Exploration in the face of Parametric and Intrinsic Uncertainties

Paper #XXX

ABSTRACT

In distributional reinforcement learning (RL), the estimated distribution of value functions model both the parametric and intrinsic uncertainties. We propose a novel, efficient exploration method for deep RL that has two components. The first is a decaying schedule to suppress the intrinsic uncertainty. The second is an exploration bonus calculated from the upper quantiles of the learned distribution. In Atari 2600 games, our method achieves 483 % average gain in cumulative rewards over QR-DQN.

KEYWORDS

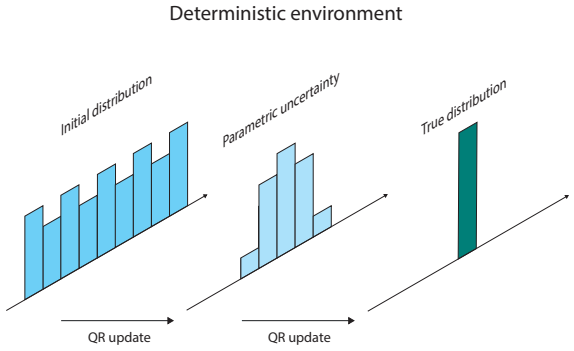
distributional reinforcement learning; exploration

1 INTRODUCTION

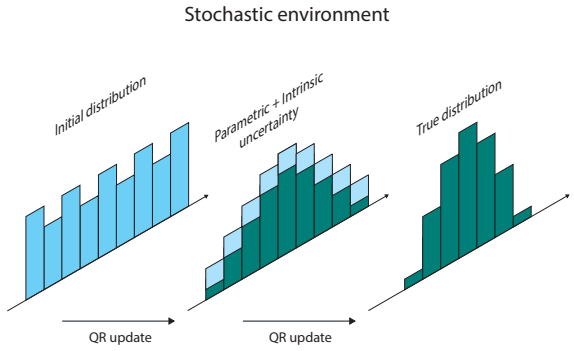
Exploration is a long standing problem in Reinforcement Learning (RL), where *optimism in the face of uncertainty* is one fundamental principle ([20, 29]). Here the uncertainty refers to *parametric uncertainty*, which arises from the variance in the estimates of certain parameters given finite samples. Both count-based methods ([2, 4, 17, 26, 33]) and Bayesian methods ([8, 17, 25]) follow this optimism principle. In this paper, we propose to use distributional RL methods to achieve this optimism.

Different from classical RL methods, where an expectation of value function is learned ([22, 30, 36]), distributional RL methods ([5, 16]) maintain a full distribution of future return. In the limit, distributional RL captures the intrinsic uncertainty of an MDP ([5, 9, 10, 28]). *Intrinsic uncertainty arises from the stochasticity of the environment*, which is parameter and sample independent. However, it is not trivial to quantify the effects of parametric and intrinsic uncertainties in distribution learning. To investigate this, let us look closer at a simple setup of distribution learning. Here we use Quantile Regression (QR) (detailed in Section 2.2), but the example presented here holds for other distribution learning methods. Here the random samples are drawn from any stationary distribution. The initial estimated distribution is set to be the uniform one (left plots). At each time step, QR updates its estimate in an on-line fashion by minimizing some loss function. In the limit the estimated QR distribution converges to the true distribution (right plots). The two middle plots examine the intermediate estimated distributions before convergence in two distinct cases.

Case 1: Figure 1a shows a deterministic environment where the data is generated by a degenerate distribution. In this case, the intermediate estimate of the distribution (middle plot) contains only the information about parametric uncertainty. Here, parametric uncertainty comes from the error in the estimation of the quantiles.



(a) Intrinsic uncertainty.



(b) Intrinsic and parametric uncertainties.

Figure 1: Uncertainties in deterministic and stochastic environments.

Case 2: Figure 1b shows a stochastic environment, where the data is generated by a non-degenerate (stationary) distribution. In this case, the intermediate estimated distribution is the result of both parametric and intrinsic uncertainties.

