Statistics and Samples in Distributional Reinforcement Learning

Mark Rowland [1] **Robert Dadashi** [2] **Saurabh Kumar** [2] **Rémi Munos** [1] **Marc G. Bellemare** [2] **Will Dabney** [1]

Abstract

We present a unifying framework for designing and analysing distributional reinforcement learning (DRL) algorithms in terms of recursively estimating statistics of the return distribution. Our key insight is that DRL algorithms can be decomposed as the combination of some statistical estimator and a method for imputing a return distribution consistent with that set of statistics. With this new understanding, we are able to provide improved analyses of existing DRL algorithms as well as construct a new algorithm (EDRL) based upon estimation of the *expectiles* of the return distribution. We compare EDRL with existing methods on a variety of MDPs to illustrate concrete aspects of our analysis, and develop a deep RL variant of the algorithm, ER-DQN, which we evaluate on the Atari-57 suite of games.

1. Introduction

In reinforcement learning (RL), a central notion is the *return*, the sum of discounted rewards. Typically, the average of these returns is estimated by a value function and used for policy improvement. Recently, however, approaches that attempt to learn the distribution of the return have been shown to be surprisingly effective (Morimura et al., 2010a;b; Bellemare et al., 2017; Dabney et al., 2017; 2018; Gruslys et al., 2018); we refer to the general approach of learning return distributions as *distributional RL* (DRL).

Despite impressive experimental performance (Bellemare et al., 2017; Barth-Maron et al., 2018; Dabney et al., 2018) and fundamental theoretical results (Rowland et al., 2018; Qu et al., 2018; Bellemare et al., 2019), it remains challenging to develop and analyse DRL algorithms. In this paper, we propose to address these challenges by phrasing DRL algorithms in terms of recursive estimation of sets of statistical functionals of the return distribution. We observe that DRL algorithms can be viewed as combining a statistical estimator with a procedure we refer to as an *imputation strategy*, which generates a return distribution consistent with the set of statistical estimates. This highly general approach (see Figure 1) requires a precise treatment of the differing roles of statistics and samples in distributional RL.

Using this framework we are able to provide new theoretical results for existing DRL algorithms as well as derive a new algorithm based on the expectiles of the return distribution. More importantly, our novel approach applies to a large class of statistical functionals and imputation strategies, suggesting several avenues for future research. Specifically, we provide answers to the following questions:

(i) Can we describe existing DRL algorithms in a unifying framework, and could such a framework be used to develop new algorithms?
(ii) What statistical functionals of the return distribution can be learnt *exactly* through Bellman updates?
(iii) If certain statistics cannot be learnt exactly, how can we estimate them in a principled manner, and give guarantees on their approximation error relative to the true values of these statistics?

After reviewing relevant background material, we begin with (i) by presenting a new framework for understanding DRL, in terms of a set of *statistics* to be learnt and an *imputation strategy* for specifying a dynamic programming update. We then formalise (ii) by introducing the notion of *Bellman closedness* for collections of statistical functionals, and show that in a wide class of such functionals, the only properties of return distributions that can be learnt exactly through Bellman updates are moments. Interestingly, this rules out statistical functionals such as quantiles that have formed the basis of successful existing DRL algorithms. However, we then address (iii) by showing that the framework allows us to give guarantees on the approximation error introduced in learning these statistics, through the notion of *approximate Bellman closedness*. We apply the framework developed in answering these questions to the case of *expectiles* to develop a new distributional RL algorithm, which we term Expectile Distributional RL (EDRL). Finally, we test these new insights on a variety of MDPs and larger-scale environments to illustrate and expand on the theoretical contributions developed earlier in the paper.

[1]DeepMind [2]Google Brain. Correspondence to: Mark Rowland <markrowland@google.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

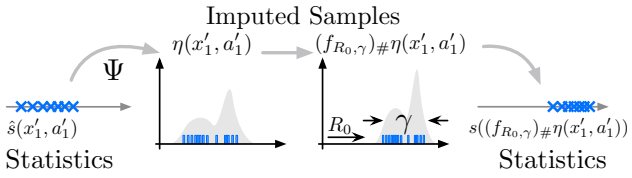

Figure 1. Illustration of learning with imputed samples from sets of statistics. Left: A distribution is imputed from the current statistical estimate. Middle: The distributional Bellman operator is applied to the imputed distribution. Right: New statistics are estimated based upon samples from the imputed distribution.

2. Background

Consider a Markov decision process $(\mathcal{X}, \mathcal{A}, p, \gamma, \mathcal{R})$ with finite state space $\mathcal{X}$, finite action space $\mathcal{A}$, transition kernel $p : \mathcal{X} \times \mathcal{A} \to \mathscr{P}(\mathcal{X})$, discount rate $\gamma \in [0, 1)$, and reward distributions $\mathcal{R}(x, a) \in \mathscr{P}(\mathbb{R})$ for each $(x, a) \in \mathcal{X} \times \mathcal{A}$. Thus, if an agent is at state $X_t \in \mathcal{X}$ at time $t \in \mathbb{N}_0$, and an action $A_t \in \mathcal{A}$ is taken, the agent transitions to a state $X_{t+1} \sim p(\cdot|X_t, A_t)$ and receives a reward $R_t \sim \mathcal{R}(X_t, A_t)$. We now briefly review two principal goals in reinforcement learning.

Firstly, given a Markov policy $\pi : \mathcal{X} \to \mathscr{P}(\mathcal{A})$, *evaluation* of π consists of computing the expected returns $Q^\pi(x, a) = \mathbb{E}_\pi \left[\sum_{t=0}^\infty \gamma^t R_t | X_0 = x, A_0 = a\right]$, where $\mathbb{E}_\pi$ indicates that at each time step $t \in \mathbb{N}$, the agent's action A_t is sampled from $\pi(\cdot|X_t)$. Secondly, the task of *control* consists of finding a policy $\pi : \mathcal{X} \to \mathscr{P}(\mathcal{A})$ for which the expected returns are maximised.

