REINFORCEMENT WITH ITERATIVE PUNISHMENT

JEFFREY A. BARRETT AND NATHAN GABRIEL

ABSTRACT. We consider the efficacy of various forms of reinforcement learning with punishment in evolving linguistic conventions in the context of Lewis-Skyrms signaling games. We show that the learning strategy of reinforcement with iterative punishment is highly effective at evolving optimal conventions in even complex signaling games. It is also robust and can be easily extended to a self-tuning variety of reinforcement learning. We briefly discuss some of the virtues of reinforcement with iterative punishment and how it may be related to learning in nature.

Keywords: reinforcement learning, Lewis-Skyrms signaling games, evolution of language

1. INTRODUCTION

We are concerned here with reinforcement learning with punishment in the context of Lewis-Skyrms signaling games.[1] In such games, punishment often contributes to both the speed and accuracy of reinforcement learning.[2]

It has been shown that a signaling game with two states, two signals, and two acts will converge to optimal signaling on simple reinforcement learning with no punishment if nature is unbiased.[3] But if one has more than two states, signals, and acts or if nature is biased, then the players often get stuck in a suboptimal pooling equilibrium. This problem becomes more pronounced in more complex games with more states, signals, and acts.[4]

Supplementing reinforcement on success with punishment on failure often facilitates the evolution of optimal player dispositions, but there is a tuning problem. If one punishes too much relative to the level of reinforcement on success, it makes it difficult for the players to learn anything at all. And if one punishes too little,

[1] David Lewis (1969) first presented this sort of game in the context of classical game theory as a way of studying how conventions might be established. Skyrms (2006, 2010) translated Lewis' signaling games into the context of evolutionary game theory. See Herrnstein (1970), Roth and Erev (1995), and Erev and Roth (1998) for a description of reinforcement learning and how it maybe used to characterize human learning in a number of salient contexts.

[2] See Barrett and Zollman (2009) for a discussion of the role of punishment in reinforcement learning.

[3] See Argiento, Pemantle, Skyrms, and Volkov (2009) for this proof. The proof is highly nontrivial for even this simple case. It is because we have very few analytic results for more complex signaling games, under even simple reinforcement, that they must be investigated primarily by simulation. See Hu, Skyrms, and Tarrès and Huttegger, Skyrms, Tarrès, and Wagner (2014) for a description of one general result.

[4] See Barrett (2006) for an early description of both of these problems and Hofbauer and Huttegger (2008) for further discussions of the latter.

one does not do much to solve the suboptimal pooling equilibrium problem. What counts as too much and too little here depends on the complexity of the game. This makes it difficult to specify a single dynamics that does well in a broad assortment of games.[5]

We will begin by considering a $4 \times 4 \times 4$ signaling game with basic reinforcement learning, then consider simple reinforcement with punishment, reinforcement with iterative punishment, and self-tuning reinforcement with iterative punishment. We will show how the latter form of reinforcement with punishment provides robust learning strategies that are effective in very different signaling games. How the virtues of reinforcement with iterative punishment pave the way for a self-tuning form of reinforcement learning will be particularly salient. We conclude by briefly discussing how reinforcement with iterative punishment may be related to learning in natural contexts.

2. SIMPLE REINFORCEMENT

A Lewis-Skyrms signaling game is a common interest evolutionary game with one sender and one receiver. On a play of an $n \times n \times n$ signaling game, the sender sees which of n states of nature obtains. She then sends one of n possible signals. The receiver cannot see the state of nature but can see the sender's signal. When she sees the signal on a play of the game, she performs one of n possible acts. Each act is appropriate to precisely one state of nature. If the receiver's act matches the current state of nature, then the play counts as a success; otherwise, it counts as a failure.

The two players update their conditional dispositions based on the outcome of each play (the sender updates her dispositions to signal conditional on the state and the receiver updates his dispositions to act conditional on the signal). Whether the agents are readily able to evolve optimal dispositions where each state of nature leads to a particular signal which leads to the act that matches the current state depends on precisely how they update their conditional dispositions given the outcome of each play. It sometimes often depends on luck as well.[6]

Let's consider concretely how this goes for a $4 \times 4 \times 4$ signaling game under basic reinforcement learning (Figure 1). One might think of the players' dispositions and how they are updated in terms of urns with balls. On each play of the game, the current state of nature is determined in a random and unbiased way. The sender has one urn for each possible state of nature $0, 1, 2, 3$. Each of these urns starts with one ball of each possible signal a, b, c, d. To determine the signal, she observes the current state of nature then draws a signal ball at random from the corresponding urn. She sends that signal. The receiver has one urn for each possible signal a, b, c, d. Each of these urns starts with one ball of each possible act $0, 1, 2, 3$, where each state of nature requires the corresponding act for success. The receiver sees the

[5] See Barrett (2006) for an investigation of various punishment strategies and Barrett and Zollman (2009) for a discussion of the role of punishment and forgetting in learning. See Alexander, Skyrms, and Zabell (2012), Barrett, Cochran, Huttegger, and Fujiwara (2017), and Cochran and Barrett (2021a) and (2021b) for discussions of other learning strategies that seek to address the problem of suboptimal pooling equilibria.

