Implicit Quantile Networks for Distributional Reinforcement Learning

Will Dabney [*1] Georg Ostrovski [*1] David Silver [1] Rémi Munos [1]

Abstract

In this work, we build on recent advances in distributional reinforcement learning to give a generally applicable, flexible, and state-of-the-art distributional variant of DQN. We achieve this by using quantile regression to approximate the full quantile function for the state-action return distribution. By reparameterizing a distribution over the sample space, this yields an implicitly defined return distribution and gives rise to a large class of risk-sensitive policies. We demonstrate improved performance on the 57 Atari 2600 games in the ALE, and use our algorithm's implicitly defined distributions to study the effects of risk-sensitive policies in Atari games.

1. Introduction

Distributional reinforcement learning (Jaquette, 1973; Sobel, 1982; White, 1988; Morimura et al., 2010b; Bellemare et al., 2017) focuses on the intrinsic randomness of returns within the reinforcement learning (RL) framework. As the agent interacts with the environment, irreducible randomness seeps in through the stochasticity of these interactions, the approximations in the agent's representation, and even the inherently chaotic nature of physical interaction (Yu et al., 2016). Distributional RL aims to model the distribution over returns, whose mean is the traditional value function, and to use these distributions to evaluate and optimize a policy.

Any distributional RL algorithm is characterized by two aspects: the parameterization of the return distribution, and the distance metric or loss function being optimized. Together, these choices control assumptions about the random returns and how approximations will be traded off. Categorical DQN (Bellemare et al., 2017, C51) combines a categorical distribution and the cross-entropy loss with the Cramér-minimizing projection (Rowland et al., 2018). For this, it assumes returns are bounded in a known range and trades off mean-preservation at the cost of overestimating variance.

C51 outperformed all previous improvements to DQN on a set of 57 Atari 2600 games in the Arcade Learning Environment (Bellemare et al., 2013), which we refer to as the Atari-57 benchmark. Subsequently, several papers have built upon this successful combination to achieve significant improvements to the state-of-the-art in Atari-57 (Hessel et al., 2018; Gruslys et al., 2018), and challenging continuous control tasks (Barth-Maron et al., 2018).

These algorithms are restricted to assigning probabilities to an a priori fixed, discrete set of possible returns. Dabney et al. (2018) propose an alternate pair of choices, parameterizing the distribution by a uniform mixture of Diracs whose locations are adjusted using quantile regression. Their algorithm, QR-DQN, while restricted to a discrete set of quantiles, automatically adapts return quantiles to minimize the Wasserstein distance between the Bellman updated and current return distributions. This flexibility allows QR-DQN to significantly improve on C51's Atari-57 performance.

In this paper, we extend the approach of Dabney et al. (2018), from learning a discrete set of quantiles to learning the full quantile function, a continuous map from probabilities to returns. When combined with a base distribution, such as $U([0, 1])$, this forms an implicit distribution capable of approximating any distribution over returns given sufficient network capacity. Our approach, *implicit quantile networks* (IQN), is best viewed as a simple distributional generalization of the DQN algorithm (Mnih et al., 2015), and provides several benefits over QR-DQN.

First, the approximation error for the distribution is no longer controlled by the number of quantiles output by the network, but by the size of the network itself, and the amount of training. Second, IQN can be used with as few, or as many, samples per update as desired, providing improved data efficiency with increasing number of samples per training update. Third, the implicit representation of the return distribution allows us to expand the class of policies to more fully take advantage of the learned distribution. Specifically, by taking the base distribution to be non-uniform, we expand the class of policies to ϵ-greedy policies on arbitrary distortion risk measures (Yaari, 1987; Wang, 1996).

[*] Equal contribution [1] DeepMind, London, UK. Correspondence to: Will Dabney <wdabney@google.com>, Georg Ostrovski <ostrovski@google.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

We begin by reviewing distributional reinforcement learning, related work, and introducing the concepts surrounding risk-sensitive RL. In subsequent sections, we introduce our proposed algorithm, IQN, and present a series of experiments using the Atari-57 benchmark, investigating the robustness and performance of IQN. Despite being a simple distributional extension to DQN, and forgoing any other improvements, IQN significantly outperforms QR-DQN and nearly matches the performance of Rainbow, which combines many orthogonal advances. In fact, in human-starts as well as in the hardest Atari games (where current RL agents still underperform human players) IQN improves over Rainbow.

2. Background / Related Work

We consider the standard RL setting, in which the interaction of an agent and an environment is modeled as a Markov Decision Process $(\mathcal{X}, \mathcal{A}, R, P, \gamma)$ (Puterman, 1994), where $\mathcal{X}$ and $\mathcal{A}$ denote the state and action spaces, R the (state- and action-dependent) reward function, $P(\cdot|x, a)$ the transition kernel, and $\gamma \in (0, 1)$ a discount factor. A policy $\pi(\cdot|x)$ maps a state to a distribution over actions.

