Modern Value Based Reinforcement Learning: A Chronological Review

MARK C. MCKENZIE, (Member, IEEE), AND MARK D. MCDONNELL, (Senior Member, IEEE)
Industrial AI Research Centre, UniSA STEM, University of South Australia, Adelaide, SA 5001, Australia
Corresponding author: Mark C. McKenzie (mark.mckenzie@mymail.unisa.edu.au)

ABSTRACT Investigation of value based Reinforcement Learning algorithms exhibited a resurgence into mainstream research in 2015 following demonstration of super-human performance when applied to Atari 2600 games. Since then, significant media attention and hype have accompanied this area, and the field of Artificial Intelligence generally, spread across distinct categories. This review paper is focused exclusively on the progression of value based Reinforcement Learning in the last five years. We aim to distill the incremental improvements to stability and performance in this period, highlighting the minimal changes to the base algorithm over this time. This holds true with all but the one exception of the Recurrent Experience Replay in Distributed Reinforcement Learning algorithm, representing a fundamental shift and increase in agent performance through an advanced memory representation. We suggest a new focus area for value based Reinforcement Learning research.

INDEX TERMS Artificial intelligence, reinforcement learning, Q-learning.

I. INTRODUCTION Reinforcement learning is a distinct sub-topic of the broader field of Machine Learning [1], relating specifically to the notion of learning by experience, gained through actions of an agent/system in its environment, to achieve maximum long term reward [2]. The promise of Reinforcement Learning is grand: a model free agent which can interpret any environment, and learn a task to superhuman ability, all with minimal user interaction. This has spurred significant interest in the field, shown most notably by the volume of papers published since its most notable successes. However what is generally experienced is that Reinforcement Learning algorithms can be very difficult to construct and train to an optimal solution, as is widely discussed, for example as found in [3] and [4].

The current enthusiasm for Reinforcement Learning research is broadly considered to have been seeded by the work of DeepMind with the creation of the Deep Q Network [5], [6], and its success at completing Atari 2600 video games through the Arcade Learning Environment (ALE) [7]. The promise of Reinforcement Learning was then further enhanced by its role in enabling machine to triumph over humans at the game of GO [8], [9] or the complex real-time strategy game of StarCraft [10].

The scope of this review is highlighted in Section I-B, and dictates the the structure of the review as follows; Section II gives a brief history of the concepts in research that bore the field in its infancy, whilst Section III details the fundamental aspects of Value based Reinforcement Learning for consideration in the subsequent review, and details concepts which are inherently included in value based Reinforcement Learning approaches. Section IV covers the early work and threads of research which formed the basis of what is considered here as modern Reinforcement Learning. Logically Section V follows on to detail the critical approaches of Value based Reinforcement Learning in greater detail, highlighting key differences as the approaches evolved, leading into a discussion of observed results in Section VI and then conclusions of the review in Section VII.

A. THE NATURE OF REINFORCEMENT LEARNING

The Reinforcement Learning approach iteratively samples its available actions for high reward and after some period of searching, where the correct answer is not known, refines a course of actions to achieve its objective. Hence in Reinforcement Learning applications, an exploratory stage is required
and instruction given based on an expected outcome, honed over subsequent experiences.

The techniques applied in reinforcement problems, both for evaluation and instructing an algorithm, are often thought of as a conflict between exploration and exploitation [2], as discussed further in Section III-A.

The most important feature distinguishing Reinforcement Learning from other machine learning approaches is that the algorithm evaluates the actions taken to achieve some objective, and does not implicitly know the best solution. This is analogous to human learning mechanisms, where unforeseen events are progressively handled within the system with experience. As such, any successes in Reinforcement Learning for solving engineering problems are arguably a fundamental step towards true artificial intelligence, and hence the interest found here in this review.

B. SCOPE OF THIS REVIEW

In this review paper, we aim to highlight the current state of Reinforcement Learning research from the perspective of fundamental value-based approaches that have led to advancements in the field. With the breadth of research undertaken across the world, it is infeasible to address all permutations of the underlying Reinforcement Learning methodology, and as such focus will be given to the most adopted or successful approaches towards the current state of the art.

This review will briefly touch on the genesis of the value based Reinforcement Learning methodology and continue the evolution of Reinforcement Learning techniques from this important milestone, making comparison back to the original, or unaltered algorithm of Q learning.

The assertion made by [11] that a critical inhibitor to the success of Reinforcement Learning progress is the lack of reproduce-ability and non-standardized approaches to assessment and reporting are considered here to be paramount. This is particularly in comparison to supervised learning progress through the various benchmarks as shown in [12], listing the current state of the art on standard supervised learning tasks such as ImageNet [13] or CIFAR-10 [14]. Particularly when compared against supervised learning, there is a lack of standardization in reporting results in Reinforcement Learning, that makes it much more difficult to assess the relative contribution to the field of Reinforcement Learning, despite the presence of many suitable approaches [15], [16], [17]. This is important in relation to scope as it is intractable to consider approaches for criticality if they are not directly comparable in terms of performance.

This review is not an exhaustive list of all Reinforcement Learning research or the applicability of each approach to the full gambit of possible use cases, rather a chronology of the major advancements to the fundamental approach of value based Reinforcement Learning, where the approach is assumed to be equally applicable across discreet or continuous action space for example. There are countless examples in the literature where there are stated gains in overall performance at a given task through minor adjustments to the application of the fundamental approach; for example the division of the state space input into multiple policies [18]; or the smoothing of the Bellman equation to reduce statistical variation in the learning phase [19]. Whilst these are valuable contributions to the overall field of research, they are not the focus of this review.

Standing out amongst recent work, we especially highlight one recent advancement of significant note, as it includes a previously unseen aspect, being the inclusion of a specific memory term, which resulted in a dramatic increase in comparative performance. What is proposed here is that this contradicts a fundamental assumption of Reinforcement Learning, which is that the Markov Model sufficiently encapsulates current and all previous state space information by virtue of its action-value mapping. Corollary to this, there is likely additional performance gains to be made by incorporating higher level cognitive concepts into the agent’s decision making. It is purported here that research in the area of Reinforcement Learning will advance from this concept as a critical point in the field.

II. THE PRE-HISTORY OF MODERN REINFORCEMENT LEARNING

Modern Reinforcement Learning is characterised by Sutton and Barto [2] as a combination of two distinct historical research areas - trial-and-error search and optimal control or dynamic programming - with a third thread spanning these two fields referred to as temporal-difference methods. All threads came together in the 1980’s to become what is now known as Reinforcement Learning. Optimal Control attempts to minimize system behaviour variation over time. The Bellman equation, taken from optimal control research, is fundamental in the field of Reinforcement Learning. It specifies a value function to discern optimal actions or controls for a given objective [20], and is now the common term to describe the action-reward pairing in Reinforcement Learning applications. Inspiration for the technique came from Hamilton and Jacobi, which was subsequently extended to a discrete stochastic version referred to as Markovian decision processes [21]. An important attribute to note, referred to as the Markov property, holds where knowledge of the current state is all that is required to decide the best course of action, as illustrated graphically in Figure 1. All information contained in every previous state is said to be inconsequential, through inherent understanding incorporated in the current state. Perhaps the most notable example of optimal control is the work of [22] and [23].

The more common thread of trial-and-error search is more intuitive to humans due to the congruence with our own learning mechanisms, and has arguably received more attention in research as a result, where it is even claimed as an essential aspect of true artificial intelligence [24]. Early influential work in trial-and-error search techniques involved the playing of simple board games repeatedly to reward or punish moves.
post game [25]. This extended to more complex tasks such as the pole balancing task [26].

The temporal-difference method is the third thread of research contributing to the unified field of modern Reinforcement Learning, where temporal difference ideas were first applied to checkers in order to modify the function used for decision making online, which included what is known as secondary reinforcers [27]. Following this, research into Reinforcement Learning slowed until linkages were made to animal psychology by [28] and carried on by [29]. The unification of these threads came when optimal control and temporal difference were utilised in the $Q$-learning approach by [30] and [31].

III. VALUE-BASED REINFORCEMENT LEARNING CONCEPTS

Value-based Reinforcement Learning is the methodology of an artificial intelligence agent acting with priority towards a high reward state, within an environment through trial-and-error, assigning current and expected future value to all intermediate state-action pairing through gradient based optimisation. The agent then seeks out high value states given an exploitation policy to maximise overall reward. The Reinforcement Learning model is described in [32] as any learning agent connected to the environment through perception of the state of that environment and action taken within it.

