A Distributional Perspective on Reinforcement Learning

Marc G. Bellemare [*1] **Will Dabney** [*1] **Rémi Munos** [1]

Abstract

In this paper we argue for the fundamental importance of the *value distribution*: the distribution of the random return received by a reinforcement learning agent. This is in contrast to the common approach to reinforcement learning which models the expectation of this return, or *value*. Although there is an established body of literature studying the value distribution, thus far it has always been used for a specific purpose such as implementing risk-aware behaviour. We begin with theoretical results in both the policy evaluation and control settings, exposing a significant distributional instability in the latter. We then use the distributional perspective to design a new algorithm which applies Bellman's equation to the learning of approximate value distributions. We evaluate our algorithm using the suite of games from the Arcade Learning Environment. We obtain both state-of-the-art results and anecdotal evidence demonstrating the importance of the value distribution in approximate reinforcement learning. Finally, we combine theoretical and empirical evidence to highlight the ways in which the value distribution impacts learning in the approximate setting.

1. Introduction

One of the major tenets of reinforcement learning states that, when not otherwise constrained in its behaviour, an agent should aim to maximize its expected utility Q, or *value* (Sutton & Barto, 1998). Bellman's equation succintly describes this value in terms of the expected reward and expected outcome of the random transition $(x, a) \to (X', A')$:

$$Q(x, a) = \mathbb{E}\, R(x, a) + \gamma\, \mathbb{E}\, Q(X', A').$$

In this paper, we aim to go beyond the notion of value and argue in favour of a distributional perspective on reinforcement learning. Specifically, the main object of our study is the random return Z whose expectation is the value Q. This random return is also described by a recursive equation, but one of a distributional nature:

$$Z(x, a) \stackrel{D}{=} R(x, a) + \gamma Z(X', A').$$

The *distributional Bellman equation* states that the distribution of Z is characterized by the interaction of three random variables: the reward R, the next state-action (X', A'), and its random return $Z(X', A')$. By analogy with the well-known case, we call this quantity the *value distribution*.

Although the distributional perspective is almost as old as Bellman's equation itself (Jaquette, 1973; Sobel, 1982; White, 1988), in reinforcement learning it has thus far been subordinated to specific purposes: to model parametric uncertainty (Dearden et al., 1998), to design risk-sensitive algorithms (Morimura et al., 2010b;a), or for theoretical analysis (Azar et al., 2012; Lattimore & Hutter, 2012). By contrast, we believe the value distribution has a central role to play in reinforcement learning.

Contraction of the policy evaluation Bellman operator. Basing ourselves on results by Rösler (1992) we show that, for a fixed policy, the Bellman operator over value distributions is a contraction in a maximal form of the Wasserstein (also called Kantorovich or Mallows) metric. Our particular choice of metric matters: the same operator is not a contraction in total variation, Kullback-Leibler divergence, or Kolmogorov distance.

Instability in the control setting. We will demonstrate an instability in the distributional version of Bellman's optimality equation, in contrast to the policy evaluation case. Specifically, although the optimality operator is a contraction in expected value (matching the usual optimality result), it is not a contraction in any metric over distributions. These results provide evidence in favour of learning algorithms that model the effects of nonstationary policies.

Better approximations. From an algorithmic standpoint, there are many benefits to learning an approximate distribution rather than its approximate expectation. The distributional Bellman operator preserves multimodality in value distributions, which we believe leads to more stable learning. Approximating the full distribution also mitigates the effects of learning from a nonstationary policy. As a whole,

[*]Equal contribution [1]DeepMind, London, UK. Correspondence to: Marc G. Bellemare <bellemare@google.com>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

we argue that this approach makes approximate reinforcement learning significantly better behaved.

We will illustrate the practical benefits of the distributional perspective in the context of the Arcade Learning Environment (Bellemare et al., 2013). By modelling the value distribution within a DQN agent (Mnih et al., 2015), we obtain considerably increased performance across the gamut of benchmark Atari 2600 games, and in fact achieve state-of-the-art performance on a number of games. Our results echo those of Veness et al. (2015), who obtained extremely fast learning by predicting Monte Carlo returns.

From a supervised learning perspective, learning the full value distribution might seem obvious: why restrict ourselves to the mean? The main distinction, of course, is that in our setting there are no given targets. Instead, we use Bellman's equation to make the learning process tractable; we must, as Sutton & Barto (1998) put it, "learn a guess from a guess". It is our belief that this guesswork ultimately carries more benefits than costs.

2. Setting

We consider an agent interacting with an environment in the standard fashion: at each step, the agent selects an action based on its current state, to which the environment responds with a reward and the next state. We model this interaction as a time-homogeneous Markov Decision Process $(\mathcal{X}, \mathcal{A}, R, P, \gamma)$. As usual, $\mathcal{X}$ and $\mathcal{A}$ are respectively the state and action spaces, P is the transition kernel $P(\cdot \,|\, x, a)$, $\gamma \in [0, 1]$ is the discount factor, and R is the reward function, which in this work we explicitly treat as a random variable. A stationary policy π maps each state $x \in \mathcal{X}$ to a probability distribution over the action space $\mathcal{A}$.

2.1. Bellman's Equations

The *return* Z^π is the sum of discounted rewards along the agent's trajectory of interactions with the environment. The value function Q^π of a policy π describes the expected return from taking action $a \in \mathcal{A}$ from state $x \in \mathcal{X}$, then acting according to π:

$$Q^\pi(x, a) := \mathbb{E}\, Z^\pi(x, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(x_t, a_t)\right], \qquad (1)$$
$$x_t \sim P(\cdot \,|\, x_{t-1}, a_{t-1}), a_t \sim \pi(\cdot \,|\, x_t), x_0 = x, a_0 = a.$$

Fundamental to reinforcement learning is the use of Bellman's equation (Bellman, 1957) to describe the value function:

$$Q^\pi(x, a) = \mathbb{E}\, R(x, a) + \gamma \mathop{\mathbb{E}}_{P,\pi} Q^\pi(x', a').$$

In reinforcement learning we are typically interested in acting so as to maximize the return. The most common approach for doing so involves the optimality equation

$$Q^*(x, a) = \mathbb{E}\, R(x, a) + \gamma\, \mathbb{E}_P \max_{a' \in \mathcal{A}} Q^*(x', a').$$

This equation has a unique fixed point Q^*, the optimal value function, corresponding to the set of optimal policies Π^* (π^* is optimal if $\mathbb{E}_{a \sim \pi^*} Q^*(x, a) = \max_a Q^*(x, a)$).