For an agent following policy π, the discounted sum of future rewards is denoted by the random variable $Z^\pi(x, a) = \sum_{t=0}^{\infty} \gamma^t R(x_t, a_t)$, where $x_0 = x$, $a_0 = a$, $x_t \sim P(\cdot|x_{t-1}, a_{t-1})$, and $a_t \sim \pi(\cdot|x_t)$. The action-value function is defined as $Q^\pi(x, a) = \mathbb{E}\left[Z^\pi(x, a)\right]$, and can be characterized by the Bellman equation

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

The objective in RL is to find an optimal policy π^*, which maximizes $\mathbb{E}[Z^\pi]$, i.e. $Q^{\pi^*}(x, a) \geq Q^\pi(x, a)$ for all π and all x, a. One approach is to find the unique fixed point $Q^* = Q^{\pi^*}$ of the Bellman optimality operator (Bellman, 1957):

$$Q(x, a) = \mathcal{T}Q(x, a) := \mathbb{E}\left[R(x, a)\right] + \gamma \mathbb{E}_P \max_{a'} Q(x', a').$$

To this end, Q-learning (Watkins, 1989) iteratively improves an estimate, Q_θ, of the optimal action-value function, Q^*, by repeatedly applying the Bellman update:

$$Q_\theta(x, a) \leftarrow \mathbb{E}\left[R(x, a)\right] + \gamma \mathbb{E}_P\left[\max_{a'} Q_\theta(x', a')\right].$$

The action-value function can be approximated by a parameterized function Q_θ (e.g. a neural network), and trained by minimizing the squared temporal difference (TD) error,

$$\delta_t^2 = \left[r_t + \gamma \max_{a' \in \mathcal{A}} Q_\theta(x_{t+1}, a') - Q_\theta(x_t, a_t)\right]^2,$$

over samples (x_t, a_t, r_t, x_{t+1}) observed while following an ϵ-greedy policy over Q_θ. This policy acts greedily with respect to Q_θ with probability $1 - \epsilon$ and uniformly at random otherwise. DQN (Mnih et al., 2015) uses a convolutional neural network to parameterize Q_θ and the Q-learning algorithm to achieve human-level play on the Atari-57 benchmark.

2.1. Distributional RL

In distributional RL, the distribution over returns (the law of Z^π) is considered instead of the scalar value function Q^π that is its expectation. This change in perspective has yielded new insights into the dynamics of RL (Azar et al., 2012), and been a useful tool for analysis (Lattimore & Hutter, 2012). Empirically, distributional RL algorithms show improved sample complexity and final performance, as well as increased robustness to hyperparameter variation (Barth-Maron et al., 2018).

An analogous distributional Bellman equation of the form

$$Z^\pi(x, a) \overset{D}{=} R(x, a) + \gamma Z^\pi(X', A')$$

can be derived, where $A \overset{D}{=} B$ denotes that two random variables A and B have equal probability laws, and the random variables X' and A' are distributed according to $P(\cdot|x, a)$ and $\pi(\cdot|x')$, respectively.

Morimura et al. (2010a) defined the distributional Bellman operator explicitly in terms of conditional probabilities, parameterized by the mean and scale of a Gaussian or Laplace distribution, and minimized the Kullback-Leibler (KL) divergence between the Bellman target and the current estimated return distribution. However, the distributional Bellman operator is not a contraction in the KL.

As with the scalar setting, a distributional Bellman optimality operator can be defined by

$$\mathcal{T}Z(x, a) \overset{D}{:=} R(x, a) + \gamma Z(X', \arg\max_{a' \in \mathcal{A}} \mathbb{E}\, Z(X', a')),$$

with X' distributed according to $P(\cdot|x, a)$. While the distributional Bellman operator for policy evaluation is a contraction in the p-Wasserstein distance (Bellemare et al., 2017), this no longer holds for the control case. Convergence to the optimal policy can still be established, but requires a more involved argument.

Bellemare et al. (2017) parameterize the return distribution as a categorical distribution over a fixed set of equidistant points and minimize the KL divergence to the projected distributional Bellman target. Their algorithm, C51, outperformed previous DQN variants on the Atari-57 benchmark. Subsequently, Hessel et al. (2018) combined C51 with enhancements such as prioritized experience replay (Schaul et al., 2016), n-step updates (Sutton, 1988), and the dueling architecture (Wang et al., 2016), leading to the Rainbow agent, current state-of-the-art in Atari-57.

The categorical parameterization, using the projected KL loss, has also been used in recent work to improve the critic of a policy gradient algorithm, D4PG, achieving significantly improved robustness and state-of-the-art performance across a variety of continuous control tasks (Barth-Maron et al., 2018).

2.2. p-Wasserstein Metric

The p-Wasserstein metric, for $p \in [1, \infty]$, plays a key role in recent results in distributional RL (Bellemare et al., 2017; Dabney et al., 2018). It has also been a topic of increasing interest in generative modeling (Arjovsky et al., 2017; Bousquet et al., 2017; Tolstikhin et al., 2017), because unlike the KL divergence, the Wasserstein metric inherently trades off approximate solutions with likelihoods.

The p-Wasserstein distance is the L_p metric on inverse cumulative distribution functions (c.d.f.), also known as quantile functions (Müller, 1997). For random variables U and V with quantile functions F_U^{-1} and F_V^{-1}, respectively, the p-Wasserstein distance is given by

$$W_p(U, V) = \left(\int_0^1 |F_U^{-1}(\omega) - F_V^{-1}(\omega)|^p d\omega \right)^{1/p}.$$

The class of optimal transport metrics express distances between distributions in terms of the minimal cost for transporting mass to make the two distributions identical. This cost is given in terms of some metric, $c\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{\geq 0}$, on the underlying space $\mathcal{X}$. The p-Wasserstein metric corresponds to $c = L_p$. We are particularly interested in the Wasserstein metrics due to the predominant use of L_p spaces in mean-value reinforcement learning.

2.3. Quantile Regression for Distributional RL

Bellemare et al. (2017) showed that the distributional Bellman operator is a contraction in the p-Wasserstein metric, but as the proposed algorithm did not itself minimize the Wasserstein metric, this left a theory-practice gap for distributional RL. Recently, this gap was closed, in both directions. First and most relevant to this work, Dabney et al. (2018) proposed the use of *quantile regression* for distributional RL and showed that by choosing the quantile targets suitably the resulting projected distributional Bellman operator is a contraction in the ∞-Wasserstein metric. Concurrently, Rowland et al. (2018) showed the original class of categorical algorithms are a contraction in the Cramér distance, the L_2 metric on cumulative distribution functions.