[6] On the games and learning dynamics we will consider, the players will typically evolve *nearly* optimal dispositions on a successful run. Their dispositions will correspond to a bijective map from states to acts, but even when they are successful, their behavior will be somewhat noisy and hence not always respect the bijective map.

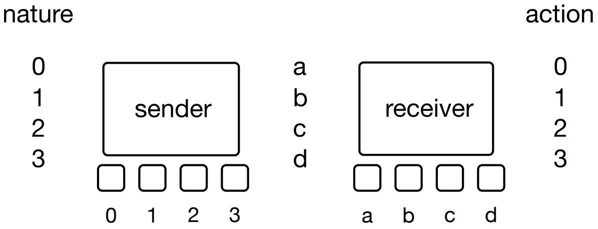

Figure 1. $4 \times 4 \times 4$ signaling game

signal then draws an act ball at random from the corresponding urn. She performs the act. If the receiver's act matches the current state, then each agent returns their ball to the urn from which it was drawn and adds a ball of the same type to the urn; otherwise, they simply replace return the ball that they drew.

The question as to how well a learning dynamics does on a particular type of problem is an empirical one. On simulation, when the modeled sender and receiver update their conditional dispositions under this dynamics, they start off randomly signaling and randomly acting. The signals initially have no meanings and there is no pattern to the receiver's action. But, as they learn from experience, about 0.797 of the time the players dispositions evolve to exhibit a cumulative success rate exceeding 0.8 with 10^6 plays per run (with 1000 runs). If they reach this level of success, then the run did not get stuck in a suboptimal pooling equilibrium as the most successful pooling equilibria for this game have cumulative success rates of 0.75. More generally, if a game evolves to do better than its best suboptimal pooling equilibria, then it tends to do monotonically better over time thereafter approaching an optimal signaling system. For this reason, the level of success that matters in each of the games we consider is the success rate of the most successful suboptimal equilibria. As for the specific case at hand, the players usually evolve a nearly optimal signaling system under simple reinforcement learning in the $4 \times 4 \times 4$ game.

Basic reinforcement learning does not do as well on more complicated signaling games. With 10^7 plays per run, the $8 \times 8 \times 8$ game exhibits a cumulative success rate over 0.875 just under half of the time 0.426. The higher cutoff here is because the most successful suboptimal pooling equilibria are better here. This is part of the reason the success rate is lower than for the $4 \times 4 \times 4$ game. But the game is also harder. Note that we are running the $8 \times 8 \times 8$ games ten times as long and still getting a significantly lower cumulative success rate.

While there is no entirely straightforward way of comparing the difficulty of evolving optimal dispositions for more and less complex games or of comparing the relative efficacy of different learning strategies, we will aim to give a sense of the relative difficulties and efficacies by describing what happens on simulation under an assortment of different game parameters. There are a few simplifying conventions that will allow for more accurate comparisons across different games. The first of these has to do with how we track success.

The most common way to measure evolutionary success in the signaling game literature is to track cumulative success over a fixed number of plays. The statistics reported above are for that.[7] An advantage of this measure is that one gets a rough sense of both how reliable the system ultimately is and how quickly it got there. In contrast, one might track the success rate over just the last few plays of the game on a run. This measure provides a better sense of the properties of the final evolved system. For the $8 \times 8 \times 8$ game we just considered, 0.592 of the runs exhibit a success rate over 0.875 when just the last 10^4 plays of each run are considered. Given the focus of the present investigation, we will measure success in this way in what follows.

When the agents do not evolve toward an optimal signaling system, they get stuck in a suboptimal pooling equilibrium. As a concrete example of what this looks like, the suboptimal equilibrium illustrated in Figure 2 sometimes arises under simple reinforcement learning in the $3 \times 3 \times 3$ game even with unbiased nature.