In essence, Reinforcement Learning is a process of improving an agent’s ability to perform some task through experience, or trial and error. A fundamental process of evaluation feedback is the action-reward pairing, giving estimated reward at some future time, based on the learning agent and its interaction with the environment. An agent is any mathematical representation of the decision making process, acting on an environment [2]. Observations are made by the actor of the current state $s_t$, and a best action $a_t$ to take is estimated according to some policy $\pi_t(s, a)$ at any given time $t$, where $s_t \in S$, the set of all states, and $a_t \in A$, the set of all actions. Mathematically,

$$V_{\pi}(s) = \mathbb{E}_a \left[ R_t \mid s_t = s \right], \quad (1)$$

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \epsilon,$$  

where $\gamma$ is a discount factor to promote short or long term actions [2], [30]. This is graphically illustrated in Figure 2 where the relationship to the Markov Property is observable. This gives the concept of a Value Function in which the best course of action is calculated, based on the intermediate state values $V$ and selection policy $\pi$, to maximise the overall reward $R_t$. The agent predicts randomly at first, given that no understanding of the pattern of inputs and how they relate to outputs is yet understood. The system is then updated to minimise the error between the calculated outputs and the class states. All value based Reinforcement Learners are predicated on this fundamental principle, maximising the value function through maximum overall reward is optimal for the success of the agent at the given task.

![FIGURE 1. An illustrative example of the Markov model, showing probability $P$ of state transitions, dependent only on the current state of the system under the Markov property.](image)

**FIGURE 2.** The widely accepted Reinforcement Learning model, showing the agent taking an action $A$ to update the state $S$ and receive reward $R$.  

A. EXPLORATION AND EXPLOITATION

A critical differentiator between Reinforcement Learning and other machine learning approaches is the requirement to explore the action space [32]. The reward which is obtained by a selected action for a given state is not known $a$-priori, and an agent must build the understanding within the model through trial and error. The issue with this is that exploration implies sub-optimal action selection to avoid local minima. There are many techniques applied to ensuring adequate exploration of the state space, the most common of which is $\epsilon$-greedy, which involves the exploitation action being the action with highest reward, being selected unless some small probability $\epsilon$ criteria is reached, in which case a random action is taken. Numerous incremental enhancements to the task of exploring the state space have been made; by using the accuracy of the reward estimate to signal an under sampled state action pair [33]; controlling the amount of available actions for a given state, thereby controlling the probability of exploration actions [34]; by introducing noise into the parameter space to introduce stochastic variation into the state-action pairing [35], [36]; or by simply tracking the relative frequency of action selection for a state and altering the probability for under sampled actions [37], [38], [39], [40].

It is considered here that the different approaches to ensuring the state space is adequately explored do not fundamentally change the Reinforcement Learning method, being an
operation equally applied across any Reinforcement Learning technique. As such, in-depth review of each approach is out of scope of this review, noting that several research articles state that it made a marked improvement to overall success of the technique, and an important area of research in it’s own right.

B. MODEL FREE AND MODEL BASED REINFORCEMENT LEARNING

Model based learning involves the determination of a complete understanding of the environment and the relative reward of each action at each state. This was the standard approach historically, and commonly referred to as dynamic programming. The critical failing of the approach is it is intractable for all but the simplest of environments. The growth from model-based to model-free Reinforcement Learning allowed the application to complex environments, and is widely applied in modern techniques in the form of neural networks. The origins of neural networks are based in pattern recognition and perceptual learning inspired by the understanding of neural systems in mammals [41], [42], [43]. At the simplest level, a single neuron provides a binary decision, which can be stacked to provide high level interpretation of information [44]. For a given input signal \( \mathbf{x} \) an output signal \( \mathbf{y} \) is calculated having passed through some number of hidden nodes as multiples of neuron link weights \( w \) and activation function \( \sigma \). Following on from the initial application of backpropagation in neural networks [45], the technique has been applied in a variety of forms, and have led to augmented algorithms which represent the current state of art in numerous machine learning fields, Reinforcement Learning included amongst them. The fundamental approach to achieve backpropagation is stochastic gradient descent [46], in which the easily measured error at output (easy in supervised learning approaches) is propagated through the layers by calculating derivatives, with the current implementation simplified using the chain rule [47].

C. DEEP AND SHALLOW REINFORCEMENT LEARNING

It is noted in [48] that the main focus at present in the field of machine learning is in the area of supervised learning, which is essentially applying a model to achieve a known category of labelled data. This is extended to Deep supervised learning through application of many layers of convolutional filtering operations applied to more complex input states. Early work on deep networks used linear regression methods applied to polynomial activation functions [49], but the objectives and benefits to augmenting the input information to emphasise underlying data structure remains the same [48]. The first architecture to incorporate the visual cortex inspirations, which are now known as convolutional neural networks, was titled the Neocognitron [50], [51] and incorporated a set of convolutional filters, as is the common practice today, with the exception of the absence of backpropagation of errors into the filter weights. The modern field of Deep Belief Networks came when the error backpropagation was combined with unsupervised pre-training [52], [53], where Restricted Boltzmann Machines [54], [55] were stacked to encode then reconstruct original data and hence provide low level discriminants to high level image data. Deep Learning has subsequently become a major field of research, with the ability to provide human level representation of data independent of in depth intervention from the human writing the algorithm [56], a fundamental weakness in previous machine learning approaches [57]. Following the initial application of neural networks to Q-Learning algorithms in Reinforcement Learning [58], the technique of deep learning was incorporated to Reinforcement Learning to play Atari 2600 games as the Deep Q Learner [5], leading to the state of the art performance which could be credited with rejuvenating the field of research under review here: value based Reinforcement Learning.

Deep Reinforcement Learning is not fundamentally different to shallow Reinforcement Learning except in that it applies the concept of convolutional neural networks to the input state, which is generally imagery within the literature. Deep Learning convolutional neural networks are neural networks that include convolutional layers, which are known to improve performance of the system through enabling translation invariant predictions and enhanced representation of input features [59]. This has broadened the potential application space significantly, but again does not significantly alter the fundamental technique applied; it is subsequently assumed in this review to be ubiquitous to the underlying Reinforcement Learning algorithm, Deep or Shallow.

IV. GENESIS OF REINFORCEMENT LEARNING

The early research into Reinforcement Learning focused on the now considered simple tasks of common board games, such as backgammon, chess or checkers. However at the time it was revolutionary when a human was beaten at these games by a machine. Whilst these techniques have been dramatically superseded by modern algorithms, many of the fundamental operations remain. The following section of this review follows the chronology of these major milestones in the application of Reinforcement Learning, and hence the growth of the field of research.

A. TEMPORAL DIFFERENCE LEARNING

Arguably the first high profile application of what is now the modern field of Reinforcement Learning is the TD-Gammon algorithm, utilising temporal difference learning to defeat human players at the game backgammon [60], [61]. The temporal difference learning was then used to defeat human players at chess [62]. Note that IBM’s Deep Blue was far more famous, defeating the chess champion, but this approach was not Reinforcement Learning based, arguably not even machine learning as it was a simple alpha-beta search using brute force [63], [64].

Temporal difference learning is described in detail by [2], where the term comes from the target update, or error term for backpropagation being essentially the difference in the measured future value and the estimated future value,
moderated by a discount factor $\gamma$. The policy target in a temporal difference learning algorithm is then the sum of the experienced reward and all future rewards, under the assumption it is known, mathematically given as

$$r_{t+1} + \gamma V(s_{t+1}).$$  \hfill (3)

Hence, the states’ value estimate is updated according to

$$V(s_t) \leftarrow V(s_t) + \alpha [r_{t+1} + \gamma V(s_{t+1}) - V(s_t)].$$  \hfill (4)

where $\alpha$ is the learning rate parameter.

TD(n) and TD($\lambda$) are an extension to the fundamental approach of temporal difference learning in which a number of steps taken are used in each bootstrap update [2]. In the latter case the relative weights of each step are altered by the value of the $\lambda$ term.

A limitation of the approach is that each state must be sampled sufficiently in order to converge to the true value at that state. This is intractable for modern applications as for most cases the environment is continuous, and hence there are an infinite number of states. As was stated previously, the approach was made famous by the work of [60] titled TD-Gammon by beating a human at backgammon, however it should be noted that backgammon does not suffer this limitation as there are only a relatively small number of states on the board. Since then, a number of minor alterations have been made to the technique for specific applications, without fundamentally altering the approach; [65], [66] for example incorporated tree search into the temporal difference technique.

A similar application of the temporal difference to the famous TD-Gammon learning approach was by [67] for the learning of the game draughts, or [62] for learning chess. As such, due to its relative utility in applications in early machine learning research and influence on successive implementations, it is considered here one of the fundamental steps toward the modern approach of Reinforcement Learning.

B. REINFORCE

A class of reinforcement algorithms called REINFORCE were proposed by [68] to maximise immediate reward, built on earlier work by the same research [69], [70], in which the major contribution to modern Reinforcement Learning comes through the application of function approximators for the statistical connection layers, and the learning through backpropagation that it facilitates. Despite this, as is noted by [71], the method fails to learn a value function and as a result learns much slower than other approaches, and for this reason has experienced much less attention in subsequent research. However it should be noted that the methods have influenced a branch of Reinforcement Learning which is outside of scope as it is not value based, referred to as Policy Gradient methods.