By estimating the quantile function at precisely chosen points, QR-DQN minimizes the Wasserstein distance to the distributional Bellman target (Dabney et al., 2018). This estimation uses *quantile regression*, which has been shown to converge to the true quantile function value when minimized using stochastic approximation (Koenker, 2005).

In QR-DQN, the random return is approximated by a uniform mixture of N Diracs,

$$Z_\theta(x, a) := \tfrac{1}{N} \sum_{i=1}^{N} \delta_{\theta_i(x,a)},$$

with each θ_i assigned a fixed quantile target, $\hat{\tau}_i = \frac{\tau_{i-1} + \tau_i}{2}$ for $1 \leq i \leq N$, where $\tau_i = i/N$. These quantile estimates are trained using the Huber (1964) quantile regression loss, with threshold κ,

$$\rho_\tau^\kappa(\delta_{ij}) = |\tau - \mathbb{I}\{\delta_{ij} < 0\}| \frac{\mathcal{L}_\kappa(\delta_{ij})}{\kappa}, \quad \text{with}$$

$$\mathcal{L}_\kappa(\delta_{ij}) = \begin{cases} \frac{1}{2}\delta_{ij}^2, & \text{if } |\delta_{ij}| \leq \kappa \\ \kappa(|\delta_{ij}| - \frac{1}{2}\kappa), & \text{otherwise} \end{cases},$$

on the pairwise TD-errors

$$\delta_{ij} = r + \gamma\theta_j(x', \pi(x')) - \theta_i(x, a).$$

At the time of this writing, QR-DQN achieves the best performance on Atari-57, human-normalized mean and median, of all agents that do not combine distributional RL, prioritized replay, and n-step updates (Dabney et al., 2018; Hessel et al., 2018; Gruslys et al., 2018).

2.4. Risk in Reinforcement Learning

Distributional RL algorithms have been theoretically justified for the Wasserstein and Cramér metrics (Bellemare et al., 2017; Rowland et al., 2018), and learning the distribution over returns, in and of itself, empirically results in significant improvements to data efficiency, final performance, and stability (Bellemare et al., 2017; Dabney et al., 2018; Gruslys et al., 2018; Barth-Maron et al., 2018). However, in each of these recent works the policy used was based entirely on the mean of the return distribution, just as in standard reinforcement learning. A natural question arises: can we expand the class of policies using information provided by the distribution over returns (i.e. to the class of risk-sensitive policies)? Furthermore, when would this larger policy class be beneficial?

Here, 'risk' refers to the uncertainty over possible outcomes, and *risk-sensitive* policies are those which depend upon more than the mean of the outcomes. At this point, it is important to highlight the difference between *intrinsic uncertainty*, captured by the distribution over returns, and *parametric uncertainty*, the uncertainty over the value estimate typically associated with Bayesian approaches such as PSRL (Osband et al., 2013) and Kalman TD (Geist & Pietquin, 2010). Distributional RL seeks to capture the former, which classic approaches to risk are built upon[1].

Expected utility theory states that if a decision policy is consistent with a particular set of four axioms regarding its choices then the decision policy behaves as though it is maximizing the expected value of some utility function U (von Neumann & Morgenstern, 1947),

$$\pi(x) = \arg\max_a \mathop{\mathbb{E}}_{Z(x,a)}[U(z)].$$

This is perhaps the most pervasive notion of risk-sensitivity. A policy maximizing a linear utility function is called *risk-neutral*, whereas concave or convex utility functions give rise to *risk-averse* or *risk-seeking* policies, respectively. Many previous studies on risk-sensitive RL adopt the utility function approach (Howard & Matheson, 1972; Marcus et al., 1997; Maddison et al., 2017).

A crucial axiom of expected utility is *independence*: given random variables X, Y and Z, such that $X \succ Y$ (X preferred over Y), any mixture between X and Z is preferred to the same mixture between Y and Z (von Neumann & Morgenstern, 1947). Stated in terms of the cumulative probability functions, $\alpha F_X + (1-\alpha)F_Z \geq \alpha F_Y + (1-\alpha)F_Z$, $\forall \alpha \in [0,1]$. This axiom in particular has troubled many researchers because it is consistently violated by human behavior (Tversky & Kahneman, 1992). The Allais paradox is a frequently used example of a decision problem where people violate the independence axiom of expected utility theory (Allais, 1990).

However, as Yaari (1987) showed, this axiom can be replaced by one in terms of convex combinations of outcome values, instead of mixtures of distributions. Specifically, if as before $X \succ Y$, then for any $\alpha \in [0,1]$ and random variable Z, $\alpha F_X^{-1} + (1-\alpha)F_Z^{-1} \geq \alpha F_Y^{-1} + (1-\alpha)F_Z^{-1}$. This leads to an alternate, dual, theory of choice than that of expected utility. Under these axioms the decision policy behaves as though it is maximizing a distorted expectation, for some continuous monotonic function h:

$$\pi(x) = \arg\max_a \int_{-\infty}^{\infty} z \frac{\partial}{\partial z}(h \circ F_{Z(x,a)})(z)\, dz.$$

Such a function h is known as a *distortion risk measure*, as it distorts the cumulative probabilities of the random variable (Wang, 1996). That is, we have two fundamentally equivalent approaches to risk-sensitivity. Either, we choose a utility function and follow the expectation of this utility. Or, we choose a reweighting of the distribution and compute expectation under this distortion measure. Indeed, Yaari (1987) further showed that these two functions are inverses of each other. The choice between them amounts to a choice over whether the behavior should be invariant to mixing with random events or to convex combinations of outcomes.

