Parallel reward and punishment control in humans and robots: safe reinforcement learning using the MaxPain algorithm

Stefan Elfwing
Dept. of Brain Robot Interface
ATR Computational Neuroscience Laboratories
2-2-2 Hikaridai, Seika-cho, Soraku-gun
Kyoto 619-0288, Japan
Email: elfwing@atr.jp

Ben Seymour
Computational and Biological Learning Lab
Dept. of Engineering at Cambridge University, UK
Center for Information and Neural Networks
National Institute for Information and
Communications Technology, Suita, Japan
Email: bjs49@cam.ac.uk

***Abstract*—An important issue in reinforcement learning systems for autonomous agents is whether it makes sense to have separate systems for predicting rewards and punishments. In robotics, learning and control are typically achieved by a single controller, with punishments coded as negative rewards. However in biological systems, some evidence suggests that the brain has a separate system for punishment. Although this may in part be due to biological constraints of implementing negative quantities, it raises the question as to whether there is any computational rationale for keeping reward and punishment prediction operationally distinct. Here we outline a basic argument supporting this idea, based on the proposition that learning best-case predictions (as in Q-learning) does not always achieve the safest behaviour. We introduce a modified RL scheme involving a new algorithm which we call 'MaxPain' - which back-ups worst-case predictions in parallel, and then scales the two predictions in a multi-attribute RL policy. i.e. independently learning 'what to do' as well as 'what not to do' and then combining this information. We show how this scheme can improve performance in benchmark RL environments, including a grid-world experiment and a delayed version of the mountain car experiment. In particular, we demonstrate how early exploration and learning are substantially improved, leading to much 'safer' behaviour. In conclusion, the results illustrate the importance of independent punishment prediction in RL, and provide a testable framework for better understanding punishment (such as pain) and avoidance in humans, in both health and disease.**

I. INTRODUCTION

Humans, animals and robots share the common problem of inhabiting a complex and dynamic world, in which survival depends on harvesting rewards in the face of frequent, occasionally catastrophic, dangers. They therefore share the requirement for a control system that is effective at widely exploring and learning about reward, whilst staying safe. Understanding how this can be achieved effectively is a central concern of both neuroscience and robotics.

In recent years it has become clear that reinforcement learning (RL) provides a robust computational basis for understanding learning and decision-making in the brain, and is well known as a powerful algorithmic framework for control in autonomous robots [1], [2]. With respect to staying safe, RL in robots conventionally treats punishment as negative reward. However there is evidence that decision-system in animals have separate systems for rewards and punishments [3], [4]. This raises the question as to whether there is any fundamental reason for keeping rewards and punishments separate. If there is, this would not only offer new insights into control systems in robots (in which staying safe is becoming more of a priority, given the cost and human-interaction of modern robots [5]), but it would provide a theoretical basis to understand punishment in the brain - both in health and disease.

In this paper, we first review the architecture of neural control systems, taking a reinforcement learning perspective, and focusing on Pavlovian (state-based prediction) and Instrumental (action-based control) processes underlying avoidance learning. In particular, we consider neurobiological evidence that suggests dissociable processes for punishment prediction in action (instrumental) systems. We then use these insights to motivate a new algorithm, which we call MaxPain, that aims to concurrently balance punishment and reward prediction within a multi-attribute RL framework. We test the algorithm on a dangerous grid-world navigational task, and a delayed version of the mountain car task. These are chosen somewhat arbitrarily, but represent perhaps the two classic RL tasks to which RL algorithms have been conventionally applied. Finally, we consider how the insights these result can be used to better understand punishment and pain systems in the brain.

Punishment control in animals and humans

Pavlovian learning represents the passive learning of the predictive value of cues through association [6]. Reward and punishment are controlled by distinct systems, each associated with core conditioned responses such as approach and withdrawal, respectively. Although conditioned responses are themselves actions [7], they do not generally change the occurrence of outcome, and as such Pavlovian learning is taken to reflect the learning of state values. Rodent studies support a single Pavlovian reward and single Pavlovian punishment system. This is based, for example, on the observation that conditioning of a cue to one type of punishment (e.g. electric shock) will block new learning of the cue to a different punishment of comparable magnitude (e.g. aversive sound) [8], [9]. Furthermore, reward and punishment systems are mutually inhibitory, and omission of a reward can excite the punishment system, and vice versa. As such 'conditioned inhibitors' of reward can block the acquisition of new punishments [10]).

Good experimental evidence supports the fact that Pavlovian rewards and punishments are learned using a reinforcement learning like process, and temporal difference models have provided a good fit to neurophysiological data (including BOLD fMRI an EEG) during the dynamic acquisition of value [11], [12]. For instance, studies of Pavlovian conditioning to rewards as pleasant tasting juice, reward prediction errors are observed in brain (fMRI BOLD) responses in the striatum [13]. And in studies of Pavlovian conditioning to pain, punishment prediction errors are observed, also in the striatum [14]. Interestingly, many classic studies of financial reward and loss have identified purely reward prediction errors, with loss coded as negative reward and no positive representation of punishment, suggesting that in some cases the full range of positive and negative outcomes is coded in a single scale within the reward system [15]–[17]. But various experimental manipulations have since shown that this appears to be due to a sort of positive framing effect, given that it is difficult to probe *real* financial loss in experimental volunteers. In experiments in which financial loss is more meaningful, it is possible to identify simultaneous coding of oppositely signed reward and punishment [17]. In these studies, therefore, what is observed is consistent with parallel mirror opponent systems [18], in which the *same* value quantity is simultaneously coded with opposite sign in reward and punishment systems.