We view value functions as vectors in $\mathbb{R}^{\mathcal{X} \times \mathcal{A}}$, and the expected reward function as one such vector. In this context, the *Bellman operator* $\mathcal{T}^\pi$ and *optimality operator* $\mathcal{T}$ are

$$\mathcal{T}^\pi Q(x, a) := \mathbb{E}\, R(x, a) + \gamma \mathop{\mathbb{E}}_{P,\pi} Q(x', a') \qquad (2)$$

$$\mathcal{T} Q(x, a) := \mathbb{E}\, R(x, a) + \gamma\, \mathbb{E}_P \max_{a' \in \mathcal{A}} Q(x', a'). \qquad (3)$$

These operators are useful as they describe the expected behaviour of popular learning algorithms such as SARSA and Q-Learning. In particular they are both contraction mappings, and their repeated application to some initial Q_0 converges exponentially to Q^π or Q^*, respectively (Bertsekas & Tsitsiklis, 1996).

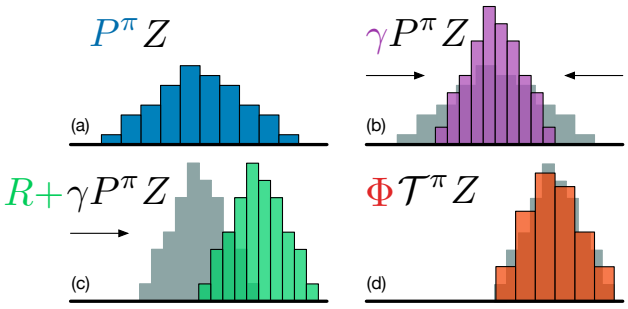


Figure 1. A distributional Bellman operator with a deterministic reward function: (a) Next state distribution under policy π, (b) Discounting shrinks the distribution towards 0, (c) The reward shifts it, and (d) Projection step (Section 4).

3. The Distributional Bellman Operators

In this paper we take away the expectations inside Bellman's equations and consider instead the full distribution of the random variable Z^π. From here on, we will view Z^π as a mapping from state-action pairs to distributions over returns, and call it the *value distribution*.

Our first aim is to gain an understanding of the theoretical behaviour of the distributional analogues of the Bellman operators, in particular in the less well-understood control setting. The reader strictly interested in the algorithmic contribution may choose to skip this section.

3.1. Distributional Equations

It will sometimes be convenient to make use of the probability space $(\Omega, \mathcal{F}, \Pr)$. The reader unfamiliar with mea-

sure theory may think of Ω as the space of all possible outcomes of an experiment (Billingsley, 1995). We will write $\|\mathbf{u}\|_p$ to denote the L_p norm of a vector $\mathbf{u} \in \mathbb{R}^{\mathcal{X}}$ for $1 \leq p \leq \infty$; the same applies to vectors in $\mathbb{R}^{\mathcal{X}\times\mathcal{A}}$. The L_p norm of a random vector $U : \Omega \to \mathbb{R}^{\mathcal{X}}$ (or $\mathbb{R}^{\mathcal{X}\times\mathcal{A}}$) is then $\|U\|_p := \left[\mathbb{E}\left[\|U(\omega)\|_p^p\right]\right]^{1/p}$, and for $p = \infty$ we have $\|U\|_\infty = \sup \|U(\omega)\|_\infty$ (we will omit the dependency on $\omega \in \Omega$ whenever unambiguous). We will denote the c.d.f. of a random variable U by $F_U(y) := \Pr\{U \leq y\}$, and its inverse c.d.f. by $F_U^{-1}(q) := \inf\{y : F_U(y) \geq q\}$.

A distributional equation $U \stackrel{D}{:=} V$ indicates that the random variable U is distributed according to the same law as V. Without loss of generality, the reader can understand the two sides of a distributional equation as relating the distributions of two independent random variables. Distributional equations have been used in reinforcement learning by Engel et al. (2005); Morimura et al. (2010a) among others, and in operations research by White (1988).

3.2. The Wasserstein Metric

The main tool for our analysis is the Wasserstein metric d_p between cumulative distribution functions (see e.g. Bickel & Freedman, 1981, where it is called the Mallows metric). For F, G two c.d.fs over the reals, it is defined as

$$d_p(F, G) := \inf_{U,V} \|U - V\|_p,$$

where the infimum is taken over all pairs of random variables (U, V) with respective cumulative distributions F and G. The infimum is attained by the inverse c.d.f. transform of a random variable $\mathcal{U}$ uniformly distributed on $[0, 1]$:

$$d_p(F, G) = \|F^{-1}(\mathcal{U}) - G^{-1}(\mathcal{U})\|_p.$$

For $p < \infty$ this is more explicitly written as

$$d_p(F, G) = \left(\int_0^1 \left|F^{-1}(u) - G^{-1}(u)\right|^p du\right)^{1/p}.$$

Given two random variables U, V with c.d.fs F_U, F_V, we will write $d_p(U, V) := d_p(F_U, F_V)$. We will find it convenient to conflate the random variables under consideration with their versions under the $\inf$, writing

$$d_p(U, V) = \inf_{U,V} \|U - V\|_p.$$

whenever unambiguous; we believe the greater legibility justifies the technical inaccuracy. Finally, we extend this metric to vectors of random variables, such as value distributions, using the corresponding L_p norm.

Consider a scalar a and a random variable A independent of U, V. The metric d_p has the following properties:

$$d_p(aU, aV) \leq |a| d_p(U, V) \tag{P1}$$
$$d_p(A + U, A + V) \leq d_p(U, V) \tag{P2}$$
$$d_p(AU, AV) \leq \|A\|_p d_p(U, V). \tag{P3}$$

We will need the following additional property, which makes no independence assumptions on its variables. Its proof, and that of later results, is given in the appendix.

