Distributional Reinforcement Learning with Dual Expectile-Quantile Regression

Sami Jullien[*1] **Romain Deffayet**[*1,2] **Paul Groth**[1] **Jean-Michel Renders**[2] **Maarten de Rijke**[1]

[1]University of Amsterdam , Amsterdam, The Netherlands
[2]Naver Labs Europe, Meylan, France

Abstract

Distributional reinforcement learning (RL) has proven useful in multiple benchmarks as it enables approximating the full distribution of returns and extracts rich feedback from environment samples. The commonly used quantile regression approach to distributional RL – based on asymmetric L_1 losses – provides a flexible and effective way of learning arbitrary return distributions. In practice, it is often improved by using a more efficient, asymmetric hybrid L_1-L_2 Huber loss for quantile regression. However, by doing so, distributional estimation guarantees vanish, and we empirically observe that the estimated distribution rapidly collapses to its mean. Indeed, asymmetric L_2 losses, corresponding to expectile regression, cannot be readily used for distributional temporal difference learning.
Motivated by the efficiency of L_2-based learning, we propose to jointly learn expectiles and quantiles of the return distribution in a way that allows efficient learning while keeping an estimate of the full distribution of returns. We prove that our proposed operator converges to the distributional Bellman operator in the limit of infinite estimated quantile and expectile fractions, and we benchmark a practical implementation on a toy example and at scale. On the Atari benchmark, our approach matches the performance of the Huber-based IQN-1 baseline after 200M training frames but avoids distributional collapse and keeps estimates of the full distribution of returns. Code: https://github.com/samijullien/ieqn

1 INTRODUCTION

Distributional reinforcement learning (RL) [Bellemare et al., 2023] aims to maintain an estimate of the full distribution of expected returns rather than only the mean. Compared to a mean-based approach, it can be used to better capture the uncertainty in the transition matrix of the environment [Bellemare et al., 2017], as well as the stochasticity of the policy being evaluated, which may enable faster and more stable training by making better use of the data samples [Mavrin et al., 2019].

Non-parametric approximations of the return distribution learned by quantile regression have proven to be very effective in several domains [Dabney et al., 2018a,b, Yang et al., 2019], when combined with deep RL agents such as deep Q-networks (DQN) [Mnih et al., 2013] or soft actor-critic (SAC) [Haarnoja et al., 2018]. They come with the major advantage of providing guarantees for the convergence of distributional policy estimation [Dabney et al., 2018b], and in certain cases, of convergence to the optimal policy [Rowland et al., 2023], all while requiring few assumptions on the shape of the return distribution and demonstrating strong empirical performance [Dabney et al., 2018a, Yang et al., 2019]. However, the best-performing quantile-based agents are often obtained by replacing the original quantile regression loss function, i.e., an asymmetric L_1 loss, by an asymmetric Huber loss, i.e., a hybrid L_1-L_2 loss. By doing so, distributional guarantees vanish, as the proofs proposed in previous work relied on the L_1-based quantile regression [Bellemare et al., 2023, Dabney et al., 2018b]. Critically, we show in Section 5.2.4 that the estimated distributions collapse to their mean in practice. In this paper, *we propose a different approach, based on both quantile and expectile regression, that matches the performance of Huber-based agents while preserving distributional estimation guarantees and avoiding distributional collapse in practice.*

We are not the first to note that asymmetric L_2 losses, i.e., that regress *expectiles* of the target distribution, tend to yield degenerate estimated distributions when training agents with

[*]Both authors contributed equally to the paper.

temporal difference learning. Rowland et al. [2019] note that expectiles of a distribution cannot be interpreted as samples from this distribution, and therefore expectiles other than the mean cannot be directly used to compute the target values in distributional temporal difference learning. Instead, they propose to generate samples from expectiles of the distribution by adding an imputation step, that requires solving a costly root-finding problem. While theoretically justified, we found this approach to be extremely slow in practice, preventing widespread use at scale. In contrast, our dual approach tackles this problem through learning, and only requires an additional two-layer neural network with the computation of a quantile loss function on top of the expectile loss function. This approach therefore adds close to no computational overheads when training Atari agents on modern GPUs.

Our contributions can be summarized as follows:

- We propose a novel dual expectile-quantile approach to distributional dynamic programming that provably converges to the true value distribution in the limit of infinite estimated quantile and expectile fractions.
- We release implicit expectile-quantile networks (IEQN),[*] a practical implementation of our dual approach based on implicit quantile networks [Dabney et al., 2018a].
- We show both on a toy example and at scale on the Atari-5 benchmark that IEQN (i) avoids distributional collapse, and (ii) matches the performance of the Huber-based IQN-1 approach.

2 BACKGROUND

2.1 DISTRIBUTIONAL REINFORCEMENT LEARNING

We consider an environment modeled by a Markov decision process (MDP) $(\mathcal{S}, \mathcal{A}, R, T, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are a state and action space, respectively, $R(s,a)$ denotes the stochastic reward obtained by taking action a in state s, $T(\cdot \mid s,a)$ is the probability distribution over possible next states after taking a in s, and γ is a discount factor. Furthermore, we write $\pi(\cdot \mid s)$ for a (potentially stochastic) policy selecting the action depending on the current state.

We consider the problem of finding a policy maximizing the average discounted return, i.e.,

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right],$$

where $a_t \sim \pi(\cdot \mid s_t)$ and $s_{t+1} \sim T(\cdot \mid s_t, a_t)$. We can define the action-value random variable for policy π as $Z^\pi : (s,a) \mapsto \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$, with $s_0 = s, a_0 = a$. We will refer to action-value variables and their estimators as Z-functions in the remainder. Note that the Q-function, as usually defined in RL [Sutton and Barto, 2018], is given by $Q^\pi(s,a) = \mathbb{E}\left[Z^\pi(s,a)\right]$. In this work, we consider approaches that evaluate policies through distributional dynamic programming, i.e., by repeatedly applying the distributional Bellman operator $\mathcal{T}^\pi$ to a candidate Z-function:

$$\mathcal{T}^\pi Z(s_t, a_t) = R(s_t, a_t) + \gamma \mathbb{E}_\pi \left[Z(s^\pi_{t+1}, a^\pi_{t+1})\right]. \quad (1)$$

This operator has been shown to be a contraction in the p-Wasserstein distance and therefore admits a unique fixed point Z^π [Bellemare et al., 2017]. A major challenge of distributional RL resides in the choice of representation for the action-value distribution, as well as the empirical implementation of the distributional Bellman operator. For simplicity, in the remainder and in line with previous work, we only consider empirical distributions [Bellemare et al., 2023, Definition 5.5] (i.e., whose representation can fit in finite memory), and refer to the empirical representation distributional Bellman operator [Bellemare et al., 2023, Algorithm 5.1] as $\mathcal{T}^\pi$.