Pavlovian systems don't themselves provide a mechanism for the flexible control of action. The nature of the learning rules that govern action learning (i.e. instrumental, or operant conditioning) for rewards are relatively well understood [19]. Again, there is good evidence for reinforcement learning like processes, inferred from both choice behaviour and observed in neurophsyiological responses across species [20], [21].

The underlying structure of systems mediating control over punishments, in particular avoidance, has been more difficult to dissect. A long-standing theory, called two-factor theory, proposes that avoidance reflects the learning of the action to escape from the acquired Pavlovian fear of punishment [22]. That is, first a Pavlovian value is learned, followed by instrumental learning of an action to escape from it. Relatedly, subsequent experiments have illustrated how the avoided state acting as a safety state, with a reward-like representation derived from the fact that it signals the absence of punishment (i.e. a conditioned inhibitor) [23]. In this way, control is provided both from escape from fear of punishment and reinforcement by safety, and avoidance can be maintained [24], [25].

The fact that avoidance can be learned and maintained without any precipitating cue, as in free-operant (Sidman) designs [26], causes problems for a simple signalled avoidance model (although it is possible to involve various complex internal timing mechanisms). This has led to expectancy-based models [27], in which explicit representations of the values and outcomes of actions can be acquired and executed according to knowledge of the outcomes, a concept which has resonance with model-based RL algorithms.

Early fMRI studies into avoidance have shown that action learning is well fit by simple temporal difference action-learning models [28], with robustly observed prediction errors seen in dorsal regions of striatum [16], although punishment (i.e. the oppositely signed response pattern) prediction errors have sometimes been seen in different brain regions (such as the insula cortex), co-occuring with and with opposite sign to reward [29] . Subsequent mixed reward-punishment designs, in which the outcome probabilities are independent of each other, have attempted to identify convergence of values related to avoidance and reward acquisition. Such models rely on simultaneously optimising two quantities - minimising pain and maximising reward, and are effectively modeled as a multi-attribute Q-learning problem, with a scaling factor reflecting the balance of incentive values of punishment and reward [3]. Notably, pharmacological manipulation selectively modulates reward values, not punishment values, suggesting that the construction of values is at least partly dissociable. Evidence in rodents has also shown that equally strong avoidance and reward actions seem to have a distinct underlying neural basis, with dopamine manipulations enhancing reward but not avoidance actions [30].

More recently, it has been shown that when modelling behaviour during avoidance learning task, using TD-learning models, it is possible to distinguish dissociable punishment (pain) and punishment omission learning rates. Furthermore, it is possible to show individual differences in the propensity to learn from punishment versus its omission [4], and importantly, people who learn primarily from omission outcomes show a reward-signed prediction error in striatal brain responses, whereas those driven by punishment outcomes show punishment-signed (aversive) prediction error response. This suggests quite directly that there may be separate action-value signals for reward and punishment, that compete to control behaviour, which converge on the striatum.

These considerations lead to the hypothesis that having dissociable systems for learning action values for reward and punishment may confer some sort of computational advantage over single-system architectures. In particular, it may offer a mechanism to enhance safety during learning. We therefore set out to test this using a novel RL algorithm, outlined below, inspired by the neural data.

II. METHOD

We consider a standard RL [1] setting. In each time step t, the agent observes a state s and selects an action a according to its stochastic policy $\pi_t(s, a)$ (i.e., the probability of selecting action $a_t = a$ in state $s_t = s$). The environment then makes a transition from the current state s to the next state s' and the agent receives a scalar reward R. The action-value function $Q^{\pi}(s,a)$ is the expected accumulated discounted reward for selecting action a in state s and thereafter following policy π:

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k} | s_t = s, a_t = a\right], \qquad (1)$$

where γ is the discount factors for future reward. The optimal action-value function is defined as $Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a)$.

In the MaxPain algorithm, the standard reward R is separated into two parts, the positive reward $r \geq 0$:

$$r = \max(R, 0), \qquad (2)$$

and the pain (or punishment) $p \geq 0$:

$$p = -\min(R, 0). \qquad (3)$$

The MaxPain algorithm learns estimates of two action-value functions: Q_r and Q_p, which try to maximize the accumulated discounted positive reward and the accumulated discounted pain, respectively. To combine the two action-value functions into a single objective, Q_w, we consider the linear combination of Q_r and $-Q_p$.

$$Q_w(s,a) = wQ_r(s,a) - (1-w)Q_p(s,a). \qquad (4)$$

Here, $0 \leq w \leq 1$ is the weight that controls the trade-off between maximising the received positive reward and minimising the experienced pain. The two action-values are updated according to

$$Q_r(s,a) \leftarrow Q_r(s,a) + \alpha_r \delta_r, \qquad (5)$$

$$Q_p(s,a) \leftarrow Q_p(s,a) + \alpha_p \delta_p, \qquad (6)$$

where α_r and α_p are learning rates. To compute the TD-error for the Q_r-values, δ_r, we use either the off-policy Q-learning algorithm:

$$\delta_r \leftarrow r + \gamma_r Q_r(s', \underset{a'}{\operatorname{argmax}}(Q_w(s',a')) - Q_r(s,a), \qquad (7)$$

or the on-policy Sarsa [31] algorithm:

$$\delta_r \leftarrow r + \gamma_r Q_r(s',a') - Q_r(s,a). \qquad (8)$$

To be able to learn an estimate of the policy that maximises the accumulated discounted pain while following a policy derived from Q_w, which tries to achieve the opposite outcome, we only use the off-policy Q-learning algorithm to compute the TD-error, δ_p, for the Q_p-values:

$$\delta_p \leftarrow p + \gamma_p Q_p(s', \underset{a'}{\operatorname{argmin}}(Q_w(s',a')) - Q_p(s,a). \qquad (9)$$