Lemma 1 (Partition lemma)**.** *Let $A_1, A_2, \ldots$ be a set of random variables describing a partition of Ω, i.e. $A_i(\omega) \in \{0, 1\}$ and for any ω there is exactly one A_i with $A_i(\omega) = 1$. Let U, V be two random variables. Then*

$$d_p\big(U, V\big) \leq \sum\nolimits_i d_p(A_i U, A_i V).$$

Let $\mathcal{Z}$ denote the space of value distributions with bounded moments. For two value distributions $Z_1, Z_2 \in \mathcal{Z}$ we will make use of a maximal form of the Wasserstein metric:

$$\bar{d}_p(Z_1, Z_2) := \sup_{x,a} d_p(Z_1(x, a), Z_2(x, a)).$$

We will use $\bar{d}_p$ to establish the convergence of the distributional Bellman operators.

Lemma 2. *$\bar{d}_p$ is a metric over value distributions.*

3.3. Policy Evaluation

In the *policy evaluation* setting (Sutton & Barto, 1998) we are interested in the value function V^π associated with a given policy π. The analogue here is the value distribution Z^π. In this section we characterize Z^π and study the behaviour of the policy evaluation operator $\mathcal{T}^\pi$. We emphasize that Z^π describes the intrinsic randomness of the agent's interactions with its environment, rather than some measure of uncertainty about the environment itself.

We view the reward function as a random vector $R \in \mathcal{Z}$, and define the transition operator $P^\pi : \mathcal{Z} \to \mathcal{Z}$

$$P^\pi Z(x, a) \stackrel{D}{:=} Z(X', A') \tag{4}$$
$$X' \sim P(\cdot \,|\, x, a),\ A' \sim \pi(\cdot \,|\, X'),$$

where we use capital letters to emphasize the random nature of the next state-action pair (X', A'). We define the distributional Bellman operator $\mathcal{T}^\pi : \mathcal{Z} \to \mathcal{Z}$ as

$$\mathcal{T}^\pi Z(x, a) \stackrel{D}{:=} R(x, a) + \gamma P^\pi Z(x, a). \tag{5}$$

While $\mathcal{T}^\pi$ bears a surface resemblance to the usual Bellman operator (2), it is fundamentally different. In particular, three sources of randomness define the compound distribution $\mathcal{T}^\pi Z$:

a) The randomness in the reward R,
b) The randomness in the transition P^π, and
c) The next-state value distribution $Z(X', A')$.

In particular, we make the usual assumption that these three quantities are independent. In this section we will show that (5) is a contraction mapping whose unique fixed point is the random return Z^π.

3.3.1. Contraction in $\bar{d}_p$

Consider the process $Z_{k+1} := \mathcal{T}^\pi Z_k$, starting with some $Z_0 \in \mathcal{Z}$. We may expect the limiting expectation of $\{Z_k\}$ to converge exponentially quickly, as usual, to Q^π. As we now show, the process converges in a stronger sense: $\mathcal{T}^\pi$ is a contraction in $\bar{d}_p$, which implies that all moments also converge exponentially quickly.

Lemma 3. *$\mathcal{T}^\pi : \mathcal{Z} \to \mathcal{Z}$ is a γ-contraction in $\bar{d}_p$.*

Using Lemma 3, we conclude using Banach's fixed point theorem that $\mathcal{T}^\pi$ has a unique fixed point. By inspection, this fixed point must be Z^π as defined in (1). As we assume all moments are bounded, this is sufficient to conclude that the sequence $\{Z_k\}$ converges to Z^π in $\bar{d}_p$ for $1 \leq p \leq \infty$.

To conclude, we remark that not all distributional metrics are equal; for example, Chung & Sobel (1987) have shown that $\mathcal{T}^\pi$ is not a contraction in total variation distance. Similar results can be derived for the Kullback-Leibler divergence and the Kolmogorov distance.

3.3.2. Contraction in Centered Moments

Observe that $d_2(U, V)$ (and more generally, d_p) relates to a coupling $C(\omega) := U(\omega) - V(\omega)$, in the sense that

$$d_2^2(U, V) \leq \mathbb{E}[(U - V)^2] = \mathbb{V}\big(C\big) + \big(\mathbb{E}\, C\big)^2.$$

As a result, we cannot directly use d_2 to bound the variance difference $|\mathbb{V}(\mathcal{T}^\pi Z(x, a)) - \mathbb{V}(Z^\pi(x, a))|$. However, $\mathcal{T}^\pi$ is in fact a contraction in variance (Sobel, 1982, see also appendix). In general, $\mathcal{T}^\pi$ is not a contraction in the p^{th} centered moment, $p > 2$, but the centered moments of the iterates $\{Z_k\}$ still converge exponentially quickly to those of Z^π; the proof extends the result of Rösler (1992).

3.4. Control

Thus far we have considered a fixed policy π, and studied the behaviour of its associated operator $\mathcal{T}^\pi$. We now set out to understand the distributional operators of the *control* setting – where we seek a policy π that maximizes value – and the corresponding notion of an optimal value distribution. As with the optimal value function, this notion is intimately tied to that of an optimal policy. However, while all optimal policies attain the same value Q^*, in our case a difficulty arises: in general there are many optimal value distributions.

In this section we show that the distributional analogue of the Bellman optimality operator converges, in a weak sense, to the set of optimal value distributions. However, this operator is not a contraction in any metric between distributions, and is in general much more temperamental than the policy evaluation operators. We believe the convergence issues we outline here are a symptom of the inherent instability of greedy updates, as highlighted by e.g. Tsitsiklis (2002) and most recently Harutyunyan et al. (2016).

Let Π^* be the set of optimal policies. We begin by characterizing what we mean by an *optimal value distribution*.

Definition 1 (Optimal value distribution)**.** *An optimal value distribution is the v.d. of an optimal policy. The set of optimal value distributions is $\mathcal{Z}^* := \{Z^{\pi^*} : \pi^* \in \Pi^*\}$.*

The mapping from policies to value distributions $\pi \mapsto Z^\pi$ is continuous. As a result, $\mathcal{Z}^*$ inherits many of the properties of Π^*: it is convex and, in finite state-action spaces, compact. We emphasize that not all value distributions with expectation Q^* are optimal: they must match the full distribution of the return under some optimal policy.