2.2 QUANTILE AND EXPECTILE REGRESSION

Let Z be a real-valued probability distribution. The α-*quantile* q_α of Z is defined as a value splitting the probability mass of Z in two parts of weights α and $1-\alpha$, respectively:

$$P(z \leq q_\alpha) = \alpha.$$

Therefore, the *quantile function* $Q_Z : \alpha \mapsto q_\alpha$ is the inverse cumulative distribution function: $Q_Z = F_Z^{-1}$. Alternatively, quantiles are given by the minimizer of an asymmetric L_1 loss:

$$q_\alpha = \arg\min_q \mathbb{E}_{z\sim Z}\left[\left(\alpha \mathbb{1}_{z>q} + (1-\alpha)\mathbb{1}_{z\leq q}\right)|z-q|\right]. \quad (2)$$

Expectiles and the *expectile function* $E_Z : \tau \mapsto e_\tau$ are defined analogously, as the τ-expectile e_τ minimizes the asymmetric L_2 loss:

$$e_\tau = \arg\min_e \mathbb{E}_{z\sim Z}\left[\left(\tau \mathbb{1}_{z>e} + (1-\tau)\mathbb{1}_{z\leq e}\right)(z-e)^2\right]. \quad (3)$$

2.3 QUANTILES AND EXPECTILES IN DISTRIBUTIONAL RL

Quantile regression has been used for distributional RL in many previous studies [see, e.g., Dabney et al., 2018a,b, Yang et al., 2019] where a parameterized quantile function $Q^\theta_Z(s,a,\alpha)$ is trained using a quantile temporal difference loss function derived from Eq. (2), i.e., for N estimated quantiles:

$$\mathcal{L}_Q\left(Q^\theta_Z(s,a,\cdot), \mathbf{z}\right) = \sum_{i=1}^{N}\sum_{j=1}^{N} l_Q(q_i, z_j), \quad (4)$$

[*] Available at https://github.com/samijullien/ieqn.

with $l_Q(q_i, z_j) = (\alpha_i \mathbb{1}_{z_j > q_i} + (1-\alpha_i)\mathbb{1}_{z_j \leq q_i})|z_j - q_i|$, where the trainable quantile values $q_i = Q_Z^\theta(s, a, \alpha_i)$ are obtained by querying the quantile function at various quantile fractions α_i, which can be either fixed by the designer [Dabney et al., 2018b], sampled from a distribution [Dabney et al., 2018a], or learned during training [Yang et al., 2019]. In quantile-based temporal difference (QTD) learning, the target samples z_j can be obtained by querying the estimated quantile function at the next state-action pair: $z_j = r + \gamma Q_Z^\theta(s', a', \alpha_j)$.* Indeed, because the true quantile function is the inverse CDF of the action-value distribution, Dabney et al. [2018b] and Bellemare et al. [2023] showed that, among N-atoms representations, quantiles at equidistant fractions minimize the 1-Wasserstein distance with the action-value distribution and that the resulting projected Bellman operator is a contraction mapping in such a distance. Rowland et al. [2023] extended these results to prove the convergence of QTD learning under mild assumptions. We refer to these studies for a more detailed convergence analysis.

In contrast, expectile-based temporal difference (ETD) learning does not allow the same training loss as the one given by Eq. (4). We first write the generic ETD loss derived from Eq. (3):

$$\mathcal{L}_E\left(E_Z^\theta(s, a, \cdot), \mathbf{z}\right) = \sum_{i=1}^{N} \sum_{j=1}^{N} l_E(e_i, z_j), \tag{5}$$

with $l_E(e_i, z_j) = (\tau_i \mathbb{1}_{z_j > e_i} + (1-\tau_i)\mathbb{1}_{z_j \leq e_i})(z_j - e_i)^2$, and $e_i = E_Z^\theta(s, a, \tau_i)$. Here, choosing $z_j = r + \gamma E_Z^\theta(s', a', \tau_j)$, analogously to QTD learning and non-distributional TD learning, would cause the update to approximate a different distribution because the expectile function is in general not the inverse CDF of the return distribution, meaning that expectiles cannot be considered as samples from the distribution. Rowland et al. [2019] formalized this idea using the concept of *Bellman-closedness*, i.e., that the projected Bellman operator yields the same statistics whether it is applied to the target distribution or to the implicit distribution given by statistics of the target distribution (i.e., in our case a uniform mixture of diracs with locations given by quantiles or expectiles).

3 RELATED WORK

3.1 DISTRIBUTIONAL REINFORCEMENT LEARNING

Distributional reinforcement learning has been shown to result in several benefits over a mean-based approach – by ascribing randomness to the value of a state-action pair, an algorithm can learn more efficiently for close states and actions [Mavrin et al., 2019], as well as capture possible stochasticity in the environment [Martin et al., 2020]. Some works also use distributional RL for risk-sensitive control [Fei et al., 2021, Lim and Malik, 2022, Greenberg et al., 2022]. Multiple families of approaches have emerged.

Estimating a parameterized distribution is a straightforward approach, and has been explored from both Bayesian [Strens, 2000, Vlassis et al., 2012] and frequentist [Jullien et al., 2023] perspectives. However, this usually requires an expensive likelihood computation, as well as making a restrictive assumption on the shape of the return distribution Z. For instance, assuming a normal distribution when the actual distribution is heavy-tailed can yield disappointing results.

Thus, approaches based on non-parametric estimation are also used to approximate the distribution. C51 [Bellemare et al., 2017] quantizes the domain where Z has non-zero density (usually in 51 atoms, hence the name), and performs weighted classification on the atoms, by computing the cross-entropy between Z and $\mathcal{T}^\pi Z$. While C51 increases performance over non-distributional RL, it requires the user to manually set the return bounds and is not guaranteed to minimize any p-Wasserstein metric with the target return distribution.

Another important non-parametric approach to the estimation of a distribution is quantile regression. Quantile regression relies on the minimization of an asymmetric $L1$ loss. Estimating quantiles allows one to approximate the action-value distribution without relying on a shape assumption. QR-DQN [Dabney et al., 2018b] introduced quantile regression as a way to minimize the 1-Wasserstein metric between Z and $\mathcal{T}^\pi Z$. ER-DQN [Rowland et al., 2019] traded the estimation of quantiles for expectiles, at the cost of a potential distribution collapse, which they prevent via a root-finding procedure. Further, implicit quantile networks (IQN) [Dabney et al., 2018a] sample and embed quantile fractions, instead of keeping them fixed, thereby improving performance. Fully parameterized quantile functions (FQF) [Yang et al., 2019] add another network generating quantiles fractions to be estimated. We build on IQN and its expectile counterpart to propose a well-performing, non-collapsing agent.

3.2 EXPECTILE REGRESSION

Expectiles were originally introduced as a family of estimators of *location parameters* for a given distribution, to palliate possible heteroskedasticity of the error terms in regression [Newey and Powell, 1987, Philipps, 2021a].

Expectiles can be seen as mean estimators under missing data [Philipps, 2021b]. Unlike quantiles, they span the entire convex hull of the distribution's support, and on this ensemble, the expectile function is strictly increasing: an expectile fraction is always associated to a unique value. Expectiles

*We can have $a' \sim \pi(\cdot \mid s')$, as in actor-critic algorithms, or $a' = \arg\max_a Q_Z^\theta(s', a, \alpha_j)$ as in Q-learning. This section is agnostic to that choice but we refer to [Bellemare et al., 2023] for convergence analysis in the latter case.

have been used in reinforcement learning successfully before [Rowland et al., 2019], but in a way that requires a slow optimization step to achieve satisfactory performance. Moreover, expectile regression is subject to the same crossing issue as quantiles, albeit empirically less so [Waltrup et al., 2015]. Expectiles have also been used in offline reinforcement learning to compute a soft maximum over potential outcomes seen in the offline data [Kostrikov et al., 2022].

Importantly for our work, it has been shown that under mild assumptions expectile regression is the best linear unbiased estimator of any location parameter within the range of the distribution, which includes any quantile of the distribution [Philipps, 2021a]. In particular, expectile regression has lower variance than quantile regression for estimating quantiles of the distribution. This theoretical property has been confirmed empirically by Waltrup et al. [2015]. These observations encourage us to use expectile regression as a way to estimate quantiles of the value distribution, which we describe in the next section. In contrast to prior works that proposed numerical solutions to the problem of mapping an estimated expectile to its corresponding quantile [Rowland et al., 2019, Waltrup et al., 2015], we propose a learning-based approach to this problem.

