NON-DECREASING QUANTILE FUNCTION NETWORK WITH EFFICIENT EXPLORATION FOR DISTRIBUTIONAL REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Although distributional reinforcement learning (DRL) has been widely examined in the past few years, two important problems still remain unsolved. One is how to ensure the monotonicity of the quantile estimates, the other is how to design an efficient exploration strategy to fully utilize the distribution information. In this paper, we propose a non-decreasing quantile function network (NDQFN) to address the crossing issue of the learned quantile curve and incorporate it with an efficient exploration method called distributional prediction error (DPE) for DRL. We demonstrate the necessity and efficiency of our method from both the theory and model implementation perspectives. Experiments on Atari 2600 Games demonstrate that our method can significantly outperforms some state-of-art DRL algorithms especially in some hard explored games.

1 INTRODUCTION

Distributional reinforcement learning (DRL) algorithms (Jaquette, 1973; Sobel, 1982; White, 1988; Morimura et al., 2010; Bellemare et al., 2017c), different from the value based methods (Watkins, 1989; Mnih et al., 2013) which focus on the expectation of the return, characterize the cumulative reward as a random variable and attempt to approximate its whole distribution. Most existing DRL methods fall into two main classes according to their ways to model the return distribution. One class, including categorical DQN (Bellemare et al., 2017a), C51, and Rainbow (Hessel et al., 2018), assumes that all the possible returns are bounded in a known range and learns the probability of each value through interacting with the environment. Another class, for example the QR-DQN (Bellemare et al., 2017b), tries to obtain the quantile estimates at some fixed locations by minimizing the Huber loss (Huber, 1992) of quantile regression (Koenker, 2005). To more precisely parameterize the entire distribution, some quantile value based methods, such as IQN (Dabney et al., 2018) and FQF (Yang et al., 2019), are proposed to learn a continuous map from any quantile fraction $\tau \in (0, 1)$ to its estimate on the quantile curve.

The theoretical validity of QR-DQN (Bellemare et al., 2017b), IQN (Dabney et al., 2018) and FQF (Yang et al., 2019) heavily depends on a prerequisite that the approximated quantile curve is non-decreasing. Unfortunately, since no global constraint is imposed when simultaneously estimating the quantile values at multiple locations, the monotonicity can not be ensured by using any existing network design. At early training stage, the crossing issue is even more severe given limited training samples. Another problem to be solved is how to design an efficient exploration method for DRL. Most existing exploration techniques are originally designed for non-distributional RL (Bellemare et al., 2016; Ostrovski et al., 2017; Tang et al., 2017; Fox et al., 2018; Machado et al., 2018), and very few of them can work for the DRL situation. Mavrin et al. (2019) proposes a DRL-based exploration method, DLTV, based on QR-DQN which outperforms QR-DQN in most Atari games within 40 million training frames. However, this approach can not be applied to quantile value based algorithms such as IQN since the quantile locations are re-sampled at each training iteration and the exploration bonus based on left truncated variance could be extremely unstable.

To address these two common issues in DRL studies, we propose a novel algorithm called non-decreasing quantile function network (NDQFN), together with an efficient exploration strategy, distributional prediction error (DPE), designed for DRL. The NDQFN architecture allows us to approx-

imate the quantile distribution of the return by using a non-decreasing piece-wise linear function. The monotonicity of $N+1$ fixed quantile levels is ensured by an incremental structure, and the quantile value at any $\tau \in (0,1)$ can be estimated as the weighted sum of its two nearest neighbors among the $N+1$ locations. DPE uses the 1-Wasserstein distance between the quantile distributions estimated by the target network and the predictor network as an additional bonus when selecting the optimal action. We describe the implementation details of the two architectures with some theoretical and empirical support. We examine the performance of the NDQFN model with DPE on Atari 2600 games, and show that it performs consistently better than IQN especially in some hard-explored games such as Venture, Montezuma Revenge and Private eyes.

For the rest of this paper, we first go through some background knowledge of distributional RL in Section 2. Then in Sections 3, we introduce NDQFN, DPE and describe their implementation details. Section 4 presents the experiment results using the Atari benchmark, investigating the empirical performance of NDQFN, DPE and their combination by comparing with some baseline methods.

2 Background and Related Work

Following the standard reinforcement learning setting, the agent-environment interactions are modeled as a Markov Decision Process, or MDP, $(\mathcal{X}, \mathcal{A}, R, P, \gamma)$ (Puterman, 2014). $\mathcal{X}$ and $\mathcal{A}$ denote a finite set of states and a finite set of actions, respectively. $R : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0,1)$ is a discounted factor. $P : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to [0,1]$ is the transition kernel.