Definition 2. *A greedy policy π for $Z \in \mathcal{Z}$ maximizes the expectation of Z. The set of greedy policies for Z is*

$$\mathcal{G}_Z := \{\pi : \sum\nolimits_a \pi(a \,|\, x)\, \mathbb{E}\, Z(x, a) = \max_{a' \in \mathcal{A}} \mathbb{E}\, Z(x, a')\}.$$

Recall that the expected Bellman optimality operator $\mathcal{T}$ is

$$\mathcal{T}Q(x, a) = \mathbb{E}\, R(x, a) + \gamma\, \mathbb{E}_P \max_{a' \in \mathcal{A}} Q(x', a'). \quad (6)$$

The maximization at x' corresponds to some greedy policy. Although this policy is implicit in (6), we cannot ignore it in the distributional setting. We will call a *distributional Bellman optimality operator* any operator $\mathcal{T}$ which implements a greedy selection rule, i.e.

$$\mathcal{T}Z = \mathcal{T}^\pi Z \text{ for some } \pi \in \mathcal{G}_Z.$$

As in the policy evaluation setting, we are interested in the behaviour of the iterates $Z_{k+1} := \mathcal{T}Z_k$, $Z_0 \in \mathcal{Z}$. Our first result is to assert that $\mathbb{E}\, Z_k$ behaves as expected.

Lemma 4. *Let $Z_1, Z_2 \in \mathcal{Z}$. Then*

$$\left\|\mathbb{E}\,\mathcal{T}Z_1 - \mathbb{E}\,\mathcal{T}Z_2\right\|_\infty \leq \gamma \left\|\mathbb{E}\, Z_1 - \mathbb{E}\, Z_2\right\|_\infty,$$

and in particular $\mathbb{E}\, Z_k \to Q^*$ exponentially quickly.

By inspecting Lemma 4, we might expect that Z_k converges quickly in $\bar{d}_p$ to some fixed point in $\mathcal{Z}^*$. Unfortunately, convergence is neither quick nor assured to reach a fixed point. In fact, the best we can hope for is pointwise convergence, not even to the set $\mathcal{Z}^*$ but to the larger set of *nonstationary optimal value distributions*.

Definition 3. *A nonstationary optimal value distribution Z^{**} is the value distribution corresponding to a sequence of optimal policies. The set of n.o.v.d. is $\mathcal{Z}^{**}$.*

Theorem 1 (Convergence in the control setting)**.** *Let $\mathcal{X}$ be measurable and suppose that $\mathcal{A}$ is finite. Then*

$$\lim_{k\to\infty} \inf_{Z^{**}\in\mathcal{Z}^{**}} d_p(Z_k(x,a), Z^{**}(x,a)) = 0 \quad \forall x, a.$$

If $\mathcal{X}$ is finite, then Z_k converges to $\mathcal{Z}^{**}$ uniformly. Furthermore, if there is a total ordering $\prec$ on Π^*, such that for any $Z^* \in \mathcal{Z}^*$,

$$\mathcal{T}Z^* = \mathcal{T}^\pi Z^* \text{ with } \pi \in \mathcal{G}_{Z^*},\ \pi \prec \pi' \ \ \forall \pi' \in \mathcal{G}_{Z^*} \setminus \{\pi\}.$$

Then $\mathcal{T}$ has a unique fixed point $Z^* \in \mathcal{Z}^*$.

Comparing Theorem 1 to Lemma 4 reveals a significant difference between the distributional framework and the usual setting of expected return. While the mean of Z_k converges exponentially quickly to Q^*, its distribution need not be as well-behaved! To emphasize this difference, we now provide a number of negative results concerning $\mathcal{T}$.

Proposition 1. *The operator $\mathcal{T}$ is not a contraction.*

Consider the following example (Figure 2, left). There are two states, x_1 and x_2; a unique transition from x_1 to x_2; from x_2, action a_1 yields no reward, while the optimal action a_2 yields $1+\epsilon$ or $-1+\epsilon$ with equal probability. Both actions are terminal. There is a unique optimal policy and therefore a unique fixed point Z^*. Now consider Z as given in Figure 2 (right), and its distance to Z^*:

$$\bar{d}_1(Z, Z^*) = d_1(Z(x_2,a_2), Z^*(x_2,a_2)) = 2\epsilon,$$

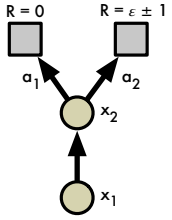
where we made use of the fact that $Z = Z^*$ everywhere except at (x_2, a_2). When we apply $\mathcal{T}$ to Z, however, the greedy action a_1 is selected and $\mathcal{T}Z(x_1) = Z(x_2, a_1)$. But

$$\begin{aligned} d_1(\mathcal{T}Z, \mathcal{T}Z^*) &= d_1(\mathcal{T}Z(x_1), Z^*(x_1)) \\ &= \tfrac{1}{2}|1-\epsilon| + \tfrac{1}{2}|1+\epsilon| > 2\epsilon \end{aligned}$$

for a sufficiently small ϵ. This shows that the undiscounted update is not a nonexpansion: $\bar{d}_1(\mathcal{T}Z, \mathcal{T}Z^*) > \bar{d}_1(Z, Z^*)$. With $\gamma < 1$, the same proof shows it is not a contraction. Using a more technically involved argument, we can extend this result to any metric which separates Z and $\mathcal{T}Z$.

Proposition 2. *Not all optimality operators have a fixed point $Z^* = \mathcal{T}Z^*$.*

To see this, consider the same example, now with $\epsilon = 0$, and a greedy operator $\mathcal{T}$ which breaks ties by picking a_2 if $Z(x_1) = 0$, and a_1 otherwise. Then the sequence $\mathcal{T}Z^*(x_1), (\mathcal{T})^2 Z^*(x_1), \ldots$ alternates between $Z^*(x_2, a_1)$ and $Z^*(x_2, a_2)$.



	x_1	x_2, a_1	x_2, a_2
Z^*	$\epsilon \pm 1$	0	$\epsilon \pm 1$
Z	$\epsilon \pm 1$	0	$-\epsilon \pm 1$
$\mathcal{T}Z$	0	0	$\epsilon \pm 1$

Figure 2. Undiscounted two-state MDP for which the optimality operator $\mathcal{T}$ is not a contraction, with example. The entries that contribute to $\bar{d}_1(Z, Z^*)$ and $\bar{d}_1(\mathcal{T}Z, Z^*)$ are highlighted.

Proposition 3. *That $\mathcal{T}$ has a fixed point $Z^* = \mathcal{T}Z^*$ is insufficient to guarantee the convergence of $\{Z_k\}$ to $\mathcal{Z}^*$.*

Theorem 1 paints a rather bleak picture of the control setting. It remains to be seen whether the dynamical eccentricities highlighted here actually arise in practice. One open question is whether theoretically more stable behaviour can be derived using stochastic policies, for example from conservative policy iteration (Kakade & Langford, 2002).