4 METHOD

4.1 DUAL TRAINING OF QUANTILES AND EXPECTILES

Expectiles have been suggested to be more efficient than quantiles for function approximation [Newey and Powell, 1987, Waltrup et al., 2015], but unlike quantiles, they cannot be directly used to generate proper samples of the estimated return distribution (z_j in Eq. (5)), which are required in distributional dynamic programming. Rowland et al. [2019] propose an *imputation strategy*, i.e., a way to generate samples of a distribution that matches the current set of estimated expectiles, by solving a convex optimisation problem. In our experiments, we found that applying this imputation strategy tends to drastically increase the runtime (around 25 times slower in our setup), making experimentation with such methods close to impossible for researchers with modest computing resources. In this paper, we propose to learn a functional mapping between expectiles and quantiles and use the predicted quantiles to generate samples.

We learn a single Z-function using expectile regression. Therefore, we have $\forall(s,a) \in \mathcal{S} \times \mathcal{A}, \tau \in [0,1],\ Z_\theta(s,a,\tau) \triangleq E_{Z(s,a)}(\tau)$, where Z is the true Z-function we wish to estimate. Then, we note that for non-deterministic $Z(s,a)$, the expectile function at a given state-action pair $E_{Z(s,a)} \in \mathbb{R}^{[0,1]}$ is a strictly increasing and continuous function that spans the entire convex hull of the distribution's support [Holzmann and Klar, 2016]. Meanwhile, the quantile function $Q_{Z(s,a)} \in \mathbb{R}^{[0,1]}$ spans the distribution's support. As a consequence, every quantile is a single expectile, i.e., there exists a functional mapping from quantile fractions to expectile fractions. In this work, we propose to learn such a mapper $m_\phi(s,a,\tau) \triangleq E^{-1}_{Z(s,a)} \circ F^{-1}_{Z(s,a)}(\tau)$ using the quantile regression loss function from Eq. (4). We then have $\forall(s,a) \in \mathcal{S} \times \mathcal{A}, \tau \in [0,1],\ Z_\theta(s,a,m_\phi(s,a,\tau)) \triangleq Q_{Z(s,a)}(\tau)$. We can then simply query our estimator of quantiles at the next state-action pair to yield a sound imputation step, while the parameters of the Z-function are learned through expectile regression.

For any tuple (s,a,s',a'), our proposed update step can be described as follows:

1. Sample fractions $\hat{\tau} \sim \mathcal{U}(0,1)$.
2. Generate approximate samples of the target distribution using the quantile representation:

$$\hat{z} = R(s,a) + \gamma Z_\theta(s',a',m_\phi(s',a',\hat{\tau})).$$

3. Use expectile regression to learn the Z-function:

$$Z_\theta(s,a,\hat{\tau}) \leftarrow \min_\theta \mathcal{L}_E\left(Z_\theta(s,a,\hat{\tau}),\hat{z}\right).$$

4. Use quantile regression to learn the mapper:

$$m_\phi(s,a,\hat{\tau}) \leftarrow \min_\phi \mathcal{L}_Q\left(Z_\theta(s,a,m_\phi(s,a,\hat{\tau})),\hat{z}\right).$$

The state-action embeddings of the mapper are copied form those of the Z-function. This way, the parameters of the Z-function (in our experiments below this includes the large image embedding networks and the overall scale of the rewards) are learned using expectile regression, while only the residual shape difference between the quantile and expectile function is learned by the mapper, using quantile regression.

The update step described above can be formalized as a distributional operator, which we define in Section 4.2. We prove that our proposed update operator converges to the distributional Bellman operator in the limit of infinite estimated quantile/expectile fractions. Then, in Section 4.3, we detail a practical implementation of dual expectile-quantile RL based on implicit quantile networks that we name IEQN.

4.2 CONVERGENCE OF THE DUAL EXPECTILE-QUANTILE OPERATOR

In this section, we prove that our proposed update operator converges to the distributional dynamic programming operator from Eq. (1) as the number of quantiles and expectiles kept in memory grows infinitely large, i.e., that the error incurred by our dual expectile-quantile operator vanishes in the limit of an infinite number of statistics to be evaluated. This result relies on several properties of the expectile function, including its absolute continuity that we establish in the following lemma:

Algorithm 1 Implicit expectile-quantile networks (IEQN) update

Require: Z-function Z_θ, mapper m_ϕ, fractions $(\tau_i)_{i=1,\dots,N} \sim \mathcal{U}([0,1])$, learning rate λ.

Collect experience (s, a, r, s')

for $i = 1, \dots, N$ **do**

Compute expectile values $e_i \leftarrow Z_\theta(s, a, \tau_i)$ and quantile values $q_i \leftarrow Z_\theta(s, a, m_\phi(\tau_i))$

Compute the greedy next-action:

$$a' \leftarrow \max_{b \in \mathcal{A}} \frac{1}{N} \sum_{i=1}^{N} Z_\theta(s', b, m_\phi(\tau_i))$$

Compute target samples:

$$z_i \leftarrow r + \gamma \cdot \text{stop\_grad}(Z_\theta(s', a', m_\phi(\tau_i)))$$

end for

Compute the expectile loss:

$$\mathcal{L}_E \leftarrow \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\tau_i \mathbb{1}_{z_j > e_i} + (1-\tau_i)\mathbb{1}_{z_j \le e_i}\right)(z_j - e_i)^2$$

Compute the quantile loss:

$$\mathcal{L}_Q \leftarrow \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\tau_i \mathbb{1}_{z_j > q_i} + (1-\tau_i)\mathbb{1}_{z_j \le q_i}\right)|z_j - q_i|$$

Update expectile function parameters: $\theta \leftarrow \theta - \lambda \nabla_\theta \mathcal{L}_E$

Update mapper parameters: $\phi \leftarrow \phi - \lambda \nabla_\phi \mathcal{L}_Q$

Lemma 1. *Let Z be a random variable taking values in $[a, b]$ with finite second moment and whose CDF admits finitely many discontinuities. Then, the expectile function $E_Z : \tau \mapsto \arg\min_e \mathbb{E}_{z\sim Z}[(\tau \mathbb{1}_{z>e} + (1-\tau)\mathbb{1}_{z\le e})(z-e)^2]$ is absolutely continuous on $[0, 1]$.*

The proofs for this lemma and all results below are included in the appendix. We are now able to prove our main result, Theorem 2, i.e., that our dual regression projection operator approximates the target distribution well in the limit of an infinite number of quantile/expectile fractions:

Theorem 2. *Let $\tau_k = \frac{2k-1}{2K}$, for $k = 1, \dots, K$, and let $\Pi_{\mathcal{M}}^K : \mathscr{P}(\mathbb{R}) \to \mathscr{P}(\mathbb{R})$ be the dual regression projection operator defined as:*

