Temporal Difference and Return Optimism in Cooperative Multi-Agent Reinforcement Learning

Mark Rowland
DeepMind

Shayegan Omidshafiei
DeepMind

Daniel Hennes
DeepMind

Will Dabney
DeepMind

Andrew Jaegle
DeepMind

Paul Muller
DeepMind

Julien Pérolat
DeepMind

Karl Tuyls
DeepMind

ABSTRACT

We describe a framework for decentralised cooperative multi-agent reinforcement learning that encompasses existing approaches based on asymmetric updates, as well as methods based on distributional reinforcement learning. Both of these strategies can be interpreted as augmenting single-agent learning algorithms with optimism; the former at the level of temporal difference (TD) errors, and the latter at the level of returns. This perspective allows for the detailed examination of fundamental differences between these two families of methods across a range of environments, and we identify several environment properties that exemplify the differences in performance that may arise in practice. Further, the unifying framework used to describe these algorithms highlights many possible variants of these basic approaches, and we introduce several new families of algorithms that can be seen as interpolating between TD-optimism and return-optimism.

KEYWORDS

Reinforcement learning, distributional reinforcement learning, optimism, game theory, multi-agent learning

1 INTRODUCTION

A key aim of artificial intelligence is to build agents that can cooperate or compete with one another effectively while acting independently, in spite of incomplete information and a dynamically-changing environment. This is the problem addressed by decentralised multi-agent reinforcement learning (MARL), where agents must learn when credit for an outcome is due to their own actions, rather than the actions of other agents or uncontrollable dynamics in the environment [2, 9, 17, 52]. A simple and well-investigated mechanism for solving this problem in cooperative environments is to learn *optimistically*, effectively operating under the assumption that other agents will themselves eventually learn to perform optimally. Under certain conditions, optimistic agents learn more accurate estimates of the value of their actions than risk-neutral (non-optimistic) counterparts [36, 37], suggesting that optimism may be a key part of the solution to this problem.

Many approaches to optimistic learning are based on single-agent algorithms such as Q-learning [53]. Optimism can be implemented at the level of *temporal difference (TD) errors* by modifying the Q-learning rule to put less weight on negative TD errors; this approach is taken by distributed Q-learning [22], lenient Q-learning [6, 35, 36], and hysteretic Q-learning [26, HQL], which are effective in a range of cooperative tasks and have been successfully scaled up in combination with deep learning to large-scale environments [32–34, 54]. However, when the domain itself is stochastic, these modified Q-learning rules can induce misplaced optimism towards uncontrollable environment dynamics, leading to sub-optimal behaviour.

In parallel, there has been a recent surge of interest in *distributional* approaches to single-agent reinforcement learning (RL), in which agents learn to predict not only expected returns, but further properties of the full distribution of returns [3, 28]. These approaches have led to state-of-the-art performance in several large-scale domains [14, 19, 31, 55]. However, most distributional RL algorithms do not explicitly use the learned distribution when choosing actions but instead simply choose the action that will maximise the expected return. Recently, distributional RL has also been applied to multi-agent learning [13, 20, 24]. In contrast to the TD-optimistic approaches described above, these methods can be understood as introducing optimism at the level of *returns*, as they act according to *risk-sensitive* summaries of the return distribution, unlike the risk-neutral mean value in standard RL.

In this paper, we investigate the fundamental differences between these TD-optimistic and return-optimistic approaches in multi-agent learning. More precisely, our key contributions are as follows:

- We prove *equivalence* of existing TD-optimistic multi-agent algorithms and distributional RL approaches in *stateless environments*.
- We then show this equivalence breaks down for environments with state, and that more generally, existing approaches such as HQL can be interpreted as inducing optimism at the level of temporal difference errors, whereas distributional RL methods naturally induce optimism at the level of the return.
- We investigate several environment features that exemplify the differences between these two approaches. One such example is environments with unequal episode lengths. TD-optimistic approaches such as HQL are prone to *miscalibrated optimism* in such environments.
- Based on considerations such as those above, we propose new families of algorithms for decentralised cooperative multi-agent learning, which interpolate between TD-optimism and return-optimism.
- Finally, we take a broader view, and sketch out several further directions for developing our understanding of TD-optimistic and return-optimistic approaches to multi-agent learning.

2 BACKGROUND

We begin by recalling particular notions from multi-agent value-based RL and distributional single-agent RL, at the required level of detail for this paper; for more detailed accounts of these two topics, see Bloembergen et al. [7], Busoniu et al. [9], Hernandez-Leal et al. [18], Littman [23], Matignon et al. [27], Panait and Luke [35], Tuyls and Weiss [52] and Bellemare et al. [3], Rowland et al. [42], respectively.

2.1 Multi-agent Markov decision processes

Markov decision processes (MDPs) are common models for agent-environment interactions in reinforcement learning. A Markov decision process $(\mathcal{X}, \mathcal{A}, P, \mathcal{R}, \gamma)$ is specified by a finite state space $\mathcal{X}$, finite action space $\mathcal{A}$, transition kernel $P : \mathcal{X} \times \mathcal{A} \to \mathscr{P}(\mathcal{X})$, reward distributions $\mathcal{R} : \mathcal{X} \times \mathcal{A} \to \mathscr{P}(\mathbb{R})$, and discount factor $\gamma \in [0, 1]$.

In this paper, we use multi-agent Markov decision processes [8, MMDP] as a model for cooperative multi-agent interactions. A MMDP is specified by a standard MDP $(\mathcal{X}, \mathcal{A}, P, \mathcal{R}, \gamma)$ whose action set has additional structure, taking the form $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$, for some integer $N \in \mathbb{N}$ and finite sets $(\mathcal{A}_n)_{n=1}^N$. The interpretation is that agents labeled 1 through N interact in this MMDP, at each timestep having access to the state $x \in \mathcal{X}$, each selecting an action $a^n \in \mathcal{A}_n$ $(n = 1, \ldots, N)$, and all receiving the common reward from the MDP. Thus, the agents interact jointly with the environment to produce trajectories $(x_t, a_t, r_t)_{t \geq 0}$ of states, joints actions, and rewards, and the agents' goals are perfectly aligned to maximise the return $\sum_{t \geq 0} \gamma^t r_t$.