4. Approximate Distributional Learning

In this section we propose an algorithm based on the distributional Bellman optimality operator. In particular, this will require choosing an approximating distribution. Although the Gaussian case has previously been considered (Morimura et al., 2010a; Tamar et al., 2016), to the best of our knowledge we are the first to use a rich class of parametric distributions.

4.1. Parametric Distribution

We will model the value distribution using a discrete distribution parametrized by $N \in \mathbb{N}$ and $V_{\text{MIN}}, V_{\text{MAX}} \in \mathbb{R}$, and whose support is the set of atoms $\{z_i = V_{\text{MIN}} + i\triangle z : 0 \le i < N\}$, $\triangle z := \frac{V_{\text{MAX}} - V_{\text{MIN}}}{N-1}$. In a sense, these atoms are the "canonical returns" of our distribution. The atom probabilities are given by a parametric model $\theta : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^N$

$$Z_\theta(x,a) = z_i \quad \text{w.p. } \ p_i(x,a) := \frac{e^{\theta_i(x,a)}}{\sum_j e^{\theta_j(x,a)}}.$$

The discrete distribution has the advantages of being highly expressive and computationally friendly (see e.g. Van den Oord et al., 2016).

4.2. Projected Bellman Update

Using a discrete distribution poses a problem: the Bellman update $\mathcal{T}Z_\theta$ and our parametrization Z_θ almost always have disjoint supports. From the analysis of Section 3 it would seem natural to minimize the Wasserstein metric (viewed as a loss) between $\mathcal{T}Z_\theta$ and Z_θ, which is also

conveniently robust to discrepancies in support. However, a second issue prevents this: in practice we are typically restricted to learning from sample transitions, which is not possible under the Wasserstein loss (see Prop. 5 and toy results in the appendix).

Instead, we project the sample Bellman update $\hat{\mathcal{T}}Z_\theta$ onto the support of Z_θ (Figure 1, Algorithm 1), effectively reducing the Bellman update to multiclass classification. Let π be the greedy policy w.r.t. $\mathbb{E}\,Z_\theta$. Given a sample transition (x, a, r, x'), we compute the Bellman update $\hat{\mathcal{T}}z_j := r + \gamma z_j$ for each atom z_j, then distribute its probability $p_j(x', \pi(x'))$ to the immediate neighbours of $\hat{\mathcal{T}}z_j$. The i^{th} component of the projected update $\Phi\hat{\mathcal{T}}Z_\theta(x, a)$ is

$$(\Phi\hat{\mathcal{T}}Z_\theta(x,a))_i = \sum_{j=0}^{N-1}\left[1 - \frac{|[\hat{\mathcal{T}}z_j]_{V_{\text{MIN}}}^{V_{\text{MAX}}} - z_i|}{\triangle z}\right]_0^1 p_j(x', \pi(x')), \tag{7}$$

where $[\cdot]_a^b$ bounds its argument in the range $[a, b]$.[1] As is usual, we view the next-state distribution as parametrized by a fixed parameter $\tilde{\theta}$. The sample loss $\mathcal{L}_{x,a}(\theta)$ is the cross-entropy term of the KL divergence

$$D_{\text{KL}}(\Phi\hat{\mathcal{T}}Z_{\tilde{\theta}}(x,a) \,\|\, Z_\theta(x,a)),$$

which is readily minimized e.g. using gradient descent. We call this choice of distribution and loss the *categorical algorithm*. When $N = 2$, a simple one-parameter alternative is $\Phi\hat{\mathcal{T}}Z_\theta(x,a) := [\mathbb{E}[\hat{\mathcal{T}}Z_\theta(x,a)] - V_{\text{MIN}})/\triangle z]_0^1$; we call this the *Bernoulli algorithm*. We note that, while these algorithms appear unrelated to the Wasserstein metric, recent work (Bellemare et al., 2017) hints at a deeper connection.

Algorithm 1 Categorical Algorithm

input A transition $x_t, a_t, r_t, x_{t+1}, \gamma_t \in [0, 1]$
  $Q(x_{t+1}, a) := \sum_i z_i p_i(x_{t+1}, a)$
  $a^* \leftarrow \arg\max_a Q(x_{t+1}, a)$
  $m_i = 0, \quad i \in 0, \ldots, N-1$
  for $j \in 0, \ldots, N-1$ **do**
    # Compute the projection of $\hat{\mathcal{T}}z_j$ onto the support $\{z_i\}$
    $\hat{\mathcal{T}}z_j \leftarrow [r_t + \gamma_t z_j]_{V_{\text{MIN}}}^{V_{\text{MAX}}}$
    $b_j \leftarrow (\hat{\mathcal{T}}z_j - V_{\text{MIN}})/\Delta z$ # $b_j \in [0, N-1]$
    $l \leftarrow \lfloor b_j \rfloor, u \leftarrow \lceil b_j \rceil$
    # Distribute probability of $\hat{\mathcal{T}}z_j$
    $m_l \leftarrow m_l + p_j(x_{t+1}, a^*)(u - b_j)$
    $m_u \leftarrow m_u + p_j(x_{t+1}, a^*)(b_j - l)$
  end for
output $-\sum_i m_i \log p_i(x_t, a_t)$ # Cross-entropy loss

5. Evaluation on Atari 2600 Games

To understand the approach in a complex setting, we applied the categorical algorithm to games from the Arcade Learning Environment (ALE; Bellemare et al., 2013). While the ALE is deterministic, stochasticity does occur in a number of guises: 1) from state aliasing, 2) learning from a nonstationary policy, and 3) from approximation errors. We used five training games (Fig 3) and 52 testing games.

For our study, we use the DQN architecture (Mnih et al., 2015), but output the atom probabilities $p_i(x, a)$ instead of action-values, and chose $V_{\text{MAX}} = -V_{\text{MIN}} = 10$ from preliminary experiments over the training games. We call the resulting architecture *Categorical DQN*. We replace the squared loss $(r + \gamma Q(x', \pi(x')) - Q(x, a))^2$ by $\mathcal{L}_{x,a}(\theta)$ and train the network to minimize this loss.[2] As in DQN, we use a simple ϵ-greedy policy over the expected action-values; we leave as future work the many ways in which an agent could select actions on the basis of the full distribution. The rest of our training regime matches Mnih et al.'s, including the use of a target network for $\tilde{\theta}$.