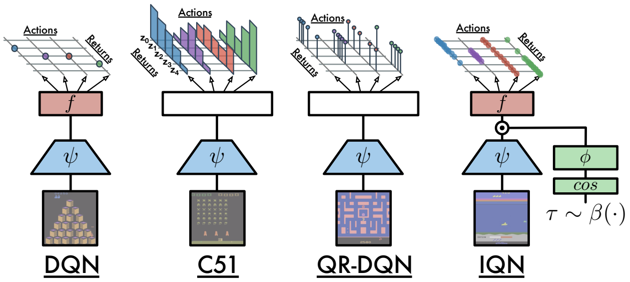

Figure 1. Network architectures for DQN and recent distributional RL algorithms.

Distortion risk measures include, as special cases, cumulative probability weighting used in cumulative prospect theory (Tversky & Kahneman, 1992), conditional value at risk (Chow & Ghavamzadeh, 2014), and many other methods (Morimura et al., 2010b). Recently Majumdar & Pavone (2017) argued for the use of distortion risk measures in robotics.

3. Implicit Quantile Networks

We now introduce the *implicit quantile network* (IQN), a deterministic parametric function trained to reparameterize samples from a base distribution, e.g. $\tau \sim U([0,1])$, to the respective quantile values of a target distribution. IQN provides an effective way to learn an implicit representation of the return distribution, yielding a powerful function approximator for a new DQN-like agent.

Let $F_Z^{-1}(\tau)$ be the quantile function at $\tau \in [0,1]$ for the random variable Z. For notational simplicity we write $Z_\tau := F_Z^{-1}(\tau)$, thus for $\tau \sim U([0,1])$ the resulting state-action return distribution sample is $Z_\tau(x,a) \sim Z(x,a)$.

We propose to model the state-action quantile function as a mapping from state-actions and samples from some base distribution, typically $\tau \sim U([0,1])$, to $Z_\tau(x,a)$, viewed as samples from the implicitly defined return distribution.

Let $\beta\colon [0,1] \to [0,1]$ be a distortion risk measure, with identity corresponding to risk-neutrality. Then, the *distorted expectation* of $Z(x,a)$ under β is given by

$$Q_\beta(x,a) := \mathop{\mathbb{E}}_{\tau \sim U([0,1])} \left[Z_{\beta(\tau)}(x,a) \right].$$

Notice that the distorted expectation is equal to the expected value of $F_{Z(x,a)}^{-1}$ weighted by β, that is, $Q_\beta = \int_0^1 F_Z^{-1}(\tau) d\beta(\tau)$. The immediate implication of this is that for any β, there exists a sampling distribution for τ such that the mean of Z_τ is equal to the distorted expectation of Z

[1] One exception is the recent work (Moerland et al., 2017) towards combining both forms of uncertainty to improve exploration.

under β, that is, any distorted expectation can be represented as a weighted sum over the quantiles (Dhaene et al., 2012). Denote by π_β the risk-sensitive greedy policy

$$\pi_\beta(x) = \arg\max_{a \in \mathcal{A}} Q_\beta(x, a). \tag{1}$$

For two samples $\tau, \tau' \sim U([0, 1])$, and policy π_β, the sampled temporal difference (TD) error at step t is

$$\delta_t^{\tau,\tau'} = r_t + \gamma Z_{\tau'}(x_{t+1}, \pi_\beta(x_{t+1})) - Z_\tau(x_t, a_t). \tag{2}$$

Then, the IQN loss function is given by

$$\mathcal{L}(x_t, a_t, r_t, x_{t+1}) = \frac{1}{N'} \sum_{i=1}^{N} \sum_{j=1}^{N'} \rho_{\tau_i}^{\kappa} \left(\delta_t^{\tau_i,\tau'_j} \right), \tag{3}$$

where N and N' denote the respective number of iid samples $\tau_i, \tau'_j \sim U([0, 1])$ used to estimate the loss. A corresponding sample-based risk-sensitive policy is obtained by approximating Q_β in Equation 1 by K samples of $\tilde{\tau} \sim U([0, 1])$:

$$\tilde{\pi}_\beta(x) = \arg\max_{a \in \mathcal{A}} \frac{1}{K} \sum_{k=1}^{K} Z_{\beta(\tilde{\tau}_k)}(x, a).$$

Implicit quantile networks differ from the approach of Dabney et al. (2018) in two ways. First, instead of approximating the quantile function at n fixed values of τ we approximate it with $Z_\tau(x, a) \approx f(\psi(x), \phi(\tau))_a$ for some differentiable functions f, ψ, and ϕ. If we ignore the distributional interpretation for a moment and view each $Z_\tau(x, a)$ as a separate action-value function, this highlights that implicit quantile networks are a type of *universal value function approximator* (UVFA) (Schaul et al., 2015). There may be additional benefits to implicit quantile networks beyond the obvious increase in representational fidelity. As with UVFAs, we might hope that training over many different τ's (goals in the case of the UVFA) leads to better generalization between values and improved sample complexity than attempting to train each separately.

Second, τ, τ', and $\tilde{\tau}$ are sampled from continuous, independent, distributions. Besides $U([0, 1])$, we also explore risk-sentive policies π_β, with non-linear β. The independent sampling of each τ, τ' results in the sample TD errors being decorrelated, and the estimated action-values go from being the true mean of a mixture of n Diracs to a sample mean of the implicit distribution defined by reparameterizing the sampling distribution via the learned quantile function.