Our approach is related to single-policy multi-objective RL [32], in particular the linear version of the framework for scalarised single-policy multi-objective RL algorithms proposed by Moffaert *et al.* [33]. The main difference is that they only consider objectives that maximize their accumulated discounted rewards.

We use softmax action selection with a Boltzmann distribution. The probability to select action a in state s is given by

$$\pi(a|s) = \frac{\exp(Q_w(s,a)/\tau)}{\sum_b \exp(Q_w(s,b)/\tau)}, \qquad (10)$$

where τ is the temperature that controls the trade-off between exploration and exploitation. We use hyperbolic annealing of the temperature, where the temperature decreases after every episode i:

$$\tau(i) = \frac{\tau_0}{1 + \tau_k i}. \qquad (11)$$

Here, τ_0 is the initial temperature and τ_k controls the rate of annealing.

Algorithm 1 shows the pseudo-code for the MaxPain algorithm using Q-learning of the Q_r-values.

Algorithm 1 MaxPain

```
Initialize $Q_r$ and $Q_p$ arbitrarily
for each episode do
  Get initial state $s$
  while $s$ is not terminal do
    % $Q_w$ is computed by (4)
    Select $a$ in $s$ based on policy derived from $Q_w$
    Take $a$, observe $r$, $p$, and $s'$
    $\delta_r \leftarrow r - Q_r(s,a)$
    $\delta_p \leftarrow p - Q_p(s,a)$
    if $s$' is not terminal then
      $\delta_r \leftarrow \delta_r + \gamma_r Q_r(s, \operatorname{argmax}_{a'} Q_w(s',a'))$
      $\delta_p \leftarrow \delta_p + \gamma_p Q_p(s, \operatorname{argmin}_{a'} Q_w(s',a'))$
    end if
    $Q_r(s,a) \leftarrow Q_r(s,a) + \alpha_r \delta_r$
    $Q_p(s,a) \leftarrow Q_p(s,a) + \alpha_p \delta_p$
    $s \leftarrow s'$
  end while
end for
```

III. EXPERIMENTS

A. Grid-world

First, we consider a painful grid-world navigation task (see the illustrations in Fig. 2). The goal is to navigate from the starting position in the southwest corner (green square) to the goal in the northeast corner (red square), while avoiding hitting the inner and the outer walls. The agent receives a positive reward of 1 for reaching the goal, and a pain of 0.1 (a negative reward of -0.1 in the case of standard RL) for hitting a wall. There are four actions that moves the agent one step north, south, east, or west. If the agent hit a wall then it remains in its current position. We tested MaxPain with w set to 0.1, 0.5, and 0.9, using Q-learning to learn the Q_r-values (see (7) and Algorithm 1), and we compared the performance with standard Q-learning. We ran 100 separate runs of 1000 episodes for each algorithm and setting of w, and we used the same settings of the meta-parameters in all experiments: $\alpha = 0.1$, $\gamma = 0.99$, $\tau_0 = 0.5$, and $\tau_k = 0.05$.

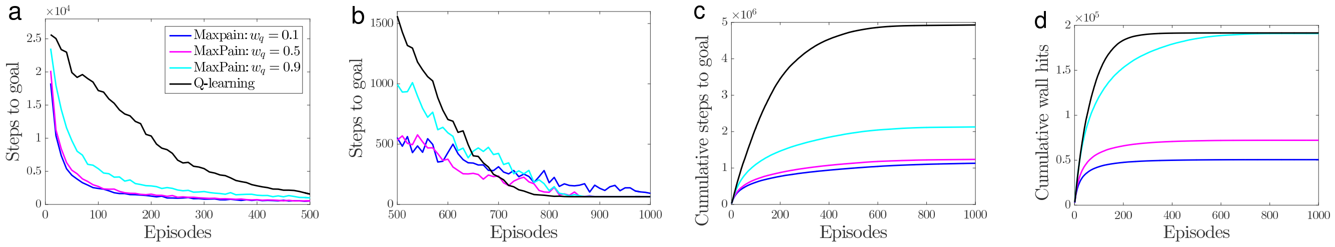

Fig. 1. Average learning curves over 100 separate runs in the grid-world navigation task for MaxPain with w set to 0.1, 0.5, and 0.9, and for Q-learning. The figure shows the number of steps to goal for the first 500 episodes (a) and the final 500 episodes (b) (shown separately for the sake of clarity), the cumulative number of steps to goal (c), and the cumulative number of wall hits (d).