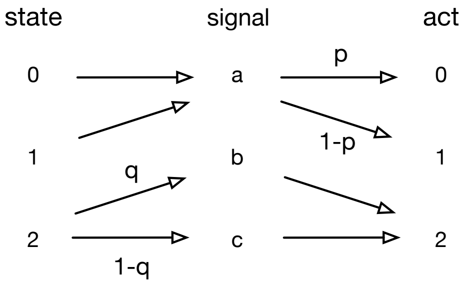

FIGURE 2. $3 \times 3 \times 3$ signaling game

An optimal signaling system requires that one have a bijection between states of nature and signals and between signals and acts and hence, taken together, between states and signals. But here the sender uses a to represent both states 0 and 1 and the receiver randomly does act 0 and 1 with probabilities p and $1-p$ when he sees an a signal. And the sender randomly sends b and c with probabilities q and $1-q$ and the receiver always does act 2 when he gets either of these two signals. If nature is unbiased, the mean success rate will be 2/3. And since unilaterally changing the value of either p or q does nothing to improve the agents' fortunes, their mixed strategies from an equilibrium under the dynamics.

Such suboptimal equilibria are increasingly common the more complex the game. On 10^6 plays per run the $16 \times 16 \times 16$ game has an end-of-run success rate on simple reinforcement learning better than 0.9375 on 0.169 of the runs. The $32 \times 32 \times 32$ game does better than its most successful pooling equilibria at 0.969 on 0.003 of the runs. With ten times the number of plays per run 10^7, the $32 \times 32 \times 32$ game meets this cutoff a slightly more often on 0.036 of the runs. But on even 2×10^7 plays per run, $64 \times 64 \times 64$ game was never found to meet the corresponding cutoff at 0.985 on 1000 runs.

A natural question is whether one might do better on such complex games under a different learning dynamics. In one sense of better this question is easy to answer. There are some learning dynamics that will with probability one eventually evolve

[7] See Barrett (2006) for further examples of this measure for various signaling games on simple reinforcement learning and reinforcement with punishment.

a strictly optimal signaling system for any finite signaling game. Consider win-stay/lose-randomize. Here the sender adopts an initial bijective map from states to signals and the receiver adopts an initial bijective map from signals to acts. On each play of the game, the sender and receiver stay with their present map unless the play fails in which case, each adopts a new bijective map independently and at random. Since there is a fixed positive probability on each play that they will happen to pick a pair of maps that fit with each other to produce a bijection between states and acts, this dynamics will eventually yield a optimal signaling system with probability one. But given the number of possible strategies in a moderately complex signaling game, it could take a cosmological long time to do so. In terms of speed this often a terrible dynamics. We will consider an illustrative example of this in the next section.

We turn now to consider how punishment might improve both the speed and accuracy of reinforcement learning in the context of signaling games.[8] This requires some care. While one can sometimes get a very fast and accurate learning dynamics by tuning the dynamics' parameters to the particular problem at hand, one should ultimately want a learning dynamics that does well in a broad range of contexts. We will aim to make progress on characterizing such a robust learning dynamics.

3. SIMPLE REINFORCEMENT WITH PUNISHMENT

A signaling game under simple reinforcement with punishment works very much like a signaling game under reinforcement except that one may also punish by removing balls from urns on failure. In the $4 \times 4 \times 4$ signaling game under $(+i, -j)$ reinforcement with punishment, one adds i balls on success and may remove up to j balls on failure. More precisely, if the receiver's act matches the current state, then each agent returns their ball to the urn from which it was drawn and adds i balls of that type to the urn; otherwise, they return the ball to the urn from which it was drawn and remove j balls of that type from that urn subject to the condition that at least one ball of the type remains in the urn. Hence, if there are j or fewer balls of the given type, all are removed but one. The aim is to keep the same number of types in each urn by not allowing any type to go extinct.

Optimal signaling often evolves much faster and more reliably under reinforcement with punishment than under reinforcement alone. On just reinforcement learning without punishment, the $8 \times 8 \times 8$ game exceeds the 0.875 final-success-rate cutoff on 0.613 of runs with 10^6 plays per run. Most of the failed runs here reflect suboptimal pooling equilibria. In contrast, $(+1, -1)$ reinforcement with punishment exceeds the 0.875 success rate cutoff on a remarkable 0.945 of runs with the same number of plays per run.

Punishment makes suboptimal pooling less likely by tending to weaken the players' suboptimal dispositions when they use them and fail. But it is also easy to punish too much. The $8 \times 8 \times 8$ game with $(+1, -2)$ learning was never found to exceed the 0.875 cutoff on 1000 runs with 10^6 plays per run. That said, increasing the level of reinforcement a bit preserves success for this particular game with 0.951 of runs meeting the cutoff for $(+2 - 1)$ learning.