This example illustrates distributions learned via distributional methods (such as distributional RL algorithms) model the randomness arising from both intrinsic and parametric uncertainties. In this paper, we study how to take advantage of distributions learned by distributional RL methods for efficient exploration in the face of uncertainty.

To be more specific, we use Quantile Regression Deep-Q-Network (QR-DQN, [10]) to learn the distribution of value function. We start with an examination of the two uncertainties and a naive solution that leaves the intrinsic uncertainty unsupressed. We construct a counter example in which this naive solution fails to learn. The intrinsic uncertainty persists and leads the naive solution to favor actions with higher variances. To suppress the intrinsic uncertainty, we apply a decaying schedule to improve the naive solution.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 2019, Montreal, Canada

https://doi.org/doi

One interesting finding in our experiments is that the distributions learned by QR-DQN can be asymmetric. By using the upper quantiles of the estimated distribution ([24]), we estimate an optimistic exploration bonus for QR-DQN.

We evaluated our algorithm in 49 Atari games ([6]). Our approach achieved 483 % average gain in cumulative rewards over QR-DQN. The overall improvement is reported in Figure 9.

In the rest of this paper, we first present some preliminaries of RL Section 2. In Section 3, we then study the challenges posed by the mixture of parametric and intrinsic uncertainties, and propose a solution to suppress the intrinsic uncertainty. We also propose a truncated variance estimation for exploration bonus in this section. In Section 4, we present empirical results in Atari games. Section 5 contains an overview of related work, and Section 6 contains conclusion.

2 BACKGROUND

2.1 Reinforcement Learning

We consider a Markov Decision Process (MDP) of a state space $\mathcal{S}$, an action space $\mathcal{A}$, a reward "function" $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, a transition kernel $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, and a discount ratio $\gamma \in [0, 1)$. In this paper we treat the reward "function" R as a random variable to emphasize its stochasticity. Bandit setting is a special case of the general RL setting, where we usually only have one state.

We use $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ to denote a stochastic policy. We use $Z^\pi(s, a)$ to denote the random variable of the sum of the discounted rewards in the future, following the policy π and starting from the state s and the action a. We have $Z^\pi(s, a) \doteq \sum_{t=0}^{\infty} \gamma^t R(S_t, A_t)$, where $S_0 = s, A_0 = a$ and $S_{t+1} \sim p(\cdot|S_t, A_t), A_t \sim \pi(\cdot|S_t)$. The expectation of the random variable $Z^\pi(s, a)$ is

$$Q^\pi(s, a) \doteq \mathbb{E}_{\pi,p,R}[Z^\pi(s, a)]$$

which is usually called the state-action value function. In general RL setting, we are usually interested in finding an optimal policy π^*, such that $Q^{\pi^*}(s, a) \geq Q^\pi(s, a)$ holds for any (π, s, a). All the possible optimal policies share the same optimal state-action value function Q^*, which is the unique fixed point of the Bellman optimality operator ([7]),

$$Q(s, a) = \mathcal{T}Q(s, a) \doteq \mathbb{E}[R(s, a)] + \gamma \mathbb{E}_{s' \sim p}[\max_{a'} Q(s', a')]$$

Based on the Bellman optimality operator, Watkins and Dayan [36] proposed Q-learning to learn the optimal state-action value function Q^* for control. At each time step, we update $Q(s, a)$ as

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$$

where α is a step size and (s, a, r, s') is a transition. There have been many work extending Q-learning to linear function approximation ([31, 32]). Mnih et al. [22] combined Q-learning with deep neural network function approximators, resulting the Deep-Q-Network (DQN). Assume the Q function is parameterized by a network θ, at each time step, DQN performs a stochastic gradient descent to update θ minimizing the loss

$$\frac{1}{2}(r_{t+1} + \gamma \max_a Q_{\theta^-}(s_{t+1}, a) - Q_\theta(s_t, a_t))^2$$

where θ^- is target network ([22]), which is a copy of θ and is synchronized with θ periodically, and $(s_t, a_t, r_{t+1}, s_{t+1})$ is a transition sampled from a experience replay buffer ([22]), which is a first-in-first-out queue storing previously experienced transitions.