C. SARSA

Up until this point, Reinforcement Learning approaches have only addressed discreet state space problems, reliant on the ability to sample each state space sufficiently to determine all transitions [72]. Progressing the field of Reinforcement Learning to be capable of generalising continuous state spaces was a major leap forward by [72] in the development of the algorithm which would later be referred to as SARSA in the work of [2], which is an acronym for State, Action, Reward, State Prime, Action Prime in reference to the inputs to the update phase. The approach presented by [72] extends directly on the work of [30] for the underlying update approach and [73] in the use of neural networks. The fundamental principle of the leap forward in Reinforcement Learning research was the update of the state-action policy at each time step in an on-line methodology, with improved convergence towards an optimal policy compared to previous approaches.

In essence the only difference between temporal difference learning and SARSA is the extension of the algorithm detailed in Equation (4), from the mapping of state to reward value $V(s_t)$ to state-action pairs $Q(s_t, a_t)$ to reward values,

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)].$$  \hfill (5)

This is an on-policy method in the fact that only visited states and chosen action pairs are updated. In the event that each state-action pair is visited an infinite number of times, the approach has been shown to converge to an optimal solution.

There have been numerous successful applications of the SARSA algorithm, for example variations of the approach detailed in [72] and [74] were used by [75] to prove bootstrapping for online Reinforcement Learning is preferable to Monte Carlo methods. Even modern research applications still use this fundamental approach, i.e. [76] and [77], and in a swarming application [78].

D. Q-LEARNING

Two papers [30], [31] are credited with the development of the Q-learning approach through the application of the Bellman equation [20], [79] to Markov decision processes. The breakthrough was a form of learning through action and response, which has seen subsequent research apply the technique to some high profile applications, including defeating human world champions at numerous tasks, some of which are highly complex. This now forms the basis of modern Reinforcement Learning.

Built on the concept of Markov decision processes, Q Learning relies on the concept of each state $s$, having a probability to transition to another state through some action $a$ gaining some reward $r$. The Markov model is graphically represented in Figure 1. If all rewards obtained for every action across all states are known, the optimal policy to achieve maximum reward is obvious.

Watkins [30] determined that the total discounted reward was the success criteria to optimise toward, based on the premise that reward now is worth more than some
future reward. This was mathematically represented by the discount factor $\gamma$ in the following mathematical expression

$$r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^n r_{t+n} + \cdots.$$  

(6)

This leads to the concept of Value where any given state has an expected value given by the highest reward achieving set of actions, articulated mathematically as the function

$$V_\pi(s) = E_t \left[ r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^n r_{t+n} + \cdots | s_t = s \right].$$  

(7)

This gave rise to the term Value based Reinforcement Learning, noting the $\pi$ term denotes the policy of selecting actions at each state. The Q function is the mapping of reward to each state-action pair,

$$Q(s, a) = E \left[ r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t = s, a_t = a, \pi \right],$$  

(8)

where the accumulated reward $Q$ is a function of the observation state $(s)$ and proposed action $(a)$. The optimal policy is then found by maximising the reward, which is as simple as always choosing the action which yields the maximum total discounted reward, i.e.

$$Q^*(s, a) = \max_{\pi} E \left[ r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t = s, a_t = a, \pi \right].$$  

(9)

The one step update of the Q network improves the policy over consecutive updates, and is mathematically shown as

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right].$$  

(10)

The optimal return $Q^*$ is the maximum reward possible given the rewards $(r)$ at each future time step $(t)$ discounted by $\gamma$. Of course this is predicated on the need to know accurately the reward expected at each state and action, which is the only shortcoming of the original research, in that it required a further leap forward in order to provide the critical mechanism to learn the reward as opposed to expecting it to be known, ultimately leading to the subcategory of artificial intelligence referred to as Reinforcement Learning. This is the endeavour of the subsequent research within modern Reinforcement Learning. Despite this, a number of successful applications of the concept in this fundamental form have been found [80], [81].

**V. MODERN REINFORCEMENT LEARNING**

The renaissance of Reinforcement Learning in research today can be linked to a highly cited body of work by Google’s DeepMind project [5], [6] identified as the Deep Q Network (DQN). In the context of this review paper, modern value based Reinforcement Learning is considered as approaches building on, or post publication of this work announcing super-human performance against Atari games and of course the DQN itself.

**A. DEEP Q LEARNING**

What is referred to here as the vanilla DQN method is introduced in the papers [5], [6] from the Google DeepMind group following on from the underlying research into Reinforcement Learning of arcade games [82], [83]. This approach generated significant attention to the field of Reinforcement Learning and Machine Learning in general due to the reported Super-human performance of the algorithm against a high proportion of tested Atari games, as summarised in Figure 3, showing the stated performance results relative to human performance normalised by the random score of each game.

Two major additions by [6] within the DQN methodology allowed the standard Q Learning approach to converge given the extreme increase in state space complexity. The first was experience replay, where essentially the network is trained periodically on a randomized historical sample batch to remove correlations and smooth over the data changes in the data distribution. This process was biologically inspired by the theory that mammalian brains re-enact past experiences during periods of low activity [85], [86]. The second technique is referred to as delayed update, where the target state is held constant for a period of time by remembering the network architecture between updates. The new or updated network’s reward is calculated for a given state using the Q update function given previously as Equation (10), and this is compared to the reward which would have been experienced by the stored network state [6]. The update to the network is then based on the loss function $L$,

$$L_i(\theta_i) = E_{(s, a, r, s') \sim (U(D))} \left[ \left( r + \gamma \max_a Q(s', a; \theta_i^+) - Q(s, a; \theta_i) \right)^2 \right].$$  

(11)

where $\theta_i$ represents the network architecture (weights) and $\theta_i^+$ represents the stored network of the delayed update step. A selection of uniformly distributed samples $(U(D))$ of the state variables $(s, a, r, s')$ is drawn from the stored sample pool and used for the assessment of the loss function. The network architecture is then updated periodically; this effectively keeps a constant target state for a period of time.

The platform used to test the Q learner was an emulation of the Atari 2600 game console which supports 49 games as listed in Figure 3. The emulator utilises $210 \times 160$ color pixels at a 60 Hz frame rate and a color palette of 128, with the Q Learner only having access to the visual image and the number of available actions in which to make decisions [6]. The actions available form the final fully connected layer of the neural network. Comparisons were made against other approaches: (i) the Best Linear Learner and (ii) the Contingency awareness (SARSA) [82], [83]. The DQN approach was largely very successful in comparison to these two methods, which in many cases could not surpass a human player [6]. This comparison shows that at the time in which it was published the approach represented the state-of-the-art for Reinforcement Learning.
The deep Q-learning method of Reinforcement Learning has been subsequently used in countless applications and studies since its inception, e.g. [87], [88], [89], [90], [91], and [92]. Furthermore, the successes made by the approach have spawned a number of advancements to the field, as will be discussed in subsequent sections.

**B. DOUBLE Q-LEARNING**

The first major alteration to the vanilla DQN approach came in the research of [93] carrying on from earlier work [94], referred to as Double Q-Learning, or Double DQN (DDQN). The research identified that there is commonly an overestimating in the value function, causing biased selection of actions already sampled. It has been suggested that the instability is a result of correlations in the observation sequence and policy changes due to small updates to the \( Q \) function, i.e. a non-stationary action-value function [95]. The DDQN approach extended on the original research by decoupling the state-action function for determining value from the target network which is updated online at each step sample of the game. In effect, the approach holds in memory a sample of the agent’s weights \( \theta \) for assessment of the state-value pairing and selection of actions, whilst bootstrapping the target network weights \( \theta' \), periodically assigning the target weights to the agent. This does not alter the \( Q \) function described in Equation (8), but changes the update function from Equation (10) as follows,

\[
Q'(s_t, a_t) \leftarrow Q'(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma \max_a Q(s_{t+1}, a) \\
- Q'(s_t, a_t) \right].
\] (12)

At the time, this incremental improvement to value based Reinforcement Learning was shown to be the state of the art, outperforming DQN at the fundamental test of Atari 2600 games [96], with comparative results as shown in Figure 4. It is important to note here that the paper included updated performance values against the expanded set of games, and is hence used for comparison going forward.

This variant of the value based Reinforcement Learning approach is a seemingly small and easily implemented change to the base approach; resultantly it has been widely applied in the literature, e.g. [97], [98], [99], and [100], with a number of suggested incremental improvements [101], [102]. Importantly, the Double DQN methodology has been widely accepted and used in subsequent critically important research, e.g. [103].