2.2 Optimistic decentralised learning in MMDPs

An optimal joint policy could be learned in a centralised manner by treating the problem as a single-agent MDP on the action space $\mathcal{A} = \prod_{n=1}^N \mathcal{A}_n$, although this scales exponentially in size with respect to the number of agents. Moreover, this requires that there is either a centralized learning system, or that all agents communicate their actions with each other at each timestep. To alleviate these issues, one can consider *independent*, or *decentralised*, learning, a paradigm in which agents learn without directly observing the actions of other agents.

In the decentralised multi-agent setting, a natural idea is to have each agent implement a single-agent RL algorithm, such as Q-learning. However, the environment is effectively non-stationary from the point of view of an individual agent, meaning that agents learning in such a manner are not guaranteed to reach an jointly optimal policy, and empirical performance can be quite poor. However, single-agent algorithms can be modified to learn in an *optimistic* manner, which often leads to theoretical and empirical improvements in performance.

One of the first algorithms in this area with theoretical guarantees is *distributed Q-learning* [22], an optimistic variant of Q-learning in which the agents are initialised with pessimistic Q-functions, and only update in response to positive temporal difference errors. Agents using distributed Q-learning are guaranteed to jointly play the optimal policy eventually in all *deterministic* MMDPs.

Much empirical progress has been made in adapting the basic approach of distributed Q-learning to *stochastic* MMDPs, with prominent examples including lenient Q-learning [36–38] and hysteretic Q-learning (HQL) [26]. HQL modifies standard Q-learning as shown in Algorithm 1; the parameters $\alpha > \beta > 0$ are typically viewed as learning rate parameters, but here we equivalently view them as part of the loss. In the control case, the action a' is selected greedily with respect to the current Q-function estimate, i.e. $a' = \arg\max_{\tilde{a}} Q(x', \tilde{a})$. HQL weights positive TD errors more heavily than negative TD errors, and can therefore be understood as interpolating between distributed Q-learning and standard Q-learning.

Algorithm 1 Hysteretic learning update.

Observe transition (x, a, r, x', a')
if x' is terminal **then**
    Compute $\Delta = r - Q(x, a)$
else
    Compute $\Delta = r + \gamma Q(x', a') - Q(x, a)$
Update $Q(x, a)$ by taking gradient of loss

$$\Delta^2 \left[\alpha \mathbb{1}_{\Delta > 0} + \beta \mathbb{1}_{\Delta < 0}\right]$$

2.3 Distributional reinforcement learning

In standard (single-agent) reinforcement learning, an agent's typical goals are either to (i) compute expected returns

$$Q^\pi(x, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, X_0 = x, A_0 = a \right], \tag{1}$$

conditional on each initial state-action pair $(x, a) \in \mathcal{X} \times \mathcal{A}$, given a policy $\pi : \mathcal{X} \to \mathscr{P}(\mathcal{A})$ (the *evaluation problem*); or (ii) compute the optimal policy $\pi^* : \mathcal{X} \to \mathscr{P}(\mathcal{A})$ and associated value function as in Equation (1), referred to as the *control problem*.

Value-based algorithms typically recursively estimate these quantities using approaches based on (approximate) dynamic programming [4, 5, 40, 50, 51]. Notably, these approaches focus on estimation of the *expected value* of the random variable $\sum_{t=0}^{\infty} \gamma^t R_t | X_0 = x, A_0 = a$, as in Equation (1).

By contrast, *distributional* approaches to RL aim to learn more information about this random variable than just its mean. These ideas can be traced back far into the RL literature [28, 29, 44] and have recently enjoyed a surge in interest due to the effectiveness of these methods in deep RL algorithms [3, 14, 15, 31, 55]. Recent approaches have focused on learning particular properties of the distributions such as cumulative probabilities [3], quantiles [15], and expectiles [42], and using certain distributional versions of dynamic programming to compute these values [3, 41, 42].

As a special case of interest in this paper, we now provide a more detailed overview of expectile-based distributional RL. Given $\tau \in [0, 1]$, the *τ-expectile* $e_\mu(\tau)$ of a scalar distribution $\mu \in \mathscr{P}(\mathbb{R})$ (with finite second moment) is defined

$$\arg\min_{y \in \mathbb{R}} \mathbb{E}_{Z \sim \mu} \left[(Z - y)^2 [\tau \mathbb{1}_{Z > y} + (1 - \tau) \mathbb{1}_{Z < y}] \right]. \tag{2}$$

Algorithm 2 Expectile distributional learning update.

Observe transition (x, a, r, x', a')
for $k = 1, \ldots, K$ **do**
  if x' is terminal **then**
    Compute $\Delta_{k,\ell} = r - Q(x, a; \tau_k) \;\; \forall \ell$
  else
    $\frac{1}{K} \sum_{\ell=1}^{K} \delta_{z_\ell} \leftarrow \texttt{Impute}(Q(x', a'; \tau_{1:K}))$
    Compute $\Delta_{k,\ell} = r + \gamma z_\ell - Q(x, a; \tau_k) \;\; \forall k$
  Update $Q(x, a; \tau_k)$ by taking gradient of loss
    $\frac{1}{K} \sum_{\ell=1}^{K} \Delta_{k,\ell}^2 \left[\tau_k \mathbb{1}_{\Delta_{k,\ell}>0} + (1 - \tau_k)\mathbb{1}_{\Delta_{k,\ell}<0}\right]$

Expectiles [30] are a key form of risk measurement in econometrics [56] and can be interpreted as non-risk-neutral generalisations of the mean of a distribution (the 1/2-expectile is precisely the mean). Any value of τ greater (resp., less than) 1/2 yields an expectile reflecting a greater degree of optimism (resp., pessimism) about the distribution in question. Expectiles also have close parallels with quantiles, which are also parametrised by $\tau \in [0, 1]$ and defined analogously to Equation (2), using an absolute value loss in place of the squared loss over $Z - y$.