Figure 4 illustrates the typical value distributions we observed in our experiments. In this example, three actions (those including the button press) lead to the agent releasing its laser too early and eventually losing the game. The corresponding distributions reflect this: they assign a significant probability to 0 (the terminal value). The safe actions have similar distributions (LEFT, which tracks the invaders' movement, is slightly favoured). This example helps explain why our approach is so successful: the distributional update keeps separated the low-value, "losing" event from the high-value, "survival" event, rather than average them into one (unrealizable) expectation.[3]

One surprising fact is that the distributions are not concentrated on one or two values, in spite of the ALE's determinism, but are often close to Gaussians. We believe this is due to our discretizing the diffusion process induced by γ.

5.1. Varying the Number of Atoms

We began by studying our algorithm's performance on the training games in relation to the number of atoms (Figure 3). For this experiment, we set $\epsilon = 0.05$. From the data, it is clear that using too few atoms can lead to poor behaviour, and that more always increases performance; this is not immediately obvious as we may have expected to saturate the network capacity. The difference in performance between the 51-atom version and DQN is particularly striking: the latter is outperformed in all five games, and in SEAQUEST we attain state-of-the-art performance. As an additional point of the comparison, the single-parameter Bernoulli algorithm performs better than DQN in 3 games out of 5, and is most notably more robust in ASTERIX.

[1] Algorithm 1 computes this projection in time linear in N.

[2] For $N = 51$, our TensorFlow implementation trains at roughly 75% of DQN's speed.

[3] Video: http://youtu.be/yFBwyPuO2Vg.

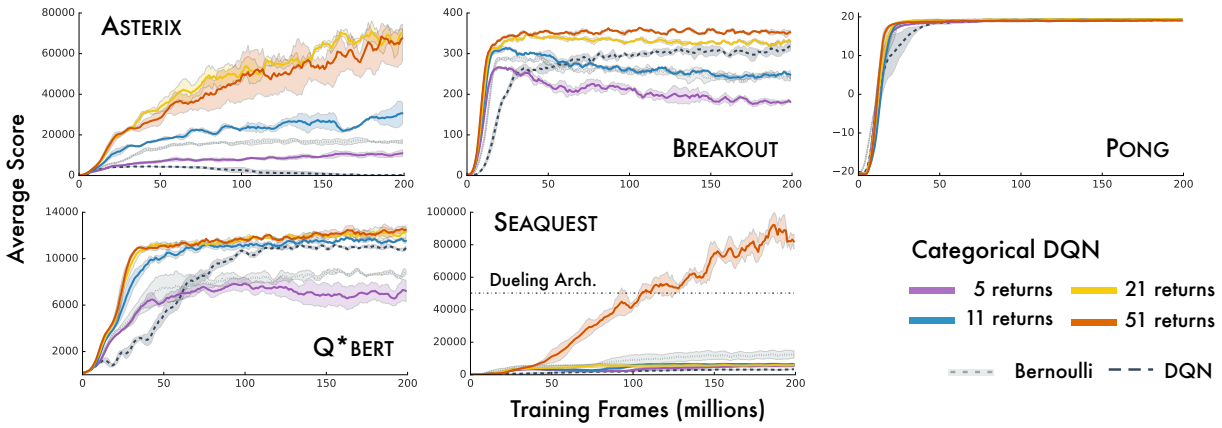


Figure 3. Categorical DQN: Varying number of atoms in the discrete distribution. Scores are moving averages over 5 million frames.

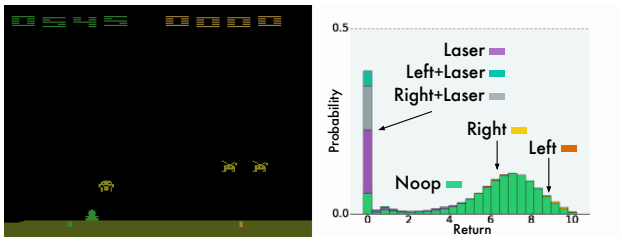


Figure 4. Learned value distribution during an episode of SPACE INVADERS. Different actions are shaded different colours. Returns below 0 (which do not occur in SPACE INVADERS) are not shown here as the agent assigns virtually no probability to them.

One interesting outcome of this experiment was to find out that our method does pick up on stochasticity. PONG exhibits intrinsic randomness: the exact timing of the reward depends on internal registers and is truly unobservable. We see this clearly reflected in the agent's prediction (Figure 5): over five consecutive frames, the value distribution shows two modes indicating the agent's belief that it has yet to receive a reward. Interestingly, since the agent's state does not include past rewards, it cannot even extinguish the prediction after receiving the reward, explaining the relative proportions of the modes.

5.2. State-of-the-Art Results

The performance of the 51-atom agent (from here onwards, C51) on the training games, presented in the last section, is particularly remarkable given that it involved none of the other algorithmic ideas present in state-of-the-art agents. We next asked whether incorporating the most common hyperparameter choice, namely a smaller training ϵ, could lead to even better results. Specifically, we set $\epsilon = 0.01$ (instead of 0.05); furthermore, every 1 million frames, we evaluate our agent's performance with $\epsilon = 0.001$.

We compare our algorithm to DQN ($\epsilon = 0.01$), Double DQN (van Hasselt et al., 2016), the Dueling architecture (Wang et al., 2016), and Prioritized Replay (Schaul et al., 2016), comparing the best evaluation score achieved during training. We see that C51 significantly outperforms these other algorithms (Figures 6 and 7). In fact, C51 surpasses the current state-of-the-art by a large margin in a number of games, most notably SEAQUEST. One particularly striking fact is the algorithm's good performance on sparse reward games, for example VENTURE and PRIVATE EYE. This suggests that value distributions are better able to propagate rarely occurring events. Full results are provided in the appendix.

We also include in the appendix (Figure 12) a comparison, averaged over 3 seeds, showing the number of games in which C51's training performance outperforms fully-trained DQN and human players. These results continue to show dramatic improvements, and are more representative of an agent's average performance. Within 50 million frames, C51 has outperformed a fully trained DQN agent on 45 out of 57 games. This suggests that the full 200 million training frames, and its ensuing computational cost, are unnecessary for evaluating reinforcement learning algorithms within the ALE.