2.1. Bellman Equations

The classical *Bellman equation* (Bellman, 1957) relates expected returns at each state-action pair $(x, a) \in \mathcal{X} \times \mathcal{A}$ to the expected returns at possible next states in the MDP by:

$$Q^\pi(x, a) = \mathbb{E}_\pi[R_0 + \gamma Q^\pi(X_1, A_1) | X_0 = x, A_0 = a]\,. \quad (1)$$

This gives rise to the following fixed-point iteration scheme

$$Q(x, a) \leftarrow \mathbb{E}_\pi[R_0 + \gamma Q(X_1, A_1) | X_0 = x, A_0 = a]\,, \quad (2)$$

for updating a collection of *approximations* $(Q(x, a)|(x, a) \in \mathcal{X} \times \mathcal{A})$ towards their true values. This fundamental algorithm, together with techniques from approximate dynamic programming and stochastic approximation, allows expected returns in an MDP to be learnt and improved upon, forming the basis of all value-based RL (Sutton & Barto, 2018).

The *distributional Bellman equation* describes a similar relationship to Equation (1) at the level of probability distributions (Morimura et al., 2010a;b; Bellemare et al., 2017). Letting $\eta_\pi(x, a) \in \mathscr{P}(\mathbb{R})$ be the *distribution* of the random return $\sum_{t=0}^\infty \gamma^t R_t \,|X_0 = x, A_0 = a$ when actions are selected according to π, we have

$$\begin{aligned} \eta_\pi(x, a) &= (\mathcal{T}^\pi \eta_\pi)(x, a), \quad (3) \\ &= \mathbb{E}_\pi[(f_{R_0,\gamma})_\# \eta_\pi(X_1, A_1) | X_0 = x, A_0 = a]\,, \end{aligned}$$

where the expectation gives a mixture distribution over next-states, $f_{r,\gamma} : \mathbb{R} \to \mathbb{R}$ is defined by $f_{r,\gamma}(x) = r + \gamma x$, and $g_\# \mu \in \mathscr{P}(\mathbb{R})$ is the pushforward of the measure μ through the function g, so that for all Borel subsets $A \subseteq \mathbb{R}$, we have $g_\# \mu(A) = \mu(g^{-1}(A))$ (Rowland et al., 2018). Stated in terms of the random return $Z^\pi(x, a)$, distributed according to $\eta_\pi(x, a)$, this takes a more familiar form with

$$Z^\pi(x, a) \overset{D}{=} R_0 + \gamma Z^\pi(X_1, A_1)\,.$$

In analogy with Expression (2), an update operation could be defined from Equation (3) to move a collection of approximate distributions $(\eta(x, a)|(x, a) \in \mathcal{X} \times \mathcal{A})$ towards the true return distributions. However, since the space of distributions $\mathscr{P}(\mathbb{R})$ is infinite-dimensional, it is typically impossible to work directly with the distributional Bellman equation, and existing approaches to distributional RL generally rely on parametric approximations to this equation; we briefly review some important examples of these approaches below.

2.2. Categorical and Quantile Distributional RL

To date, the main approaches to DRL employed at scale have included learning discrete *categorical* distributions (Bellemare et al., 2017; Barth-Maron et al., 2018; Qu et al., 2018), and learning distribution *quantiles* (Dabney et al., 2017; 2018; Zhang et al., 2019); we refer to these approaches as CDRL and QDRL respectively. We give brief accounts of the dynamic programming versions of these algorithms here, with full descriptions of stochastic versions, related results, and visualisations given in Appendix Section A for completeness. We note also that other approaches, such as learning mixtures of Gaussians, have been explored (Barth-Maron et al., 2018).

CDRL. CDRL assumes a *categorical* form for return distributions, taking $\eta(x, a) = \sum_{k=1}^K p_k(x, a)\delta_{z_k}$, where δ_z denotes the Dirac distribution at location z. The values $z_1 < \cdots < z_K$ are an evenly spaced, fixed set of supports, and the probability parameters $p_{1:K}(x, a)$ are learnt. The corresponding Bellman update takes the form

$$\eta(x, a) \leftarrow (\Pi_\mathcal{C} \mathcal{T}^\pi \eta)(x, a),$$

where $\Pi_\mathcal{C} : \mathscr{P}(\mathbb{R}) \to \mathscr{P}(\{z_1, \ldots, z_K\})$ is a projection operator which ensures the right-hand side of the expression above is a distribution supported only on $\{z_1, \ldots, z_K\}$; full details are reviewed in Appendix Section A.

QDRL. In contrast, QDRL assumes a parametric form for return distributions $\eta(x, a) = \frac{1}{K} \sum_{k=1}^K \delta_{z_k(x,a)}$, where now

$z_{1:K}(x,a)$ are learnable parameters. The Bellman update is given by moving the atom location $z_k(x,a)$ in $\eta(x,a)$ to the τ_k-quantile (where $\tau_k = \frac{2k-1}{2K}$) of the target distribution $\mu := (\mathcal{T}^\pi \eta)(x,a)$, defined as the minimiser $q^* \in \mathbb{R}$ of the quantile regression loss

$$\mathrm{QR}(q;\mu,\tau_k) = \mathbb{E}_{Z\sim\mu}[[\tau_k \mathbb{1}_{Z>q} + (1-\tau_k)\mathbb{1}_{Z\leq q}]\,|Z-q|]\,.$$

3. The Role of Statistics in Distributional RL

In this section, we describe a new perspective on existing distributional RL algorithms, with a focus on learning sets of statistics, rather than approximate distributions. We begin with a precise definition.

Definition 3.1 (Statistical functionals). *A* statistical functional *(SF) is a function $s : \mathscr{P}(\mathbb{R}) \to \mathbb{R}$. We also allow statistical functionals to be defined on subsets of $\mathscr{P}(\mathbb{R})$, in situations where an assumption (such as finite moments) is required for the statistic to be defined. We use the term* statistic *to refer to a value taken by an SF.*

The QDRL update described in Section 2.2 is readily interpreted from the perspective of learning SFs; the update extracts the values of a finite set of quantiles from the target distribution, and all other information about the target is lost. It is less obvious whether the CDRL update can also be interpreted as keeping track of a finite set of SFs, but the following lemma shows that this is indeed the case.