**C. PRIORITIZED EXPERIENCE REPLAY**

Up until this point in the timeline of Value based Reinforcement Learning, the concept of Experience Replay was well accepted and implemented as it provided fundamental benefits to the technique, in that it reduced the high variance associated with individual updates of the \( Q \) score, as well as breaking the correlation between samples by mixing the training data set in time, more appropriately approximating the independent and identically distributed sampling assumption [104]. The failing of the approach is simply that each sample in the training data set is given equal weight, and this does not give proper appreciation to state-action pairings which result in significant improvements to the total reward in comparison to events which have little or no affect. Building on top of the Double DQN methodology discussed previously, the work of [103] introduced an incremental advancement to the field of value based Reinforcement Learning through a technique termed Prioritized Experience Replay, and as the name suggests, gives priority in agent updates to samples with greater training effect when drawing samples from the experience replay buffer.
FIGURE 4. First four relative performances of Reinforcement Learning research, data provided by [84]. Any games for which the score increased or decreased by factors larger than 10 are truncated in the figures.
The underlying concept of Prioritized Experience Replay stems from idea that transitions, i.e. the selection of action based on state, yielding a predicted reward, can be surprising in that the prediction is wildly wrong, redundant in that it has been observed regularly, or highly relevant such as if it yields high reward [103]. The research suggests to select training samples with a high expectation of learning progress, proportional to the temporal-difference (TD) error δ, a process first suggested by [105]. Recalling earlier that TD-error is the difference between expected and experienced reward in an update, representing the update component in Equation (4). The paper further states that this can cause a loss of diversity in training samples, requiring a normalisation process. Mathematically the process to determine sampling probability P of sample i is

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}. \quad (13)$$

The priority of a sample $p_i$ is calculated as the ranking of TD-errors, i.e. the greater the error the higher the priority, or the absolute value of the error itself; ranking is more robust to outliers [103]. The $\alpha$ term is a hyper-parameter which is strictly greater than zero, and controls the strength of the prioritization selection. Bias in the estimation is introduced in the process by changing the sampling distribution, and removed by applying the importance-sampling weights to the TD-error term in all updates of the agent, which are calculated according to the function

$$w_i = \left(\frac{1}{N} \cdot \frac{1}{P(i)}\right)^\beta. \quad (14)$$

Once again, the state of the art in value based Reinforcement Learning has been progressed through this incremental improvement, combining Double DQN with prioritizing experience replay [103], demonstrated on the field’s seemingly accepted metric of success, the Atari 2600 games, as shown in Figure 4b.

There are numerous fields of application for the approach of value based Reinforcement Learning presented as Prioritized Experience Replay, such as online object recognition [106], robot navigation [107], or simple network and traffic routing as stated in [108]. A similar concept has also been proposed using strata of experiences and selecting amongst the stratus to promote important samples [109]. A similar concept was proposed by [90], referred to as Hindsight Experience Replay, in that it makes better use of the experience replay buffer, but different in that it focuses on the case where multiple goals are inherent in the agents task. This approach was later combined with subsequent research to again claim state-of-the-art status as the Prioritized Dueling Experience Replay approach [110].

D. DUELING DEEP Q NETWORKS

Early work by [111] suggested that separating the advantage and value functions improved performance of the Q Learning Reinforcement Learning algorithm. An extension of the work for deep convolutional neural networks [84] was applied to the common Atari 2600 baseline, achieving state-of-the-art results at the time of publication (noting that this was achieved when combined with previous concepts such as DDQN and Prioritized Experience Replay) to instantiate the method commonly referred to as Duelling Deep Q Networks; see Figure 4c for relative performance assessment against the previous state-of-the-art at the time, Prioritized Experience Replay.

The major contribution to the field comes in the separation of the Advantage term from the Value term in agents network. This concept is considered here to be one of the first proposed methods involving a derivative of the standard feed forward convolutional network topology commonly applied in research. A clear comparison is made by [84] in Figure 5, where the scalar output of Value V estimate for a given state is determined separately from the vector of Advantage A terms. These terms are necessarily combined in selecting an optimal action to take for the given state.

![FIGURE 5. A comparison of network architectures for the standard Q Network approach to date, and the discussed Dueling network architecture.](image)

Most important in this approach, is an understanding of what the Advantage term actually represents, where mathematically it is simply the difference between the expected value for all actions at a given state and the actual value for the optimal policy $\pi$ action set,

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s). \quad (15)$$

The Advantage nomenclature reflects the relative update contribution, in that the optimal action to take for a given state would estimate the value to be the closest to the true value (even if there were discrepancies due to insufficient training samples to accurately estimate the true value at that state), providing the least Advantage in learning. Conversely, a state-action pairing which was significantly different from the true value, resulting in a larger Advantage output from
the network. This is stated to have the benefit to training the agent through better utilisation of these non optimal state-action pairings [84], accelerating learning and reducing uncertainty in training.

A critical aspect of the Dueling Deep Q Network approach is in the combination of the Advantage and Value terms, which again is simple mathematically, i.e.

\[
Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + \left( A(s, a; \theta, \alpha) - \frac{1}{|A|} \sum_{a'} A(s, a'; \theta, \alpha) \right).
\] (16)

The important aspects to note here are in that the state-action updates are dependent not only on the Value predicted by the agent, but also its perceived Advantage, relative to the advantages across the entire set of actions. This is considered to add stability to updates, whilst not directly altering the selection of actions, which is still based on expected value of current and future rewards.

The technique of Dueling Deep Q Networks has been widely applied in the literature, in diverse fields of application such as traffic signalling [112], estimating algorithmic parameters [113], UAV control [114], and administering pain relief [115]. This is perhaps an indication of the stability of the approach resulting in widespread adaptation. In addition, there are instances in literature where the process has been extended [116], but also incorporation into other approaches to address the Reinforcement Learning field of research [35], [117], [118].

E. NOISY NETWORKS

Another highly successful approach to achieving the task of value based Reinforcement Learning was proposed by [35] and self labelled as NoisyNet. This work leveraged a well researched concept from supervised learning in which noise is added to the neural network to improve overall classification success following class optimization [119], [120], but is the first to apply the concept to Reinforcement Learning, which is an arguably more complex subgroup of artificial intelligence. It was identified by [121] that the act of randomly perturbing the selected action is unlikely to fully access all states of the environment in any form of efficient manner. Whilst this seems relatively intuitive, and despite the large body of work to find more effective methods [122], [123], [124] a vast majority of applications still use the simple e-greedy approach. The major contribution of the work by [35] is considered here to be the instantiation of a problem-specific exploration strategy without the need for random perturbation of action selection or the need for researcher input, replacing the fixed and often decreasing exploration strategy. This major contribution was in addition to the reported state-of-the-art performance against the Atari 2600 games benchmark (as shown in Figure 4d) comparing the NoisyNet concept applied to both vanilla DQN and Dueling DQN respectively. It is interesting to note that the combination of NoisyNet with Dueling showed a bigger net improvement over the Dueling network than the alternate comparison against the vanilla DQN approach.

In essence the NoisyNet methodology adds parametric noise \( \epsilon^w \) to the fully connected layer weights (\( \mu \) in this nomenclature as opposed to the standard \( w \)) and biases of a deep or shallow neural network, to cause variation in the resultant output from the layer, and hence in the value based Reinforcement Learning context. In turn this produces variation in the estimated value of the current and future states given some policy and hence in the optimal action selection. The crucial element however is in the progressing update to that parametric noise contribution \( \sigma^w \). Noting that the standard representation of a fully connected layer neuron with input \( X \), bias \( b \), layer weights \( w \) and output \( y \) is given mathematically as

\[
y = f \left( wX + b \right),
\] (17)

where the function represents any number of continuous differentiable functions, such as rectified linear units. The NoisyNet approach extends this as

\[
y \overset{df}{=} f \left( (\mu^w + \sigma^w \circ \epsilon^w)X + \mu^b + \sigma^b \circ \epsilon^b \right).
\] (18)

Comparing the two representations of the neuron for the standard and noisy fully connected layers, it is observed that the NoisyNet variation is in the addition of the noise component, where the \( \sigma \) terms are progressively updated at each learning update.

Like the methods discussed up until now, the NoisyNet Reinforcement Learning approach has wide applicability across multiple tasks, for example UAV navigation [125], autonomous ground vehicle control [126] or even conversational dialogue [127].

F. DISTRIBUTIONAL REINFORCEMENT LEARNING

A subsequent methodology proposed by [128], and identified here as the Distributional Reinforcement Learning method (often referred to as C51), raised the issue that the value estimate is not a singular value for a given state-action pairing. Instead it is a distribution of probable values in which the expected value of that distribution is what has been optimised against up until this point in the Reinforcement Learning field of research. This work by [128] acknowledges the work of [129], [130], and [131] in identifying the underlying relationship, but extends the research importantly by applying it to the value estimate, noting that a Gaussian distribution was applied previously by [132] and [133]. The result was state-of-the-art performance against the Atari 2600 benchmark, with relative performance against the highest results obtained by DQN, Double DQN, Dueling DQN or Prioritized Dueling Experience Replay cited in [110], as shown in Figure 6.