$\forall \eta \in \mathscr{P}(\mathbb{R})$,

$$\Pi_{\mathcal{M}}^K(\eta) = \frac{1}{K} \sum_{k=1}^{K} \delta_{E_\eta\left(\text{floor}^K\left(E_\eta^{-1}(F_\eta^{-1}(\tau_k))\right)\right)}$$
$$= \frac{1}{K} \sum_{k=1}^{K} \delta_{E_\eta\left(\frac{2\lfloor K M(\tau_k)+1/2\rfloor - 1}{2K}\right)},$$

where $E_\eta : [0,1] \to \mathbb{R}$ is the expectile function of η, $F_\eta^{-1} : [0,1] \to \mathbb{R}$ is the inverse CDF – i.e., the quantile function – of η, and $\text{floor}^K(x) = \tau_{\lfloor Kx + \frac{1}{2}\rfloor}$. Let $\eta \in \mathscr{P}(\mathbb{R})$ be a bounded-support probability distribution with finite second moment and whose CDF admits finitely many discontinuities, and let W_1 be the 1-Wasserstein distance. Then:

$$\lim_{K\to\infty} W_1(\Pi_{\mathcal{M}}^K \eta, \eta) = 0 \ .$$

Reusing the notation from the theorem, we can formally define our dual expectile-quantile operator. Let $\pi \in \mathscr{P}(\mathcal{A})^{\mathcal{S}}$ be a policy, we have:

$$\mathcal{T}_{\mathcal{M}^K}^\pi = \Pi_{\mathcal{M}}^K \mathcal{T}^\pi \ ,$$

where $\mathcal{T}^\pi : Z(s_t, a_t) = R(s_t, a_t) + \gamma \mathbb{E}_\pi\left[Z(s_{t+1}^\pi, a_{t+1}^\pi)\right]$ is the distributional Bellman operator (see Section 2.1). We can now derive a key corollary in the context of distributional RL training:

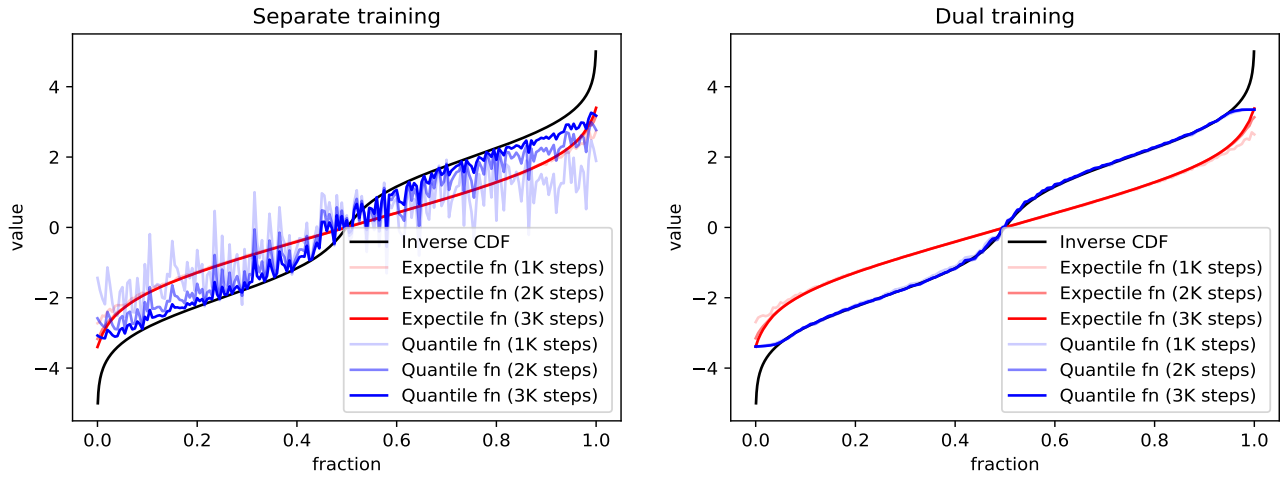
Corollary 3. *On Markov decision processes with bounded rewards and $\gamma < 1$, the dual expectile-quantile operator converges pointwise to the distributional Bellman operator:*

$$\lim_{K\to\infty} \mathcal{T}_{\mathcal{M}^K}^\pi = \mathcal{T}^\pi \text{ pointwise.}$$

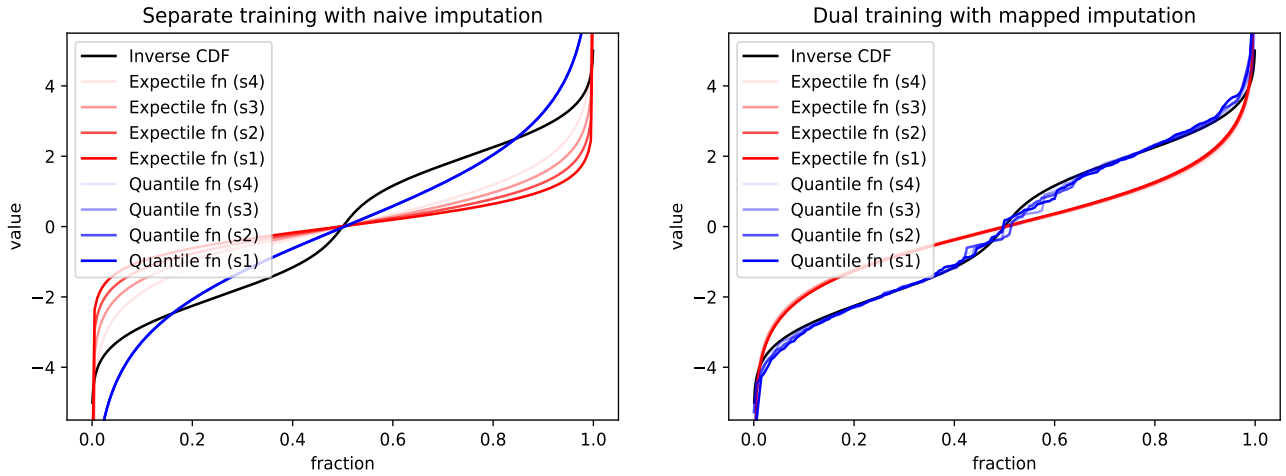
This result comes in contrast to the failure of the naive expectile operator [Rowland et al., 2019] to match the distributional Bellman operator. We now present a practical implementation of an agent using our dual approach.

4.3 A PRACTICAL IMPLEMENTATION: IEQN

We use the principle described in Section 4.1 to implement IEQN (Algorithm 1), a new distributional RL agent based on



(a) Approximating a distribution with separate and dual training.



(b) Tabular distributional RL with separate and dual training.

Figure 1: **(a)** Approximating a bimodal distribution with quantile and expectile regression. Quantile regression approximates the inverse CDF, albeit with high variance, especially on extreme values (left, blue curves). Expectiles converge very quickly to the expectile function (left, red curves). When training a mapper to generate quantiles from expectiles, quantile estimation becomes much more efficient (right). **(b)** Distributional RL with function approximation in a chain MDP with 4 states, and a bimodal reward distribution at the last state. The expectile function collapses as the temporal difference error propagates to previous states (left, red curves) while the quantile function is a poor approximation of the inverse CDF (left, blue curves). Our dual method solves both problems (right).

implicit quantile networks (IQN) [Dabney et al., 2018a]. The Z-function is modeled as a neural network inputting a state and a fraction $\tau \sim \mathcal{U}(0, 1)$, and outputting τ-expectile values for all actions. Its parameters are learned via an asymmetric L_2 loss, i.e., expectile regression. We also use a neural network to implement the mapper between quantile fractions and expectile fractions, and learn its parameters via an asymmetric L_1 loss, i.e., quantile regression.

5 EXPERIMENTS

We first demonstrate on a toy MDP the benefits of learning quantiles and expectiles together. We then describe our experimental setup and results on the Atari Arcade Learning Environment (ALE).