[8] See Erev and Roth (1998), Barrett and Zollman (2009), Alexander, Skyrms, and Zabell (2012), Barrett, Cochran, Huttegger, and Fujiwara (2017), and Cochran and Barrett (2021a) and (2021b) for other promising learning strategies.

While $(+1, -1)$ learning works well in the $8 \times 8 \times 8$ game, it is less successful in more complex games. The $16 \times 16 \times 16$ game was never observed to exceed the 0.9375 cutoff on $(+1, -1)$ learning with 10^6 plays per run and 1000 runs. The coordination problem posed by the game is harder here. As a result, that level of punishment is too high relative to the level of reinforcement to allow for the effective evolution of optimal dispositions. Successful dispositions cannot get traction against the high level of punishment here. But the $16 \times 16 \times 16$ game does exceed the cutoff 0.952 of the time on $(+2, -1)$ learning with 10^6 plays per run. Here the reinforcement on success is enough for the players to get traction on optimal dispositions early in the evolution. Indeed, the evolution of near optimal dispositions for these learning parameters is very fast. The cutoff is exceeded on 0.903 of the runs with just 10^5 plays per run.

While $(+2, -1)$ learning works well in these two games it does not work at all in more complex games. The $32 \times 32 \times 32$ game was never found to exceed a 0.969 cut off with this dynamics on 1000 runs with 10^6 plays per run. But this game did very well with $(+3, -1)$ learning. Here it exceeded the cutoff on 0.879 of the runs. Again, one needs sufficient reinforcement relative to punishment to get traction early in the run. And success is highly sensitive to both parameter settings. Raising the level of punishment even slightly can make a striking difference. In the present game, $(+3, -2)$ learning was never observed to succeed.

Similar phenomena are seen for yet more complex games. The $64 \times 64 \times 64$ game was never observed to exceed a 0.984375 cutoff on $(+3, -1)$ learning on 1000 runs with 2×10^6 plays per run. But it was observed to exceed this strict cutoff on 0.604 of the runs with $(+4, -1)$ learning. Indeed, it appears that the only thing limiting its success here was the relatively low number of plays per run given the complexity of the game as $(+4, -1)$ learning did better than the just slightly lower 0.969 cutoff on 0.953 of the runs even with 2×10^6 plays per run.

Determining what is happening in such simulations requires care. The $128 \times 128 \times 128$ game was never observed to exceed a 0.995 cutoff on $(+5, -1)$ learning with 2×10^6 plays per run. But again it appears that the issue here was just one of run length given the high complexity of the game and high cutoff. This is strongly suggested by the fact that 0.780 of the runs exceed the cutoff with 3×10^7 plays per run and 0.830 do with 5×10^7 plays per run for the same reinforcement and punishment parameters.

The recurring theme here is that if the level of punishment is too low given the complexity of the game and the level of reinforcement, then one tends to get stuck in suboptimal pooling equilibria; and if it is too high given the complexity of the game and the level of reinforcement, then one cannot get the traction needed to evolve optimal dispositions. But both of these points also require care.[9]

On reinforcement learning with punishment, it is always logically possible to escape from a suboptimal pooling equilibrium. This might happen if one is lucky enough to fail, and hence weaken the dispositions that led to that failure, whenever one uses suboptimal dispositions. And if one is lucky enough, it is always logically possible to get the traction necessary to evolve optimal dispositions with even a very

[9] A similar point can be made with respect to reinforcement values. While $(+5, -1)$ learning in the $128 \times 128 \times 128$ game met its cutoff in 0.830 of runs (with 5×10^7 plays per run), $(+2, -0.4)$ never met the cutoff in the same game with the same number of plays per run. Just as when punishment is too high one cannot get the traction needed to evolve optimal dispositions, when reinforcement is too low one may not get the traction needed to evolve optimal dispositions.

high level of punishment. A closer look at this second point will help to illustrate the role of tuning in simple reinforcement with punishment.

Consider a learning dynamics with a high level of punishment relative to the level of reinforcement. Regardless of how complex the game may be, given the randomly determined states of nature, it is in principle possible that they get lucky on the random dynamics that determines their actions and always succeed in the early plays of the game. If so, the punishment level does not matter early in the run since the players will evolve nearly optimal dispositions by chance; and it does not matter later in the run either since the players will then have nearly optimal dispositions and hence rarely fail and be punished.