Lemma 3.2. *CDRL updates, with distributions supported on $z_1 < \ldots < z_K$, can be interpreted as learning the values of the following statistical functionals of return distributions:*

$$s_{z_k,z_{k+1}}(\mu) = \mathbb{E}_{Z\sim\mu}\left[h_{z_k,z_{k+1}}(Z)\right] \text{ for } k = 1,\ldots,K-1\,,$$

where for $a < b$, $h_{a,b} : \mathbb{R} \to \mathbb{R}$ is a piecewise linear function defined so that $h_{a,b}(x)$ is equal to 1 for $x \leq a$, equal to 0 for $x \geq b$, and linearly interpolating between $h_{a,b}(a)$ and $h_{a,b}(b)$ for $x \in [a,b]$.

Although viewing distributional RL as approximating the return distribution with some parameterisation is intuitive from an algorithmic standpoint, there are advantages to thinking in terms of statistical functionals and their recursive estimation; this perspective allows us to precisely quantify what information is being passed through successive distributional Bellman updates. This in turn leads to new insights in the development and analysis of DRL algorithms. Before addressing these points, we first consider a motivating example where a lack of precision could lead us astray.

3.1. Expectiles

Motivated by the success of QDRL, we consider learning *expectiles* of return distributions, a family of SFs introduced by Newey & Powell (1987) which have proven to be important as risk measures in financial mathematics (Ziegel, 2016). Expectiles generalise the mean in analogy with how quantiles generalise the median; as the goal of RL is to maximise mean returns, we conjectured that expectiles, in particular, might lead to successful DRL algorithms. We begin with a formal definition.

Definition 3.3 (Expectiles). *Given a distribution $\mu \in \mathscr{P}(\mathbb{R})$ with finite second moment, and $\tau \in [0,1]$, the τ-expectile of μ is defined to be the minimiser $q^* \in \mathbb{R}$ of the expectile regression loss $\mathrm{ER}(q;\mu,\tau)$, given by*

$$\mathrm{ER}(q;\mu,\tau) = \mathbb{E}_{Z\sim\mu}\left[[\tau \mathbb{1}_{Z>q} + (1-\tau)\mathbb{1}_{Z\leq q}]\,(Z-q)^2\right].$$

For each $\tau \in [0,1]$, we denote the τ-expectile of μ by $e_\tau(\mu)$.

We remark that: (i) the expectile regression loss is an asymmetric version of the squared loss, just as the quantile regression loss is an asymmetric version of the absolute value loss; and (ii) the $1/2$-expectile of μ is simply its mean. Because of this, we can attempt to derive an algorithm by replacing the quantile regression loss in QDRL with the expectile regression loss in Definition 3.3, so as to learn the expectiles corresponding to $\tau_1,\ldots,\tau_K \in [0,1]$.

Following this logic, we again take approximate distributions of the form $\eta(x,a) = \frac{1}{K}\sum_{k=1}^K \delta_{z_k(x,a)}$, and we perform updates according to

$$z_k(x,a) \leftarrow \operatorname*{argmin}_{q\in\mathbb{R}} \mathrm{ER}(q;\mu,\tau_k)\,, \tag{4}$$

where $\mu = (\mathcal{T}^\pi\eta)(x,a)$ is the target distribution.

In practice, however, this algorithm does not perform as we might expect, and in fact the variance of the learnt distributions collapses as training proceeds, indicating that the algorithm does not approximate the true expectiles in any reasonable sense. In Figure 2, we illustrate this point by comparing the learnt statistics for this "naive" approach with those of CDRL and our proposed algorithm EDRL (introduced in Section 3.3). All methods accurately approximate the immediate reward distribution (right), but as successive Bellman updates are applied the different algorithms show characteristic approximation errors. The CDRL algorithm overestimates the variance of the return distribution due to the projection Π_C splitting probability mass across the discrete support; by contrast, the naive expectile approach underestimates the true variance, quickly converging to a single Dirac.

We observe that there is a *"type error"* present in Expression (4); the parameter being updated, $z_k(x,a)$, has the semantics of a *statistic*, as the minimiser of the ER loss, whilst the parameters appearing in the target distribution $(\mathcal{T}^\pi\eta)(x,a)$ have the semantics of *outcomes/samples*. A

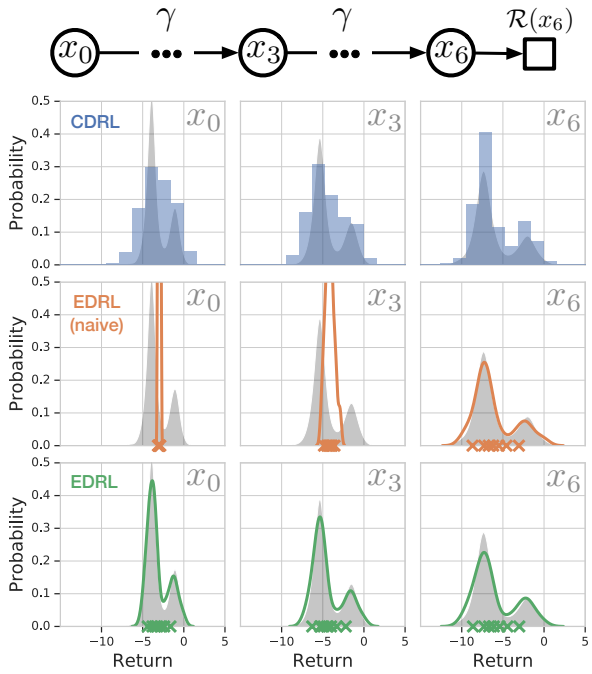

Figure 2. Chain MDP, one action, with bimodal reward distribution at absorbing state x_6 and $\gamma = 0.9$. CDRL (top, blue) fits the true return distribution (grey) well, but overestimates the variance. A naive approach to EDRL (middle, orange) accurately fits the immediate reward distribution at x_6, but quickly collapses to zero variance with successive Bellman updates. Our proposed approach, EDRL, using imputation strategies (bottom, green) provides an accurate approximation through many Bellman updates.

crucial message of this paper is the need to distinguish between statistics and samples in distributional RL; in the next section, we describe a general framework for achieving this.