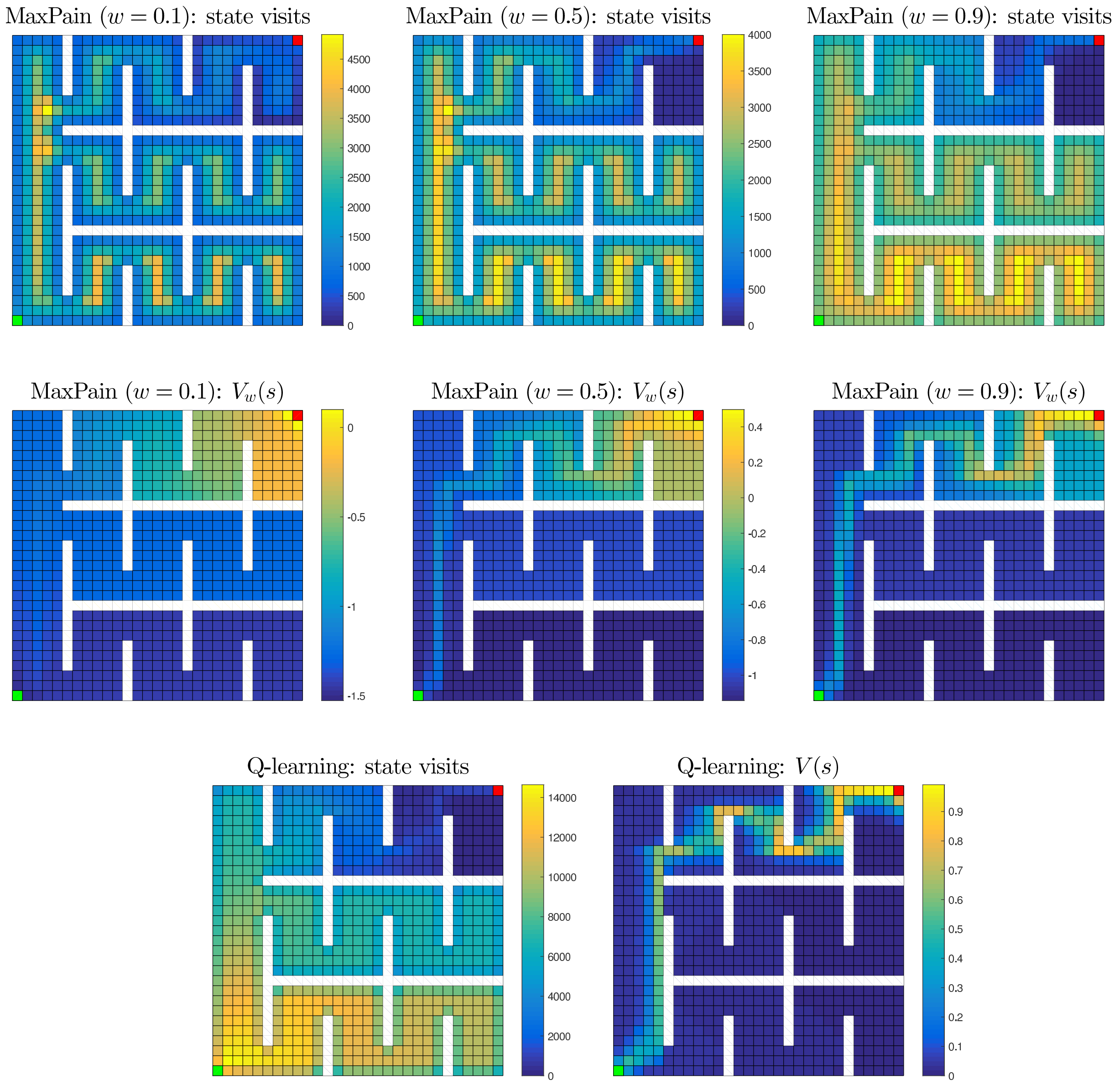

Fig. 2. The top row shows the average number state visits and the middle row shows the average final state values, $V_w(s) = \max_a Q_w(s,a)$, for MaxPain with w set to 0.1, 0.5, and 0.9 in the grid-world task. The bottom row shows the average number of state visits and the average final state values, $V(s) = \max_a Q(s,a)$, for Q-learning. The average values were computed over 100 separate runs.

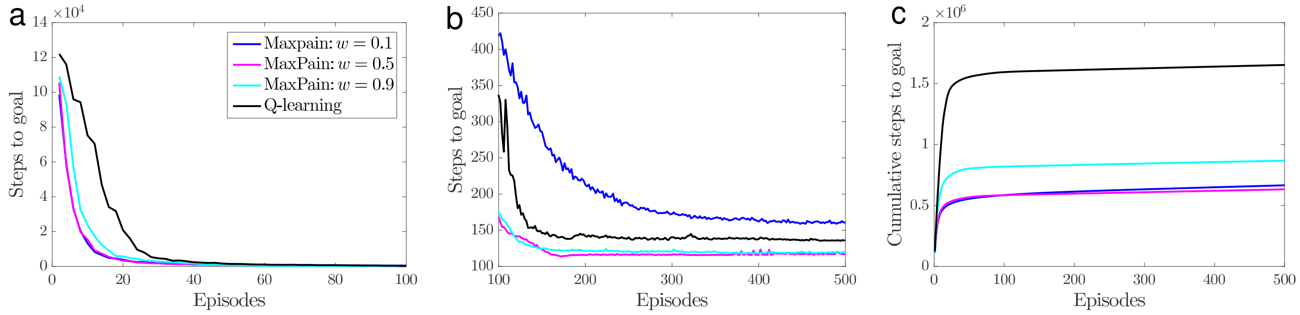

Fig. 3. Average learning curves over 100 separate runs in the mountain car task for MaxPain with w set to 0.1, 0.5, and 0.9, and for Q-learning. The figure shows the number of steps to goal for the first 100 episodes (a) and the final 400 episodes (b) (shown separately for the sake of clarity), and the cumulative number of steps to goal (c).