5.1 CHAIN MDP: A TOY EXAMPLE

We start by observing the effect of our proposed operator in a toy environment. The MDP comprises 4 states, each pointing to the next through a unique action and without

accumulating any reward, until the last state s_4, where the episode terminates and the agent obtains a reward sampled from a bimodal distribution $r \sim (\frac{1}{2}\mathcal{N}(-2,1) + \frac{1}{2}\mathcal{N}(+2,1))$ (see the Appendix for a visual description).

Figure 1a highlights the advantageous properties of expectile regression that were introduced in prior work [Philipps, 2021a,b, Waltrup et al., 2015]. When trying to approximate the distribution of terminal rewards directly from samples (left), we can see that expectile regression yields more accurate estimates than quantile regression in the low-data regime (recall that the quantile function is the inverse CDF while the expectile function is in general not). Interestingly, coupling expectile regression with our mapper (right) allows us to recover the quantile function much more efficiently than quantile regression itself. We can therefore confirm the findings from prior work [Philipps, 2021a, Waltrup et al., 2015] and conclude that our learning-based procedure for mapping estimated expectiles to their corresponding quantile fraction is effective.

In Figure 1b, we instantiate the problem in a typical dynamic programming setting, to illustrate the deficiencies of regular quantile and expectile dynamic programming. We can observe (left) that quantile function learning is sample-inefficient and fails to approximate the distribution within the given evaluation budget.* However, the distribution information is propagated correctly through temporal difference updates, since the quantile functions estimated at each state coincide. In contrast, the expectile function collapses to the mean as the error propagates from s_4 to s_1. This is due to the fact that expectile values at the next state-action pair cannot be used as pseudo-samples of the return distribution $Z(s_{t+1})$ [Rowland et al., 2019]. Finally, Figure 1b (right) shows that our dual training method, where the pseudo-samples of $Z(s', a')$ are the estimated quantiles $Z_\theta(s_{t+1}, m_\phi(\tau))$, solves both issues: the expectile function does not collapse anymore and the quantile function approximation is an accurate estimation of the inverse CDF.

5.2 EXPERIMENTS ON THE ATARI ARCADE LEARNING ENVIRONMENT

5.2.1 Baselines

We experimented with the following baselines to evaluate our approach:

IQN-0, IQN-1 We approximate quantiles using the general approach described in IQN [Dabney et al., 2018a], respectively without and with a Huber loss.

IEN-Naive We use a similar approach as for IQN, but trained with an expectile loss and a naive imputation step as described in [Rowland et al., 2019], i.e., expectile values are used as target for the temporal difference loss. The solver-based implementation described by the authors was too slow on our setup, as it was approximately 25 times slower than the other baselines.

*In this figure, we learn quantile and expectile functions parameterized by neural networks, as opposed to Figure 1a where each statistic is learned independently from others. This explains why the quantile function's appearance is smoother in this figure.

5.2.2 Environments

We opted to conduct our experiments with the Atari Learning Environment (ALE) [Bellemare et al., 2013], following the setup of Machado et al. [2018], notably including a 25% chance to perform a sticky action at each step, i.e., repeating the latest action instead of using the action predicted by the agent. This creates stochasticity in the environment, which should be captured by distributional RL agents. In order to accommodate for limited computing resources, we constrained ourselves to the Atari-5 subbenchmark [Aitchison et al., 2023], yet using 5 seeds to reduce the uncertainty in our results. We perform 25 validation episodes every 1M steps to generate our performance curves. As is common with ALE, we report human-normalized scores, rather than raw game scores, and we aggregate them using the interquartile mean (IQM), as it *is a better indicator of overall performance* (compared to sample median) [Agarwal et al., 2021], due to its robustness to scale across tasks and to outliers. It is especially needed, as the presence of sticky actions increases the number of outlier seeds.

5.2.3 Implementation details

We base all baselines and our method on the same underlying neural network, implemented in JAX [Bradbury et al., 2018]. Its architecture follows the structure detailed by Dabney et al. [2018a]. We used the training loop composition of CleanRL [Huang et al., 2022]. Hyperparameters can be found in the appendix. We implemented the Z-function for all agents as a feed-forward neural network with layer normalization. We did not use the fraction proposal network introduced with FQF [Yang et al., 2019], as our method can be seen as complementary to it, and we focus on the effect of the choice of statistics. Finally, we found that using layer normalization increased performance for both our method and baselines.

As described in Algorithm 1, we only use the expectile loss to update the Z-function for our agent, while we use the quantile loss to update our mapper. The mapper is implemented as a two layer, residual fully-connected neural network with ReLU and Tanh activations. Since it is queried to obtain both the candidate and target values, we use a mapper-specific target network updated less frequently than the live network, using Polyak averaging [Polyak and Juditsky, 1992] with a weight of 0.5. We share the parameters across all states, to simplify its architecture. We detail the implications of this

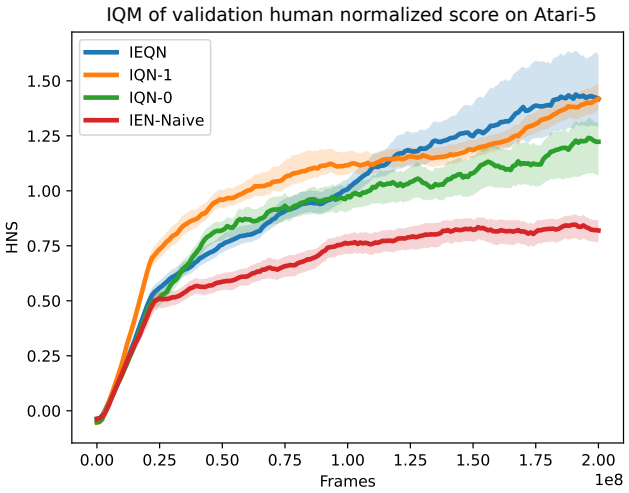


Figure 2: Interquartile mean of the human normalized score of distributional RL agents on the Atari-5 benchmark with 5 random seeds per environment. Shaded areas correspond to the 25-th and 75-th percentiles of a bootstrap distribution. A rolling average with window size of $20M$ frames is performed to enhance readability.

choice in the appendix.

5.2.4 Results

In this section, we verify that our dual approach also provides benefits at scale, on a classic benchmark.

We first present, in Figure 2, the aggregated results over 5 seeds on the Atari-5 benchmark. We can see that despite a slower start, IEQN ends up matching the performance of IQN-1. To get statistically stronger results, we also performed a bootstrap hypothesis test on the difference of IQMs at the end of training (we average scores from the last 5 validation epochs to be robust to instabilities). We found that our method surpasses the performance of both the quantile approach (achieved significance level 0.0117), and naive expectile approach (achieved significance level 0), thereby demonstrating the benefits of dual regression over single regression of either quantiles or expectiles on the final performance.

Furthermore, we verify in Table 1 that IEQN avoids distributional collapse in practice. In fact, while IQN-1's estimated distribution is much narrower than IQN-0's – a confirmation that the Huber loss causes distributional collapse, despite its better efficiency – IEQN's quantile spread is much larger than IQN-1's. Moreover, the expectile spread of IEQN is much larger and more stable than that of IEN-Naive, suggesting that expectile distributional RL yields degenerate distributions, as noted by Rowland et al. [2019], but that dual expectile-quantile distributional RL avoids this collapse.

Table 1: Average and standard deviation of the distance between quantile (respectively expectile) 0.1 and 0.9, relatively to the scale of the Q-function, at the end of training.