3.2. Imputation Strategies

If we had access to full return distribution estimates $\eta(x', a')$ at each possible next state-action pair (x', a'), we would be able to avoid the conflation between samples and statistics described in the previous section. Denoting the approximation to the value of a statistical functional s_k at a state-action pair $(x, a) \in \mathcal{X} \times \mathcal{A}$ by $\hat{s}_k(x, a)$, we would like to update according to:

$$\hat{s}_k(x, a) \leftarrow s_k\left((\mathcal{T}^\pi \eta)(x, a)\right). \tag{5}$$

Thus, a principled way in which to design DRL algorithms for collections of SFs is to include an *additional* step in the algorithm in which for any state-action pair (x', a') that we would like to backup from, the estimated statistics $\hat{s}_{1:K}(x', a')$ are converted into a consistent distribution $\eta(x', a')$. This would then allow backups of the form in Expression (5) to be carried out. This notion is formalised in the following definition.

Definition 3.4 (Imputation strategies). *Given a set of statistical functionals $\{s_1, \ldots, s_K\}$, an* imputation strategy *is a function $\Psi : \mathbb{R}^K \to \mathscr{P}(\mathbb{R})$ that maps each vector of statistics to a distribution that has those statistics. Mathematically, Ψ is such that $s_i(\Psi(\sigma_{1:K})) = \sigma_i$, for each $i \in \{1, \ldots, K\}$ and each collection of statistics $\sigma_{1:K} \in \mathbb{R}^K$.*

Thus, an imputation strategy is simply a function that takes in a collection of values for certain statistical functionals, and returns a probability distribution with those values; in some sense, it is a pseudo-inverse of $s_{1:K}$.

Example 3.5 (Imputation strategies in CDRL and QDRL). *In QDRL, the imputation strategy is given by $\Psi(\sigma_{1:K}) = \frac{1}{K}\sum_{k=1}^{K} \delta_{\sigma_k}$. In CDRL, given approximate statistics $\hat{s}_{z_k, z_{k+1}}(x, a)$ for $k = 1, \ldots, K-1$, the imputation strategy is given by selecting the distribution $\sum_{k=1}^{K} p_k \delta_{z_k}$ such that $p_1 = \hat{s}_{z_1, z_2}(x, a)$, $p_k = \hat{s}_{z_k, z_{k+1}}(x, a) - \hat{s}_{z_{k-1}, z_k}(x, a)$ for $k = 2, \ldots, K-1$, and $p_K = 1 - \sum_{k<K} p_k$.*

We now have a general framework for defining principled distributional RL algorithms: (i) select a family of statistical functionals to learn; (ii) select an imputation strategy; (iii) perform (or approximate) updates of the form in Expression (5). We summarise this in Algorithm 1.

Algorithm 1 Generic DRL update algorithm.

Require: Statistic estimates $\hat{s}_{1:K}(x, a)\ \forall (x, a) \in \mathcal{X} \times \mathcal{A}$ and $k = 1, \ldots, K$, imputation strategy Ψ.
Select state-action pair $(x, a) \in \mathcal{X} \times \mathcal{A}$ to update.
Impute distribution at each possible next state-action pair:
$\eta(x', a') = \Psi(\hat{s}_{1:K}(x', a')), \quad \forall (x', a') \in \mathcal{X} \times \mathcal{A}.$
Update statistics at $(x, a) \in \mathcal{X} \times \mathcal{A}$:
$\hat{s}_k(x, a) \leftarrow s_k\left((\mathcal{T}^\pi \eta)(x, a)\right).$

3.3. Expectile Distributional Reinforcement Learning

We now apply the general framework of statistics and imputation strategies developed in Section 3.2 to the specific case of *expectiles*, introduced in Section 3.1. We will define an imputation strategy so that updates of the form given in Expression (5) can be applied to learn expectiles.

The imputation strategy has the task of accepting as input a collection of expectile values $\epsilon_1, \ldots, \epsilon_K$, corresponding to $\tau_1, \ldots, \tau_K \in (0, 1)$, and computing a probability distribution μ such that $e_{\tau_i}(\mu) = \epsilon_i$ for $i = 1, \ldots, K$. Since $\mathrm{ER}(q; \mu, \tau)$ is strictly convex as a function of q, this can be restated as finding a probability distribution μ satisfying the first-order optimality conditions

$$\nabla_q \mathrm{ER}(q; \mu, \tau_i)\big|_{q=\epsilon_i} = 0 \quad \forall i \in [K]. \tag{6}$$

This defines a root-finding problem, but may equivalently be formulated as a minimisation problem, with objective

$$\sum_{i=1}^{K} \left(\nabla_q \mathrm{ER}(q; \mu, \tau_i)\big|_{q=\epsilon_i} \right)^2 . \tag{7}$$

By constraining the distribution μ to be of the form $\frac{1}{N}\sum_{n=1}^{N} \delta_{z_n}$ and viewing the minimisation objective above as a function of $z_{1:N}$, it is straightforwardly verifiable that this minimisation problem is convex. The imputation strategy is thus defined implicitly, by stating that $\Psi(\epsilon_{1:K})$ is given by a minimiser of (7) of the form $\frac{1}{N}\sum_{n=1}^{N} \delta_{z_n}$. We remark that other parametric choices for μ are possible, but the mixture of Dirac deltas described above leads to a particular tractable optimisation problem.

Having established an imputation strategy Ψ, Algorithm 1 now yields a full DRL algorithm for learning expectiles, which we term EDRL. Returning to Figure 2, we observe that EDRL (bottom row) is able to accurately represent the true return distribution, even after many Bellman updates through the chain, and does not exhibit the collapse observed with the naive approach in Section 3.1.