A dynamic programming algorithm, expectile distributional reinforcement learning (EDRL), which aims to learn the expectiles of the return distributions in an MDP for a prespecified collection of expectile parameters $\tau_1, \ldots, \tau_K \in [0, 1]$, is presented in Algorithm 2 [42]. In this algorithm and the others that follow, the action a' that appears in the transition (x, a, r, x', a') may be obtained according to some greedy procedure with respect to the current Q-function Q (as in Q-learning), or by sampling according to some fixed policy $a' \sim \pi(\cdot|x')$ (as in policy evaluation); both cases are covered by stating the algorithm in this way. The `Impute` subroutine called in Algorithm 2 takes in a collection of expectile values, and outputs a distribution that has those expectiles. This is an under-determined problem, and so can in principle be implemented in different ways, as described in Rowland et al. [42].

2.4 Applying distributional reinforcement learning to multi-agent problems

The two previous sections have addressed separate learning problems: decentralised cooperative multi-agent learning in MMDPs, and distributional reinforcement learning in single-agent MDPs. Several recent works have applied distributional reinforcement learning to MMDPs, either swapping or augmenting learning rules such as hysteretic Q-learning with techniques from distributional RL [13, 20, 24]. These papers have produced performance gains by using distributional approaches in a variety of empirical settings. The basic application of distributional RL to decentralised cooperative multi-agent learning has each agent implementing a single-agent distributional RL algorithm, and selecting actions according to some optimistic summary of the distribution, such as a quantile.

Our goal in this paper is to develop an understanding of the fundamental similarities and differences between multi-agent learning algorithms such as hysteretic Q-learning, and distributional approaches.

3 DISTRIBUTIONAL REINFORCEMENT LEARNING IN COOPERATIVE MULTI-AGENT ENVIRONMENTS

With preliminaries and background established in Section 2, we are ready to establish theoretical connections between existing optimistic approaches to multi-agent valued-based learning, and distributional RL.

3.1 Stateless environments

Our first result highlights the similarities between distributional and hysterestic approaches to decentralised multi-agent learning. The result concerns *stateless environments*; MMDPs in which there is only one state, and all episodes finish after a single time-step. These are also referred to as matrix games in the game theory literature.

Proposition 3.1 (Equivalence of hysteretic learning and EDRL in stateless environments.) Consider an environment with a single state that terminates immediately. Then the hysteretic update rule with parameters (α, β) is equivalent (up to a change in learning rate) to the expectile distributional update for a single expectile with parameter $\tau = \alpha/(\alpha + \beta)$.

Proof. In stateless environments, learner n maintain a Q-value $Q(a)$ for each action $a \in \mathcal{A}_n$. An observation consists of an action a taken by a player, and a corresponding observed reward r. In this case, the HQL loss reduces to

$$(r - Q(a))^2 \left[\alpha \mathbb{1}_{r>Q(a)} + \beta \mathbb{1}_{r<Q(a)}\right], \tag{3}$$

since there is no next state to bootstrap from. Now consider the associated EDRL loss corresponding to the expectile level $\tau \in (0, 1)$. The EDRL loss reduces to

$$(r - Q(a))^2 \left[\tau \mathbb{1}_{r>Q(a)} + (1 - \tau)\mathbb{1}_{r<Q(a)}\right]. \tag{4}$$

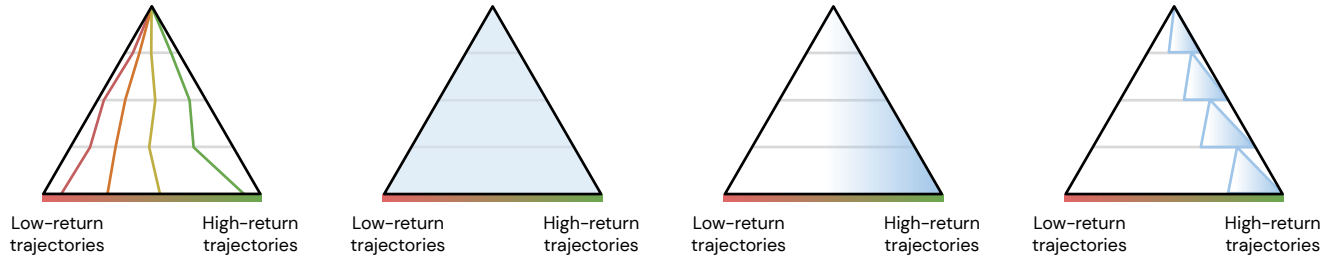
Now observe that if $\tau = \alpha/(\alpha + \beta)$, then Equation (4) is equal to Equation (3) multiplied by the constant $(\alpha + \beta)^{-1}$, as required. □

There are several important consequences to this simple result. First, the result makes clear the *type* of optimism that HQL induces over each TD error; it is precisely an expectile of the reward distribution, dictated by the choice of α and β.

Additionally, there is nothing particular about using expectiles as a means of optimistic value summary. Distributional RL has made use of a variety of other distribution summaries such as quantiles, and this result is suggestive of a wider space of TD-optimistic algorithms making use of other distribution summaries. This is an early indicator of the general framework of optimistic algorithms developed in this paper; moving to consider the general case of MMDPs with state opens this space of possible algorithms up even further.

3.2 Environments with state

Despite the equivalence established between EDRL and hysteretic learning in the stateless case, the central observation made in this paper establishes the contrary in stateful environments.