The experimental results (see Fig. 1) show that MaxPain could achieve fast initial learning speed, safe exploration, and near-optimal final performance. Compared with Q-learning, MaxPain with $w = 0.5$ reduced the average total number of steps to goal over the 1000 episodes by 75 % (from 4,928,982 to 1,236,884 steps), reduced the average total number of wall hits by 62 % (from 191,483 to 72,141 hits), and it reached a final average performance of 64.75 steps to goal (64.02 for Q-learning), which is close to the shortest path to the goal of 64 steps. The setting of $w = 0.5$ provided a good trade-off between safe exploration and final performance. With $w = 0.1$, the MaxPain agent achieved fast initial learning while hitting the walls the fewest times of any of the agents, but it needed 92.30 steps on average to reach the goal at the end of learning. With $w = 0.9$, the MaxPain agent reached a near-optimal final average performance of 64.70 steps to goal, but it hit the walls close to as many times as the Q-learning agent and it used almost twice as many steps in total to reach the goal as the other two MaxPain agents.

The learning of the Q_p-values was complementary to the learning of the Q_r-values, as shown in the heat maps of the final average state values ($V_w(s) = \max_a Q_w(s, a)$) in Fig. 2. The learned Q_p-values created a potential field with larger values the further away the MaxPain agents were from the goal, which helped to steer them towards the goal during exploration. This explains the faster learning speeds of the MaxPain agents compared with the Q-learning agent, and also the slower learning speed for $w = 0.9$ compared with the two lower settings of w. The heat maps of the average number of state visits in Fig. 2 show the safer learning of the MaxPain agents with $w = 0.1$ and $w = 0.5$, achieved by concentrating their exploration to the safest states (i.e., states with at least 2 steps from any wall).

B. Mountain car

Second, we consider a more difficult, delayed-reward, version of the mountain car benchmark task. In the mountain car task, the agent has to drive an under-powered car up a steep mountain slope. Since the car's engine is weak, the agent has to back up the opposite slope and then start to accelerate forward to gain enough momentum to be able to drive to the top of the mountain. The car moves according to the simplified physics defined in [34]. The car's position x_t and velocity $\dot{x}_t$ is updated by

$$\begin{aligned} x_{t+1} &= \text{bound}[x_t + \dot{x}_{t+1}], && (12) \\ x_{t+1} &= \text{bound}[\dot{x}_t + 0.001 a_t - 0.0025 \cos(3 x_t)], && (13) \end{aligned}$$

where the bound operations enforce $-1.2 \leq x_{t+1} \leq 0.5$ and $-0.07 \leq \dot{x}_{t+1} \leq 0.07$. If x_{t+1} hits the left bound then $\dot{x}_{t+1}$ is reset to zero. An episode ends when x_{t+1} hits the right bound and the goal is reached. To increase the difficulty of the exploration, we add two actions a_t, half throttle forward (+0.5) and half throttle reverse (-0.5), to the standard three actions, full throttle forward (+1), full throttle reverse (-1), and zero throttle (0).

Instead of the standard setting where the agent receives a negative reward of -1 in each time step until the car reaches the goal, we use a delayed-reward setting where the agent receives a reward of +1 for reaching the goal. Each episode starts with the car standing still ($\dot{x}_0 = 0$) in the bottom of the valley ($x_0 = -\pi/6$). To be able to test the MaxPain algorithm in this setting, we introduce a pain of 0.1 (a negative reward of -0.1 in the case of standard RL) when the car is stuck close to the bottom of valley at low speeds:

$$\begin{aligned} -0.01 - \pi/6 < \quad & x_{t+1} < 0.01 - \pi/6, && (14) \\ -0.005 < \quad & \dot{x}_{t+1} < 0.005. && (15) \end{aligned}$$

Otherwise, the agent receives a zero reward.

We tested MaxPain with w set to 0.1, 0.5, and 0.9, using Sarsa(λ) to learn the Q_r-values, and we compared the performance with standard Sarsa(λ). We used radial basis function (RBF) networks to approximate all action-value functions, using 16 equidistant Gaussian basis functions, ϕ_i, in each of the two state space dimensions. The approximate action-value function $Q(s, a|\boldsymbol{\theta})$ with network weights $\boldsymbol{\theta}$ is computed by

$$\begin{aligned} Q(s, a|\boldsymbol{\theta}) &= \sum_i \theta_{ai} \phi_i(s), && (16) \\ \phi_i(s) &= \exp\left(-\frac{\|s - c_i\|^2}{2\sigma_i^2}\right). && (17) \end{aligned}$$

Here, θ_{ai} is the network weight connecting basis function ϕ_i and the output Q-value unit for action a, and c_i is the centre