3.4. Stochastic Approximation

Practically speaking, it is often not possible to compute the updates in Expression (5), owing to MDP dynamics being unknown and/or intractable to integrate over. Because of this, it is often necessary to apply stochastic approximation. Let (r, x', a') be a sample of the random variables (R_0, X_1, A_1), obtained by direct interaction with the environment. Then, we update $\hat{s}_k(x, a)$ using the gradient of a loss function $L_k : \mathbb{R} \times \mathscr{P}(\mathbb{R}) \to \mathbb{R}$:

$$\nabla_{\hat{s}_k(x,a)} L_k(\hat{s}_k(x, a); (f_{r,\gamma})_{\#}\eta(x', a')) . \tag{8}$$

For EDRL, a natural such loss function for the estimated statistic $\hat{s}_k(x, a)$ is the expectile regression loss of Definition 3.3 at τ_k; this yields a stochastic version of EDRL, described in Algorithm 2.

To ensure convergence of these stochastic gradient updates to the correct statistic, it should be the case that the expectation of the (sub-)gradient (8) at the true value of the statistic is equal to 0. It can be verified that this is the case whenever (i) the true statistic q^* of a distribution μ satisfies $q^* = \mathrm{argmin}_{q\in\mathbb{R}} L_k(q; \mu)$, and (ii) the loss L_k is *affine* in the probability distribution argument. M-estimator losses and their associated statistical functionals (Huber & Ronchetti, 2009) satisfy these conditions, and thus represent a large family of SFs to which this approach to DRL could immediately be applied; the SFs in CDRL, QDRL and EDRL are all special cases of M-estimators.

Algorithm 2 Stochastic EDRL update algorithm.

Require: Expectile estimates $\hat{s}_k(x, a)$ for each $(x, a) \in \mathcal{X} \times \mathcal{A}$ and $k = 1, \ldots, K$.
Collect sample (x, a, r, x', a').
Impute distribution $\frac{1}{K}\sum_{k=1}^{K} \delta_{z_k}$ from target expectiles $\hat{s}_{1:K}(x', a')$ by solving (6) or minimising (7).
Scale/translate samples $z_i \leftarrow r + \gamma z_i \ \forall i$.
Update estimated expectiles at $(x, a) \in \mathcal{X} \times \mathcal{A}$ by computing the gradients

$$\nabla_{\hat{s}_k(x,a)} \textstyle\sum_{k=1}^{K} \mathrm{ER}(\hat{s}_k(x, a); \frac{1}{N}\sum_{n=1}^{N} \delta_{z_n}, \tau_k)$$

for each $k = 1, \ldots, K$.

4. Analysing Distributional RL

We now use the framework of statistics and imputations strategies developed in Section 3 to build a deeper understanding of the accuracy with which statistics in distributional RL may be learnt via Bellman updates.

4.1. Bellman Closedness

The classical Bellman equation (1) shows that there is a closed-form relationship between expected returns at each state-action pair of an MDP; if the goal is to learn expected returns, we are not required to keep track of any other statistics of the return distributions. This well-known observation, together with the new interpretation of DRL algorithms as learning collections of statistics of return distributions, motivates a more general question:

> "Given a set of statistical functionals $\{s_1, \ldots, s_K\}$, if we want to learn the values $s_{1:K}(\eta_\pi(x, a))$ for all $(x, a) \in \mathcal{X} \times \mathcal{A}$ via dynamic programming, is it sufficient to keep track of *only* these values?"

The following definition formalises this question.

Definition 4.1 (Bellman closedness). *A set of statistical functionals $\{s_1, \ldots, s_K\}$ is* Bellman closed *if for each $(x, a) \in \mathcal{X} \times \mathcal{A}$, the statistics $s_{1:K}(\eta_\pi(x, a))$ can be expressed, in an MDP-independent manner, in terms of the random variables R_0 and $s_{1:K}(\eta_\pi(X_1, A_1))|X_0 = x, A_0 = a$, and the discount factor γ. We refer to any such expression for a Bellman closed set of statistical functionals as a* Bellman equation*, and write $\mathcal{T}^\pi : (\mathbb{R}^K)^{\mathcal{X}\times\mathcal{A}} \to (\mathbb{R}^K)^{\mathcal{X}\times\mathcal{A}}$ for the corresponding operator such that the Bellman equation can be written*

$$\mathbf{s}^\pi = \mathcal{T}^\pi \mathbf{s}^\pi , \tag{9}$$

where $\mathbf{s}^\pi = (s_{1:K}(\eta_\pi(x, a))|(x, a) \in \mathcal{X} \times \mathcal{A})$.

Thus, the singleton set consisting of the mean SF is Bellman closed; the corresponding Bellman equation is Equation (1).

It is also known that the set consisting of the mean and variance SFs are Bellman closed (Sobel, 1982). Given a Bellman closed set of SFs $\{s_1, \ldots, s_K\}$ with contractive Bellman operator $\mathcal{T}^\pi$, the true statistics of the return distributions of an MDP can be calculated via a fixed-point iteration scheme. In contrast, if a collection of SFs $s_{1:K}$ is *not* Bellman closed, there is no Bellman equation relating the statistics of the return distributions, and consequently it is not possible to learn the statistics *exactly* using dynamic programming in a self-contained way; the set of SFs must either be enlarged to make it Bellman closed, or an imputation strategy can be used to perform backups as described in Section 3.2.

An important class of Bellman closed sets of statistical functionals are given in the following result (Sobel, 1982; Lattimore & Hutter, 2012).

Lemma 4.2. *For each $K \in \mathbb{N}$, the set of statistical functionals consisting of the first K moments is Bellman closed.*

The next result shows that across a wide range of SFs, collections of moments are effectively the only finite sets of statistics that are Bellman closed; the proof relies on a result of Engert (1970) which characterises finite-dimensional vector spaces of measurable functions closed under translation.