3.1. Implementation

Consider the neural network structure used by the DQN agent (Mnih et al., 2015). Let $\psi\colon \mathcal{X} \to \mathbb{R}^d$ be the function computed by the convolutional layers and $f\colon \mathbb{R}^d \to \mathbb{R}^{|\mathcal{A}|}$ the subsequent fully-connected layers mapping $\psi(x)$ to the estimated action-values, such that $Q(x, a) \approx f(\psi(x))_a$. For our network we use the same functions ψ and f as in DQN, but include an additional function $\phi\colon [0, 1] \to \mathbb{R}^d$ computing an embedding for the sample point τ. We combine these to form the approximation $Z_\tau(x, a) \approx f(\psi(x) \odot \phi(\tau))_a$, where $\odot$ denotes the element-wise (Hadamard) product.

As the network for f is not particularly deep, we use the multiplicative form, $\psi \odot \phi$, to force interaction between the convolutional features and the sample embedding. Alternative functional forms, e.g. concatenation or a 'residual' function $\psi \odot (1 + \phi)$, are conceivable, and $\phi(\tau)$ can be parameterized in different ways. To investigate these, we compared performance across a number of architectural variants on six Atari 2600 games (ASTERIX, ASSAULT, BREAKOUT, MS.PACMAN, QBERT, SPACE INVADERS). Full results are given in the Appendix. Despite minor variation in performance, we found the general approach to be robust to the various choices. Based upon the results we used the following function in our later experiments, for embedding dimension $n = 64$:

$$\phi_j(\tau) := \mathrm{ReLU}(\sum_{i=0}^{n-1} \cos(\pi i \tau) w_{ij} + b_j). \tag{4}$$

After settling on a network architecture, we study the effect of the number of samples, N and N', used in the estimate terms of Equation 3.

We hypothesized that N, the number of samples of $\tau \sim U([0, 1])$, would affect the sample complexity of IQN, with larger values leading to faster learning, and that with $N = 1$ one would potentially approach the performance of DQN. This would support the hypothesis that the improved performance of many distributional RL algorithms rests on their effect as auxiliary loss functions, which would vanish in the case of $N = 1$. Furthermore, we believed that N', the number of samples of $\tau' \sim U([0, 1])$, would affect the variance of the gradient estimates much like a mini-batch size hyperparameter. Our prediction was that N' would have the greatest effect on variance of the long-term performance of the agent.

We used the same set of six games as before, with our chosen architecture, and varied $N, N' \in \{1, 8, 32, 64\}$. In Figure 2 we report the average human-normalized scores on the six games for each configuration. Figure 2 (left) shows the average performance over the first ten million frames, while (right) shows the average performance over the last ten million (from 190M to 200M).

As expected, we found that N has a dramatic effect on early performance, shown by the continual improvement in score as the value increases. Additionally, we observed that N'

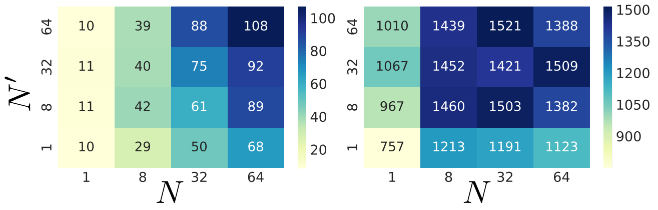

Figure 2. Effect of varying N and N', the number of samples used in the loss function in Equation 3. Figures show human-normalized agent performance, averaged over six Atari games, averaged over first 10M frames of training (left) and last 10M frames of training (right). Corresponding values for baselines: DQN $(32, 253)$ and QR-DQN $(144, 1243)$.

affected performance very differently than expected: it had a strong effect on early performance, but minimal impact on long-term performance past $N' = 8$.

Overall, while using more samples for both distributions is generally favorable, $N = N' = 8$ appears to be sufficient to achieve the majority of improvements offered by IQN for long-term performance, with variation past this point largely insignificant. To our surprise we found that even for $N = N' = 1$, which is comparable to DQN in the number of loss components, the longer term performance is still quite strong ($\approx 3\times$ DQN).

In an informal evaluation, we did not find IQN to be sensitive to K, the number of samples used for the policy, and have fixed it at $K = 32$ for all experiments.

4. Risk-Sensitive Reinforcement Learning

In this section, we explore the effects of varying the distortion risk measure, β, away from identity. This only affects the policy, π_β, used both in Equation 2 and for acting in the environment. As we have argued, evaluating under different distortion risk measures is equivalent to changing the sampling distribution for τ, allowing us to achieve various forms of risk-sensitive policies. We focus on a handful of sampling distributions and their corresponding distortion measures. The first one is the cumulative probability weighting parameterization proposed in cumulative prospect theory (Tversky & Kahneman, 1992; Gonzalez & Wu, 1999):

$$\mathrm{CPW}(\eta, \tau) = \frac{\tau^\eta}{(\tau^\eta + (1-\tau)^\eta)^{\frac{1}{\eta}}}.$$

In particular, we use the parameter value $\eta = 0.71$ found by Wu & Gonzalez (1996) to most closely match human subjects. This choice is interesting as, unlike the others we consider, it is neither globally convex nor concave. For small values of τ it is locally concave and for larger values of τ it becomes locally convex. Recall that concavity corresponds to risk-averse and convexity to risk-seeking policies.

Second, we consider the distortion risk measure proposed by Wang (2000), where Φ and Φ^{-1} are taken to be the standard Normal cumulative distribution function and its inverse:

$$\mathrm{Wang}(\eta, \tau) = \Phi(\Phi^{-1}(\tau) + \eta).$$

For $\eta < 0$, this produces risk-averse policies and we include it due to its simple interpretation and ability to switch between risk-averse and risk-seeking distortions.