But while it is always *possible* for optimal signaling to evolve with even very high levels of punishment, one should expect simple reinforcement with punishment to be *extremely slow* with high levels of punishment relative to reinforcement, and particularly so for more complex games. In this, simple reinforcement with severe punishment behaves much like the win-stay/lose-randomize strategy discussed earlier.[10]

Given this, one might put the recurring theme in a somewhat more accurate positive form: simple reinforcement with punishment allows for the effective evolution of nearly optimal dispositions in even complex signaling games if one uses *just right* levels of reinforcement and punishment for the complexity of the game.[11] It is worth reflecting on how remarkable this success may be before worrying further over tuning.

Setting aside the potentially unbounded number of mixed strategies available to the two players in a $128 \times 128 \times 128$ game, there are 128^{128} (more than 10^{269}) maps from states of nature to signals and the same number from signals to acts. But the vast majority of these maps are suboptimal. An optimal signaling system requires the sender to have a bijective map from states of nature to signals and the receiver to have a matching bijective map from signals to acts. There are 128! (more than 10^{215}) bijective maps from states of nature to signals for the sender and the same number of bijective maps from signals to acts for the receiver. For a given sender bijective map, there is *only one* receiver map that will allow for optimal signaling. As a consequence, evolving optimal dispositions for such a game would be extraordinarily unlikely on something like win-stay/lose-randomize learning at supercomputing speeds and cosmological time scales even if the players strategies are restricted to just bijective maps.[12] That $(+5-1)$ learning usually succeeds in finding a bijective map for the sender and a matching map for the receiver in just 5×10^7 plays is a nontrivial virtue of simple reinforcement with punishment.

But while simple reinforcement with punishment may be remarkably effective with just-right learning parameters, one should want a *single* learning dynamics

[10] As we will see in the next section, reinforcement with iterative punishment allows for success with much higher levels of punishment since the early evolution is protected from punishment. Indeed, this will prove to be the key to the remarkable effectiveness of reinforcement with iterative punishment.

[11] There is likely a best setting of reward and punishment parameters for a particular game run at a specified length. Further, such best parameters are likely not whole numbers as we have been using here. While we will seek to address the tuning problem in another way here, another approach would be to try to say something about how to best determine these parameter settings either when presented with a problem or, arguably more promising, dynamically while evolving a solution.

[12] The universe is likely younger than $5 \times 10^{17} seconds$.

that does well on a broad assortment of games. To formulate something like this, we will start by considering reinforcement with iterative punishment. We will then reflect on how this dynamics might allow the players to tune how they learn as they learn to meet the demands of the signaling game at hand.

4. REINFORCEMENT WITH ITERATIVE PUNISHMENT

On reinforcement with iterative punishment, one alternates between periods where the learning dynamics is mostly reinforcing and periods where it is mostly punishing. To most clearly illustrate how the dynamics works, we will suppose here that reinforcement phases involve pure reinforcement on success with no punishment on failure and that punishment phases involve pure punishment on failure with no reinforcement on success.

During a reinforcement phase, the players will form dispositions that together map from states of nature to receiver acts. Some parts of the map may be nearly optimal. On these a state of nature will produce the corresponding act with high probability. Other parts may be suboptimal. On these the state may produce the corresponding action, but it does not do so reliably. A subsequent punishment phase will act to reset the dispositions associated with the suboptimal mixed parts of the map, since their lower expected success rate is associated with more expected punishment, without undoing the nearly optimal bijective parts. Then the players will have another chance to get the suboptimal parts that were just erased right when they return to reinforcement. Inasmuch as they need only build the remaining parts of the bijective map on subsequent reinforcements, the players face an ever easier task after each punishment phase and hence stand a yet better chance of getting things right. When successful, they evolve nearly optimal dispositions by assembling the full bijective map in parts.

Tuning still matters for reinforcement with iterated punishment. One needs a sufficient level of reinforcement and needs to run it sufficiently long to get strong enough dispositions for the bijective parts of the map that it is unlikely that they will be erased by subsequent punishment. And one needs a sufficient level punishment and needs to run it long enough that it is likely that the parts of the map that are not bijective are erased. But one does not want the punishment to be so high or so long as to erase the nearly optimal bijective parts of the map that were hard-won on previous reinforcements.

Tuning in reinforcement with iterative punishment and in simple reinforcement with punishment are, however, quite different. In short, reinforcement with iterative punishment is much more robust. This is in part because it allows for a higher level of punishment. And this has to do with the core idea behind the dynamics.