	Quantiles spread	Expectiles spread
IQN-0	1.25 ± 0.198	-
IQN-1	0.144 ± 0.072	-
IEN-Naive	-	0.174 ± 0.195
IEQN	0.721 ± 0.142	0.465 ± 0.086

6 CONCLUSION

We have proposed a statistics-based approach to distributional reinforcement learning that uses the simultaneous estimation of quantiles and expectiles of the action-value distribution. Previous work only estimated quantiles or expectiles separately. Our new approach presents the advantage of leveraging the efficiency of the expectile-based loss for both expectile and quantile estimation while solving the theoretical shortcomings of expectile-based distributional reinforcement learning, which often lead to a collapse of the expectile function in practice.

We have shown on a toy environment how the dual optimization affects the statistics recovered in distributional RL: in short, the quantile function is estimated more accurately than with vanilla quantile regression and the expectile function remains consistent after several steps of temporal difference training. We have also benchmarked our approach at scale, on the Atari-5 benchmark. Our model, IEQN, matches the performance of the Huber-based IQN-1 and surpasses that of both expectile and quantile-based agents, demonstrating its effectiveness in practical scenarios.

We open possibilities for future research to use a distributional approach that performs well and does not collapse. For future work, we plan to investigate how the dual approach can be used in risk-aware decision-making problems, and how it performs when the goal is to optimize risk metrics such as (conditional) value-at-risk. Moreover, we plan to gather insights into what type of behavior is favored by the quantile and expectile loss, respectively.

Acknowledgments

This research was partially supported by Ahold Delhaize, through AIRLab Amsterdam, by the Dutch Research Council (NWO), under project numbers 024.004.022, NWA.1389.20.183, and KICH3.LTP.20.006, and by the European Union's Horizon Europe program under grant agreement No 101070212. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc Bellemare. Deep Reinforcement Learning at the Edge of the Statistical Precipice. *Advances in Neural Information Processing Systems*, 34, 2021.

Matthew Aitchison, Penny Sweetser, and Marcus Hutter. Atari-5: Distilling the Arcade Learning Environment down to Five Games. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 421–438. PMLR, 23–29 Jul 2023.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research*, 47:253–279, jun 2013.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A Distributional Perspective on Reinforcement Learning. In *ICML*, pages 449–458. PMLR, 2017.

Marc G. Bellemare, Will Dabney, and Mark Rowland. *Distributional Reinforcement Learning*. MIT Press, 2023.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: Composable Transformations of Python+NumPy Programs, 2018. URL http://github.com/google/jax.

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit Quantile Networks for Distributional Reinforcement Learning. In *ICML*, pages 1096–1105. PMLR, 2018a.

Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional Reinforcement Learning with Quantile Regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018b.

Yingjie Fei, Zhuoran Yang, and Zhaoran Wang. Risk-Sensitive Reinforcement Learning with Function Approximation: A Debiasing Approach. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3198–3207. PMLR, 18–24 Jul 2021.

Ido Greenberg, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Efficient Risk-Averse Reinforcement Learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 32639–32652. Curran Associates, Inc., 2022.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *arXiv preprint arXiv:1801.01290*, 2018.

Hajo Holzmann and Bernhard Klar. Expectile Asymptotics. *Electronic Journal of Statistics*, 10(2):2355 – 2371, 2016.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. CleanRL: High-quality Single-file Implementations of Deep Reinforcement Learning Algorithms. *JMLR*, 23(274):1–18, 2022.

Sami Jullien, Mozhdeh Ariannezhad, Paul Groth, and Maarten de Rijke. A Simulation Environment and Reinforcement Learning Method for Waste Reduction. *TMLR*, 2023.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline Reinforcement Learning with Implicit Q-Learning. In *ICLR*, 2022.

Shiau Hong Lim and Ilyas Malik. Distributional Reinforcement Learning for Risk-Sensitive Policies. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 30977–30989. Curran Associates, Inc., 2022.

Nikolai Luzin. *The Integral and Trigonometric Series (Russian)*. PhD thesis, Moscow State University, 1915.

Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew J. Hausknecht, and Michael Bowling. Revisiting the Arcade Learning Environment: Evaluation Protocols and Open Problems for General Agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

John Martin, Michal Lyskawinski, Xiaohu Li, and Brendan Englot. Stochastically Dominant Distributional Reinforcement Learning. In *ICML*, volume 119, pages 6745–6754. PMLR, 13–18 Jul 2020.

Borislav Mavrin, Hengshuai Yao, Linglong Kong, Kaiwen Wu, and Yaoliang Yu. Distributional Reinforcement Learning for Efficient Exploration. In *ICML*, pages 4424–4434. PMLR, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning. *arXiv preprint arXiv:1312.5602*, 2013.

Whitney K. Newey and James L. Powell. Asymmetric Least Squares Estimation and Testing. *Econometrica*, 55(4): 819–847, 1987.

Colin Philipps. When is an Expectile the Best Linear Unbiased Estimator? *SSRN*, 2021a.

Collin Philipps. Interpreting Expectiles. *SSRN*, 2021b.

Boris T. Polyak and Anatoli B. Juditsky. Acceleration of Stochastic Approximation by Averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Mark Rowland, Robert Dadashi, Saurabh Kumar, Rémi Munos, Marc G. Bellemare, and Will Dabney. Statistics and Samples in Distributional Reinforcement Learning. In *ICML*, pages 5528–5536. PMLR, 2019.

Mark Rowland, Rémi Munos, Mohammad Gheshlaghi Azar, Yunhao Tang, Georg Ostrovski, Anna Harutyunyan, Karl Tuyls, Marc G. Bellemare, and Will Dabney. An Analysis of Quantile Temporal-Difference Learning. *arXiv preprint arXiv:2301.04462*, 2023.

Malcolm Strens. A Bayesian Framework for Reinforcement Learning. In *ICML*, volume 2000, pages 943–950, 2000.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

Nikos Vlassis, Mohammad Ghavamzadeh, Shie Mannor, and Pascal Poupart. Bayesian Reinforcement Learning. *Reinforcement Learning: State-of-the-Art*, pages 359–386, 2012.

Linda Schulze Waltrup, Fabian Sobotka, Thomas Kneib, and Göran Kauermann. Expectile and Quantile Regression—David and Goliath? *Statistical Modelling*, 15(5): 433–456, 2015.

Xingli Wei. *Parameter Estimation and Prediction Interval Construction for Location-Scale Models with Nuclear Applications*. PhD thesis, McMaster University, 2014.

Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu. Fully Parameterized Quantile Function for Distributional Reinforcement Learning. *NeurIPS*, 32, 2019.

Qiwli Yao and Howell Tong. Asymmetric Least Squares Regression Estimation: A Nonparametric Approach. *Journal of Nonparametric Statistics*, 6(2-3):273–292, 1996.

Moisej A. Zaretsky. On One Theorem on Absolutely Continuous Functions (Russian). *Doklady Rossiiskoi Akademii Nauk*, 1925.

APPENDIX

This appendix has the following sections:

A HYPERPARAMETERS, CODE AND IMPLEMENTATION DETAILS

A.1 HYPERPARAMETERS

We use JAX [Bradbury et al., 2018] to train our models. A full training procedure of 200M training frames and corresponding validation epochs takes approximately 50 hours in our setup.

Table 2: Z-function hyperparameters.