2.2 Quantile Regression

The core idea behind QR-DQN is the Quantile Regression introduced by the seminal paper [18]. This approach gained significant attention in the field of Theoretical and Applied Statistics and might not be well known in other fields. For that reason we give a brief introduction here. Let us first consider QR in the supervised learning. Given data $\{(x_i, y_i)\}_i$, we want to compute the quantile of y corresponding the quantile level τ. linear quantile regression loss is defined as:

$$L(\beta) = \sum_i \rho_\tau(y_i - x_i\beta) \tag{1}$$

where

$$\rho_\tau(u) = u(\tau - I_{u<0}) = \tau|u|I_{u\geq 0} + (1 - \tau)|u|I_{u<0} \tag{2}$$

is the weighted sum of residuals. Weights are proportional to the counts of the residual signs and order of the estimated quantile τ. For higher quantiles positive residuals get higher weight and vice versa. If $\tau = \frac{1}{2}$, then the estimate of the median for y_i is $\theta_1(y_i|x_i) = x_i\hat{\beta}$, with $\hat{\beta} = \arg\min L(\beta)$.

2.3 Distributional RL

Instead of learning the expected return Q, distributional RL focuses on learning the full distribution of the random variable Z directly ([5, 16]). There are various approaches to represent a distribution in RL setting ([3, 5, 9]). In this paper, we focus on the quantile representation ([10]) used in QR-DQN, where the distribution of Z is represented by a uniform mix of N supporting quantiles:

$$Z_\theta(s, a) \doteq \frac{1}{N} \sum_{i=1}^{N} \delta_{\theta_i(s,a)}$$

where δ_x denote a Dirac at $x \in \mathbb{R}$, and each θ_i is an estimation of the quantile corresponding to the quantile level (a.k.a. quantile index) $\hat{\tau}_i \doteq \frac{\tau_{i-1}+\tau_i}{2}$ with $\tau_i \doteq \frac{i}{N}$ for $0 \leq i \leq N$. The state-action value $Q(s, a)$ is then approximated by $\frac{1}{N}\sum_{i=1}^{N} \theta_i(s, a)$. Such approximation of a distribution is referred to as quantile approximation.

Similar to the Bellman optimality operator in mean-centered RL, we have the distributional Bellman optimality operator for control in distributional RL,

$$\mathcal{T}Z(s, a) \doteq R(s, a) + \gamma Z(s', \arg\max_{a'} \mathbb{E}_{p,R}[Z(s', a')])$$
$$s' \sim p(\cdot|s, a)$$

Based on the distributional Bellman optimality operator, Dabney et al. [10] proposed to train quantile estimations (i.e., $\{q_i\}$) via the Huber quantile regression loss ([15]). To be more specific, at time step t the loss is

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{i'=1}^{N}\left[\rho^\kappa_{\hat{\tau}_i}\left(y_{t,i'} - \theta_i(s_t, a_t)\right)\right]$$

where $y_{t,i'} \doteq r_t + \gamma\theta_{i'}\left(s_{t+1}, \arg\max_{a'} \sum_{i=1}^{N} \theta_i(s_{t+1}, a')\right)$ and $\rho^\kappa_{\hat{\tau}_i}(x)$ $\doteq |\hat{\tau}_i - \mathbb{I}\{x < 0\}|\mathcal{L}_\kappa(x)$, where $\mathbb{I}$ is the indicator function and $\mathcal{L}_\kappa$

is the Huber loss,

$$\mathcal{L}_\kappa(x) \doteq \begin{cases} \frac{1}{2}x^2 & \text{if } x \le \kappa \\ \kappa(|x| - \frac{1}{2}\kappa) & \text{otherwise} \end{cases}$$

3 ALGORITHM

In this section we present our method. First, we study the issue of the mixture of parametric and intrinsic uncertainties in the estimated distributions learned by QR approach. We show that the intrinsic uncertainty has to be suppressed in calculating exploration bonus and introduce a decaying schedule to achieve this.

Second, in a simple example where the distribution is asymmetric, we show exploration bonus from truncated variance outperforms bonus from the variance. In fact, we did find that the distributions learned by QR-DQN (in Atari games) can be asymmetric. Thus we combine the truncated variance for exploration in our method.

