett, Varun Kanade, Yevgeny Seldin, and Csaba Szepesvári. 2013. Online learning in Markov decision processes with adversarially chosen transition probability distributions. In *Advances in neural information processing systems*. 2508–2516.

[52] Jia Yuan Yu and Shie Mannor. 2009. Online learning in Markov decision processes with arbitrarily changing rewards and transitions. In *2009 International Conference on Game Theory for Networks*. IEEE, 314–322.

[53] Jia Yuan Yu, Shie Mannor, and Nahum Shimkin. 2009. Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research* 34, 3 (2009), 737–757.

[54] Martin Zinkevich, Amy Greenwald, and Michael L Littman. 2006. Cyclic equilibria in Markov games. In *Advances in Neural Information Processing Systems*. 1641–1648.

[55] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. 2008. Regret minimization in games with incomplete information. In *Advances in neural information processing systems*. 1729–1736.