Third, we consider a simple power formula for risk-averse ($\eta < 0$) or risk-seeking ($\eta > 0$) policies:

$$\mathrm{Pow}(\eta, \tau) = \begin{cases} \tau^{\frac{1}{1+|\eta|}}, & \text{if } \eta \geq 0 \\ 1 - (1-\tau)^{\frac{1}{1+|\eta|}}, & \text{otherwise} \end{cases}.$$

Finally, we consider conditional value-at-risk (CVaR):

$$\mathrm{CVaR}(\eta, \tau) = \eta\tau.$$

CVaR has been widely studied in and out of reinforcement learning (Chow & Ghavamzadeh, 2014). Its implementation as a modification to the sampling distribution of τ is particularly simple, as it changes $\tau \sim U([0,1])$ to $\tau \sim U([0,\eta])$. Another interesting sampling distribution, not included in our experiments, is denoted $\mathrm{Norm}(\eta)$ and corresponds to τ sampled by averaging η samples from $U([0,1])$.

In Figure 3 (right) we give an example of a distribution (Neutral) and how each of these distortion measures affects the implied distribution due to changing the sampling distribution of τ. $\mathrm{Norm}(3)$ and $\mathrm{CPW}(.71)$ reduce the impact of the tails of the distribution, while Wang and CVaR heavily shift the distribution mass towards the tails, creating a risk-averse or risk-seeking preference. Additionally, while CVaR entirely ignores all values corresponding to $\tau > \eta$, Wang gives these non-zero, but vanishingly small, probability.

By using these sampling distributions we can induce various risk-sensitive policies in IQN. We evaluate these on the same set of six Atari 2600 games previously used. Our algorithm simply changes the policy to maximize the distorted expectations instead of the usual sample mean. Figure 3 (left) shows our results in this experiment, with average scores reported under the usual, risk-neutral, evaluation criterion.

Intuitively, we expected to see a qualitative effect from risk-sensitive training, e.g. strengthened exploration from a risk-seeking objective. Although we did see qualitative differences, these did not always match our expectations. For two of the games, ASTERIX and ASSAULT, there is a very significant advantage to the risk-averse policies. Although CPW tends to perform almost identically to the standard risk-neutral policy, and the risk-seeking $\mathrm{Wang}(1.5)$ performs as well or worse than risk-neutral, we find that both risk-averse policies improve performance over standard IQN. However, we also observe that the more risk-averse of the

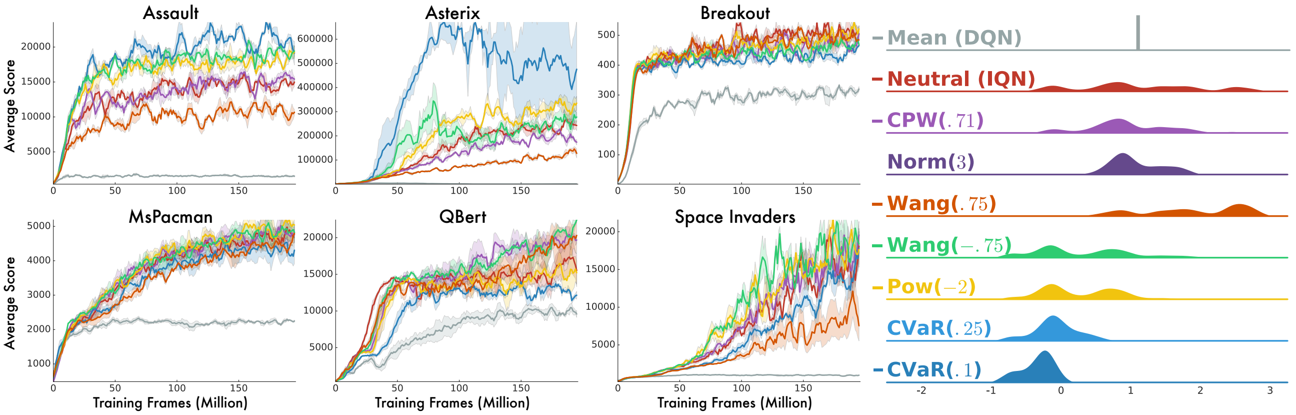

Figure 3. Effects of various changes to the sampling distribution, that is various cumulative probability weightings.

two, $\mathrm{CVaR}(0.1)$, suffers some loss in performance on two other games (QBERT and SPACE INVADERS).

Additionally, we note that the risk-seeking policy significantly underperforms the risk-neutral policy on three of the six games. It remains an open question as to exactly why we see improved performance for risk-averse policies. There are many possible explanations for this phenomenon, e.g. that risk-aversion encodes a heuristic to stay alive longer, which in many games is correlated with increased rewards.

5. Full Atari-57 Results

Finally, we evaluate IQN on the full Atari-57 benchmark, comparing with the state-of-the-art performance of Rainbow, a distributional RL agent that combines several advances in deep RL (Hessel et al., 2018), the closely related algorithm QR-DQN (Dabney et al., 2018), prioritized experience replay DQN (Schaul et al., 2016), and the original DQN agent (Mnih et al., 2015). Note that in this section we use the risk-neutral variant of the IQN, that is, the policy of the IQN agent is the regular ϵ-greedy policy with respect to the mean of the state-action return distribution.