Key	Value
Discount factor	0.99
Batch size	32
Fraction distribution	$\mathcal{U}([0,1])$
Learning rate	$1e^{-4}$
Random frames before training	200000
Size of convolutional layers	[32, 64, 64]
Size of fully-connected layer	512
Critic updates per sample	2
Buffer size	1e6
Frames between target network updates	35000
Target network update rate	1.0

Table 3: Mapper hyperparameters.

Key	Value
Layer size	64
Learning rate	$7e^{-5}$
Target network update rate	0.5

A.2 CODE

Our training and evaluation loop is based on CleanRL [Huang et al., 2022]. The code base is available on `https://github.com/samijullien/ieqn`.

B SHARING THE MAPPER'S PARAMETERS

Sharing the mapper's parameters across states and actions allows us to lighten the computational burden, which is part of the goal of this paper. We found this technique to work well in practice on the Atari-5 benchmark, although it requires additional assumptions in theory. We review these assumptions in this section.

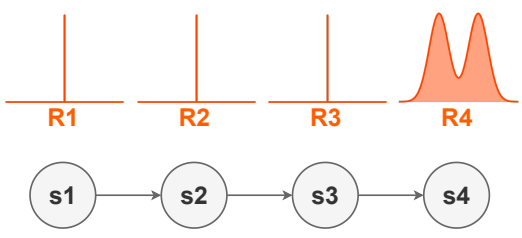
Yao and Tong [1996] show that there exists such a shared mapping between quantiles and expectiles when the regression follows a location-scale model, i.e., for random variables X and Y:

$$Y = \mu(X) + \sigma(X)\varepsilon,$$

where μ and σ are continuous functions, ε is centered and finite-variance, and ε, X are independent. When the return distribution follows this model, X being the state-action variable in this context, sharing the mapper's parameters is theoretically valid. While this may seem limiting, it does not require all state-action pairs to be allocated the same distributions, only that they share a common shape. Moreover, the location-scale family is quite broad, as it includes, e.g., Normal, Student, Cauchy, GEV distributions, and more [Wei, 2014].

In many distributional reinforcement learning scenarios, the assumption may be satisfied. For instance, when the environmental stochasticity emerges from small, independent perturbations, i.e., normally-distributed errors, the return distribution at every state will still be normally distributed as convolutions of Gaussian distributions are also Gaussian. On the other hand, this assumption can fail under high-frequency transition distributions, i.e., branching behaviors, where the same state-action pair can yield drastically different outcomes and the reward-next-state distribution has non-continuous support. We leave for future work the investigation of when sharing the mapper's parameters across state-action pairs fails in practice.

C TOY MARKOV DECISION PROCESS



$\gamma = 1$

Figure 3: Toy Markov decision process.

D PROOF OF LEMMA 1

Our proof of Theorem 2 requires the absolute continuity of the expectile function. Therefore, we first prove the following lemma:

Lemma 1. *Let Z be a random variable taking values in $[a, b]$ with finite second moment and whose CDF admits finitely many discontinuities. Then, the expectile function $E_Z : \tau \mapsto \arg\min_e \mathbb{E}_{z\sim Z}[(\tau \mathbb{1}_{z>e} + (1-\tau)\mathbb{1}_{z\leq e})(z-e)^2]$ is absolutely continuous on $[0, 1]$.*

Proof. Our proof relies on the Banach-Zarecki theorem [Zaretsky, 1925], which states that any real-valued function f defined on a real bounded closed interval is absolutely continuous if and only if on this interval:

(i) f is continuous;

(ii) f has bounded variation; and

(iii) f follows the Luzin N property [Luzin, 1915], i.e., the image by f of a set with null Lebesgue measure also has null Lebesgue measure.

It is well-known that the expectile function is continuous on $[0, 1]$ [Holzmann and Klar, 2016, Philipps, 2021b]. Therefore, (i) is satisfied.

E_Z is monotonically increasing and takes values in the finite support of Z. Therefore it has bounded variation and (ii) is satisfied.

In order to prove (iii), we first note that any function that is differentiable on a co-countable set has the Luzin N property [Luzin, 1915]. We therefore use our assumption that Z admits a finite number of discontinuities in the following.

Let F_Z be the CDF of Z and $D = \{z \in [a, b] : \lim_{x\to z} F_Z(x) \neq F_Z(z)\}$ be the finite set of points at which F_Z is not continuous. D is a finite set within a metric space and therefore closed. As a consequence, its complement $C_{[a,b]} = [a, b] \setminus D$ is open in $[a, b]$, i.e., $\forall z \in C_{[a,b]}, \exists \varepsilon > 0$ such that $\forall x \in [a, b] d(x, z) < \varepsilon \Rightarrow x \in C_{[a,b]}$. In other words, if F_Z is continuous at a point within $[a, b]$, it is also continuous in a neighborhood of that point within $[a, b]$. By assumption, the set $C^N_{[a,b]} = \left\{z \in [a, b] : \exists \varepsilon > 0, \forall x \in [a, b], d(x, z) < \varepsilon \Rightarrow x \in C_{[a,b]}\right\}$ of points where F_Z is continuous in a neighborhood of said point is therefore co-finite.

It has been shown that the expectile function E_Z is continuously differentiable at any point $\tau \in [0, 1]$ such that F_Z is continuous in a neighborhood of $E_Z(\tau)$ [Holzmann and Klar, 2016, Newey and Powell, 1987]. The expectile function is bijective [Philipps, 2021b] so the set of points where E_Z is differentiable $\mathcal{D}^{E_Z}_{[a,b]} = E_Z^{-1}\left(C^N_{[a,b]}\right)$ is also a co-finite set.

The expectile function is differentiable on a co-finite (and thus co-countable) set, i.e., it has the Luzin N property [Luzin, 1915], which yields (iii).

We can finally apply the Banach-Zarecki theorem and conclude that the expectile function E_Z is absolutely continuous on $[0, 1]$. □

E PROOF OF THEOREM 2

We can now use the absolute continuity of the expectile function under our assumptions to prove the following theorem:

Theorem 2. *Let $\tau_k = \frac{2k-1}{2K}$, for $k = 1, \dots, K$, and let $\Pi_{\mathcal{M}}^K : \mathscr{P}(\mathbb{R}) \to \mathscr{P}(\mathbb{R})$ be the dual regression projection operator defined as:*
$\forall \eta \in \mathscr{P}(\mathbb{R})$,

$$\begin{aligned}\Pi_{\mathcal{M}}^K(\eta) &= \frac{1}{K}\sum_{k=1}^{K} \delta_{E_\eta\left(\text{floor}^K\left(E_\eta^{-1}(F_\eta^{-1}(\tau_k))\right)\right)} \\ &= \frac{1}{K}\sum_{k=1}^{K} \delta_{E_\eta\left(\frac{2\lfloor K M(\tau_k)+1/2\rfloor-1}{2K}\right)},\end{aligned}$$

where $E_\eta : [0,1] \to \mathbb{R}$ is the expectile function of η, $F_\eta^{-1} : [0,1] \to \mathbb{R}$ is the inverse CDF – i.e., the quantile function – of η, and $\text{floor}^K(x) = \tau_{\lfloor Kx+\frac{1}{2}\rfloor}$. Let $\eta \in \mathscr{P}(\mathbb{R})$ be a bounded-support probability distribution with finite second moment and whose CDF admits finitely many discontinuities, and let W_1 be the 1-Wasserstein distance. Then:

$$\lim_{K\to\infty} W_1(\Pi_{\mathcal{M}}^K\eta, \eta) = 0\ .$$

Proof. Thanks to the triangle inequality, we have :

$$W_1(\Pi_{\mathcal{M}}^K\eta, \eta) \leqslant W_1(\Pi_{\mathcal{M}}^K\eta, \Pi_Q^K\eta) + W_1(\Pi_Q^K\eta, \eta)\ ,$$

where Π_Q^K is the projected quantile regression estimator defined as:

$$\forall \eta \in \mathscr{P}(\mathbb{R}),\quad \Pi_Q^K(\eta) = \frac{1}{K}\sum_{k=1}^{K} \delta_{F_\eta^{-1}(\tau_k)}\ .$$