The most recent version of the ALE contains a stochastic execution mechanism designed to ward against trajectory overfitting.Specifically, on each frame the environment rejects the agent's selected action with probability $p = 0.25$. Although DQN is mostly robust to stochastic execution, there are a few games in which its performance is reduced. On a score scale normalized with respect to the random and DQN agents, C51 obtains mean and median score improvements of 126% and 21.5% respectively, confirming the benefits of C51 beyond the deterministic setting.

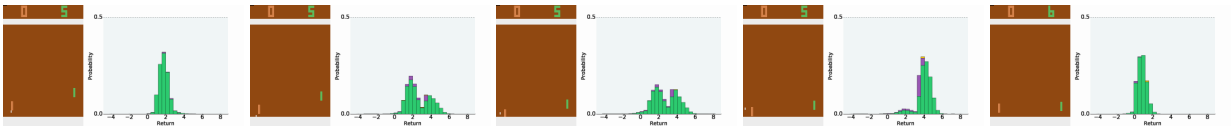


Figure 5. Intrinsic stochasticity in PONG.

	Mean	Median	> H.B.	> DQN
DQN	228%	79%	24	0
DDQN	307%	118%	33	43
DUEL.	373%	151%	37	50
PRIOR.	434%	124%	39	48
PR. DUEL.	592%	172%	39	44
UNREAL†	880%	**250%**	-	-
C51	**1010%**	178%	**40**	**50**

Figure 6. Mean and median scores across 57 Atari games, measured as percentages of human baseline (H.B., Nair et al., 2015).

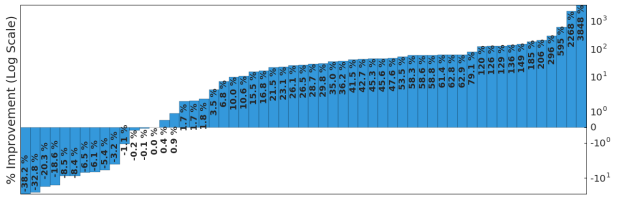


Figure 7. Percentage improvement, per-game, of C51 over Double DQN, computed using van Hasselt et al.'s method.

6. Discussion

In this work we sought a more complete picture of reinforcement learning, one that involves value distributions. We found that learning value distributions is a powerful notion that allows us to surpass most gains previously made on Atari 2600, without further algorithmic adjustments.

6.1. Why does learning a distribution matter?

It is surprising that, when we use a policy which aims to maximize expected return, we should see any difference in performance. The distinction we wish to make is that *learning distributions matters in the presence of approximation*. We now outline some possible reasons.

Reduced chattering. Our results from Section 3.4 highlighted a significant instability in the Bellman optimality operator. When combined with function approximation, this instability may prevent the policy from converging, what Gordon (1995) called *chattering*. We believe the gradient-based categorical algorithm is able to mitigate these effects by effectively averaging the different distributions, similar to conservative policy iteration (Kakade & Langford, 2002). While the chattering persists, it is integrated to the approximate solution.

State aliasing. Even in a deterministic environment, state aliasing may result in effective stochasticity. McCallum (1995), for example, showed the importance of coupling representation learning with policy learning in partially observable domains. We saw an example of state aliasing in PONG, where the agent could not exactly predict the reward timing. Again, by explicitly modelling the resulting distribution we provide a more stable learning target.

A richer set of predictions. A recurring theme in artificial intelligence is the idea of an agent learning from a multitude of predictions (Caruana 1997; Utgoff & Stracuzzi 2002; Sutton et al. 2011; Jaderberg et al. 2017). The distributional approach naturally provides us with a rich set of auxiliary predictions, namely: the probability that the return will take on a particular value. Unlike previously proposed approaches, however, the accuracy of these predictions is tightly coupled with the agent's performance.

Framework for inductive bias. The distributional perspective on reinforcement learning allows a more natural framework within which we can impose assumptions about the domain or the learning problem itself. In this work we used distributions with support bounded in $[V_{\text{MIN}}, V_{\text{MAX}}]$. Treating this support as a hyperparameter allows us to change the optimization problem by treating all extremal returns (e.g. greater than V_{MAX}) as equivalent. Surprisingly, a similar value clipping in DQN significantly degrades performance in most games. To take another example: interpreting the discount factor γ as a proper probability, as some authors have argued, leads to a different algorithm.

Well-behaved optimization. It is well-accepted that the KL divergence between categorical distributions is a reasonably easy loss to minimize. This may explain some of our empirical performance. Yet early experiments with alternative losses, such as KL divergence between continuous densities, were not fruitful, in part because the KL divergence is insensitive to the values of its outcomes. A closer minimization of the Wasserstein metric should yield even better results than what we presented here.

In closing, we believe our results highlight the need to account for distribution in the design, theoretical or otherwise, of algorithms.

† The UNREAL results are not altogether comparable, as they were generated in the asynchronous setting with per-game hyperparameter tuning (Jaderberg et al., 2017).

Acknowledgements

The authors acknowledge the important role played by their colleagues at DeepMind throughout the development of this work. Special thanks to Yee Whye Teh, Alex Graves, Joel Veness, Guillaume Desjardins, Tom Schaul, David Silver, Andre Barreto, Max Jaderberg, Mohammad Azar, Georg Ostrovski, Bernardo Avila Pires, Olivier Pietquin, Audrunas Gruslys, Tom Stepleton, Aaron van den Oord; and particularly Chris Maddison for his comprehensive review of an earlier draft. Thanks also to Marek Petrik for pointers to the relevant literature.

References

Azar, Mohammad Gheshlaghi, Munos, Rémi, and Kappen, Hilbert. On the sample complexity of reinforcement learning with a generative model. In *Proceedings of the International Conference on Machine Learning*, 2012.

Bellemare, Marc G, Naddaf, Yavar, Veness, Joel, and Bowling, Michael. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Bellemare, Marc G., Danihelka, Ivo, Dabney, Will, Mohamed, Shakir, Lakshminarayanan, Balaji, Hoyer, Stephan, and Munos, Rémi. The cramer distance as a solution to biased wasserstein gradients. *arXiv*, 2017.

Bellman, Richard E. *Dynamic programming*. Princeton University Press, Princeton, NJ, 1957.