The fundamental variation proposed by the method of Distributional Reinforcement Learning is the change from a Q function representing a state-action estimate of expected current and future value, to a value distribution, denoted \( Z \),
(a) Distributional Reinforcement Learning agent performance baselined against previous state-of-the-art performance of Noisy Network Architecture, against Atari 2600 games.

(b) Stated performance improvement of the APE-X Reinforcement Learning agent, relative to the previous state of the art Distributional Reinforcement Learning agent, against Atari 2600 games.

(c) Stated performance improvement of the RAINBOW Reinforcement Learning agent, relative to the previous state of the art APE-X agent, against Atari 2600 games.

(d) Stated performance improvement of the QR-DQN Reinforcement Learning agent, relative to the previous state of the art APE-X agent, against Atari 2600 games.

**FIGURE 6.** Second four relative performances of Reinforcement Learning research, data supplied by [118], [128], [134]. Any games for which the score increased or decreased by factors larger than 10 are truncated in the figures.
in which the expectation is the Q score or value in previous methods [128]. This is expressed mathematically as

$$Z(s, a) \equiv R(s, a) + \gamma Z(s', a')$$ \hspace{0.2cm} (19)$$

It was considered here somewhat self-evident that the expected value for any state-action pairing would in fact be a distribution, as opposed to a single numeric value. However, what was interesting to note was that the relative improvement over the previously set benchmarks was minor (<5%) in most cases, suggesting that the value distribution is approximated reasonably well by the expectation value, at least for the deterministic Atari 2600 games.

The Distributional Reinforcement Learning approach was later extended to include other assistive techniques, namely Prioritized Experience Replay to form the Distributed Prioritized Experience Replay approach [135], and to include quantile regression [136]. Additionally, there exists a number of examples of the algorithms direct utilisation for real world learning tasks, such as resource management [137], robot control [138] and financial investing [139].

G. ASYNCHRONOUS & DISTRIBUTED REINFORCEMENT LEARNING

Building on work by [140] and [141], a class of Reinforcement Learning algorithms were first proposed by [142] (and similarly by [143]) involving the parallelism of learning agents across CPU threads, acting independently from one another, and labelled as the General Reinforcement Learning Architecture or Gorila algorithm. Whilst the approach uses the fundamental Double DQN algorithm of value based Reinforcement Learning in its application with no fundamental changes to speak of, it is notable due to the stated improvements in speed of learning primarily, but also improved performance against the base Double DQN agent against most of the Atari 2600 benchmark, as shown in Figure 6b.

The GORILA approach maintains an agent as well as a second target agent as in the Double DQN approach, along with an experience replay buffer which is either local or global in scope. The difference this approach has in comparison with previous methods is in the bank of N network parameters being maintained, each acting independently of each other on an instance of the environment, and processed asynchronously across the machines and a common parameter server. Periodic update of the target network is made from the bank of parameters being maintained. It was stated that the approach reached the results up to 20 times faster than the standard GPU based approach [117].

The concept of applying Reinforcement Learning across distributed processes was further developed by [117] in what was described as asynchronous Reinforcement Learning. The approach was applied across multiple variants of the Reinforcement Learning task, including value based one-step DQN agent, although the most noted or successful application was in the policy based variant labelled as Asynchronous Advantage Actor-Critic (A3C) which is out of scope for this review, being focused on value based Reinforcement Learning. The contribution of the asynchronous method to Reinforcement Learning is in the use of multiple CPU threads instead of multiple machines, drastically simplifying implementation. It was also noted that the approach varied the exploration policy across threads, implemented by varying the epsilon greedy parameter, and most importantly, the absence of an experience replay buffer. This is of critical importance to note as it is the presence of this buffer which is considered here as a primary contributor to the computational load of the Reinforcement Learning problem, not the processing of the agent. What this showed was that sampling widely enough in a bootstrapping process could mitigate the need for maintaining a large sample of older experiences, but did not address if there was a benefit to be had by doing so.

H. DISTRIBUTED PRIORITIZED EXPERIENCE REPLAY

Ape-X is an approach to Reinforcement Learning introduced by [135] which has combined the concept of massive parallel processing of the interaction of the agent with the environment (a processing bottleneck in Reinforcement Learning) with the concepts of Prioritized Experience Replay, which determined that not all experiences of an agent are of equal importance for the value update in back-propagation. A critical aspect of the approach to note is that the parallel ‘actor’ agents are building and maintaining a local experience replay buffer, which are drawn from globally by the common ‘learner’ agent. The individual ‘actor’ agents maintain the local buffer in terms of updating the priorities in accordance with the target network updates. The paper by [135] showed state-of-the-art performance on a number of the Atari benchmark games, and substantial improvements to learning rates, proclaimed to be a result of the increased exploration of the environment state space, but it is also noted here that the prioritization of experiences is a major contributor to performance over the asynchronous or GORILA methods. Importance Weighted Actor-Learner Architecture (IMPALA) [144] extended on this to also make the ‘learner’ agent asynchronous using a V-trace method to map experiences to gradients in asynchronous updates of the agent. This was only applied at that time to a policy based Reinforcement Learning approach, but could be applied to value based approaches, however this is again out of scope for this review.

I. RAINBOW REINFORCEMENT LEARNING

At the time of its publication, the Rainbow algorithm [10] was reportedly the state of the art in value based Reinforcement Learning approaches. The major advancement on previous incremental improvements to the value based Reinforcement Learning methodology was to combine several previous approaches into a single algorithm; namely the double DQN, prioritized experience replay, duelling DQN, multi-step learning, distributional Q networks and noisy networks. Each of these approaches have been discussed at length here, where each contribute toward the
overall success of the agent in correcting for different issues within the original DQN method. The only contradiction to this comment is that it was stated by [118] that the contribution of the double DQN to overall performance was negligible, based on ablation studies, i.e. simply removing that section of the code from the processing. Most likely the effect of the double network was superseded by one or more other included approaches.

The most interesting observation of the Rainbow methodology is that it dramatically improves on a number of published benchmarks of performance across the individual components of which it is constructed, and notably the vanilla DQN method itself, whilst remaining in essence a variation on the original algorithm. Figure 6c shows the relative performance of the RAINBOW Reinforcement Learning algorithm against the previous state-of-the-art results of the NoisyNet algorithm.

The only major variations in each individual sub-component of the RAINBOW algorithm from the original research, is the temporal difference error was not applied to determine the priority of the experiences for replay, instead the Kullback–Leibler loss is used, and that both streams of the fully connected layers in the dueling network architecture has the distributional probabilities applied [118].

Despite the improved success of the RAINBOW Reinforcement Learning approach over the widely applied DQN variants, there have been limited published applications in the real world, but some examples of the extension to the technique are found in [134], [145], and [146]. This is undoubtedly a factor of how recently it has been first published.

J. IMPLICIT QUANTILE NETWORKS

The work by [136] expanded the concepts of the value function being a distribution, which should be modelled or learned explicitly [128] as well as the purported benefits of parallel processing of learner and actor agents [117], [142] to create the Quantile Regression DQN (QR-DQN) approach. The most significant change proposed by the approach over Distributional DQN is to separate the return distribution into $N$ quantiles, taking advantage of the increase to speed and sampling distribution increase identified previously. This approach has resulted in state-of-the-art level performance at the time of publication against the Atari 2600 benchmark [136], with results shown in Figure 6d.

Implicit Quantile Networks is stated by [134] to be an extension of the distributed Reinforcement Learning approach [128] and the authors own work [136], where parameter sets are distributed amongst the learner agents, returning an implicitly defined (hence the name) return distribution, resulting in a risk-sensitive policy. Here risk is defined as the possibility of under-sampling, or not sampling the entire breadth of possible state-action pairing. The benefit of the approach was stated as being the increased sampling space with improved learning efficiency, giving an overall improvement within the same learning time as previous techniques.

K. RECURRENT EXPERIENCE REPLAY IN DISTRIBUTED REINFORCEMENT LEARNING

The approach of Recurrent Experience Replay in Distributed Reinforcement Learning by [147] has been coined as the R2D2 algorithm. Fundamentally the algorithm utilises a prioritized experience replay with a double DQN agent, and leverages the benefits discussed previously in sampling efficiency and depth provided by distributed concepts [134], [135], [136], [144], i.e. multiple actors operating on environment instances. The advancement made by the research of [147] is in the application of Long Short Term Memory [148] to build a deeper memory representation, in excess of what was commonly applied as a frame buffer of recent states to some level (typically four), although the use of LSTM had been applied to Reinforcement Learning previously as the Deep Recurrent Q-Network (DRQN) [149] and in other work [117], [144], [150], the key point of difference here is in the use in conjunction with the other advances in Reinforcement Learning, i.e. experience replay. The R2D2 also applied the principles of partially observable Markov decision processes (POMDP) [151], although in the context of the Atari benchmark, it is noted that this would be an insignificant contributor to success as the task is fully observable [147]. What was profoundly noticeable from this alteration to the previous Value based approaches described

![Figure 7. Chronology of mean performance of value based reinforcement learning, normalised to the DQN benchmarked scores.](image-url)
here, is the dramatic increase in overall performance against the Atari 2600 benchmark, as shown in Figure 8b.