(a) Raw trajectories (b) Risk-neutral (c) Return-optimistic (d) TD-optimistic

Figure 1: Schematic illustration of the differences between TD-optimistic and return-optimistic learning algorithms. (a) A schematic illustration of the game tree for a particular environment. Trajectories begin at the root node at the top, and eventually reach an absorbing leaf node at the bottom of the diagram. Trajectories are ordered so that the right-most trajectories are the highest rewarding. (b) A risk-neutral approach to estimating the value of an action at the root node considers all possible trajectories in accordance with their probability of arising, as indicated by the constant shading of the diagram. (c) A return-optimistic summary of value gives more weight to high-return trajectories, as indicated by the shading more heavily on the right-hand side of the diagram. (d) A TD-optimistic summary of value compounds optimistic estimates of the outcome of each step of a trajectory, as indicated by the stacked triangles on the right-hand side of the plot.

Key Observation

HQL and EDRL are non-equivalent in stateful environments, and can therefore learn distinct values and policies. This difference is due to the fact that HQL induces optimism at the level of TD errors, whilst EDRL induces optimism at the level of returns.

This key distinction between HQL and EDRL in stateful environments is in fact a special case of a more general difference between algorithms that are optimistic at the level of TD errors, and algorithms that are optimistic at the level of returns. We provide a schematic illustration of this difference in Figure 1, to provide additional intuition as to how value estimates are affected by these two types of optimism.

The simple analysis above of the similarities and differences of EDRL and HQL in stateless and stateful cases has identified two axes of variation for optimistic algorithms: (i) the *optimism target*, the quantity in the learning process that is subject to optimism, and (ii) the *optimism type*, the type of optimistic summary the learning algorithm uses over this quantity. These axes are illustrated in Figure 2; as described in the caption, this table already reveals several algorithmic approaches that to our knowledge have not yet been explored, and can also be extended both vertically and horizontally to reveal yet more unexplored approaches, as described in later sections.

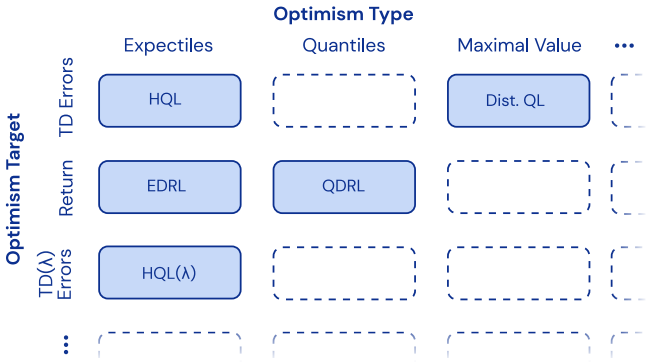


Figure 2: Two axes of variation for decentralised multi-agent RL algorithms based on optimism: (i) *Optimism target*: 'What quantity is the algorithm optimistic about?'; (ii) *Optimism type*: 'What type of optimism is used over the selected target?'. Proposition 3.1 shows that in stateless cases, TD errors and returns are equivalent, and so the 'TD errors' and 'returns' rows coincide. QDRL refers to quantile distributional reinforcement learning [15]. Blank cells indicate approaches which to our knowledge have not yet been investigated. There are many other types of optimism that could be considered, such as the use of higher moments or conditional-value-at-risk, which would correspond to additional columns in this table, leading to further approaches to explore. In Section 5, we describe several methods that extend this table vertically as well, introducing algorithms that interpolate between TD- and return-optimism, including HQL(λ), which induces optimism at the level of TD(λ) errors, as indicated above.

4 DIFFERENCES BETWEEN TD-OPTIMISM AND RETURN-OPTIMISM

In the previous section, we established that HQL and EDRL, and more broadly TD-optimistic and return-optimistic methods, induce optimism by manipulating different underlying quantities. We now identify several characteristics of environments under which these two algorithmic approaches lead to different outcomes and conduct a preliminary analysis of the learnt behaviours in instances of these environments.

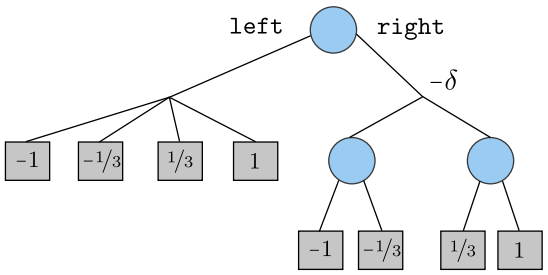


Figure 3: The environment used in the experiments in Section 4.1. Blue circles indicate states of the environment, whilst grey squares indicate terminal rewards.

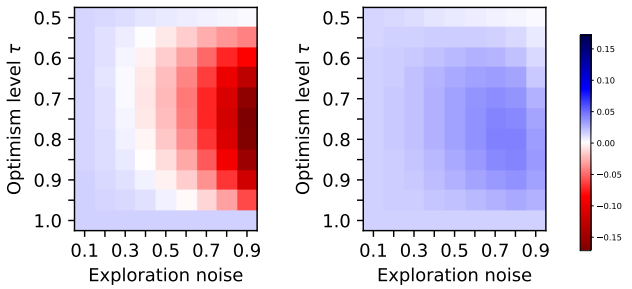


Figure 4: HQL (left) and EDRL (right) differences in `left` and `right` action values at the root node of the game tree in Figure 3. Blue corresponds to the learnt value for `left` being greater than that of `right`, which is the correct ordering, whilst red corresponds to the opposite ordering. HQL prefers the sub-optimal action, particularly in the presence of high exploration noise, whereas EDRL always identifies the correct optimal action.