On the agent side, a stochastic policy π maps state x to a distribution over action $a \sim \pi(\cdot|x)$ regardless of the time step t. The discounted cumulative rewards is denoted by a 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 objective is to find an optimal policy π^* to maximize the expectation of $Z^\pi(x,a)$, $\mathbb{E}Z^\pi(x,a)$, which is denoted by the state-action value function $Q^\pi(x,a)$. A common way to obtain π^* is to find the unique fixed point $Q^* = Q^{\pi^*}$ of the Bellman optimality operator $\mathcal{T}$ (Bellman, 1966):

$$Q^\pi(x,a) = \mathcal{T}Q^\pi(x,a) := \mathbb{E}[R(x,a)] + \gamma \mathbb{E}_P \max_{a'} Q^*(x', a'). \tag{1}$$

The state-action value function Q can be approximated by a parameterized function Q_θ (e.g. a neural network). Q -learning (Watkins, 1989) iteratively updates the network parameters by minimizing the squared temporal difference (TD) error

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

on a sampled transition (x_t, a_t, r_t, x_{t+1}), collected by running an ϵ-greedy policy over Q_θ.

Distributional RL algorithms, instead of directly estimating the mean $\mathbb{E}Z^\pi(x,a)$, focus on the distribution $Z^\pi(x,a)$ to sufficiently capture the intrinsic randomness. The distributional Bellman operator, which has a similar structure with (1), is defined as follows,

$$\mathcal{T}Z^\pi(x,a) \overset{D}{=} R(x,a) + \gamma Z^\pi(X', A'),$$

where $X' \sim P(\cdot|x,a)$ and $A' \sim \pi(\cdot|X')$. $A \overset{D}{=} B$ denotes the equality in probability laws.

3 Non-decreasing Quantile Function Network

In this section, we propose a novel Non-decreasing Quantile Function Network (NDQFN) to address the crossing issue of DRL. We begin by discussing the importance of ensuring the monotonicity of the learned quantile function. Then we give the formal definition of NDQFN and describe its implementation details in practice. Last but not least, we introduce a general exploration approach named distributional prediction error (DPE), which can be incorporated with NDQFN or other DRL algorithms to improve the training efficiency of the DRL agent.

3.1 Non-monotonicity in Distirbutional Reinforcement Learning

The crossing issue of the quantile estimates has been broadly studied by the statistics community (He, 1997; Chernozhukov et al., 2010; Dette & Volgushev, 2008). In Distributional RL, with limited observations at each quantile level, it is much likely that the obtained quantile curve for some

state-action pair (x, a) is non-monotonic as Figure 1(a) illustrates. The key reason behind this phenomenon is that the quantile values at multiple quantile fractions are separately estimated without applying any global constraint to ensure the monotonicity. Ignoring the non-decreasing property of the learned quantile function leaves a theory-practice gap, which results in the potentially non-optimal action selection and abnormal policy gradient in practice. How to ensure the monotonicity of the approximated quantile function is a challenging task, especially to some quantile value based algorithms such as IQN and FQF, which do not focus on fixed quantile locations during the training process. To address this issue, we propose NDQFN, an incremental network structure, to determinately output non-decreasing quantile estimates for any set of quantile fractions.

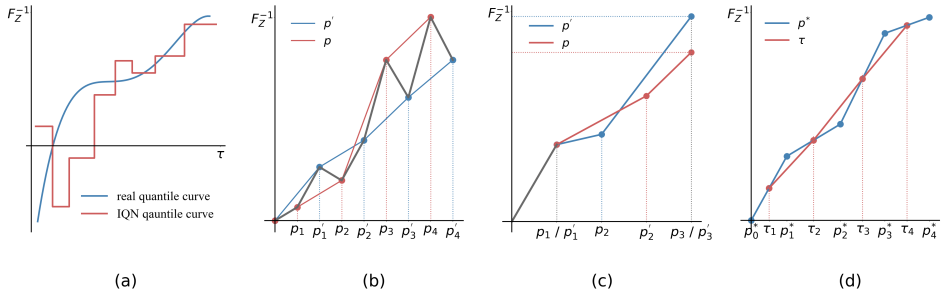

Figure 1: Quantile estimates obtained by (a) IQN, (b)(c) a simple incremental structure proposed in (3), and (d) NDQFN

3.2 Non-decreasing Quantile Function

Let $\boldsymbol{p} = (p_0, \cdots, p_N)^\top$ be a vector of $N+1$ quantile fractions, satisfying $0 \leq p_i \leq p_{i+1} \leq 1$ for each $i \in \{1, 2, \ldots, N-1\}$. In distributional RL studies, the distribution of the total return $Z(x, a)$ is usually parameterized by a mixture of N Diracs, such that

$$Z_{\boldsymbol{\theta},\boldsymbol{p}}(x, a) := \sum_{i=0}^{N-1} (p_{i+1} - p_i)\delta_{\theta_i(x,a)}, \tag{2}$$