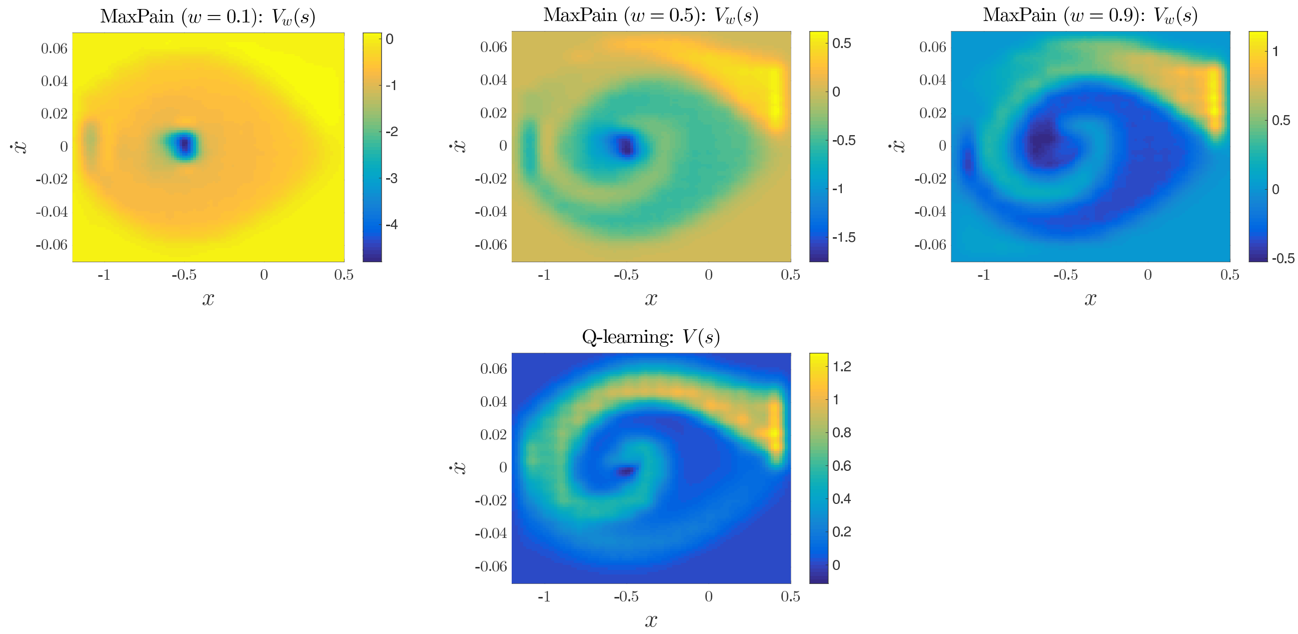

Fig. 4. The top row shows the average final state values, $V_w(s) = \max_a Q_w(s,a)$, for MaxPain with w set to 0.1, 0.5, and 0.9 in the mountain car task. The bottom row shows the average final state values, $V(s) = \max_a Q(s,a)$, for Q-learning. The average values were computed over 100 separate runs.

of ϕ_i with width σ_i. The Sarsa(λ) update of the network parameters is given by

$$\boldsymbol{e} \leftarrow \gamma\lambda\boldsymbol{e} + \nabla_{\boldsymbol{\theta}} Q(s,a|\boldsymbol{\theta}), \tag{18}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\delta\boldsymbol{e}, \tag{19}$$

where $\boldsymbol{e}$ is the eligibility traces ($\boldsymbol{e}_0 = \boldsymbol{0}$), λ is the trace-decay rate, $\nabla_{\theta_{ai}} Q(s,a|\boldsymbol{\theta}) = \phi_i(s)$, and δ is computed as in (8). We ran 100 separate runs of 500 episodes for each algorithm and setting of w, and we used the same settings of the meta-parameters in all experiments: $\alpha = 0.1$, $\gamma = 0.995$, $\lambda = 0.8$, $\tau_0 = 1$, and $\tau_k = 1$.

The experimental results are similar to the grid-world task (see Fig. 3). MaxPain with $w = 0.5$ achieved fast initial learning speed and high final performance. Compared with Sarsa(λ), it reduced the average total number of steps to the goal by 62 % (from 1,652,862 to 632,343 steps) and it reached the best average final performance of 116.8 steps to goal (135.9 for Sarsa(λ)). With $w = 0.1$, the MaxPain agent achieved fast initial learning speed but it needed 160.2 steps on average to reach the goal at the end of learning. With $w = 0.9$, the MaxPain achieved high average final performance of 118.9 steps to goal but the initial learning speed was slower than for the two lower settings of w. The heat maps in Fig. 4 of the average final state-values show that the learning of the two action-value functions in MaxPain was complementary. The learning of the Q_p-values steered the car away from the bottom of the valley while the learning of the Q_r-values directed the car towards the goal. However, in the case of the lowest setting of $w = 0.1$ there was almost no guidance towards to goal, which explains the worse final performance.

IV. Discussion

The results demonstrate the advantage gained by learning separate reward and punishment action values. It allows significantly safer exploration, as well as effective learning and near-optimal long-term performance. The Maxpain algorithm achieves this by appropriately combining values to balance the relative incentives to harvest rewards and avoid punishment. This is demonstrated across two very different, but classic learning problems - dangerous gridworld and the delayed mountain car problem.