In single-agent reinforcement learning, exploration affects only the rate of convergence to the optimal policy, but in multi-agent settings, exploration also affects *which* policies agents converge to. In light of this, the performance of TD-optimistic and return-optimistic approaches depend on the exploration strategy used, and we therefore study the performance of these algorithms under a variety of exploration noise and optimism parameter values. In order to avoid introducing additional hyperparameters such as learning rates that are required in sample-based algorithms, we focus on operator-based implementations of TD-optimistic and return-optimistic algorithms in the experiments that follow, using dynamic programming. Specifically, the operator version of HQL updates values to the minimiser of expected loss in Algorithm 1, and the operator version of EDRL computes an *exact* distributional Bellman operator update, using expectiles to define the greedy policy.

4.1 Compounding optimism and variable-length trajectories

Recalling Figure 1, TD-optimistic methods such as HQL induce optimism in value estimates at every temporal difference error. We

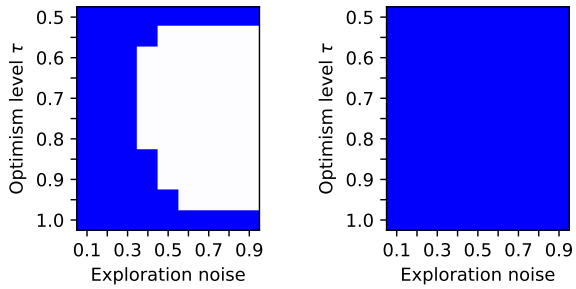


Figure 5: Whether or not a joint optimal policy is learnt under HQL (left) and EDRL (right) for both agents in the game illustrated in Figure 3. Blue indicates that a joint optimal policy is learnt, whilst white indicates that it is not.

therefore hypothesise that for environments in which trajectories of different lengths are possible, TD-optimistic methods may end up preferring actions that yield longer trajectories. This is because longer trajectories include more temporal differences (and hence more total optimism) than shorter trajectories. We refer to this phenomenon as *compounding optimism*.

As a preliminary test of this hypothesis, we consider a simple environment in which one agent selects either the `left` or `right` action. Under the `left` action, a second agent selects which of the rewards $-1, -1/3, 1/3, 1$ the two agents will receive. Under the `right` action, a cost of $\delta = 0.01$ is paid, and the second agent then makes two decisions that lead to one of the rewards $-1, -1/3, 1/3, 1$ again; see Figure 3. We use the discount factor $\gamma = 1$; given the -0.01 cost, the optimal choice for the first agent is to take the `left` action.

We consider the case where the second agent's policy is fixed at an ε-greedy version of its optimal policy, and in Figure 4, we summarise the first agent's learnt action values at the root node. Specifically, we plot the value of the `right` action subtracted from that of the `left` action, for both HQL and EDRL. We vary both the exploration noise and the optimism parameter of the learning agent. Whilst EDRL agents identify the correct optimal action under all parameter combinations, many HQL agents prefer the suboptimal action, particularly in the presence of high levels of exploration noise. In other words, the deeper game tree that corresponds to the `right` action leads to *compounded optimism* in TD-optimistic approaches but not in return-optimistic approaches.

We also verify under which conditions a joint optimal policy is obtained when both agents simultaneously update their value estimates. Results are displayed in Figure 5; two learners using return-optimistic approaches learn an optimal policy in all tested settings of exploration and optimism, whilst the TD-optimistic learners do not learn the optimal policy in the presence of reasonably large exploration noise.

4.2 Repeated states

In contrast to the previous example, many MMDPs have a transition structure that permits a state to be visited multiple times in a single trajectory. There are also qualitative differences in the ways that TD-optimistic and return-optimistic methods learn in these settings.

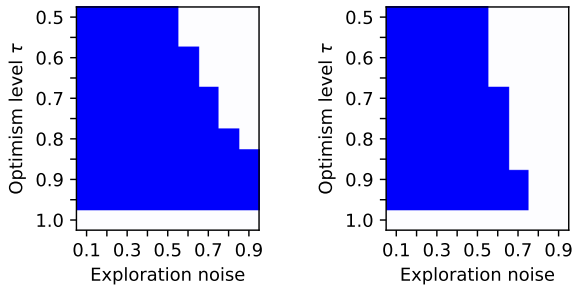


Figure 6: Whether or not a joint optimal policy is learnt under HQL (left) and EDRL (right) for both agents in the repeated climbing game. Blue indicates that a joint optimal policy is learnt, whilst white indicates that it is not.

The return may reflect randomness in other agents' actions at a given state several times in the same trajectory, and each instance is treated as a separate source of randomness. We might therefore also expect different behaviour from TD-optimistic and return-optimistic approaches in environments in which repeated states are common.

To explore this intuition, we consider a repeated version of the partially-stochastic variant [27] of the *climbing game*, originally introduced by Claus and Boutilier [12]. In our environment, there is a single state which repeatedly transitions back to itself, with the reward at each timestep being specified by the payoffs of the climbing game. Computing the exact distributional Bellman update is impractical in this setting, since the environment has an infinite horizon, so we use a 100-quantile approximation [15]. Further, we use a discount factor of $\gamma = 0.8$. Convergence results for HQL and EDRL are presented in Figure 6, with initialisation at the optimal policy, to test the stability of this equilibrium; overall, the return-optimistic approach is more sensitive to the optimism parameter. In contrast to the case of variable-length trajectories, this suggests that TD-optimistic approaches may be preferable relative to return-optimistic methods in environments in which states are often encountered multiple times within the same trajectory.

5 NEW ALGORITHMS FOR DECENTRALISED COOPERATIVE MULTI-AGENT LEARNING

Our earlier discussions make clear that hysteretic approaches to decentralised cooperative multi-agent learning induce optimism over TD errors, whilst distributional approaches induce optimism over the return, and further that these approaches can lead to different learning outcomes. A natural question to ask is whether it is possible to induce optimism elsewhere in the agent's update. Here, we propose two new families of approaches that interpolate between TD-optimism and return-optimism; these methods can be viewed as further extending Figure 2.