3.1 The issue of intrinsic uncertainty

A naive approach to exploration would be to use the variance of the estimated distribution as a bonus. We provide an illustrative counter example. Consider a multi-armed bandit environment with 10 arms where each arm's reward follows normal distribution $\mathcal{N}(\mu_k, \sigma_k)$. In each run, means $\{\mu_k\}_k$ are drawn from standard normal. Standard deviation of the best arm is set to 1.0, other arms' standard deviations are set to 5. In the setting of multi-armed bandits, this approach leads to picking the arm a such that

$$a = \arg\max_k \bar{\mu}_k + c\sigma_k \tag{3}$$

where $\bar{\mu}_k$ and σ_k^2 are the estimated mean and variance of the k-th arm, computed from the corresponding quantile distribution estimation.

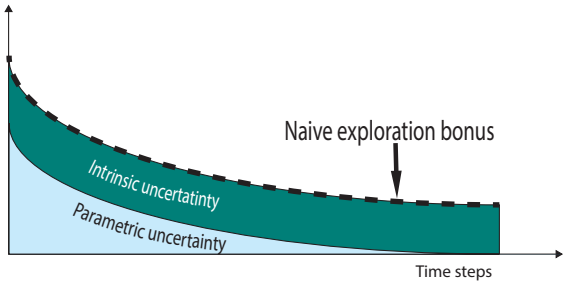
Figure 3 shows that naive exploration bonus fails. Figure 2a illustrates the reason for the failure of naive exploration bonus. The estimated QR distribution is a mixture of parametric and intrinsic uncertainties. Recall, as learning progresses the parametric uncertainty vanishes and the intrinsic uncertainty stays (Figure 2b). Therefore, this naive exploration bonus will tend to be biased towards intrinsic variation, which hurts performance. Note that the best arm has a low intrinsic variation. It is not chosen since its exploration bonus term is much smaller than the other arms as parametric uncertainty vanishes in all arms.

The major obstacle in using the estimated distribution by QR for exploration is the composition of parametric and intrinsic uncertainties, whose variance is measured by the term σ_k^2 in (3). To suppress the intrinsic uncertainty, we propose a decaying schedule in the form of a multiplier to σ_k^2:

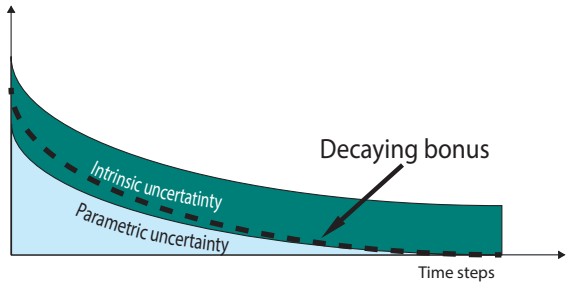
$$a = \arg\max_k \bar{\mu}_k + c_t\bar{\sigma}_k \tag{4}$$

Figure 2b depicts the exploration bonus resulting from the application of decaying schedule. From the classical QR theory ([19]), it is known that the parametric uncertainty decays at the following rate:

$$c_t = c\sqrt{\frac{\log t}{t}} \tag{5}$$



(a) Naive exploration bonus.



(b) Decaying exploration bonus.

Figure 2: Exploration in the face of intrinsic and parametric uncertainties.

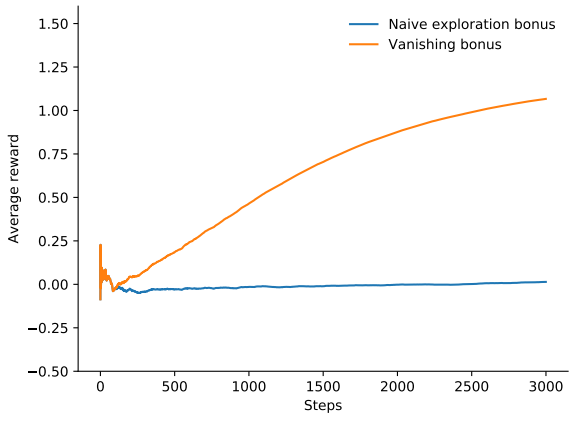


Figure 3: Performance of naive exploration and decaying exploration bonus in the counter example.

where c is a constant factor. We apply this new schedule to the counter example where the naive solution fails. As shown in Figure 3, this decaying schedule significantly outperforms the naive exploration bonus.