where each θ_i denotes the quantile estimation at $\hat{p}_i = \frac{p_i + p_{i+1}}{2}$, and $\boldsymbol{\theta} = (\theta_0, \ldots, \theta_{N-1})^\top$. We let $F_Z^{-1}(\tau) = \inf\{z \in \mathbb{R} : \tau \leq F_Z(z)\}$ be the quantile function, which is the the inverse of the cumulative distribution function $F_Z(z) = Pr(Z < z)$. IQN (Dabney et al., 2018) estimates the quanitle function $F^{-1}_{Z(x,a)}(\tau)$ by $f(\psi(x) \odot \phi(\tau))_a$, where $\psi : \mathcal{X} \to \mathbb{R}^d$ and $\phi : [0, 1] \to \mathbb{R}^d$ represent the embedding for state x and quantile fraction τ. $f : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{A}|}$ maps the element-wise product of $\psi(x)$ and $\phi(\tau)$ to the final quantile estimate. As Figure 1(a) illustrates, the monotonicity of the learned quantile values at multiple quantile fractions can not be ensured since they are individually estimated without adding a global non-crossing constraint.

To address this crossing issue, one potential way is to estimate the increase of the quantile values between two nearby fractions p_{i-1} and p_i, i.e., $\Delta_i(x, a; \boldsymbol{p}) = F_Z^{-1}(p_i) - F_Z^{-1}(p_{i-1})$. Then each $F_Z^{-1}(p_i)$ can be represented as the cumulative sum of these non-negative increments. Let $g : \mathbb{R}^{2d} \to [0, \infty)^{|\mathcal{A}|}$ be a non-negative function. The baseline $\Delta_0(x, a; \boldsymbol{p}) := F^{-1}_{Z(x,a)}(p_0)$ at p_0 and the N non-negative increments $\Delta_i(x, a; \boldsymbol{p})$ for $i \in \{1, \cdots, N\}$ are approximated by

$$\begin{aligned} &\Delta_0(x, a; \boldsymbol{p}) \approx \Delta_{0,\omega}(x, a; \boldsymbol{p}) = f(\psi(x))_a, \\ &\Delta_i(x, a; \boldsymbol{p}) \approx \Delta_{i,\omega}(x, a; \boldsymbol{p}) = g(\psi(x), \phi(p_i), \phi(p_{i-1}))_a, \ i = 1, \cdots, N. \end{aligned} \tag{3}$$

where f is defined in the same way as above and ω includes all the parameters to be learned. The two functions f and g here share the same state embedding $\psi(x)$. By limiting the output range of g_a in $[0, \infty)$, the obtained $N+1$ quantile estimates are non-decreasing.

Unfortunately, the incremental structure introduced in (3) cannot fundamentally solve the crossing problem. It indeed ensures the non-decreasing order of the sampled quantile fractions, such as p_1 to p_4 in Figure 1(b), at each training iteration. However, the monotonicity of another set of uncovered quantile levels, such as p'_1 to p'_4 in Figure 1(b), can not be guaranteed. Another weakness of (3) is

that the estimate of the increment $\Delta_{i,\omega}(x, a; \boldsymbol{p})$ highly depends on the base location p_{i-1} it uses. As Figure 1(c) shows, it is much likely to obtain two different quantile estimate for the same fraction p_3 by using two different bases p_2 and p_2' in practice.

To overcome the two main shortcomings of (3), we introduce our method called Non-Decreasing Quantile Function Network (NDQFN) to compute the quantile value $F^{-1}_{Z(x,a)}(\tau)$ at any $\tau \in (0, 1)$ as a linear combination of the quantile estimates at $N + 1$ pre-defined quantile fractions $\boldsymbol{p}^* = (p_0^*, p_1^*, \ldots, p_N^*)^\top$. We let $p_0^* = \varepsilon$ and $p_N^* = 1 - \varepsilon$ with ε being a small positive number, and $0 < p_i^* < p_{i+1}^* < 1$ for each $i \in \{0, 1, 2, \ldots, N-1\}$. To be more specific, we use $\Pi_{\boldsymbol{p}^*, \boldsymbol{\Delta}_\omega}$ to denote a projection operator that projects the quantile function onto a piece-wise linear function supported by a fixed quantile fraction vector $\boldsymbol{p}^*$ and an incremental structure $\boldsymbol{\Delta}_\omega = (\Delta_{0,\omega}, \cdots, \Delta_{N,\omega})^\top$. The $\Delta_{i,\omega}$'s here are defined in the same way as (3). For any quantile level $\tau \in (0, 1)$, its projection-based quantile estimate is given by

$$F^{-1,\boldsymbol{p}^*,\boldsymbol{\Delta}_\omega}_{Z(x,a)}(\tau) = \Pi_{\boldsymbol{p}^*,\Delta_\omega} F^{-1}_{Z(x,a)}(\tau) = \Delta_{0,\omega}(x, a; \boldsymbol{p}^*) + \frac{1}{N} \sum_{i=0}^{N-1} G_{i,\boldsymbol{\Delta}_\omega}(\tau, x, a; \boldsymbol{p}^*), \quad (4)$$

where $G_{i,\boldsymbol{\Delta}_\omega}$ is a linear function satisfying

$$G_{i,\boldsymbol{\Delta}_\omega}(\tau, x, a; \boldsymbol{p}^*) = \left\{ \sum_{j=1}^{i} \Delta_{j,\omega}(x, a; \boldsymbol{p}^*) + \frac{\tau - p_i^*}{p_{i+1}^* - p_i^*} \Delta_{i+1,\omega}(x, a; \boldsymbol{p}^*) \right\} I(p_i^* \leq \tau < p_{i+1}^*),$$