5.1 Multi-step hysteretic learning

A simple way to prevent optimism from compounding across many timesteps is to effectively increase the bootstrapping horizon of the learning algorithm. A straightforward modification to TD-optimistic approaches to achieve this is to use n-step, rather than one-step, temporal differences. For example, a 2-step version of HQL is actually equivalent to a return-optimistic approach in the environment considered in Section 4.1. More generally, a n-step TD-optimistic approach is equivalent to a return-optimistic approach in any environment with episodes of length at most n, and using different n-step returns offers a spectrum of trade-offs between TD-optimism and return-optimism. Recently, n-step returns have been considered empirically in a MARL setting by [16].

Algorithm 3 HQL(λ)

1: Observe trajectory $(x_t, a_t, r_t)_{t=0}^{\infty}$
2: Compute

$$\Delta = (r_0 + \gamma \max_{a' \in \mathcal{A}} Q(x_1, a') - Q(x_0, a_0)) + \sum_{t=1}^{\infty} (\gamma\lambda)^t (r_t + \gamma \max_{a' \in \mathcal{A}} Q(x_{t+1}, a') - \max_{a' \in \mathcal{A}} Q(x_t, a'))$$

3: Update $Q(x_0, a_0)$ by taking gradient of loss

$$\Delta^2 \left[\alpha \mathbb{1}_{\Delta > 0} + \beta \mathbb{1}_{\Delta < 0}\right]$$

A distinct trade-off can be made using λ-returns [48–50], yielding the HQL(λ) method as described in Algorithm 3, which can be interpreted as a risk-sensitive variant of Peng's Q(λ) [39]. A schematic illustration of multistep TD-optimistic methods is given in Figure 7, in analogy with Figure 1. With respect to Figure 2, these multi-step method can be thought of as introducing additional rows, specifying optimism targets distinct from returns and single-step TD errors.

The performance of this algorithm on the environment from Section 4.1 is illustrated in Figure 8; as expected, its performance interpolates between TD-optimistic and return-optimitic methods.

5.2 Single-step distributional learning

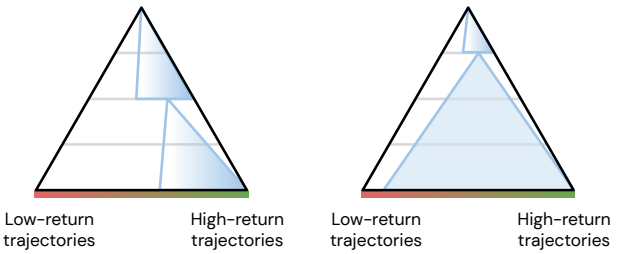
The unequal optimism that TD-optimism induces in trajectories of differing lengths can be seen to arise from the form of the temporal difference error itself. TD-optimistic algorithms bootstrap from an already-optimistic estimate of value at the next state, and then apply an optimistic loss to the resulting temporal difference error. In contrast to multi-step learning, a different approach to interpolating between TD-optimism and return-optimism is to modify approaches like HQL to bootstrap from a risk-neutral value estimate, thereby preventing the accumulation of optimism through multiple back-ups. This idea leads to the approach shown in Algorithm 4, which learns two estimates of value: (i) a risk-neutral estimate Q, and (ii) an optimistic estimate Q_o.

Since the loss for Q_o bootstraps from Q, which is a risk-neutral value estimate, Q_o is optimistic only about the immediate reward and transition that occur. This prevents the accumulation of optimism across many timesteps, as was the case with HQL, and for this reason, we term this algorithm *neutral-bootstrap HQL* (NB-HQL). This new algorithm can be reconciled with Figure 2 through the addition of an extra axis of variation: whether or not the bootstrap quantity is an optimistic summary of value. We present the performance of this new algorithm for the environment described

Algorithm 4 Neutral-bootstrap HQL.

1: Observe transition (x, a, r, x', a')
2: **if** x' is terminal **then**
3: Compute $\Delta = \Delta_o = r - Q(x, a)$
4: **else**
5: Compute $\Delta = r + \gamma Q(x', a') - Q(x, a)$
6: Compute $\Delta_o = r + \gamma Q(x', a') - Q_o(x, a)$
7: Update $Q(x, a)$ by taking gradient of loss Δ^2
8: Update $Q_o(x, a)$ by taking gradient of loss

$$\Delta_o^2 \left[\alpha \mathbb{1}_{\Delta_o > 0} + \beta \mathbb{1}_{\Delta_o < 0}\right]$$



(a) Multistep TD-optimistic **(b) TD-optimistic with neutral bootstrap**

Figure 7: Schematic illustration of (a) multistep TD-optimistic and (b) neutral-bootstrap TD-optimistic learning algorithms, following the same conventions as Figure 2.

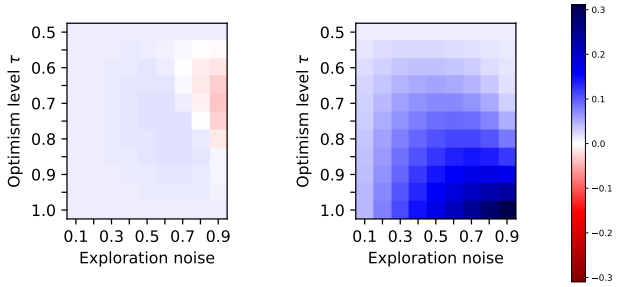


Figure 8: HQL(λ) with $\lambda = 0.5$ (left) and NB-HQL (right) differences in `left` and `right` action values at the root node of the game tree in Figure 3.

in Section 4.1, in Figure 8. The single-step optimism corrects the miscalibrated optimism that was present in HQL.