3.2 Assymetry and truncated variance

QR has no restriction on the family of distributions it can represent. In fact, the learned distribution can be *asymmetric*, defined by mean ≠ median. From Figure 5 it can be seen that the distribution estimated by QR-DQN-1 is mostly asymmetric. At the end of training, agent achieved nearly maximum score. Hence, the distributions correspond to the near-optimal policy, but they are not symmetric.

In order to account for asymmetry we consider Truncated Variance. For the sake of the argument consider a simple decomposition of the variance of the QR's estimated distribution, not necessarily asymmetric, into the two truncated variances: the *Right Truncated* (σ^2_{rt}) and the *Left Truncated* (σ^2_{lt}) variances [1]. To simplify notation we assume N is even.

$$\begin{aligned}\sigma^2 =& \frac{1}{N}\sum_{i=1}^{N}(\bar{\theta}-\theta_i)^2 \\ =& \frac{2}{N}\sum_{i=1}^{\frac{N}{2}}(\bar{\theta}-\theta_i)^2 + \frac{2}{N}\sum_{i=\frac{N}{2}+1}^{N}(\bar{\theta}-\theta_i)^2 \\ =& \sigma^2_{rt} + \sigma^2_{lt}\end{aligned}$$

If the distribution is symmetric, the *Right Truncated* and *Left Truncated* variances are equivalent. However, in the case of asymmetric distribution σ^2_{rt} and σ^2_{lt} might not be equal. To see this consider a discrete distribution with support $\{-1, 0, 2\}$ and probability atoms $\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$. In this case $\sigma^2_{rt} = \frac{17}{27} \neq \frac{25}{27} = \sigma^2_{lt}$. Furthermore, σ^2_{rt} contains the information about the lower tail variability whereas σ^2_{lt} about upper tail variability. In our example

We should note that Truncated Variance is equivalent to the Tail Conditional Variance (TCV):

$$TCV_x(\theta) = Var(\theta - \bar{\theta} | \theta > x) \tag{6}$$

defined in [35]. For instantiating optimism in the face of uncertainty, the upper tail variability is more relevant than the lower tail one, especially if the estimated distribution is asymmetric [35]. Intuitively speaking, σ^2_{lt} is more optimistic. σ^2_{lt} is biased towards positive rewards. To increase stability, we use the left truncated measure of the variability σ^2_+ based on the median rather than the mean due to its well-known statistical robustness ([14], [12]):

$$\sigma^2_+ = \frac{1}{2N}\sum_{i=\frac{N}{2}}^{N}(\bar{\theta}-\theta_i)^2 \tag{7}$$

where θ_i's are $\frac{i}{N}$-th quantiles. By combining decaying schedule from (5) with σ^2_+ from (7) we obtain a new exploration bonus for picking an action, which we call Decaying Left Truncated Variance (DLTV).

In order to empirically validate our new approach we employ a multi-armed bandits environment with asymmetrically distributed rewards. In each run the means of arms $\{\mu_k\}_k$ are drawn from standard normal distribution. The best arm's reward follow $\mu_k + E[LogNormal(0,1)] - LogNormal(0,1)$. Other arms rewards follow $\mu_k + LogNormal(0,1) - E[LogNormal(0,1)]$. We compare the performance of both exploration methods in another, symmetric environment with rewards following the normal distribution centered at corresponding means (same as the asymmetric environment) with unit variance.

The results are presented in Figure 4. With asymmetric reward distributions, the truncated variance exploration bonus significantly outperforms the naive variance exploration bonus. In addition, the performance of truncated variance is slightly better in the symmetric case.

3.3 DLTV for Deep RL

So far, we introduced the decaying schedule to control the parametric part of the composite uncertainty. Additionally, we introduced a truncated variance to improve performance in environments with asymmetric distributions. These ideas generalize in a straightforward fashion to the Deep RL setting. Algorithm 1 outlines DLTV for Deep RL. Action selection step in line 2 of Algorithm 1 uses exploration bonus in the form of σ^2_+ defined in (7) and schedule c_t defined in (5).