Theorem 4.3. *The only finite sets of SFs of the form $s(\mu) = \mathbb{E}_{Z\sim\mu}[h(Z)]$ that are Bellman closed are given by collections of SFs $s_1, \ldots, s_K : \mathscr{P}(\mathbb{R}) \to \mathbb{R}$ with the property that the linear span $\{\sum_{k=0}^{K} \alpha_k s_k | \alpha_k \in \mathbb{R}\ \forall k\}$ is equal to the linear span of the set of moment functionals $\{\mu \mapsto \mathbb{E}_{Z\sim\mu}\left[Z^l\right] | l = 0, \ldots, L\}$, for some $L \leq K$, where s_0 is the constant functional equal to 1.*

We believe this to be an important novel result, which helps to highlight how rare it is for sets of SFs to be Bellman closed. One important corollary of Theorem 4.3, given the characterisation of CDRL as learning expectations of return distributions in Lemma 3.2, is that the sets of SFs learnt in CDRL are *not* Bellman closed. A similar result holds for QDRL, and we record these facts in the following result.

Lemma 4.4. *The sets of statistical functionals learnt under (i) CDRL, and (ii) QDRL, are not Bellman closed.*

The immediate upshot of this is that in general, the *learnt* values of statistics in distributional RL algorithms need not correspond exactly to the true underlying values for the MDP (even in tabular settings), as the statistics propagated through DRL dynamic programming updates are not sufficient to determine the statistics we seek to learn. This inexactness was noted specifically for CDRL and QDRL in the original papers (Bellemare et al., 2017; Dabney et al., 2017). In this paper, our analysis and experiments confirm that these artefacts arise even with tabular agents in fully-observed domains, thus representing intrinsic properties of the distributional RL algorithms concerned. However, empirically the distributions learnt by these algorithms are often accurate. In the next section, we provide theoretical guarantees that describe this phenomenon quantitatively.

4.2. Approximate Bellman Closedness

In light of the results on Bellman closedness in Section 4.1, we might ask in what sense the values of the statistical functionals learnt by DRL algorithms relate to the corresponding true underlying values for the MDP concerned. A key task in this analysis is to formalise the notion of *low approximation error* in DRL algorithms that seek to learn collections of SFs that are not Bellman closed. Perhaps surprisingly, in general it is not possible to simultaneously achieve low approximation error on all SFs in a non-Bellman closed set; we give several examples in Appendix Section C.

Due to the fact that it is in general not possible to learn SFs uniformly well, we formalise the notion of approximate closedness in terms of the *average* approximation error across a collection of SFs, as described below.

Definition 4.5 (Approximate Bellman closedness). *A collection of statistical functionals $s_1, \ldots, s_K$, together with an imputation strategy Ψ, are said to be* ε-approximately Bellman closed *for a class $\mathcal{M}$ of MDPs if, for each MDP $M = (\mathcal{X}, \mathcal{A}, p, \gamma, \mathcal{R})$ in $\mathcal{M}$ and every policy $\pi \in \mathscr{P}(\mathcal{A})^{\mathcal{X}}$, we have*

$$\sup_{(x,a)\in\mathcal{X}\times\mathcal{A}} \frac{1}{K} \sum_{k=1}^{K} |s_k(\eta_\pi(x,a)) - \hat{s}_k(x,a)| \leq \varepsilon\,,$$

where $\hat{s}_k(x,a)$ denotes the learnt value of the SF s_k for the return distribution at the state-action pair $(x,a) \in \mathcal{X}\times\mathcal{A}$.

We can now study the approximation errors of CDRL and QDRL in light of this new concept. Whilst the analysis in Section 4.1 shows that CDRL and QDRL necessarily induce some approximation error due to lack of Bellman closedness, the following results reassuringly show that the approximation error can be made arbitrarily small by increasing the number of learnt SFs.

Theorem 4.6. *Consider the class $\mathcal{M}$ of MDPs with a fixed discount factor $\gamma \in [0,1)$, and immediate reward distributions supported on $[-R_{max}, R_{max}]$. The set of SFs and imputation strategy corresponding to CDRL with evenly spaced bin locations at $-R_{max}/(1-\gamma) = z_1 < \cdots < z_K = R_{max}/(1-\gamma)$ is ε-approximately Bellman closed for $\mathcal{M}$, where $\varepsilon = \frac{\gamma}{2(1-\gamma)(K-1)}$.*

Theorem 4.7. *Consider the class of MDPs $\mathcal{M}$ with a fixed discount factor $\gamma \in [0,1)$, and immediate reward distributions supported on $[-R_{max}, R_{max}]$. Then the collection of quantile SFs $s_k(\mu) = F_\mu^{-1}(\frac{2k-1}{2K})$ for $k = 1, \ldots, K$, with the standard QDRL imputation strategy, is ε-approximately Bellman closed for $\mathcal{M}$, where $\varepsilon = \frac{2R_{max}(5-2\gamma)}{(1-\gamma)^2 K}$.*

Both of these extend existing analyses for CDRL and QDRL. In particular, Theorem 4.6 improves on the bound of Rowland et al. (2018), and Theorem 4.7 is the first approximation result for QDRL; existing results dealt solely with contraction mappings under W_∞ (Dabney et al., 2017).

4.3. Mean Consistency

So far, our discussion has been focused around *evaluation*. For *control*, it is important to correctly estimate *expected returns*, so that accurate policy improvement can be performed. We analyse to what extent expected returns are correctly learnt in existing DRL algorithms in the following result. The result for CDRL has been shown previously (Rowland et al., 2018; Lyle et al., 2019), but our proof here gives a new perspective in terms of statistics.

Lemma 4.8. *(i) Under CDRL updates using support locations $z_1 < \cdots < z_K$, if all approximate reward distributions have support bounded in $[z_1, z_K]$, expected returns are exactly learnt. (ii) Under QDRL updates, expected returns are not exactly learnt.*

Importantly, for EDRL, as long as the $1/2$-expectile (i.e. the mean) is included in the set of SFs, expected returns are learnt exactly; we return to this point in Section 5.2.

5. Experimental Results

We first present results with a tabular version of EDRL to illustrate and expand upon the theoretical results presented in Sections 3 and 4. We then combine the EDRL update with a DQN-style architecture to create a novel deep RL algorithm (ER-DQN), and evaluate performance on the Atari-57 environments. There are several ways in which the root-finding/optimisation problems (6) and (7) may be solved in practice. In our experiments, we use a SciPy optimisation routine (Jones et al., 2001).