It is important to remember that Rainbow builds upon the distributional RL algorithm C51 (Bellemare et al., 2017), but also includes prioritized experience replay (Schaul et al., 2016), Double DQN (van Hasselt et al., 2016), Dueling Network architecture (Wang et al., 2016), Noisy Networks (Fortunato et al., 2017), and multi-step updates (Sutton, 1988). In particular, besides the distributional update, n-step updates and prioritized experience replay were found to have significant impact on the performance of Rainbow. Our other competitive baseline is QR-DQN, which is currently state-of-the-art for agents that do not combine distributional updates, n-step updates, and prioritized replay.

Thus, between QR-DQN and the much more complex Rainbow we compare to the two most closely related, and best performing, agents in published work. In particular, we would expect that IQN would benefit from the additional enhancements in Rainbow, just as Rainbow improved significantly over C51.

Figure 4 shows the mean (left) and median (right) human-normalized scores during training over the Atari-57 benchmark. IQN dramatically improves over QR-DQN, which itself improves on many previously published results. At 100 million frames IQN has reached the same level of performance as QR-DQN at 200 million frames. Table 1 gives a comparison between the same methods in terms of their best, human-normalized, scores per game under the 30 random no-op start condition. These are averages over the given number of seeds. Additionally, using human-starts, IQN achieves 162% median human-normalized score, whereas Rainbow reaches 153% (Hessel et al., 2018), see Table 2.

	Mean	**Median**	**Human Gap**	**Seeds**
DQN	228%	79%	0.334	1
PRIOR.	434%	124%	0.178	1
C51	701%	178%	0.152	1
RAINBOW	**1189%**	**230%**	0.144	2
QR-DQN	864%	193%	0.165	3
IQN	1019%	218%	**0.141**	5

Table 1. Mean and median of scores across 57 Atari 2600 games, measured as percentages of human baseline (Nair et al., 2015). Scores are averages over number of seeds.

Human-starts (median)

DQN	PRIOR.	A3C	C51	RAINBOW	IQN
68%	128%	116%	125%	153%	**162%**

Table 2. Median human-normalized scores for human-starts.

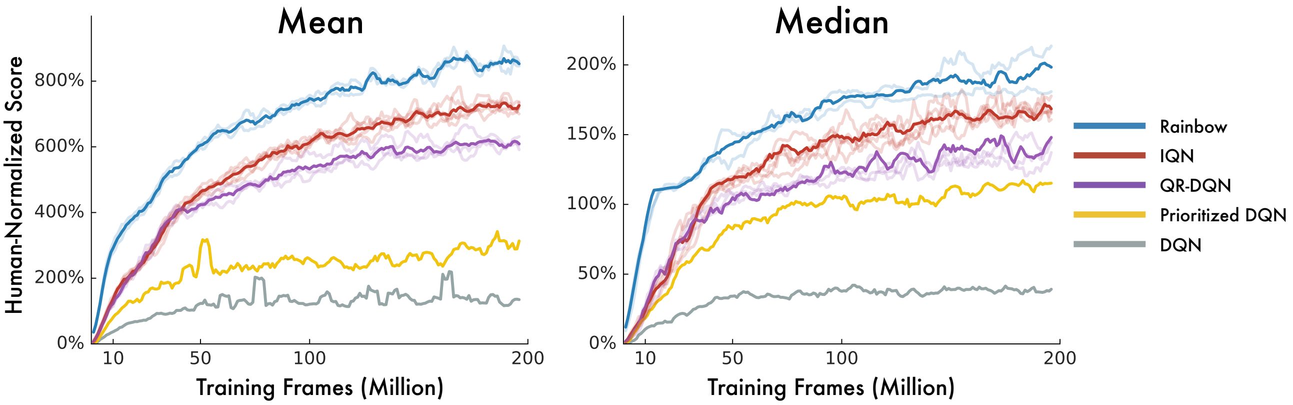

Figure 4. Human-normalized mean (left) and median (right) scores on Atari-57 for IQN and various other algorithms. Random seeds shown as traces, with IQN averaged over 5, QR-DQN over 3, and Rainbow over 2 random seeds.

Finally, we took a closer look at the games in which each algorithm continues to underperform humans, and computed, on average, how far below human-level they perform[2]. We refer to this value as the *human-gap*[3] metric and give results in Table 1. Interestingly, C51 outperforms QR-DQN in this metric, and IQN outperforms all others. This shows that the remaining gap between Rainbow and IQN is entirely from games on which both algorithms are already super-human. The games where the most progress in RL is needed happen to be the games where IQN shows the greatest improvement over QR-DQN and Rainbow.

6. Discussion and Conclusions

We have proposed a generalization of recent work based around using quantile regression to learn the distribution over returns of the current policy. Our generalization leads to a simple change to the DQN agent to enable distributional RL, the natural integration of risk-sensitive policies, and significantly improved performance over existing methods. The IQN algorithm provides, for the first time, a fully integrated distributional RL agent without prior assumptions on the parameterization of the return distribution.

IQN can be trained with as little as a single sample from each state-action value distribution, or as many as computational limits allow to improve the algorithm's data efficiency. Furthermore, IQN allows us to expand the class of control policies to a large class of risk-sensitive policies connected to distortion risk measures. Finally, we show substantial gains on the Atari-57 benchmark over QR-DQN, and even halving the distance between QR-DQN and Rainbow.

Despite the significant empirical successes in this paper there are many areas in need of additional theoretical analysis. We highlight a few particularly relevant open questions we were unable to address in the present work. First, sample-based convergence results have been recently shown for a class of categorical distributional RL algorithms (Rowland et al., 2018). Could existing sample-based RL convergence results be extended to the QR-based algorithms?