Figure 1(d) visualizes the quantile values at a support vector $\boldsymbol{p}^* = (p_1^*, p_2^*, p_3^*, p_4^*)^\top$ and the NDQFN estimates at another vector of quantile levels $\boldsymbol{\tau} = (\tau_1, \tau_2, \tau_3, \tau_4)^\top$ obtained from (4). Different from the results in Figure 1(b), all the 8 quantile estimates in Figure 1(d) remain non-decreasing. For any other vector $\boldsymbol{\tau}' \neq \boldsymbol{\tau}$, the monotonicity can still be satisfied by using the same support $\boldsymbol{p}^*$ and the linear combinations of the quantile values at the support vector $\boldsymbol{p}^*$. Compared to (3), the piece-wise linear structure guarantees the monotonicity of any two quantile fractions τ_1, τ_2 in $(0, 1)$ regardless of whether they are included in the training process or not. The expected return, also known as the Q-function, based on NDQFN is then given by

$$Q_\omega(x, a) = \int_{p_0^*}^{p_N^*} F^{-1,\boldsymbol{p}^*,\boldsymbol{\Delta}_\omega}_{Z(x,a)}(\tau) d\tau. \quad (5)$$

3.3 Network Training and Implementation Details

Based on (4), we incorporate n-step updates (Sutton, 1988), double Q learning (Hasselt, 2010), quantile regression (Koenker, 2005) and distributional Bellman update (Bellemare et al., 2017b) into the training of Non-decreasing Quantile Function Network (NDQFN). For notational simplicity, we let $P_{\tau,\omega}(x, a) = F^{-1,\boldsymbol{p}^*,\boldsymbol{\Delta}_\omega}_{Z(x,a)}(\tau)$, given a support vector $\boldsymbol{p}^*$. Following the idea of IQN, two random vectors of quantile fractions $\boldsymbol{\tau}' = (\tau_1', \cdots, \tau_{N_1}')^\top, \boldsymbol{\tau}^* = (\tau_1^*, \cdots, \tau_{N_2}^*)^\top$ are independently drawn from a uniform distribution $U(0, 1)$ at each training iteration. In this case, for each $i \in \{1, \ldots, N_1\}$ and each $j \in \{1, \ldots, N_2\}$, the corresponding temporal difference (TD) error (Dabney et al., 2018) with n-step updates on $(x_t, a_t, r_t, \cdots, r_{t+n-1}, x_{t+n})$ is computed as follows,

$$\delta_{i,j} = \sum_{i=0}^{n-1} \gamma^i r_{t+i} + \gamma^n P_{\tau_j^*,\omega'} \left(x_{t+n}, \arg\max_{a' \in \mathcal{A}} Q_{\omega'}(x_{t+n}, a') \right) - P_{\tau_i',\omega}(x_t, a_t), \quad (6)$$

where ω and ω' denote the online network and the target network, respectively. In particular, since NDQFN employs the piece-wise linear approximations defined in (4), the expected return Q_ω can be estimated by

$$\begin{aligned} Q_\omega = \int_{p_0^*}^{p_N^*} F^{-1,\boldsymbol{p}^*,\boldsymbol{\Delta}_\omega}_{Z(x,a)}(\tau) d\tau &= \sum_{i=0}^{N-1} (p_{i+1}^* - p_i^*) F^{-1,\boldsymbol{p}^*,\boldsymbol{\Delta}_\omega}_{Z(x,a)}\left(\frac{p_i^* + p_{i+1}^*}{2}\right) \\ &= \sum_{i=0}^{N-1} \frac{(p_{i+1}^* - p_i^*)}{2} \left[F^{-1,\boldsymbol{p}^*,\boldsymbol{\Delta}_\omega}_{Z(x,a)}(p_{i+1}^*) + F^{-1,\boldsymbol{p}^*,\boldsymbol{\Delta}_\omega}_{Z(x,a)}(p_i^*) \right] \end{aligned}$$

Thus, we can train the whole network by minimizing the Huber quantile regression loss (Huber, 1992) as follows,

$$\mathcal{L}(x_t, a_t, r_t, \cdots, r_{t+n-1}, x_{t+n}) = \frac{1}{N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \rho_{\tau_i'}^{\kappa} \left(\delta_{i,j}\right), \tag{7}$$

where

$$\rho_{\tau}^{\kappa} \left(\delta_{i,j}\right) = \left|\tau - I\left(\delta_{i,j} < 0\right)\right| \frac{L_{\kappa}\left(\delta_{i,j}\right)}{\kappa},$$

$$L_{\kappa}\left(\delta_{i,j}\right) = \begin{cases} \frac{1}{2}\delta_{i,j}^2, & \text{if } |\delta_{i,j}| \leq \kappa \\ \kappa\left(|\delta_{i,j}| - \frac{1}{2}\kappa\right), & \text{otherwise} \end{cases},$$