The results have significance for our understanding of learning and decision-making in the brain. In particular, the fact that many real-word environments are both potentially dangerous, and novel or dynamic, places precedence on an algorithmic strategy that achieves rapid but safe learning, over one that might have a slightly better payoff in the extreme long-run. This provides an important insight into our understanding of avoidance learning, since the notion that avoidance incorporates an action-specific punishment prediction has not been widely considered. There are several reasons for this. First, attention has been drawn into the debate on the reinforcing role of the safety state, and how this is evoked in different situations (signalled versus unsignalled avoidance) [23]. But this has always assumed a single positively reinforcing action learning process, and not an inhibitory effect from an aversive action memory. However, learning 'what to do' and learning 'what not to do' are not perfectly reciprocal, especially when the action space is large. Second, most experiments consider only one-step avoidance, and not the multi-step sequential decision-making problems well studied in robotics, which are more similar to real-world problems. But it is multi-step problems that MaxPain provides the key advantage, as it then that a single channel reward pathway fails to back up the specific memory of what not to do.

There are other potential ways of achieving safe, early exploration, including implementing non-linear utility functions and overtly risk-averse value functions (e.g. mean-variance model) [5], [35]–[37]. But these all incorporate their cautiousness into a single value function, unlike Maxpain. From a biological perspective, another method that might approximate some features of MaxPain would be to have a robust *Pavlovian* memory for punishment, and then use this to bias actions using behavioural phenomena such as Pavlovian-instrumental transfer (such as conditioned suppression [38]). Although to our knowledge this has not been demonstrated in higher-order settings, it may well be possible. However, differentiating Pavlovian from instrumental values can be experimentally difficult, as they are often correlated, a point which has confounded animal learning-theoretic experiments for decades.

A key aspect of the MaxPain model is the scaling parameter in the multi-attribute policy, which determines the relative weight applied to reward and punishment. Not only does offer potential insight into behavioural phenomena in normal individuals, such as the framing effect [39], [40], but it also might be an important parameter in certain psychiatric diseases. For instance, it might encourage compulsive avoidance in Obsessive Compulsive disorder [41], or excessive avoidance in the fear-avoidance model of chronic pain [42]. Or it might simply convey excessively negative behaviour in disorders such as depression and anxiety disorder.

Finally, the results also illustrate how biologically inspired learning architectures have the capacity to inform control systems for autonomous robots. This may be useful for robotics applications that involve expensive or delicate robots, for instance in which repeated wall-strikes might cause substantial damage.

ACKNOWLEDGMENT

SE is supported by the project commissioned by the New Energy and Industrial Technology Development Organization (NEDO). BS is supported by the Wellcome Trust (UK) and the National Institute for Information and Communications Technology (Japan).

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 1998.

[2] N. D. Daw and K. Doya, "The computational neurobiology of learning and reward," *Current Opinion in Neurobiology*, vol. 16, no. 2, pp. 199–204, 2006.

[3] B. Seymour, N. D. Daw, J. P. Roiser, P. Dayan, and R. Dolan, "Serotonin selectively modulates reward value in human decision-making," *Journal of Neuroscience*, vol. 32, no. 17, pp. 5833–5842, 2012.

[4] E. Eldar, T. U. Hauser, P. Dayan, and R. J. Dolan, "Striatal structure and function predict individual biases in learning to avoid pain," *Proceedings of the National Academy of Sciences*, vol. 113, no. 17, pp. 4812–4817, 2016.

[5] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[6] N. J. Mackintosh, *Conditioning and associative learning*. Clarendon Press Oxford, 1983.

[7] R. C. Bolles, "Species-specific defense reactions and avoidance learning," *Psychological review*, vol. 77, no. 1, pp. 32–48, 1970.

[8] L. J. Kamin, "Attention-like processes in classical conditioning," in *Miami symposium on the prediction of behavior: Aversive stimulation*, 1968, pp. 9–31.

[9] R. A. Rescorla, "Informational variables in pavlovian conditioning," *Psychology of learning and motivation*, vol. 6, pp. 1–46, 1972.

[10] A. Dickinson and M. F. Dearing, "Appetitive-aversive interactions and inhibitory processes," *Mechanisms of learning and motivation*, pp. 203–231, 1979.

[11] Y. Niv, "Reinforcement learning in the brain," *Journal of Mathematical Psychology*, vol. 53, no. 3, pp. 139–154, 2009.

[12] J. F. Cavanagh, M. J. Frank, T. J. Klein, and J. J. Allen, "Frontal theta links prediction errors to behavioral adaptation in reinforcement learning," *Neuroimage*, vol. 49, no. 4, pp. 3198–3209, 2010.

[13] J. P. O'Doherty, P. Dayan, K. Friston, H. Critchley, and R. J. Dolan, "Temporal difference models and reward-related learning in the human brain," *Neuron*, vol. 38, no. 2, pp. 329–337, 2003.

[14] B. Seymour, J. P. O'doherty, P. Dayan, M. Koltzenburg, A. K. Jones, R. J. Dolan, K. J. Friston, and R. S. Frackowiak, "Temporal difference models describe higher-order learning in humans," *Nature*, vol. 429, no. 6992, pp. 664–667, 2004.

[15] S. M. Tom, C. R. Fox, C. Trepel, and R. A. Poldrack, "The neural basis of loss aversion in decision-making under risk," *Science*, vol. 315, no. 5811, pp. 515–518, 2007.

[16] H. Kim, S. Shimojo, and J. P. O'Doherty, "Is avoiding an aversive outcome rewarding? neural substrates of avoidance learning in the human brain," *PLoS Biol*, vol. 4, no. 8, p. e233, 2006.