We can interpret this new method as a distributional RL algorithm; replacing the imputation step of the EDRL update (Line 6 of Algorithm 2) by setting each sample $z_k = Q(x', a'; 1/2)$, we obtain a version of Algorithm 4 that learns several levels of optimism about the immediate reward and transition simultaneously. The idea of using techniques from distributional RL to learn about the distribution of partial segments of the return has recently been proposed in the context of risk-neutral single-agent reinforcement learning [1].

6 DISCUSSION AND CONCLUSION

In this paper, we have highlighted the differences between well-established approaches to decentralised cooperative multi-agent learning, which use optimism over temporal difference errors as a means to encourage cooperation, and approaches based on distributional reinforcement learning, which use optimism over returns.

This has led to improved understanding as to how these algorithms perform, and which characteristics of environments tend to amplify the differences between these two families of approaches. The tendency of TD-optimistic approaches to miscalibrate optimism in environments with varying episode lengths led us to propose several new families of learning algorithms that can be interpreted as interpolating between TD-optimism and return-optimism.

There are many natural avenues for future work in developing our understanding of this aspect of multi-agent learning. First, Table 2 makes clear that there are many unexplored possibilities, within both TD-optimistic and return-optimistic families of algorithms, to develop new hysteretic algorithms based on quantiles or other optimistic distribution summaries such as conditional-value-at-risk. Similarly, while we have focused on hysteretic Q-learning as an exemplar TD-optimistic approach, our analysis can also be applied to other approaches such as lenient Q-learning [36]. There is also clearly scope for investigating the combination of these algorithms with deep learning architectures in large-scale environments, with scheduled variation in exploration rates and optimism levels throughout training.

In addition to new algorithms within the TD-optimistic and return-optimistic families, another natural source of future work is further investigation of the newly-introduced families of methods that interpolate between TD-optimism and return-optimism, as well as extending Figure 2 along the vertical axis, for example by considering CVaR-based optimism [10, 11, 21, 43]. There are also many other approaches to cooperative MARL not studied in this paper that could be naturally combined with the optimistic approaches discussed here [18, 20, 25, 45–47].

We also remark that whilst the focus of this paper is on decentralised *cooperative* multi-agent learning, the key structure of the environments that makes optimism an effective means of learning is also shared by two-player sequential-move zero-sum games, such as Chess and Go. Since this new setting is adversarial, one must replace optimism with pessimism, but the underlying situation is fundamentally the same; in deterministic environments with this structure, maximal pessimism is an effective means of learning (and in fact corresponds to minimax Q-learning [23]), but when environments are stochastic, this pessimism must be softened. Empirical investigation of TD-pessimistic, return-pessimistic, and interpolating approaches is, therefore, an additional interesting future direction.

In summary, this work has explored the properties of TD-optimistic and return-optimistic approaches to cooperative multiagent learning, and has introduced a unifying framework that encompasses these families of algorithms. Our findings identify several key properties of environments that distinguish the behaviour of these algorithms in environments with state, and the framework also naturally suggests unexplored combinations of optimism targets and types which could be explored in future work.

ACKNOWLEDGMENTS

We thank Bart De Vylder, Romuald Elie, and Thore Graepel for feedback on earlier versions of this paper. We also thank the anonymous reviewers for useful comments during the review process.