Bertsekas, Dimitri P. and Tsitsiklis, John N. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

Bickel, Peter J. and Freedman, David A. Some asymptotic theory for the bootstrap. *The Annals of Statistics*, pp. 1196–1217, 1981.

Billingsley, Patrick. *Probability and measure*. John Wiley & Sons, 1995.

Caruana, Rich. Multitask learning. *Machine Learning*, 28 (1):41–75, 1997.

Chung, Kun-Jen and Sobel, Matthew J. Discounted mdps: Distribution functions and exponential utility maximization. *SIAM Journal on Control and Optimization*, 25(1): 49–62, 1987.

Dearden, Richard, Friedman, Nir, and Russell, Stuart. Bayesian Q-learning. In *Proceedings of the National Conference on Artificial Intelligence*, 1998.

Engel, Yaakov, Mannor, Shie, and Meir, Ron. Reinforcement learning with gaussian processes. In *Proceedings of the International Conference on Machine Learning*, 2005.

Geist, Matthieu and Pietquin, Olivier. Kalman temporal differences. *Journal of Artificial Intelligence Research*, 39:483–532, 2010.

Gordon, Geoffrey. Stable function approximation in dynamic programming. In *Proceedings of the Twelfth International Conference on Machine Learning*, 1995.

Harutyunyan, Anna, Bellemare, Marc G., Stepleton, Tom, and Munos, Rémi. Q(λ) with off-policy corrections. In *Proceedings of the Conference on Algorithmic Learning Theory*, 2016.

Hoffman, Matthew D., de Freitas, Nando, Doucet, Arnaud, and Peters, Jan. An expectation maximization algorithm for continuous markov decision processes with arbitrary reward. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2009.

Jaderberg, Max, Mnih, Volodymyr, Czarnecki, Wojciech Marian, Schaul, Tom, Leibo, Joel Z, Silver, David, and Kavukcuoglu, Koray. Reinforcement learning with unsupervised auxiliary tasks. *Proceedings of the International Conference on Learning Representations*, 2017.

Jaquette, Stratton C. Markov decision processes with a new optimality criterion: Discrete time. *The Annals of Statistics*, 1(3):496–505, 1973.

Kakade, Sham and Langford, John. Approximately optimal approximate reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2002.

Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. *Proceedings of the International Conference on Learning Representations*, 2015.

Lattimore, Tor and Hutter, Marcus. PAC bounds for discounted MDPs. In *Proceedings of the Conference on Algorithmic Learning Theory*, 2012.

Mannor, Shie and Tsitsiklis, John N. Mean-variance optimization in markov decision processes. 2011.

McCallum, Andrew K. *Reinforcement learning with selective perception and hidden state*. PhD thesis, University of Rochester, 1995.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A, Veness, Joel, Bellemare, Marc G, Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K, Ostrovski, Georg, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Morimura, Tetsuro, Hachiya, Hirotaka, Sugiyama, Masashi, Tanaka, Toshiyuki, and Kashima, Hisashi.

Parametric return density estimation for reinforcement learning. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2010a.

Morimura, Tetsuro, Sugiyama, Masashi, Kashima, Hisashi, Hachiya, Hirotaka, and Tanaka, Toshiyuki. Nonparametric return distribution approximation for reinforcement learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 799–806, 2010b.

Nair, Arun, Srinivasan, Praveen, Blackwell, Sam, Alcicek, Cagdas, Fearon, Rory, De Maria, Alessandro, Panneershelvam, Vedavyas, Suleyman, Mustafa, Beattie, Charles, and Petersen, Stig et al. Massively parallel methods for deep reinforcement learning. In *ICML Workshop on Deep Learning*, 2015.

Prashanth, LA and Ghavamzadeh, Mohammad. Actor-critic algorithms for risk-sensitive mdps. In *Advances in Neural Information Processing Systems*, 2013.

Puterman, Martin L. *Markov Decision Processes: Discrete stochastic dynamic programming*. John Wiley & Sons, Inc., 1994.

Rösler, Uwe. A fixed point theorem for distributions. *Stochastic Processes and their Applications*, 42(2):195–214, 1992.

Schaul, Tom, Quan, John, Antonoglou, Ioannis, and Silver, David. Prioritized experience replay. In *Proceedings of the International Conference on Learning Representations*, 2016.

Sobel, Matthew J. The variance of discounted markov decision processes. *Journal of Applied Probability*, 19(04): 794–802, 1982.

Sutton, Richard S. and Barto, Andrew G. *Reinforcement learning: An introduction*. MIT Press, 1998.

Sutton, R.S., Modayil, J., Delp, M., Degris, T., Pilarski, P.M., White, A., and Precup, D. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *Proceedings of the International Conference on Autonomous Agents and Multiagents Systems*, 2011.

Tamar, Aviv, Di Castro, Dotan, and Mannor, Shie. Learning the variance of the reward-to-go. *Journal of Machine Learning Research*, 17(13):1–36, 2016.

Tieleman, Tijmen and Hinton, Geoffrey. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 2012.

Toussaint, Marc and Storkey, Amos. Probabilistic inference for solving discrete and continuous state markov decision processes. In *Proceedings of the International Conference on Machine Learning*, 2006.

Tsitsiklis, John N. On the convergence of optimistic policy iteration. *Journal of Machine Learning Research*, 3:59–72, 2002.

Utgoff, Paul E. and Stracuzzi, David J. Many-layered learning. *Neural Computation*, 14(10):2497–2529, 2002.

Van den Oord, Aaron, Kalchbrenner, Nal, and Kavukcuoglu, Koray. Pixel recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*, 2016.

van Hasselt, Hado, Guez, Arthur, and Silver, David. Deep reinforcement learning with double Q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

Veness, Joel, Bellemare, Marc G., Hutter, Marcus, Chua, Alvin, and Desjardins, Guillaume. Compress and control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

Wang, Tao, Lizotte, Daniel, Bowling, Michael, and Schuurmans, Dale. Dual representations for dynamic programming. *Journal of Machine Learning Research*, pp. 1–29, 2008.

Wang, Ziyu, Schaul, Tom, Hessel, Matteo, Hasselt, Hado van, Lanctot, Marc, and de Freitas, Nando. Dueling network architectures for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2016.

White, D. J. Mean, variance, and probabilistic criteria in finite markov decision processes: a review. *Journal of Optimization Theory and Applications*, 56(1):1–29, 1988.