[17] B. Seymour, M. Maruyama, and B. De Martino, "When is a loss a loss? excitatory and inhibitory processes in loss-related decision-making," *Current Opinion in Behavioral Sciences*, vol. 5, pp. 122–127, 2015.

[18] N. D. Daw, S. Kakade, and P. Dayan, "Opponent interactions between serotonin and dopamine," *Neural Networks*, vol. 15, no. 4, pp. 603–616, 2002.

[19] A. Dickinson and B. Balleine, "Motivational control of goal-directed action," *Animal Learning & Behavior*, vol. 22, no. 1, pp. 1–18, 1994.

[20] W. Schultz, P. Dayan, and P. R. Montague, "A neural substrate of prediction and reward," *Science*, vol. 275, no. 5306, pp. 1593–1599, 1997.

[21] J. O'doherty, P. Dayan, J. Schultz, R. Deichmann, K. Friston, and R. J. Dolan, "Dissociable roles of ventral and dorsal striatum in instrumental conditioning," *Science*, vol. 304, no. 5669, pp. 452–454, 2004.

[22] O. Mowrer, "Learning theory and behavior." 1960.

[23] J. A. Dinsmoor, "Stimuli inevitably generated by behavior that avoids electric shock are inherently reinforcing," *Journal of the experimental analysis of behavior*, vol. 75, no. 3, pp. 311–333, 2001.

[24] T. V. Maia, "Two-factor theory, the actor-critic model, and conditioned avoidance," *Learning & behavior*, vol. 38, no. 1, pp. 50–67, 2010.

[25] R. Weisman and J. Litner, "The course of pavlovian excitation and inhibition of fear in rats." *Journal of comparative and physiological psychology*, vol. 69, no. 4p1, p. 667, 1969.

[26] R. A. Rescorla, "Pavlovian conditioned fear in sidman avoidance learning." *Journal of Comparative and Physiological Psychology*, vol. 65, no. 1, p. 55, 1968.

[27] C. M. Gillan, G. P. Urcelay, and T. W. Robbins, "An associative account of avoidance," *The Wiley Handbook on the Cognitive Neuroscience of Learning*, p. 442, 2016.

[28] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, vol. 3, no. 1, pp. 9–44, 1988.

[29] M. Pessiglione, B. Seymour, G. Flandin, R. J. Dolan, and C. D. Frith, "Dopamine-dependent prediction errors underpin reward-seeking behaviour in humans," *Nature*, vol. 442, no. 7106, pp. 1042–1045, 2006.

[30] A. Fernando, G. Urcelay, A. Mar, A. Dickinson, and T. Robbins, "Comparison of the conditioned reinforcing properties of a safety signal and appetitive stimulus: effects of d-amphetamine and anxiolytics," *Psychopharmacology*, vol. 227, no. 2, pp. 195–208, 2013.

[31] G. A. Rummery and M. Niranjan, "On-line Q-learning using connectionist systems," Cambridge University Engineering Department, Tech. Rep. CUED/F-INFENG/TR 166, 1994.

[32] Z. Gabor, Z. Kalmar, and C. Szepesvari, "Multi-criteria reinforcement learning," in *Proceedings of the International Conference on Machine Learning (ICML-1998)*, 1998.

[33] K. V. Moffaert, M. M. Drugan, and A. Nowé, "Scalarized multi-objective reinforcement learning: Novel design techniques," in *Proceedings of the symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 2013, pp. 191–199.

[34] S. P. Singh and R. S. Sutton, "Reinforcement learning with replacing eligibility traces," *Machine Learning*, vol. 22, no. 1, pp. 123–158, 1996.

[35] J. Garcia and F. Fernández, "Safe exploration of state and action spaces in reinforcement learning," *Journal of Artificial Intelligence Research*, vol. 45, pp. 515–564, 2012.

[36] P. Geibel and F. Wysotzki, "Risk-sensitive reinforcement learning applied to control under constraints." *J. Artif. Intell. Res.(JAIR)*, vol. 24, pp. 81–108, 2005.

[37] Y. Shen, M. J. Tobia, T. Sommer, and K. Obermayer, "Risk-sensitive reinforcement learning," *Neural computation*, vol. 26, no. 7, pp. 1298–1328, 2014.

[38] L. Kamin, C. Brimer, and A. Black, "Conditioned suppression as a monitor of fear of the cs in the course of avoidance training." *Journal of comparative and physiological psychology*, vol. 56, no. 3, p. 497, 1963.

[39] A. Tversky and D. Kahneman, "The framing of decisions and the psychology of choice," in *Environmental Impact Assessment, Technology Assessment, and Risk Analysis*. Springer, 1985, pp. 107–129.

[40] B. De Martino, D. Kumaran, B. Seymour, and R. J. Dolan, "Frames, biases, and rational decision-making in the human brain," *Science*, vol. 313, no. 5787, pp. 684–687, 2006.

[41] T. U. Hauser, E. Eldar, and R. J. Dolan, "Neural mechanisms of harm-avoidance learning: A model for obsessive-compulsive disorder?" *JAMA psychiatry*, vol. 73, no. 11, pp. 1196–1197, 2016.

[42] G. Crombez, C. Eccleston, S. Van Damme, J. W. Vlaeyen, and P. Karoly, "Fear-avoidance model of chronic pain: the next generation," *The Clinical journal of pain*, vol. 28, no. 6, pp. 475–483, 2012.