Second, can the contraction mapping results for a fixed grid of quantiles given by Dabney et al. (2018) be extended to the more general class of approximate quantile functions studied in this work? Finally, and particularly salient to our experiments with distortion risk measures, theoretical guarantees for risk-sensitive RL have been building over recent years, but have been largely limited to special cases and restricted classes of risk-sensitive policies. Can the convergence of the distribution of returns under the Bellman operator be leveraged to show convergence to a fixed-point in distorted expectations? In particular, can the control results of Bellemare et al. (2017) be expanded to cover some class of risk-sensitive policies?

There remain many intriguing directions for future research into distributional RL, even on purely empirical fronts. Hessel et al. (2018) recently showed that distributional RL agents can be significantly improved, when combined with other techniques. Creating a Rainbow-IQN agent could yield even greater improvements on Atari-57. We also recall the surprisingly rich return distributions found by Barth-Maron et al. (2018), and hypothesize that the continuous control setting may be a particularly fruitful area for the application of distributional RL in general, and IQN in particular.

[2] Details of how this is computed can be found in the Appendix.

[3] Thanks to Joseph Modayil for proposing this metric.

References

Allais, M. Allais paradox. In *Utility and Probability*, pp. 3–9. Springer, 1990.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Azar, M. G., Munos, R., and Kappen, H. J. On the sample complexity of reinforcement learning with a generative model. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.

Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., TB, D., Muldal, A., Heess, N., and Lillicrap, T. Distributional policy gradients. In *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 34th International Conference on Machine Learning (ICML)*, 2017.

Bellman, R. E. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

Bousquet, O., Gelly, S., Tolstikhin, I., Simon-Gabriel, C.-J., and Schoelkopf, B. From optimal transport to generative modeling: the vegan cookbook. *arXiv preprint arXiv:1705.07642*, 2017.

Chow, Y. and Ghavamzadeh, M. Algorithms for CVaR optimization in MDPs. In *Advances in Neural Information Processing Systems*, pp. 3509–3517, 2014.

Dabney, W., Rowland, M., Bellemare, M. G., and Munos, R. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Dhaene, J., Kukush, A., Linders, D., and Tang, Q. Remarks on quantiles and distortion risk measures. *European Actuarial Journal*, 2(2):319–328, 2012.

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., et al. Noisy networks for exploration. *arXiv preprint arXiv:1706.10295*, 2017.

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

Gonzalez, R. and Wu, G. On the shape of the probability weighting function. *Cognitive Psychology*, 38(1):129–166, 1999.

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

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Howard, R. A. and Matheson, J. E. Risk-sensitive markov decision processes. *Management Science*, 18(7):356–369, 1972.

Huber, P. J. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

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

Koenker, R. *Quantile Regression*. Cambridge University Press, 2005.

Lattimore, T. and Hutter, M. PAC bounds for discounted MDPs. In *International Conference on Algorithmic Learning Theory*, pp. 320–334. Springer, 2012.

Maddison, C. J., Lawson, D., Tucker, G., Heess, N., Doucet, A., Mnih, A., and Teh, Y. W. Particle value functions. *arXiv preprint arXiv:1703.05820*, 2017.

Majumdar, A. and Pavone, M. How should a robot assess risk? Towards an axiomatic theory of risk in robotics. *arXiv preprint arXiv:1710.11040*, 2017.

Marcus, S. I., Fernández-Gaucherand, E., Hernández-Hernandez, D., Coraluppi, S., and Fard, P. Risk sensitive markov decision processes. In *Systems and Control in the Twenty-First Century*, pp. 263–279. Springer, 1997.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Moerland, T. M., Broekens, J., and Jonker, C. M. Efficient exploration with double uncertain value networks. *arXiv preprint arXiv:1711.10789*, 2017.

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

Morimura, T., Sugiyama, M., Kashima, H., Hachiya, H., and Tanaka, T. Nonparametric return distribution approximation for reinforcement learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 799–806, 2010b.

Müller, A. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29 (2):429–443, 1997.

Nair, A., Srinivasan, P., Blackwell, S., Alcicek, C., Fearon, R., De Maria, A., Panneershelvam, V., Suleyman, M., Beattie, C., and Petersen, S. e. a. Massively parallel methods for deep reinforcement learning. In *ICML Workshop on Deep Learning*, 2015.

Osband, I., Russo, D., and Van Roy, B. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pp. 3003–3011, 2013.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

Rowland, M., Bellemare, M. G., Dabney, W., Munos, R., and Teh, Y. W. An analysis of categorical distributional reinforcement learning. In *AISTATS*, 2018.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. In *International Conference on Machine Learning*, pp. 1312–1320, 2015.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

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

Sutton, R. S. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

Tversky, A. and Kahneman, D. Advances in prospect theory: cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, 5(4):297–323, 1992.

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

von Neumann, J. and Morgenstern, O. *Theory of Games and Economic Behavior*. Princeton University Press, 1947.

Wang, S. Premium calculation by transforming the layer premium density. *ASTIN Bulletin: The Journal of the IAA*, 26(1):71–92, 1996.

Wang, S. S. A class of distortion operators for pricing financial and insurance risks. *Journal of Risk and Insurance*, pp. 15–36, 2000.

Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., and de Freitas, N. Dueling network architectures for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

Watkins, C. J. C. H. *Learning from delayed rewards*. PhD thesis, King's College, Cambridge, 1989.

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.

Wu, G. and Gonzalez, R. Curvature of the probability weighting function. *Management Science*, 42(12):1676–1690, 1996.

Yaari, M. E. The dual theory of choice under risk. *Econometrica: Journal of the Econometric Society*, pp. 95–115, 1987.

Yu, K.-T., Bauza, M., Fazeli, N., and Rodriguez, A. More than a million ways to be pushed. a high-fidelity experimental dataset of planar pushing. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 30–37. IEEE, 2016.