κ is a pre-defined positive constant, and $|\cdot|$ denotes the absolute value of a scalar. To investigate the convergence of the proposed algorithm, we introduce the following theorem, which can be seen an extension of Proposition 2 in Bellemare et al. (2017b)

Theorem 1. *Let $\Pi_{\boldsymbol{p}^*, \boldsymbol{\Delta}_\omega}$ be the quantile projection defined as above with non-decreasing quantile function $F_Z^{-1, \boldsymbol{p}^*, \boldsymbol{\Delta}_\omega}$. For any two value distribution $Z_1, Z_2 \in \mathcal{Z}$ for an MDP with countable state and action spaces,*

$$\bar{d}_\infty \left(\Pi_{\boldsymbol{p}^*, \boldsymbol{\Delta}_\omega} \mathcal{T} F_{Z_1}^{-1, \boldsymbol{p}^*, \boldsymbol{\Delta}_\omega}, \Pi_{\boldsymbol{p}^*, \boldsymbol{\Delta}_\omega} \mathcal{T} F_{Z_2}^{-1, \boldsymbol{p}^*, \boldsymbol{\Delta}_\omega}\right) \leq \gamma \bar{d}_\infty \left(F_{Z_1}^{-1, \boldsymbol{p}^*, \boldsymbol{\Delta}_\omega}, F_{Z_2}^{-1, \boldsymbol{p}^*, \boldsymbol{\Delta}_\omega}\right), \tag{8}$$

where $\mathcal{T}$ denotes the distributional bellman operator, $\bar{d}_k(F_{Z_1}^{-1}, F_{Z_2}^{-1}) := sup_{x,a} W_k(F_{Z_1(x,a)}^{-1}, F_{Z_2(x,a)}^{-1})$ and $W_k(\cdot, \cdot)$ denotes the k-Wasserstein metric.

By Theorem 1, we conclude that $\Pi_{\boldsymbol{p}^*, \boldsymbol{\Delta}_\omega} \mathcal{T}$ have a unique fixed point and the repeated application of $\Pi_{\boldsymbol{p}^*, \boldsymbol{\Delta}_\omega} \mathcal{T}$ converges to the fixed point. With $\bar{d}_k \leq \bar{d}_\infty$, the convergence occurs for all $k \in [1, \infty)$. It ensures that we can obtain a consistent estimator for F_Z^{-1}, denoted as $F_Z^{-1, \boldsymbol{p}^*, \boldsymbol{\Delta}_\omega}$, by minimizing the Huber quantile regression loss defined in (7).

Now we discuss the implementation details of NQDFN. For the support $\boldsymbol{p}^*$ used in this work, we let $p_i^* = i/N$ for $i \in \{1, \cdots, N-1\}$ and $\varepsilon = 0.001$. NDQFN models the state embedding $\psi(x)$ and the τ-embedding $\phi(\tau)$ in the same way as IQN. The baseline function f in (3) consists of two fully-connected layers and a sigmoid activation function, which takes $\psi(x)$ as the input and returns an unconstrained value for $\Delta_0(x, a; \boldsymbol{p}^*)$. The incremental function g shares the same structure with f but uses the ReLU activation instead to ensure the non-negativity of all the N increments $\Delta_i(x, a; \boldsymbol{p}^*)$'s. g takes $\psi(x) \odot \phi(p_i^*)$ and $\phi(p_i^*) - \phi(p_{i-1}^*)$ as input, such that

$$\Delta_{i,\omega}(x, a; \boldsymbol{p}^*) = g(\psi(x) \odot \phi(p_i^*), \phi(p_i^*) - \phi(p_{i-1}^*))_a, \; i = 1, \cdots, N, \tag{9}$$

which to some extent captures the interactions among $\psi(x)$, $\phi(p_i^*)$ and $\phi(p_{i-1}^*)$. Although $\psi(x) \odot \phi(p_i^*) \odot \phi(p_{i-1}^*)$ may be another potential input form, the empirical results show that the combination of $\psi(x) \odot \phi(p_i^*)$ and $\phi(p_i^*) - \phi(p_{i-1}^*)$ is more preferred in practice considering its outperformance in Atari games. More details are provided in Section C of the Supplement file.

3.4 Exploration method for distributional RL

Although exploration is an indispensable field in reinforcement learning, very few methods have been developed for Distributional RL. Some most recent work such as DLTV(Mavrin et al., 2019) is built upon QR-DQN and is not applicable to IQN or FQF. To fill this gap, we design a general exploration approach for DRL algorithms called distributional prediction error (DPE), motivated by the Random Network Distillation (RND) method (Burda et al., 2018).