Reinforcement with iterative punishment builds strong dispositions during the reinforcement phase, then uses punishment to erase the suboptimal parts of the map. Since the dispositions associated with the optimal parts of the bijective map are firmly established during reinforcement, even severe subsequent punishment will tend to leave these successful dispositions intact. So the players learn freely and without constraint during the reinforcement phase, and while they may evolve bad habits along the way, these are easily eliminated by severe subsequent punishment. In contrast, on simple reinforcement with punishment, punishment is always present and hence occurs before successful dispositions have been firmly established. As a result, weak dispositions that might ultimately form part of a successful bijective map if they were allowed to strengthen may be accidentally erased before they can prove their worth if the learning parameters are not just right. A high level of punishment makes it difficult for the players to build any dispositions at all. And a low level of punishment does not address the suboptimal pooling problem.

We will consider a few concrete examples of how reinforcement with iterative punishment works. While the reinforcement phases might in principle involve some degree of punishment and the punishment phases might involve some degree of reinforcement, we will consider here only dynamics with periods of pure reinforcement punctuated by periods of pure punishment. Using square brackets to represent reinforcement with iterative punishment, $[+i, -j]$ learning alternates between periods of pure reinforcement of magnitude i and periods of pure punishment of magnitude j. The length of these periods will matter as well. We will note that in each case.

Consider the $32 \times 32 \times 32$ game under $[+2, -4]$ reinforcement with iterative punishment. The first thing to note is that under $(+2, -4)$ *simple* reinforcement with punishment, this level of punishment is too high for the players to get traction on a successful set of dispositions and they get nowhere. Indeed, as we saw earlier, even $(+2, -1)$ simple reinforcement with punishment involves too much punishment for this complex a game. The players were never found to exceed the 0.969 success-rate cutoff. But $[+2, -4]$ reinforcement with *iterative* punishment may do very well on this game depending on the length of the reinforcement and punishment phases. With a fixed run length of 10^6 plays per run and two iterations (one reinforces for a quarter of the run, then punishes for a quarter of the run, then repeats the reinforcement and punishment), the players exceed this high success-rate cutoff 0.252 of the time. With the same run length and four iterations, they exceed the cutoff 0.409 of the time. With eight iterations 0.645. And with sixteen iterations 0.918. The players are able to evolve nearly optimal dispositions very quickly on sixteen reinforcement/punishment blocks because the high punishment level erases any evolved suboptimal dispositions, and the high level of punishment does not undermine their optimal dispositions because it only kicks in after those dispositions are firmly established. This illustrates the radically different properties of reinforcement with iterative punishment. It also shows how the players build the full bijective map in parts on iterated punishment.

Importantly, the players do not continue to do better with more iterations with the fixed total run length. Alternating quickly between pure $+2$ reinforcement and pure -4 punishment closely approximates $(+2, -4)$ simple reinforcement with punishment where the players are not at all effective in establishing successful dispositions.

In the same setup, the $32 \times 32 \times 32$ game evolves more slowly on $[+1, -2]$ than on $[+2, -4]$ reinforcement with iterative punishment. With a fixed run length of 10^6 plays per run and two iterations, the players exceed the 0.969 success-rate cutoff 0.028 of the time. With the same run length and four iterations, they exceed the cutoff 0.050 of the time. With eight iterations 0.103. And with sixteen iterations 0.203. Part of the problem here is that the level of punishment is too low to fully erase the bad habits formed over a long period of reinforcement. Another is that we are not allowing the dynamics enough chances to get things right.

The $32 \times 32 \times 32$ game on $[+1, -2]$ reinforcement with iterative punishment provides a good example of the core idea behind reinforcement with iterative punishment and why it is robust. Consider this dynamics with a fixed reinforcement/punishment block length of 1.25×10^5 plays. Eight such blocks exceeds the high 0.969 success-rate cutoff 0.008 of the time. Sixteen blocks exceeds the cutoff 0.323 of the time. Twenty-four blocks exceeds it 0.906 of the time. And thirty-two blocks 0.957 of the time. Here the players are observed to improve monotonically with more reinforcement/punishment blocks. In each block sequence, the reinforcement phase builds potentially optimal dispositions and the punishment phase tends to erase suboptimal dispositions. The block length here is long enough to forge reasonably strong dispositions but short enough that the punishment phases accomplish significant cleanup. And repeating the process allows for incremental improvement.