Rowland et al. [2019, Lemma 3.2] showed that $W_1(\Pi_Q^K\eta, \eta) = O\left(\frac{1}{K}\right)$. We now turn to the first term:

$$\begin{aligned}W_1(\Pi_{\mathcal{M}}^K\eta, \Pi_Q\eta) &= \sum_{i=0}^{K-1}\frac{1}{K}\left|E_\eta\left(\text{floor}^K\left(E_\eta^{-1}\left(F_\eta^{-1}\left(\frac{2i+1}{2K}\right)\right)\right)\right) - F_\eta^{-1}\left(\frac{2i+1}{2K}\right)\right| \\ &= \sum_{i=0}^{K-1}\frac{1}{K}\left|E_\eta\left(\text{floor}^K\left(E_\eta^{-1}\left(F_\eta^{-1}\left(\frac{2i+1}{2K}\right)\right)\right)\right) - E_\eta\left(E_\eta^{-1}\left(F_\eta^{-1}\left(\frac{2i+1}{2K}\right)\right)\right)\right| \\ &\leqslant \sum_{i=0}^{K-1}\frac{1}{K}\left|E_\eta\left(\text{floor}^K\left(E_\eta^{-1}\left(F_\eta^{-1}\left(\frac{2i+1}{2K}\right)\right)\right)\right) - E_\eta\left(\text{floor}^K\left(E_\eta^{-1}\left(F_\eta^{-1}\left(\frac{2i+1}{2K}\right)\right)\right) + \frac{1}{K}\right)\right|,\end{aligned}$$

where the last inequality is obtained thanks to the monotonicity of the expectile function. By absolute continuity of the expectile function under our assumptions (proven in Lemma 1), we have:

$$\lim_{K\to\infty}\left|E_\eta\left(\text{floor}^K\left(E_\eta^{-1}\left(F_\eta^{-1}\left(\frac{2i+1}{2K}\right)\right)\right)\right) - E_\eta\left(\text{floor}^K\left(E_\eta^{-1}\left(F_\eta^{-1}\left(\frac{2i+1}{2K}\right)\right)\right) + \frac{1}{K}\right)\right| = 0,$$

from which we can deduce $\lim_{K\to\infty} W_1(\Pi_{\mathcal{M}}^K\eta, \Pi_Q\eta) = 0$ and finally $\lim_{K\to\infty} W_1(\Pi_{\mathcal{M}}^K\eta, \eta) = 0$. □

F PROOF OF COROLLARY 3

Finally, we can derive our main result for the use of distributional dynamic programming with both quantiles and expectiles:

Corollary 3. *On Markov decision processes with bounded rewards and $\gamma < 1$, the dual expectile-quantile operator converges pointwise to the distributional Bellman operator:*

$$\lim_{K\to\infty} \mathcal{T}^{\pi}_{\mathcal{M}^K} = \mathcal{T}^{\pi} \text{ pointwise.}$$

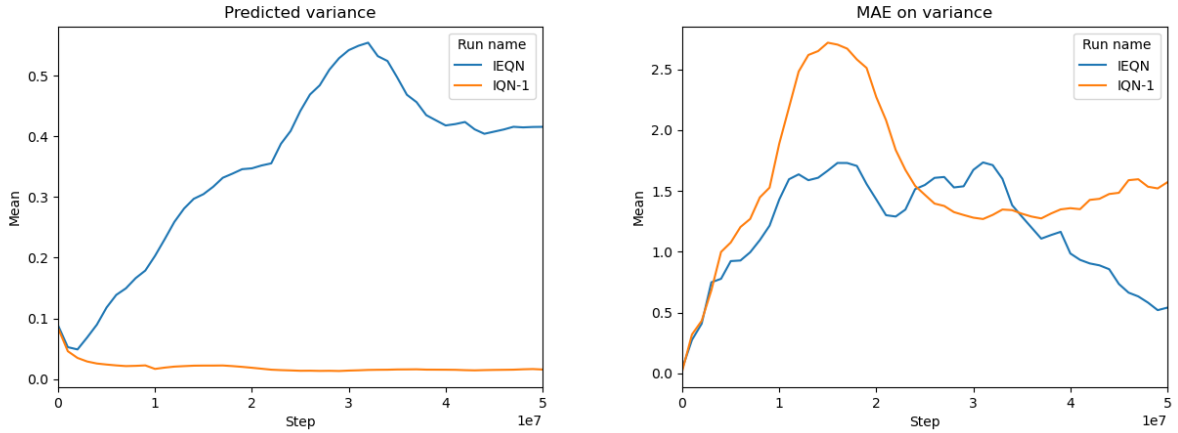
Proof. We have $\mathcal{T}^{\pi}_{\mathcal{M}^K} = \Pi^K_{\mathcal{M}} \mathcal{T}^{\pi}$. Bellemare et al. [2023] have shown that the set of empirical distributions $\mathcal{F}_E$ is closed under the operator $\mathcal{T}^{\pi}$ (Proposition 5.7). Thus, for any empirical return distribution $\eta \in \mathcal{F}_E$, $\mathcal{T}^{\pi}\eta$ is also empirical and its CDF admits finitely many discontinuities. Moreover, it has bounded support. Indeed, if, without loss of generality, we consider that the reward distribution take values in $[0, R_{\max}]$, we have that every possible return distribution η takes values in $[0, \frac{R_{\max}}{1-\gamma}]$, and therefore $\mathcal{T}^{\pi}\eta$ takes values in $[0, R_{max} + \gamma\frac{R_{\max}}{1-\gamma}] = [0, \frac{R_{\max}}{1-\gamma}]$.
We can now apply Theorem 2:

$$\forall \eta \in \mathcal{F}_E \ , \lim_{K\to\infty} W_1(\Pi^K_{\mathcal{M}} \mathcal{T}^{\pi}\eta, \mathcal{T}^{\pi}\eta) = 0,$$

and the result immediately follows. □

G ANALYSIS OF THE ESTIMATED VARIANCE

In this section, we perform an additional experiment to better assess the quality of the value distribution on the Atari task. The distribution learned in our method as well as all baselines estimates the optimal Z-function, i.e., the return distribution of the optimal policy, which we cannot have ground truth for on large-scale tasks. We may however assume that the greedy policy gets closer to the optimal policy towards the end of training. If we do so, then we can compare the learned Z-function with the return distribution obtained by unfolding our agent's policy. Below, we show the variance of the learned Z-function (Figure 4a), and the average deviation between this prediction and the observed squared differences when rolling out the policy (Figure 4b), throughout the first 50M steps of training on Battlezone.



(a) Predicted variance on Battlezone during training.

(b) Error on variance on Battlezone during training.

Figure 4: Comparison of estimated variance against observed variance of unfolding the greedy policy.

We can see that (i) IQN-1 predicts a very low variance compared to IEQN, and (ii) using the approximation that the current policy is close to the optimal policy, IEQN's prediction gets closer to the observed variance than IQN-1's, as training progresses.