The results of the R2D2 algorithm against the common Atari 2600 benchmark is a substantial improvement over previous efforts, with a minor combination of some previous concepts and a concerted effort to construct an effective combination, and the discovery of the need to ‘Burn in’ or avoid zero state initiation of the Recurrent Neural Network. The research also suggests that the impact of adding in recurrence into the neural network is more significant than simply improving the memory of the agent, improving somehow the learning representation, although this is not elaborated further. What is obvious is that the common understanding that the Markov model contains sufficient state information or representation to encapsulate all previous information should be challenged.

FIGURE 8. Final three relative performances of Reinforcement Learning research, data supplied by [134] and [147]. Any games for which the score increased or decreased by factors larger than 10 are truncated in the figures.
Despite the well noted success of model based Reinforcement Learning algorithms such as the famous AlphaGo [8] or Checkers [152], it was considered that model-based Reinforcement Learning approaches were out of scope for this review, as it is not a fundamental deviation from the standard value based approach. This position was taken as it was considered that having access to a model of the environment or rule set was a significant advantage, and somewhat contrived due to the absence of such knowledge in...
real world applications. The MuZero algorithm proposed by [153] however, demonstrated substantial improvements over the model-free approaches discussed throughout this review, and exceeds the previous state-of-the-art performance of the R2D2 algorithm by combining forward looking into a number of steps of a environment dynamic model to choose the best course of action in a given state.

The MuZero approach conducts planning by estimating the policy, the value function and the reward for the current transition [153], extending on work by [154], [155], and [156].
which represented a change in approach to model-based Reinforcement Learning by predicting the value function. The focus on learning a model of the value function makes it relevant to the context of this review, as it represents a possible step change in value based Reinforcement Learning. While noting that the MuZero itself is actually a policy based approach, it is worth comparing the performance of MuZero approach to that of the R2D2 algorithm, as shown in Figure 8c.

VI. COMPARISON OF MAJOR ADVANCEMENTS IN VALUE BASED REINFORCEMENT LEARNING PERFORMANCE

What has been discussed throughout the sections of this review, is a history of some important advancements in
value based Reinforcement Learning methodology. But as has been highlighted in research into benchmarking performance, there is a lack of commonality and repeatability to the results shown against the Atari 2600 test set [11], [16]. This makes direct and fair comparison of the performance or efficacy of the various approaches in literature difficult. Commonly the benchmark of success is the notion of superhuman level of performance, i.e. by how much does the algorithm surpass a human’s ability to complete the task. This ratio of the DQN performance against the level of a human player is shown in Figure 3, but it is important to note that here, what is or isn’t considered superhuman performance is not in question, where the interest lies in how much the field has advanced in comparison to its inception. The analysis of human relative performance across the value based reinforcement learning approaches suggests that there are broad sweeping advancements across the techniques, across the suite of games. Comparison of the various reinforcement learning techniques, relative to the previous state of art shows that in the vast majority of cases, significant gains were localised to a few select instances or games. Comparing all methodologies presented in this review, relative to the scores presented in the original DQN research, as shown in Figure 9 highlights this concept. What can be observed here is the clustering of results around each game, where most algorithms perform similarly well against any given game, of course with some limited number of exceptions such as Frostbite or Seaquest, where one particular algorithm performed exceedingly well in comparison to other approaches, but where this was not reflected across all games.

Figure 7 shows the mean score across the Atari 2600 games, normalised by the difference between the human score and the random agent, and displayed as a value relative to the original DQN results [6], ordered in time. What this shows, is that over many years since the groundbreaking research of the DQN agent, and the resultant insatiable interest in reinforcement learning and artificial intelligence in general from both academia, and the media, the level of overall performance has not had the dramatic improvements which are observed in other areas of artificial intelligence, image classification for example, or commensurate with the hyped claims of the field within mainstream media. That is of course, until the recent inclusion of the R2D2 algorithm, which as discussed previously, is considered to be one of the few major step changes in value based reinforcement learning, with the inclusion of higher level representation of memory. That is not to say the other incremental improvements aren’t important, just that they focused on improving the algorithm through higher efficiency in sampling, stability of value function representation, or depth of sampling space. To further emphasise this point, consider these relative scores separated out into the individual games, firstly ignoring the R2D2 algorithm in Figure 9, and secondly including it as in Figure 10. Here the axes limits have been clipped at 100 times the performance of DQN; above that, exactly how far it exceeds the performance is somewhat irrelevant for this discussion. The results shown in Figure 9 show that minimal improvement (<10%) has been made across the techniques, with the glaring exception of a handful of specific games, Asterix and Private Eye most notably. Compare that to the results of R2D2, as shown in Figure 10, where there are a number of clear examples of performance well above that of the original DQN, if not the majority. What is not observed, is a general improvement across all games, due in part to some games being already solved by the original DQN algorithm. To make this issue clear, consider Figure 11, which shows the performance improvement of all algorithms relative to the DQN algorithm, normalised by human level performance. This shows which games were not easily solved through reinforcement learning originally, and also human players. Figure 11 shows a measure by which an algorithm is outperforming both the DQN algorithm, and the human level performance. This highlights the issue where the reinforcement learning agent is good at outperforming humans by incredible amounts, when the task is simple, and hence a model of required behaviour is shallow and easily constructed. The majority of improvements are incremental in nature, with some small improvement over the base algorithm (in overall performance not considering learning efficiency), with a handful of examples of considerable improvement in one or more particular game constituting the majority of the mean improvement observed across time, as shown in Figure 7. Also interesting to note is that the games showing the most improvement in R2D2 compared to clusters of previous algorithms are predominantly the games which failed to exceed human performance initially, and failed to gain notable improvement across subsequent iterations of the algorithm up until that point.

VII. CONCLUSION

A subtle observation made when analysing the performance of value based deep reinforcement learning approaches, since the renaissance in this niche area of artificial intelligence, is the small pool of researchers contributing to the field. This is not true in terms of broad application of the concept, in which there is an impressive amount of effort across a broad and diverse number of researchers and research establishments. However, in terms of fundamental improvements to the techniques, which are benchmarked sufficiently against a common standard, and show some notable performance gain, there are relatively few researchers, whom are then common across multiple examples of the fundamental research. This is exacerbated when considering the affiliations of these core researchers, where there is a strong centre of gravity in a particular centre of excellence.

Of particular interest to note within this review of the field of value based reinforcement learning was that the improvements observed across the majority of value based reinforcement learning techniques were quite modest in relative terms when compared to the baseline DQN algorithm. This is considered a function of the relatively minor changes applied, where the most significant change is considered to
be the representation of the value as a distribution instead of a single value. This of course only holds true until considering the addition of an advanced memory representation into the R2D2 algorithm, which is considered here to be the major paradigm shift applied across deep value based Reinforcement Learning since its first and groundbreaking implementation. What this suggests is that the long held assumption that the Markov model being sufficient to contain a full representation of historical states, should be questioned. It is further asserted, and a key recommendation from this review, that the addition of other high level representation to other, more complex tasks as well as to improve on these benchmark levels of the Atari 2600 games.