Among the $[+i, -j]$ parameters that were observed to do well on 10^6 plays per run are $[+2, -3]$ (run success rate 0.839 with sixteen blocks), $[+2, -6]$ (run success rate 0.837 with eight blocks), and $[+2, -10]$ (run success rate 0.837 with four blocks). This illustrates the robustness of the dynamics under very different $[+i, -j]$ parameters. It also illustrates how the dynamics with fixed $[+i, -j]$ parameters may be tuned to the complexity of the game at hand by varying block length alone. There is further evidence of this latter sort of robustness.

Under reinforcement with iterative punishment, precisely the same $[+i, -j]$ parameters, with a fixed block length, work well for games of very different complexity. On $[+4, -12]$ learning and a block length of 10^6, the $32 \times 32 \times 32$ game meets its cutoff on 0.623 of the runs after 3×10^7 plays per run; the $64 \times 64 \times 64$ game meets its cutoff on 0.465 of the runs after 5×10^7 plays per run;[13] the $128 \times 128 \times 128$ game meets its cutoff on 0.903 of the runs after 5×10^7 plays per run.[14] And in all of the runs for each of these three games (1000 runs each) the final success rate was observed to be better than 0.9. This last point indicates that the dynamics avoids both the suboptimal-pooling and the traction-under-high-punishment problems here. In doing so, it exhibits a radically different behavior from simple reinforcement with punishment, where the $(+i, -j)$ parameters must be carefully tuned to the complexity of the signaling game.

So how might one choose block lengths that allow for monotonic improvement toward an optimal signaling system on iteration? Given the structure of reinforcement with iterated punishment, this can be done dynamically. What one wants is to run the reinforcement phase until one sees diminishing returns, then run the punishment phase until one sees diminishing returns, then repeat. Diminishing returns in the case of reinforcement means that one's local success rate is no longer improving significantly on repeated play. When this happens one typically has some firmly established optimal dispositions with some mixed suboptimal dispositions. For punishment it means that one's local success rate is beginning to decline significantly. When this happens, one has erased the suboptimal dispositions and is

[13] $[+4, -12]$ learning performs yet better with a block length that is tuned to the complexity of the game. On $[+4, -12]$ learning with 3×10^7 plays per run, the $64 \times 64 \times 64$ game meets its cutoff on 0.694 and 0.912 of the runs (1000 runs each) with block lengths of 5×10^5 and 2.5×10^5 respectively.

[14] As in most of the paper, cutoffs here are $n - 1/n$. Hence the cutoff for the $32 \times 32 \times 32$ game is 0.969, for the $64 \times 64 \times 64$ game is 0.984, and for the $128 \times 128 \times 128$ game is 0.992.

beginning to erase optimal dispositions. Given the robustness of reinforcement with iterative punishment, one has considerable leeway in making these determinations.

Inasmuch as such determinations may be made during play, we have the framework for self-tuning reinforcement with iterative punishment. On this dynamics, one chooses a reinforcement level and a relatively severe punishment level, then one runs reinforcement until one sees diminishing returns, then runs punishment until one sees diminishing returns, then repeats. While there remains a significant tuning issue in the choice of reinforcement and punishment levels, the dynamical determination of block length provides a significant degree of self-tuning.

There is more to say regarding how one might best detect diminishing returns to facilitate the rabid evolution of optimal dispositions. One should also want to consider how to choose reinforcement and punishment levels dynamically to this end. These are topics for further research.

5. DISCUSSION

While basic *reinforcement* learning allows for the evolution of optimal dispositions on simple signaling games, it often leads to suboptimal pooling equilibria, particularly in more complex games. *Simple reinforcement with punishment* helps to prevent suboptimal pooling, but it encounters a tuning problem. If the punishment level is too low, it does little to prevent suboptimal pooling; and if the punishment level is too high, successful dispositions fail to evolve at all.

In contrast, *reinforcement with iterated punishment* is tolerant of high levels of punishment. Here reinforcement establishes a mixed set of dispositions, and subsequent punishment tends to erase those dispositions that contribute to the suboptimal parts of the map from states to signals to acts. The aspect of signaling games that makes this dynamics effective is that an optimal bijective map can be built incrementally. The dynamics will move toward optimal dispositions if the reinforcement phase is long enough to accumulate firmly established dispositions and the punishment phase is long enough to erase the suboptimal dispositions but short enough not to do significant damage to the bijective parts of the map.

Understanding how reinforcement with iterative punishment works suggests *self-tuning reinforcement with iterated punishment.* Here one reinforces until progress begins to stall, punishes until the overall success of the system begins to decline, then repeats. While this still requires one to choose initial reinforcement and punishment levels, the players themselves determine the length of reinforcement/punishment blocks dynamically and hence have significant control of how much reinforcement and punishment in fact takes place. As a result, this dynamics is self-tuning over a broad class of signaling games.