5.1. Tabular Policy Evaluation

We consider a variant of the classic N-chain domain (see Figure 3), with rewards and dynamics chosen to yield multimodal reward distributions; see Appendix E for a full specification. We consider evaluation of the optimal policy π^* using both EDRL and QDRL.

EDRL. We ran two DRL algorithms on this N-Chain environment: (i) EDRL, using a SciPy optimisation routine to impute target samples at each step; and (ii) *EDRL-Naive*, using the update described in Section 3.1. In Figure 3 we illustrate the collapse of the start state expectiles learned by the EDRL-Naive algorithm with 9 expectiles, which leads to high expectile estimation error, measured as in Definition 4.5. In Figure 4, we show that this error grows as the distance to the goal state and number of expectiles learned increase. In contrast, under EDRL this error remains relatively low for varying numbers of expectiles and distances to the goal. In Appendix E, we show that this observation generalises to other return distributions in the N-Chain.

QDRL. In practical implementations, QDRL often minimises the Huber-quantile loss

$$\operatorname*{argmin}_{q \in \mathbb{R}} \mathbb{E}_{Z \sim \mu}[(\tau \mathbb{1}_{Z>q} + (1-\tau)\mathbb{1}_{Z<q}) H_\kappa(Z-q)], \quad (10)$$

rather than the quantile loss for numerical stability, where H_κ is the Huber loss function with width parameter κ, as in Dabney et al. (2017) (we set $\kappa = 1$). As with naive EDRL, simply replacing the quantile regression loss in QDRL with Expression (10) conflates samples and statistics, leading to worse approximation of the distribution. We propose a new algorithm for learning Huber quantiles, *Huber-QDRL-Imputation*, that incorporates an imputation strategy by solving an optimisation problem analogous to (7) in the case of the Huber quantile loss. In Figure 5, we compare this to *Huber-QDRL-Naive*, the standard algorithm for learning Huber quantiles, on the N-chain environment. As in the case of expectiles, the Huber quantile estimation error is vastly reduced when using an imputation strategy.

5.2. Tabular Control

In Section 4.3 we argued for the importance of mean consistency. In Figure 6a we give a simple, five state MDP in which the learned control policy is directly affected by mean consistency. At start state x_0 the agent has the choice of two actions, leading down two paths and culminating in two different reward distributions. The rewards at terminal states x_3 and x_4 are sampled from (shifted) exponential distributions with densities $e^{-\lambda}$ ($\lambda \geq 0$) and $e^{\lambda+1.85}$ ($\lambda \leq 1.85$), respectively. Transitions are deterministic, and $\gamma = 1$. For CDRL, we take bin locations at $(z_1, z_2, z_3) = (0, 1, 2)$.

Figure 6b shows the true return distributions, their expectations, and the means estimated by CDRL, QDRL and EDRL. Due to a lack of mean consistency both CDRL and QDRL learn a sub-optimal greedy policy. For CDRL, this is due to the true return distributions having support outside $[0, 2]$, and for QDRL, this is due to the quantiles not capturing tail behaviour. In contrast, EDRL correctly learns the means of both return distributions, and so is able to act optimally.

5.3. Expectile Regression DQN

To demonstrate the effectiveness of EDRL at scale, we combine the EDRL update in Algorithm 2 with the architecture of QR-DQN to obtain a new deep RL agent, expectile regression DQN (ER-DQN). Precise experimental details and results are given in Appendix Section D. We evaluate ER-DQN on the Arcade Learning Environment (Bellemare et al., 2013). In Figure 7, we plot mean and median hu-

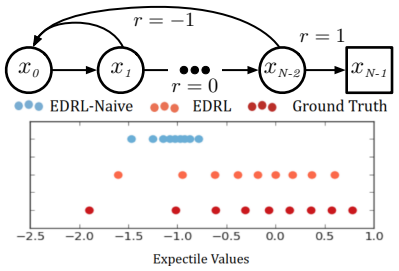

Figure 3. N-Chain environment. Bottom: Expectiles for state x_0 on 15-Chain for π^*.

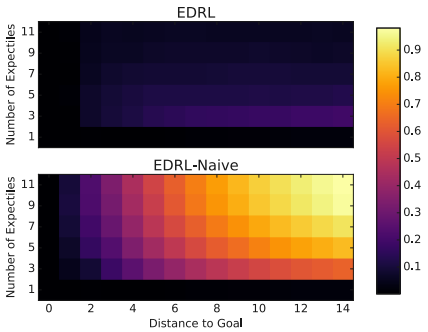

Figure 4. Expectile error for varying numbers of expectiles and different chain lengths.

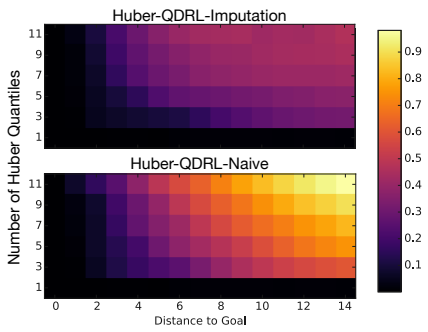

Figure 5. Huber quantile error, varying number of statistics and chain lengths.

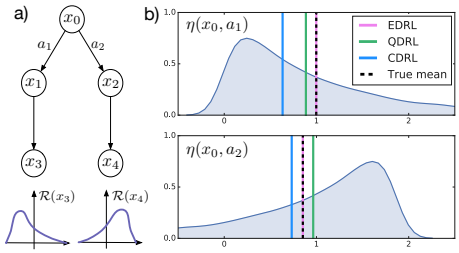

Figure 6. (a) 5-state MDP, reward zero except x_3 and x_4, which have stochastic rewards. (b) True return distributions $\eta(x_0, a_1)$, $\eta(x_0, a_2)$, and value estimated by CDRL, QDRL, and EDRL.