REFERENCES

[1] S. Marsland, Machine Learning: An Algorithmic Perspective. London, U.K.: Chapman & Hall/CRC, 2014.
[2] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, MA: MIT Press, 1998.
[3] A. Irpan. (2018). Deep Reinforcement Learning Doesn’t Work Yet. [Online]. Available: https://alexirpan.com/2018/02/14/rl-hard.html
[4] A. Kurenkov. (2018). Reinforcement Learning’s Foundational Flaw. [Online]. Available: https://thegradient.pub/why-rl-is-flawed/
[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” 2013, arXiv:1312.5002.
[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529–533, 2015.
[7] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, “The arcade learning environment: An evaluation platform for general agents,” J. Artif. Intell. Res., vol. 47, pp. 253–279, Jun. 2015.
[8] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, “Mastering the game of go with deep neural networks and tree search,” Nature, vol. 529, no. 7587, p. 484, Jan. 2016.
[9] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, “Mastering the game of go without human knowledge,” Nature, vol. 550, no. 7676, pp. 354–359, 2017.
[10] O. Vinyals et al., “Grandmaster level in starcraft ii using multi-agent reinforcement learning,” Nature, vol. 575, no. 7782, pp. 350–354, 2019.
[11] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, “Assistant,” 2017, arXiv:1711.09874.
[12] O. Nachum, M. Norouzi, and D. Schuurmans. (2018). Learning Gaussian Policies From Smoothed Action Value Functions. [Online]. Available: https://openreview.net/forum
[13] R. Bellman, “The theory of dynamic programming,” Bull. Amer. Math. Soc., vol. 60, no. 6, pp. 503–515, 1954.
[14] R. Bellman, “A Markovian decision process,” Indiana Univ. Math. J., vol. 6, no. 4, pp. 679–684, 1957.
[15] A. Y. Ng, A. Coates, M. Diehl, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, “Autonomous inverted helicopter flight via reinforcement learning,” in Experimental Robotics IX, Berlin, Germany: Springer, 2006, pp. 363–372.
[16] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, “An application of reinforcement learning to aerobatic helicopter flight,” in Proc. Adv. Neural Inf. Process. Syst., 2007, pp. 1–8.
[17] D. Michie, On Machine Intelligence. Ultimo, NSW, Australia: Halsted Press, 1986.
[18] D. Michie, “Trial and error,” Sci. Surv., vol. 2, pp. 129–145, 1961.
[19] D. Michie and R. A. Chambers, “‘Boxes’ an experiment in adaptive control,” Mach. Inf., vol. 2, no. 2, pp. 137–152, 1968.
[20] A. L. Samuel, “Some studies in machine learning using the game of checkers,” IBM J. Res. Develop., vol. 3, no. 3, pp. 210–219, Jul. 1959.
[21] A. H. Klopf, Brain Function and Adaptive Systems: A Heterostatic Approach. U.K.: Chapman & Hall/CRC, 2014.
[22] A. Y. Ng, A. Coates, M. Diehl, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, “Autonomous inverted helicopter flight via reinforcement learning,” in Experimental Robotics IX, Berlin, Germany: Springer, 2006, pp. 363–372.
[23] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, “An application of reinforcement learning to aerobatic helicopter flight,” in Proc. Adv. Neural Inf. Process. Syst., 2007, pp. 1–8.
[24] D. Michie, “Trial and error,” Sci. Surv., vol. 2, pp. 129–145, 1961.
[25] D. Michie and R. A. Chambers, “‘Boxes’ an experiment in adaptive control,” Mach. Inf., vol. 2, no. 2, pp. 137–152, 1968.
[26] A. L. Samuel, “Some studies in machine learning using the game of checkers,” IBM J. Res. Develop., vol. 3, no. 3, pp. 210–219, Jul. 1959.
[27] A. H. Klopf, Brain Function and Adaptive Systems: A Heterostatic Approach. U.K.: Chapman & Hall/CRC, 2014.
[28] A. Y. Ng, A. Coates, M. Diehl, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, “Autonomous inverted helicopter flight via reinforcement learning,” in Experimental Robotics IX, Berlin, Germany: Springer, 2006, pp. 363–372.
[29] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, “An application of reinforcement learning to aerobatic helicopter flight,” in Proc. Adv. Neural Inf. Process. Syst., 2007, pp. 1–8.
[30] D. Michie, On Machine Intelligence. Ultimo, NSW, Australia: Halsted Press, 1986.
[31] D. Michie, “Trial and error,” Sci. Surv., vol. 2, pp. 129–145, 1961.
[32] D. Michie and R. A. Chambers, “‘Boxes’ an experiment in adaptive control,” Mach. Inf., vol. 2, no. 2, pp. 137–152, 1968.
[33] A. L. Samuel, “Some studies in machine learning using the game of checkers,” IBM J. Res. Develop., vol. 3, no. 3, pp. 210–219, Jul. 1959.
[34] A. H. Klopf, Brain Function and Adaptive Systems: A Heterostatic Approach. U.K.: Chapman & Hall/CRC, 2014.
[35] A. Y. Ng, A. Coates, M. Diehl, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, “Autonomous inverted helicopter flight via reinforcement learning,” in Experimental Robotics IX, Berlin, Germany: Springer, 2006, pp. 363–372.
[36] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, “An application of reinforcement learning to aerobatic helicopter flight,” in Proc. Adv. Neural Inf. Process. Syst., 2007, pp. 1–8.
B. Widrow and M. A. Lehr, “30 years of adaptive neural networks: Perceptron, madaline, and backpropagation,” Proc. IEEE, vol. 78, no. 9, pp. 1415–1442, Sep. 1990.

P. J. Werbos, “Applications of advances in nonlinear sensitivity analysis,” in System Modeling and Optimization. Berlin, Germany: Springer, 1982, pp. 762–770.

J. Hadamard, Mémoire Sur le Problème d’analyse Relatif à l’équilibre des Plaques élastiques Encastrées, vol. 33. Paris, France: Imprimerie Nationale, 1908.

S. Dreyfus, “The numerical solution of variational problems,” J. Math. Anal. Appl., vol. 5, no. 1, pp. 30–45, 1962.

J. Schmidhuber, “Deep learning in neural networks: An overview,” Neural Netw., vol. 61, pp. 85–117, Jan. 2014.

A. G. Ivakhnenko, “Polynomial theory of complex systems,” IEEE Trans. Syst., Man, Cybern., vol. SMC-1, no. 4, pp. 364–378, Oct. 1971.

K. Fukushima, “Neural network model for a mechanism of pattern recognition unaffected by shift in position-neocognitron,” IEICE Tech. Rep., A, vol. 62, no. 10, pp. 658–665, 1979.

K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” Biol. Cybern., vol. 36, no. 4, pp. 193–202, Apr. 1980.

G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” Science, vol. 313, no. 5786, pp. 504–507, 2006.

G. Hinton, S. Osindero, and Y. Teh, “A fast learning algorithm for deep belief nets,” Neural Comput., vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

P. Smolensky, J. McClelland, and D. Rumelhart, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, London, U.K.: MIT Press, 1986.

J. L. McClelland and D. E. Rumelhart, Parallel Distributed processing in Explorations in the Microstructure of Cognition: Psychological and Biological Models, vol. 2. Cambridge, MA, USA: MIT Press, 1987.

Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, p. 436, 2015.

Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

M. Riedmiller, “Neural fitted Q iteration—First experiences with a data driven methods for reinforcement learning with function approximation,” in Proc. Adv. Neural Inf. Process. Syst., 1996, pp. 1038–1044.

R. S. Sutton, “Generalization in reinforcement learning: Successful examples using sparse coarse coding,” in Proc. Adv. Neural Inf. Process. Syst., 1996, pp. 1038–1044.

N. Sprague, “Multiple-goal reinforcement learning with modular sarsa (0),” in Proc. 18th Int. Joint Conf. Artif. Intell., 2003, pp. 1445–1447.

D. Zhao, H. Wang, K. Shao, and Y. Zhu, “Deep reinforcement learning with experience replay based on SARSA,” in Proc. IEEE Symp. Ser. Comput. Intell. (SSCI), Dec. 2016, pp. 1–6.

H. Lima and Y. Kurucz, “Swarm reinforcement learning algorithms based on sarsa method,” in Proc. SICE Annu. Conf., Aug. 2008, pp. 2045–2049.

S. E. Dreyfus and R. Bellman, Applied Dynamic Programming, Princeton, NJ, USA: Princeton Univ. Press, 1962.

A. G. Barto and S. P. Singh, “On the computational economics of reinforcement learning,” in Connectionist Models. Amsterdam, The Netherlands: Elsevier, 1991, pp. 35–44.

R. S. Sutton, “Integrated architectures for learning, planning, and reacting based on approximating dynamic programming,” in Proc. 7th Int. Conf. Amsterdam, The Netherlands: Elsevier, 1990, pp. 216–224.

M. G. Bellemare, J. Veness, and M. Bowling, “Investigating contingency awareness using Atari 2600 games,” in Proc. 26th AAAI Conf. Artif. Intell., 2012, pp. 1–8.

M. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, “The arcade learning environment: An evaluation platform for general agents,” 2012, arXiv:1207.4708.

Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, “Dueling network architectures for deep reinforcement learning,” 2015, arXiv:1511.06581.

J. L. McClelland, B. L. McNaughton, and R. C. O’Reilly, “Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory,” Psychol. Rev., vol. 102, no. 3, p. 419, 1995.

J. O’Neill, B. Pleydell-Bouverie, D. Dupret, and J. Cisarci, “Play it again: Reactivation of waking experience and memory,” Trends Neurosciences, vol. 33, no. 5, pp. 220–229, May 2010.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” 2015, arXiv:1509.02971.

B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, “Building machines that learn and think like people,” Behav. Brain Sci., vol. 40, pp. 1–58, Nov. 2017.

Y. Zhu, R. Mottaghi, E. Kolve, J. L. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, “Target-driven visual navigation in indoor scenes using deep reinforcement learning,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2017, pp. 3357–3364.

M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. P. Abbeel, and W. Zaremba, “Hindsight experience replay,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5048–5058.

M. McKenzie, P. Loxley, W. Billingsley, and S. Wong, “Competitive reinforcement learning in Atari games,” in Proc. Australas. Joint Conf. Artif. Intell. Cham, Switzerland: Springer, 2017, pp. 14–26.