RND involves two networks which have the same architecture but different network parameters. The target network ω' is randomly initialized and fixed, while the predictor network ω^* is dynamically updated. The prediction error, which measures the L_2 distance between the outputs of the target network and the predictor network, $l(x|\omega')$ and $l(x|\omega^*)$, is defined as follows,

$$i(x) = \|l(x|\omega') - l(x|\omega^*)\|^2. \tag{10}$$

The error term is minimized to distill the predictor network to the target network using all available transitions in the collected sample. The prediction error would be high for an unobserved state than for a frequently visited one. Therefore, $i(x)$ can be treated as the exploration bonus when selecting the optimal action, which encourages the agent to explore unknown states.

Different from Burda et al. (2018), DPE has one more network, the online network ω. The online network ω and the target network ω' use the same initialization, while the predictor network ω^* is separately initialized. The online network and the predictor network are trained together on the sampled data by minimizing the quantile Huber loss (7), while the target network is periodically synchronized by the online network. We use 1-Wasserstein metric $W_1(\cdot,\cdot)$ to measure the distance between the two quantile distributions associated with the predictor network and the target network. DPE and its empirical approximation based on NDQFN are defined as follows,

$$
\begin{aligned}
i^*(x,a) &= \int_{p_0^*}^{p_N^*} W_1\left(P_{\tau,\omega'}(x,a) - P_{\tau,\omega^*}(x,a)\right) d\tau \\
&= \sum_{i=0}^{N-1} \int_{p_i^*}^{p_{i+1}^*} \left| \frac{[\Delta_{i+1,\omega'}(x,a) - \Delta_{i+1,\omega*}(x,a)](\tau - p_i^*)}{p_{i+1}^* - p_i^*} + D_{i,\omega',\omega^*}(x,a) \right| d\tau \\
&= \sum_{i=0}^{N-1} \frac{p_{i+1}^* - p_i^*}{2} H_{i,\omega',\omega^*}(x,a)
\end{aligned}
$$

where

$$
\begin{aligned}
D_{i,\omega',\omega^*}(x,a) &= \sum_{j=0}^{i} [\Delta_{j,\omega'}(x,a) - \Delta_{j,\omega^*}(x,a)] \\
H_{i,\omega',\omega^*}(x,a) &= \begin{cases} |D_{i,\omega',\omega^*}(x,a) + D_{i+1,\omega',\omega^*}(x,a)|, \text{ if } \dfrac{D_{i,\omega',\omega^*}(x,a)}{D_{i+1,\omega',\omega^*}(x,a;)} > 0 \\ \dfrac{D^2_{i,\omega',\omega^*}(x,a) + D^2_{i+1,,\omega',\omega^*}(x,a)}{|D_{i,\omega',\omega^*}(x,a)| + |D_{i+1,\omega',\omega^*}(x,a)|}, \text{ otherwise} \end{cases}.
\end{aligned}
$$

With $i^*(x_t, a_t)$ being the exploration bonus, the optimal action is determined by

$$
a_t = \max_a [Q_\omega(x_t, a) + c_t i^*(x_t, a)], \tag{11}
$$

where c_t is the bonus rate. As might be expected, a more reasonable quantile estimate would highly enhance the exploration efficiency according to our exploration setting. To more clearly demonstrate this point, we compare the performance of DPE based on NDQFN and IQN in Section 4.2.

4 EXPERIMENTS

In this section, we evaluate the empirical performance of the proposed NDQFN with DPE exploration on 55 Atari games using the Arcade Learning Environment (ALE) (Bellemare et al., 2013). The baseline algorithms we compare include IQN (Dabney et al., 2018), QR-DQN (Bellemare et al., 2017b), C51 (Bellemare et al., 2017a), prioritized experience replay (Schaul et al., 2015) and Rainbow (Hessel et al., 2018). The whole architecture of our method is implemented by using the IQN baseline in the Dopamine framework (Castro et al., 2018), which performs slightly worse than reported in Dabney et al. (2018). Note that the whole method could be built upon other quantile value based baselines such as FQF.

Our hyper-parameter setting is aligned with IQN for fair comparison. The size of $\boldsymbol{p}^*$ and the number of sampled quantile levels ,$\boldsymbol{\tau}'$ and $\boldsymbol{\tau}^*$, are 32, i.e. $N = N_1 = N_2 = 32$. We use ϵ-greedy policy with $\epsilon = 0.01$ for training. At the evaluation stage, we test the agent for every 0.125 million frames with $\epsilon = 0.001$. For the DPE exploration, we fix $c_t = 1$ without decay (which is the optimal setting based on experiment results). We run all agents with 200 million frames, on NVIDIA 2080ti 11GB graphics cards. All training curves in this paper are smoothed with a moving average of 10 to improve the readability. With the same hyper-parameter setting, NDQFN with DPE is about 25% slower than IQN at training stage due to the added predictor network.