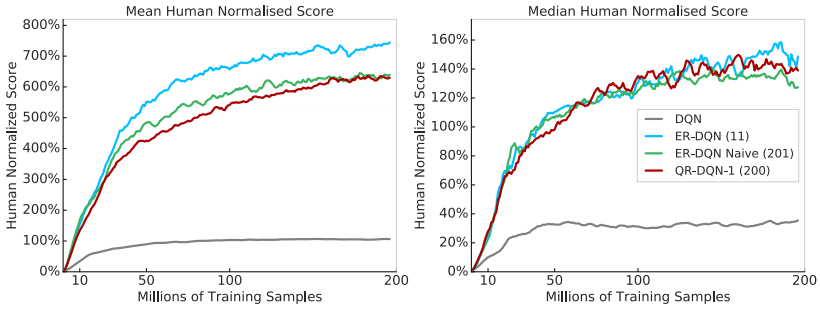

Figure 7. Mean and median human normalised scores across 57 Atari games. Number of statistics learnt for each algorithm indicated in parentheses.

man normalised scores for ER-DQN with 11 expectiles, and compare against DQN, QR-DQN (which learns 200 Huber-quantile SFs), and a naive implementation of ER-DQN that doesn't use an imputation strategy, learning 201 expectiles. All methods were re-run for this paper, and results were averaged over 3 seeds. In practice, we found that with 11 expectiles, ER-DQN already offers strong performance relative to other approaches, and at this level, the additional training overhead due to SciPy optimiser calls is low.

In terms of mean human normalised score, ER-DQN represents a substantial improvement over both QR-DQN and the naive version of ER-DQN that does not use an imputation strategy. We hypothesise that the mean consistency of EDRL (in contrast to other DRL methods; see Section 4.3) is partially responsible for these improvements, and leave further investigation of the role of mean consistency in DRL as a direction for future work. We also remark that the performance of ER-DQN shows that there may be significant practical value in applying the framework developed in this paper to other families of SFs. It remains to be seen if the presence of partial observability induces non-trivial distributions, which could also explain ER-DQN's improved performance in some games. Investigation into the robustness of ER-DQN with regards to the precise imputation strategy used is also a natural question for future work.

6. Conclusion

We have developed a unifying framework for DRL in terms of statistical estimators and imputation strategies. Through this framework, we have developed a new algorithm, EDRL, as well as proposing algorithmic adjustments to an existing approach. We have also used this framework to define the notion of Bellman closedness, and provided new approximation guarantees for existing algorithms.

This paper also opens up several avenues for future research. Firstly, the framework of imputation strategies has the potential to be applied to a wide range of collections of SFs, opening up a large space of new algorithms to explore. Secondly, our analysis has shown that a lack of Bellman closedness necessarily introduces a source of approximation error into many DRL algorithms; it will be interesting to see how this interacts with errors introduced by function approximation. Finally, we have focused on DRL algorithms that can be interpreted as learning values of a finite collection of SFs in this paper. One notable alternative is implicit quantile networks (Dabney et al., 2018), which attempt to learn an uncountable collection of quantiles with a finite-capacity function approximator; it will also be interesting to extend our analysis to this setting.

Acknowledgements

The authors acknowledge the vital contributions of their colleagues at DeepMind, as well as the anonymous reviewers for helpful comments. Thanks to Hado van Hasselt for detailed comments on an earlier draft, and to Georg Ostrovski for useful suggestions regarding the implementation of ER-DQN.

References

Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., TB, D., Muldal, A., Heess, N., and Lillicrap, T. Distributional policy gradients. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

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

Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

Bellemare, M. G., Le Roux, N., Castro, P. S., and Moitra, S. Distributional reinforcement learning with linear function approximation. *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Bellman, R. *Dynamic Programming*. Princeton University Press, 1st edition, 1957.

Dabney, W., Rowland, M., Bellemare, M. G., and Munos, R. Distributional reinforcement learning with quantile regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

Dabney, W., Ostrovski, G., Silver, D., and Munos, R. Implicit quantile networks for distributional reinforcement learning. *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

Engert, M. Finite dimensional translation invariant subspaces. *Pacific Journal of Mathematics*, 32(2):333–343, 1970.

Gruslys, A., Dabney, W., Azar, M. G., Piot, B., Bellemare, M., and Munos, R. The reactor: A fast and sample-efficient actor-critic agent for reinforcement learning. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Huber, P. J. and Ronchetti, E. *Robust Statistics*. Wiley New York, 2nd edition, 2009.

Jones, E., Oliphant, T., and Peterson, P. SciPy: Open source scientific tools for Python, 2001. URL http://www.scipy.org/.

Lattimore, T. and Hutter, M. PAC bounds for discounted MDPs. In *International Conference on Algorithmic Learning Theory (ALT)*, 2012.

Lyle, C., Castro, P. S., and Bellemare, M. G. A comparative analysis of expected and distributional reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

Morimura, T., Sugiyama, M., Kashima, H., Hachiya, H., and Tanaka, T. Nonparametric return distribution approximation for reinforcement learning. *Proceedings of the International Conference on Machine Learning (ICML)*, 2010a.

Morimura, T., Sugiyama, M., Kashima, H., Hachiya, H., and Tanaka, T. Parametric return density estimation for reinforcement learning. *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010b.

Newey, W. K. and Powell, J. L. Asymmetric least squares estimation and testing. *Econometrica: Journal of the Econometric Society*, pp. 819–847, 1987.

Qu, C., Mannor, S., and Xu, H. Nonlinear distributional gradient temporal-difference learning. *arXiv preprint arXiv:1805.07732*, 2018.

Rowland, M., Bellemare, M. G., Dabney, W., Munos, R., and Teh, Y. W. An analysis of categorical distributional reinforcement learning. *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

Sobel, M. J. The variance of discounted Markov decision processes. *Journal of Applied Probability*, 19(4):794–802, 1982.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

Zhang, S., Mavrin, B., Yao, H., Kong, L., and Liu, B. QUOTA: The quantile option architecture for reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

Ziegel, J. F. Coherence and elicitability. *Mathematical Finance*, 26(4):901–918, 2016.