Reinforcement with iterative punishment is more robust than simple reinforcement with punishment in two closely related ways. First, very different $[+i, -j]$ parameters are found to evolve optimal dispositions in the same game. And second, precisely the same $[+i, -j]$ parameters are found to evolve optimal dispositions in very different games. It is this that allows for the success of self-tuning reinforcement with iterative punishment. This self-tuning form of reinforcement learning clearly deserves further study.

Reinforcement with iterative punishment mirrors a learning structure that is often observed in natural contexts. During reinforcement periods, players learn quickly and establish a rich set of dispositions. Such low-risk free play is often punctuated by the higher-risk application of what the players have learned. On this dynamics, while they may make progress toward an optimal set of dispositions during free play, they may also pick up some bad habits along the way and end up with suboptimal dispositions.

Perhaps as a result of their efficacy, such learning strategies have come to be implemented in traditional pedagogy. Low-risk formative learning for most of a course might act as reinforcement building new dispositions. Formative learning is then punctuated with high-stakes summative assessments that compare the students' dispositions against benchmark standards for the subject matter of the course. Failure in the context of such assessments might act as punishment that weakens those dispositions that do not accord with the standards.

In reinforcement with iterated punishment learning, if higher-risk application tends to eliminate the suboptimal dispositions and if this happens in way that does not undermine the positive fruits of free play, subsequent free play allows for new, potentially optimal, dispositions to be forged in the place of the bad habits that did not stand the test in application.[15]

[15] We would like to thank Brian Skyrms and Calvin Cochran for comments on an earlier draft of this paper.

References

Alexander, J. M., Skyrms, B., & Zabell, S. L. (2012). Inventing new signals. *Dynamic Games and Applications*, 2, 129–145.

Argiento, R., Pemantle, R., Skyrms, B., & Volkov, S. (2009). Learning to signal: analysis of a micro-level reinforcement model. *Stochastic Processes and their Applications*, 119(2), 373–390

Barrett, J. A. (2007a). Dynamic partitioning and the conventionality of kinds. *Philosophy of Science*, 74, 527–546.

Barrett, J. A. (2007b). The evolution of coding in signaling games. *Theory and Decision*, 67(2), 223–237.

Barrett, J. A. (2006). Numerical simulations of the Lewis signaling game: learning strategies, pooling equilibria, and the evolution of grammar. *UC Irvine Institute for Mathematical Behavioral Sciences Technical Report.* https://www.imbs.uci.edu/research/technical.php. Accessed 8 September 2021.

Barrett, J. A. & Zollman, K. (2009). The role of forgetting in the evolution and learning of language. *Journal of Experimental and Theoretical Artificial Intelligence*, 21(4), 293–309.

Barrett, J. A., Cochran, C. T., Huttegger, S., & Fujiwara, N. (2017). Hybrid learning in signaling games. *Journal of Experimental & Theoretical Artificial Intelligence*, 29(5), 1–9.

Cochran, C. T. & Barrett, J. A. (2021a). The efficacy of human learning in Lewis-Skyrms signaling games, draft.

Cochran, C. T. & Barrett, J. A. (2021b). How signaling conventions are established, *Synthese*, https://link.springer.com/article/10.1007/s11229-020-02982-9

Erev, I. & Roth, A. E. (1998). Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria, *American Economic Review*, 88, 848–81.

Herrnstein, R. J. (1970). On the law of effect. *Journal of the Experimental Analysis of Behavior*, 13, 243–266.

Hofbauer, J. & Huttegger, S. (2008). Feasibility of communication in binary signaling games, *Journal of Theoretical Biology*, 254(4), 843–849.

Hu Y., Skyrms, B. Tarrès, P. (2011). Reinforcement learning in signaling games, https://arxiv.org/abs/1103.5818. Accessed 8 September 2021.

Huttegger, S., Skyrms, B., Tarrès, P. & Wagner, E. (2014) Some dynamics of signaling games, *Proceedings of the National Academy of Sciences of the USA*, 111 (Supplement 3), 10873–10880.

Lewis, D. (1969). *Convention.* Harvard University Press.

Roth, A. E. & Erev, I. (1995). Learning in extensive form games: experimental data and simple dynamical models in the intermediate term, *Games and Economic Behavior*, 8, 164–212.

Skyrms, B. (2010). *Signals: evolution, learning, & information.* Oxford University Press.

Skyrms, B. (2006). Signals. *Philosophy of Science*, 75(5), 489–500.