REFERENCES

[1] Mastane Achab. 2020. *Ranking and risk-aware reinforcement learning.* Ph.D. Dissertation. Institut Polytechnique de Paris.
[2] Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. 2020. The Hanabi challenge: A new frontier for AI research. *Artificial Intelligence* 280 (2020), 103216.
[3] Marc G Bellemare, Will Dabney, and Rémi Munos. 2017. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning (ICML).*
[4] Richard Bellman. 1957. *Dynamic Programming.* Dover Publications.
[5] Dimitri P Bertsekas and John N Tsitsiklis. 1996. *Neuro-dynamic programming.* Vol. 5. Athena Scientific Belmont, MA.
[6] Daan Bloembergen, Michael Kaisers, and Karl Tuyls. 2011. Empirical and theoretical support for lenient learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS).*
[7] Daan Bloembergen, Karl Tuyls, Daniel Hennes, and Michael Kaisers. 2015. Evolutionary dynamics of multi-agent learning: A survey. *Journal of Artificial Intelligence Research* 53 (2015), 659–697.
[8] Craig Boutilier. 1996. Planning, Learning and Coordination in Multiagent Decision Processes. In *Conference on Theoretical Aspects of Rationality and Knowledge (TARK).*
[9] Lucian Busoniu, Robert Babuska, and Bart De Schutter. 2008. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Trans. Systems, Man, and Cybernetics, Part C* 38, 2 (2008), 156–172.
[10] Yinlam Chow and Mohammad Ghavamzadeh. 2014. Algorithms for CVaR Optimization in MDPs. In *Neural Information Processing Systems (NIPS).*
[11] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. 2017. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research* 18, 1 (2017), 6070–6120.
[12] Caroline Claus and Craig Boutilier. 1998. The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI Conference on Artificial Intelligence.*
[13] Felipe Leno Da Silva, Anna Helena Reali Costa, and Peter Stone. 2019. Distributional Reinforcement Learning Applied to Robot Soccer Simulation. In *Adapative and Learning Agents Workshop, AAMAS.*
[14] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. 2018. Implicit quantile networks for distributional reinforcement learning. *International Conference on Machine Learning (ICML)* (2018).
[15] Will Dabney, Mark Rowland, Marc G Bellemare, and Rémi Munos. 2018. Distributional reinforcement learning with quantile regression. In *AAAI Conference on Artificial Intelligence.*
[16] Xudong Gong, Bo Ding, Jie Xu, Huaimin Wang, Xing Zhou, and Hongda Jia. 2019. Synchronous n-step method for independent Q-learning in multi-agent deep reinforcement learning. In *IEEE SmartWorld.*
[17] Pablo Hernandez-Leal, Michael Kaisers, Tim Baarslag, and Enrique Munoz de Cote. 2017. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv* (2017).
[18] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. 2019. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems* 33, 6 (2019), 750–797.
[19] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. 2018. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI Conference on Artificial Intelligence.*
[20] Jian Hu, Seth Austin Harding, Haibin Wu, and Shih-wei Liao. 2021. QR-MIX: Distributional Value Function Factorisation for Cooperative Multi-Agent Reinforcement Learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS).*
[21] Ramtin Keramati, Christoph Dann, Alex Tamkin, and Emma Brunskill. 2020. Being optimistic to be conservative: Quickly learning a CVaR policy. In *AAAI Conference on Artificial Intelligence.*
[22] Martin Lauer and Martin Riedmiller. 2000. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *International Conference on Machine Learning (ICML).*
[23] Michael L Littman. 1994. Markov games as a framework for multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML).*
[24] Xueguang Lyu and Christopher Amato. 2020. Likelihood Quantile Networks for Coordinating Multi-Agent Reinforcement Learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS).*
[25] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. 2019. MAVEN: Multi-Agent Variational Exploration. In *Neural Information Processing Systems (NeurIPS).*
[26] Laëtitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. 2007. Hysteretic Q-learning: An algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *IEEE International Conference on Intelligent Robots and Systems.*
[27] Laëtitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. 2012. Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems. *The Knowledge Engineering Review* 27, 1 (2012), 1–31.
[28] Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. 2010. Nonparametric return distribution approximation for reinforcement learning. In *International Conference on Machine Learning (ICML).*
[29] Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. 2010. Parametric return density estimation for reinforcement learning. *Conference on Uncertainty in Artificial Intelligence (UAI)* (2010).
[30] Whitney K Newey and James L Powell. 1987. Asymmetric least squares estimation and testing. *Econometrica: Journal of the Econometric Society* (1987), 819–847.
[31] Thanh Tang Nguyen, Sunil Gupta, and Svetha Venkatesh. 2021. Distributional Reinforcement Learning with Maximum Mean Discrepancy. In *AAAI Conference on Artificial Intelligence.*
[32] Shayegan Omidshafiei, Jason Pazis, Christopher Amato, Jonathan P How, and John Vian. 2017. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *International Conference on Machine Learning (ICML).*
[33] Gregory Palmer, Rahul Savani, and Karl Tuyls. 2019. Negative Update Intervals in Deep Multi-Agent Reinforcement Learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS).*
[34] Gregory Palmer, Karl Tuyls, Daan Bloembergen, and Rahul Savani. 2018. Lenient multi-agent deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS).*
[35] Liviu Panait and Sean Luke. 2005. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems* 11, 3 (2005), 387–434.
[36] Liviu Panait, Keith Sullivan, and Sean Luke. 2006. Lenient learners in cooperative multiagent systems. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS).*
[37] Liviu Panait, Karl Tuyls, and Sean Luke. 2008. Theoretical Advantages of Lenient Learners: An Evolutionary Game Theoretic Perspective. *Journal of machine learning research* 9 (2008), 423–457.
[38] Liviu Panait, R Paul Wiegand, and Sean Luke. 2003. Improving coevolutionary search for optimal multiagent behaviors. In *International Joint Conference on Artificial Intelligence (IJCAI).*
[39] Jing Peng and Ronald J. Williams. 1996. Incremental Multi-Step Q-Learning. *Machine Learning* 22, 1–3 (1996), 283–290.
[40] Martin L Puterman. 2014. *Markov decision processes: discrete stochastic dynamic programming* (4 ed.). John Wiley & Sons.
[41] Mark Rowland, Marc G Bellemare, Will Dabney, Rémi Munos, and Yee Whye Teh. 2018. An analysis of categorical distributional reinforcement learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS).*
[42] Mark Rowland, Robert Dadashi, Saurabh Kumar, Rémi Munos, Marc G Bellemare, and Will Dabney. 2019. Statistics and Samples in Distributional Reinforcement Learning. In *International Conference on Machine Learning (ICML).*
[43] Rahul Singh, Qinsheng Zhang, and Yongxin Chen. 2020. Improving Robustness via Risk Averse Distributional Reinforcement Learning. In *Learning for Dynamics and Control.*
[44] Matthew J Sobel. 1982. The variance of discounted Markov decision processes. *Journal of Applied Probability* 19, 4 (1982), 794–802.
[45] Kyunghwan Son, Sungsoo Ahn, Roben Delos Reyes, Jinwoo Shin, and Yung Yi. 2020. QTRAN++: Improved Value Transformation for Cooperative Multi-Agent Reinforcement Learning. *arXiv* (2020).
[46] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Hostallero, and Yung Yi. 2019. QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement learning. In *International Conference on Machine Learning (ICML).*
[47] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinícius Flores Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS).*
[48] Richard S Sutton. 1984. *Temporal Credit Assignment in Reinforcement Learning.* Ph.D. Dissertation.
[49] Richard S Sutton. 1988. Learning to predict by the methods of temporal differences. *Machine learning* 3, 1 (1988), 9–44.
[50] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second ed.). The MIT Press.
[51] Csaba Szepesvári. 2010. *Algorithms for reinforcement learning.* Morgan & Claypool Publishers.
[52] Karl Tuyls and Gerhard Weiss. 2012. Multiagent Learning: Basics, Challenges, and Prospects. *AI Magazine* 33, 3 (2012), 41–52.

[53] Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning* 8, 3-4 (1992), 279–292.
[54] Ermo Wei and Sean Luke. 2016. Lenient learning in independent-learner stochastic cooperative games. *Journal of Machine Learning Research* 17, 1 (2016), 2914–2955.
[55] Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu. 2019. Fully Parameterized Quantile Function for Distributional Reinforcement Learning. In *Neural Information Processing Systems (NeurIPS)*.
[56] Johanna F Ziegel. 2016. Coherence and elicitability. *Mathematical Finance* 26, 4 (2016), 901–918.