4.1 NDQFN VS IQN

In this section, we compare NDQFN with its baseline IQN (Dabney et al., 2018) to verify the improvement it achieves by employing the non-decreasing structure. The two methods are fairly compared using n-step updates and without the DPE exploration.

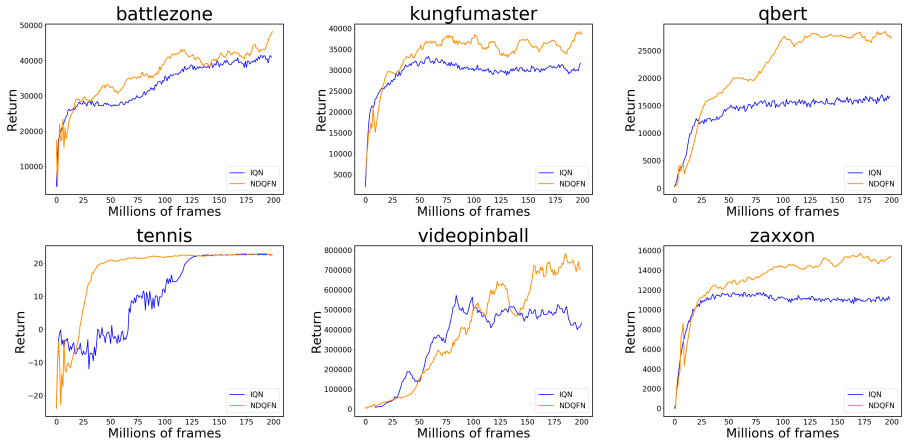

Figure 2: Training curve on Atari games for NDQFN and IQN.

Figure 2 visualizes the training curves of six randomly selected easy explored Atari games. NDQFN significantly outperforms IQN in most cases, including KungFu Master, Qbert, Video Pinball and Zaxxon. Although NDQFN achieves similar final scores with IQN in Battle Zone and Tennis, it learns much faster. This tells that the crossing issue, which usually happens in the early training stage with limited training samples, can be sufficiently addressed by the incremental design in NDQFN to ensure the monoticity of the quantile estimates.

4.2 EFFECTS OF NDQFN ON DPE

In this section, we show how the NDQFN design can help improve the exploration efficiency of DPE compared to using the IQN baseline. We pick three hard explored games, Montezuma Revenge, Private Eye, Venture, and three easy explored games, Ms. Pac-Man, River Raid and Up and Down. The main results are summarized in Figure 3.

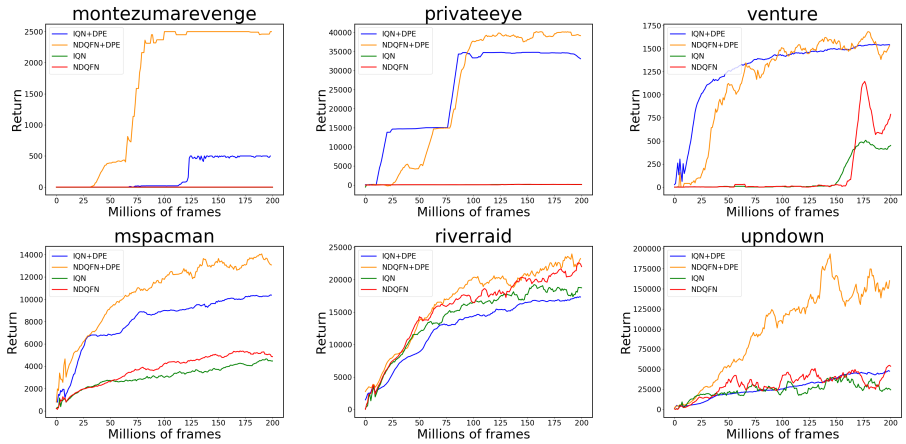

Figure 3: Comparison between NDQFN+DPE and IQN+DPE

As Figure 3 illustrates, DPE is an effective exploration method which highly increases the training performance of both IQN and NDQFN in three hard explored games. In the three easy explored games, DPE still performs slightly better than the baseline methods using ϵ-greedy, which to some

extent demonstrates the stability and robustness of DPE. On the other hand, we find that NDQFN + DPE significantly outperforms IQN + DPE especially in some hard-explored games. This agrees with our conclusion in Section 3.4 that NDQFN obtains a more reasonable quantile estimate by adding the non-decreasing constraint which helps to increase the exploration efficiency.

4.3 Full Atari Results

In this part, we evaluate the performance of the whole network which incorporates NDQFN with DPE in all the 55 Atari games. Following the IQN setting (Dabney et al., 2018), all evaluations start with 30 non-optimal actions to align with previous distributional RL works.