M. McKenzie and M. D. McDonnell, “Dedegradation of performance in reinforcement learning with state measurement uncertainty,” in Proc. Mil. Commun. Conf. Syst. (MUCS), Nov. 2019, pp. 1–5.

H. van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double Q-learning,” 2015, arXiv:1509.06461.

H. V. Hasselt, “Double Q-learning,” in Adv. Neural Inf. Process. Syst., 2010, pp. 2613–2621.

N. T. Tsidikis and B. Van Roy, “Analysis of temporal-difference learning with function approximation,” in Adv. Neural Inf. Process. Syst., 1997, pp. 1075–1081.

H. van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double Q-learning,” in Proc. 13th AAAI Conf. Artif. Intell., 2016, pp. 1–7.
J. Foerster, N. Nardelli, G. Farquhar, T. Afouras, P. H. Torr, P. Kohli, P. Nguyen, T. Akiyama, and H. Ohashi, “Experience filtering for robot navigation using deep reinforcement learning,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 4026–4034.

A. Yu, R. Palefsky-Smith, and R. Bedi, “Deep reinforcement learning for simulated autonomous vehicle control,” Course Project Rep., Winter, pp. 1–7, Tech. Rep., 2016.

E. Van der Pol and F. A. Oliehoek, “Coordinated deep reinforcement learners for traffic light control,” in Proc. Workshop Learn., Inference Control Multi-Agent Syst., 2016, pp. 1–9.

Y. Zhang, P. Sun, Y. Yin, L. Lin, and X. Wang, “Human-like autonomous vehicle speed control by deep reinforcement learning with double Q-learning,” in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2018, pp. 1251–1256.

O. Ansche1, N. Baram, and N. Shimkin, “Averaged-DQN: Variance reduction and stabilization for deep reinforcement learning,” in Proc. 34th Int. Conf. Mach. Learning, vol. 70, 2017, pp. 176–185.

N. Levine, T. Zahavy, D. J. Mankowitz, A. Tamar, and S. Mannor, “Shallow updates for deep reinforcement learning,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 3135–3145.

T. Schaul, J. Quan, I. Antonoglou, and D. Silver, “Prioritized experience replay,” 2015, arXiv:1511.05952.

L.-J. Lin, “Self-improving reactive agents based on reinforcement learning, planning and teaching,” Mach. Learn., vol. 8, nos. 3–4, pp. 293–321, May 1992.

H. Van Seijen and R. S. Sutton, “Efficient planning in MADPs by small backups,” in Proc. 30th Int. Conf. Mach. Learn., vol. 28, 2013, pp. 1–9.

X. Han, H. Liu, F. Sun, and D. Yang, “Active object detection using dueling-DQN and prioritized experience replay,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2018, pp. 1–7.

P. Nguyen, T. Akiyama, and H. Ohashi, “Experience filtering for robot navigation using deep reinforcement learning,” in Proc. 10th Int. Conf. Agents Artif. Intell., 2018, pp. 243–249.

J. Foerster, N. Nardelli, G. Farquhar, T. Afouras, P. H. Torr, P. Kohli, and S. Whiteson, “Stabilising experience replay for deep multi-agent reinforcement learning,” in Proc. 34th Int. Conf. Mach. Learn., vol. 70, 2017, pp. 1146–1155.

S.-Y. Chen, Y. Yu, Q. Da, J. Tan, H.-K. Huang, and H.-H. Tang, “Stabilizing reinforcement learning in dynamic environment with application to online recommendation,” in Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, Jul. 2018, pp. 1–7.

Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, “Dueling network architectures for deep reinforcement learning,” in Proc. 33rd Int. Conf. Mach. Learning, New York, NY, USA, ACM, 2016, pp. 1995–2003.

L. C Baird II, “Advantage updating,” Wright Lab., Wright-Patterson Air Force Base, Columbus, OH, USA, Tech. Rep., 1993.

S. Fang, F. Chen, and H. Liu, “Dueling double deep Q-network for adaptive traffic signal control with low exhaust emissions in a single intersection,” IOP Conf. Mater. Sci. Eng., vol. 612, no. 5, 2019, Art. no. 052039.

Y. Liu and C. Zhang, “Application of dueling DQN and DECGA for parameter estimation in variogram models,” IEEE Access, vol. 8, pp. 38112–38122, 2020.

W. Liu, P. Si, E. Sun, M. Li, C. Fang, and Y. Zhang, “Green mobility management in UAV-assisted IoT based on dueling DQN,” in Proc. IEEE Int. Conf. Commun. (ICC), May 2019, pp. 1–6.

D. Lopez-Martinez, P. Eschenfeldt, S. Ostvar, M. Ingram, C. Hur, and P. Picard, “Deep reinforcement learning for optimal critical care pain management with morphine using dueling double-deep Q networks,” in Proc. 41st Annua. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC), Jul. 2019, pp. 3960–3963.

Y. Huang, G. Wei, and Y. Wang, “V-D D3QN: The variant of double deep Q-learning network with dueling architecture,” in Proc. 57th Chin. Control Conf. (CCC), Jul. 2018, pp. 9130–9135.

V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 1928–1937.

M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, “Rainbow: Combining improvements in deep reinforcement learning,” in Proc. AAAI Conf. Artif. Intell., Apr. 2018, vol. 32, no. 1, pp. 1–8.

H. Mobahi, “Training recurrent neural networks by diffusion,” 2016, arXiv:1601.04114.

E. Hazan, K. Y. Levy, and S. Shalev-Shwartz, “On graduated optimization for stochastic non-convex problems,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 1833–1841.

I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, “Deep exploration via bootstrapped DQN,” in Proc. Adv. Neural Inf. Proc. Process., 2016, pp. 4026–4034.

A. Yu, R. Palefsky-Smith, and R. Bedi, “Deep reinforcement learning for simulated autonomous vehicle control,” Course Project Rep., Winter, pp. 1–7, Tech. Rep., 2016.
L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, A. Miliouddin, R. Sepassi, G. Tucker, and H. Michalewski, “Model-based reinforcement learning for atari,” 2019, arXiv:1903.00374.

S. Kapturowski, G. Ostrovski, J. Quan, R. Munos, and W. Dabney, “Recurrent experience replay in distributed reinforcement learning,” in Proc. Int. Conf. Learn. Represent., 2018, pp. 1–19.

S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

M. Hausknecht and P. Stone, “Deep recurrent Q-learning for partially observable MDPs,” 2015, arXiv:1507.06527.

A. Gruslys, W. Dabney, M. Ghseshlaghi Azar, B. Piot, M. Bellemare, and R. Munos, “The reactor: A fast and sample-efficient actor-critic agent for reinforcement learning,” 2017, arXiv:1704.04651.

P. J. Werbos, “Backpropagation through time: What it does and how to do it,” Proc. IEEE, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

J. Schaeffer, J. Culberson, N. Treloar, B. Knight, P. Lu, and D. Szafron, “A world championship caliber checkers program,” Artif. Intell., vol. 53, nos. 2–3, pp. 273–289, Feb. 1992.

J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and D. Silver, “Mastering atari, go, chess and shogi by planning with a learned model,” 2019, arXiv:1911.08265.

J. Oh, S. Singh, and H. Lee, “Value prediction network,” in Adv. Neural Inf. Process. Syst., 2017, pp. 6118–6128.

A. Tamar, Y. Wu, G. Thomas, S. Levine, and P. Abbeel, “Value iteration networks,” in Proc. 26th Int. Joint Conf. Artif. Intell., Aug. 2017, pp. 2154–2162.

D. Silver, H. van Hasselt, M. Hessel, T. Schaul, A. Guez, T. Harley, G. Dutilh-Arnold, D. Reichert, N. Rabinowitz, A. Barreto, and T.Degris, “The prediction: End-to-end learning and planning,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 3191–3199.

MARK C. MCKENZIE (Member, IEEE) received the B.Sc. degree in electronic engineering, the B.Sc. degree (Hons.) in applied mathematics, and the Ph.D. degree in electronic engineering from The University of Adelaide, Australia, in 1998, 2001, and 2006, respectively. In 2007, he joined the University of South Australia, where by 2021, he became an Associate Professor and the Principal Investigator of the Computational Learning Systems Laboratory. Currently, he is a Machine Learning Specialist and an AI Capability Lead at Consilium Technology and an Adjunct Associate Professor at The University of Adelaide. He has authored over 100 papers, including several review articles and a book Stochastic Resonance (Cambridge University Press). He has served as a Guest Editor of Proceedings of the IEEE and Frontiers in Computational Neuroscience. His research interests include the intersection between machine learning and neurobiological learning, with a specific focus on the influence of random noise on learning. His contributions to this area have been recognized by the Award of an Australian Research Fellowship from the Australian Research Council, in 2010, and a South Australian Tall Poppy Award for Science, in 2008. He has served as the Vice President and the Secretary of the IEEE South Australia Section, Joint Communications and Signal Processing Chapter.