	Mean	Median	>Human
DQN	221%	79%	24
PRIOR.	580%	124%	39
C51	701%	178%	40
RAINBOW	1213%	227%	42
QR-DQN	902%	193%	41
IQN	1112%	218%	39
NDQFN+DPE	**1959%**	**237%**	**42**

Table 1: Mean and median of scores across 55 Atari 2600 games, measured as percentages of human baseline. Scores of previous work are referenced from Yang et al. (2019).

Table 1 presents the mean and median human normalized scores by the seven compared methods across 55 Atari games, which shows that NDQFN with DPE significantly outperforms the IQN baseline and other state-of-the-art algorithms such as Rainbow. In particular, Figure 4 plots the relative improvement of NDQFN+DPE over IQN ((NDQFN-random)/(IQN-random)-1) in testing score for all the 55 Atari games. We observe that our method consistently outperforms the IQN baseline in most situations, especially for some hard games with extremely sparse reward spaces, such as Private Eye and Montezuma Revenge. More comparison results including the training curves for all the 55 games and the raw scores are provided in Sections D and E in the supplement.

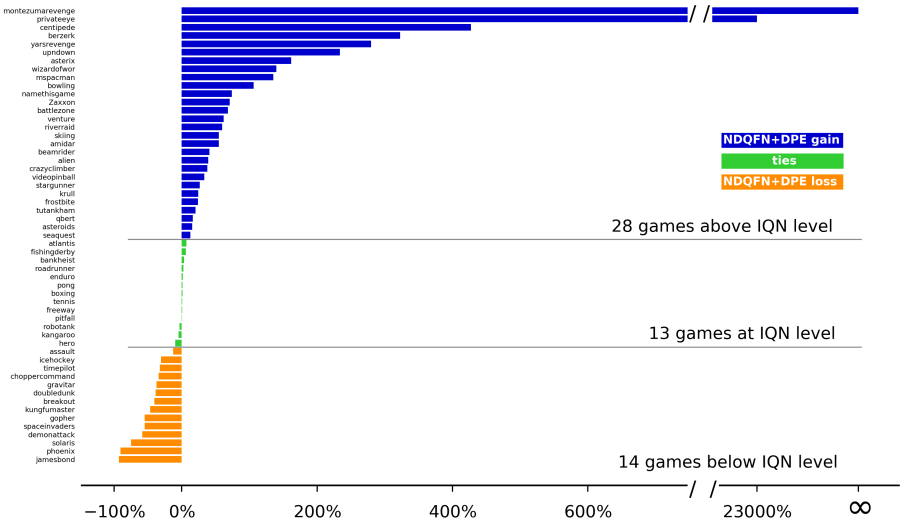

Figure 4: Cumulative rewards performance comparison of NDQFN with DPE and IQN. The bars represent relative gain/loss of NDQFN with DPE over IQN.

REFERENCES

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in neural information processing systems*, pp. 1471–1479, 2016.

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, 2013.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 449–458, 2017a.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458, 2017b.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. *arXiv preprint arXiv:1707.06887*, 2017c.

Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2018.

Pablo Samuel Castro, Subhodeep Moitra, Carles Gelada, Saurabh Kumar, and Marc G. Bellemare. Dopamine: A Research Framework for Deep Reinforcement Learning. 2018. URL `http://arxiv.org/abs/1812.06110`.

Victor Chernozhukov, Iván Fernández-Val, and Alfred Galichon. Quantile and probability curves without crossing. *Econometrica*, 78(3):1093–1125, 2010.

Will Dabney, Georg Ostrovski, David Silver, and Remi Munos. Implicit quantile networks for distributional reinforcement learning. In *International Conference on Machine Learning*, pp. 1096–1105, 2018.

Holger Dette and Stanislav Volgushev. Non-crossing non-parametric estimates of quantile curves. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):609–627, 2008.

Lior Fox, Leshem Choshen, and Yonatan Loewenstein. Dora the explorer: Directed outreaching reinforcement action-selection. In *International Conference on Learning Representations*, 2018.

Hado V Hasselt. Double q-learning. In *Advances in neural information processing systems*, pp. 2613–2621, 2010.

Xuming He. Quantile curves without crossing. *The American Statistician*, 51(2):186–192, 1997.

Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Gheshlaghi Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI*, 2018.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pp. 492–518. Springer, 1992.

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

Koenker. *Quantile regression*. Cambridge University Press, 2005.

Marlos C Machado, Marc G Bellemare, and Michael Bowling. Count-based exploration with the successor representation. *arXiv preprint arXiv:1807.11622*, 2018.

Borislav Mavrin, Shangtong Zhang, Hengshuai Yao, Linglong Kong, Kaiwen Wu, and Yaoliang Yu. Distributional reinforcement learning for efficient exploration. *arXiv preprint arXiv:1905.06125*, 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.

Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. Nonparametric return distribution approximation for reinforcement learning. In *ICML*, 2010.

Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International Conference on Machine Learning*, pp. 2721–2730, 2017.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

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

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

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in neural information processing systems*, pp. 2753–2762, 2017.

Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

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

Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu. Fully parameterized quantile function for distributional reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 6193–6202, 2019.