Algorithm 1 DLTV for Deep RL

Input: $w, w^-, (x, a, r, x'), \gamma \in [0, 1)$ ▷ network weights, sampled transition, discount factor

1: $Q(x', a') = \sum_j q_j \theta_j(x', a'; w^-)$
2: $a^* = \arg\max_{a'}(Q(x, a') + c_t\sqrt{\sigma^2_+})$
3: $\mathcal{T}\theta_j = r + \gamma\theta_j(x', a^*; w^-)$
4: $L(w) = \sum_i \frac{1}{N}\sum_j[\rho_{\hat{\tau}_i}(\mathcal{T}\theta_j - \theta_i(x, a; w))]$
5: $w' = \arg\min_w L(w)$

Output: w' ▷ Updated weights of $\theta()$

Figure 6 presents naive and decaying exploration bonus term from DLTV of QR-DQN during training in Atari Pong. Comparison of Figure 6 to Figure 2b reveals the similarity in the behavior of the naive exploration bonus and the decaying exploration bonus. This shows what the raw variance looks like in Atari 2600 game and the suppressed intrinsic uncertainty leading to a decaying bonus as illustrated in Figure 2b.

4 ATARI 2600 EXPERIMENTS

We evaluated DLTV on the set of 49 Atari games initially proposed by [22]. Algorithms were evaluated on 40 million frames[2] 3 runs per game. The summary of the results is presented in Figure 9. Our approach achieved 483 % average gain in cumulative rewards [3] over QR-DQN-1. Notably the performance gain is obtained in hard games such as Venture, PrivateEye, Montezuma Revenge and Seaquest. The median of human normalized performance reported in Figure 7 shows a significant improvement of DLTV over QR-DQN-1. We present learning curves for all 49 games in the Appendix.

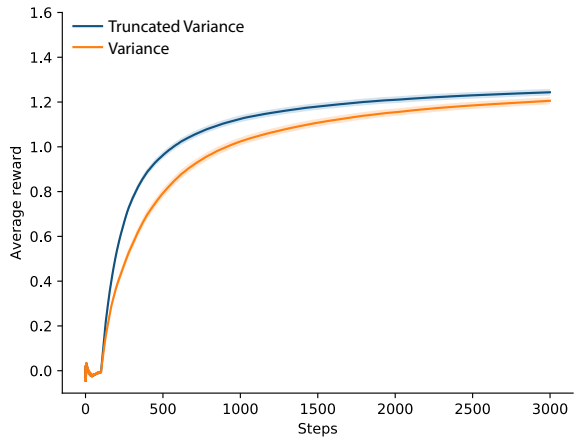
The architecture of the network follows [10]. For our experiments we chose the Huber loss with $\kappa = 1$ [4] in the work by [10] due to its smoothness compared to $L1$ loss of QR-DQN-0. (Smoothness

[1] Note: Right truncation means dropping *left* part of the distribution wrt to the mean

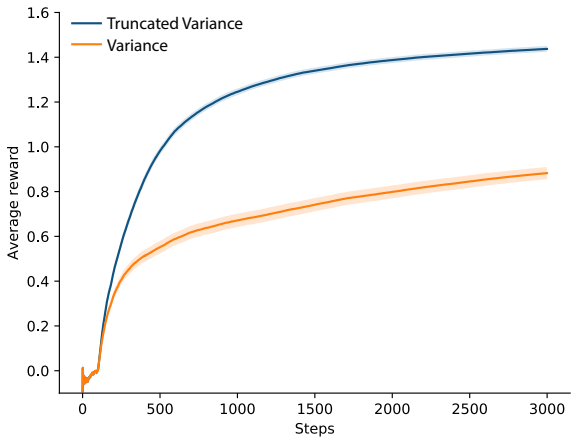
[2] Equivalently, 10 million agent steps.

[3] The cumulative reward is a suitable performance measure for our experiments, since none of the learning curves exhibit plummeting behaviour. Plummeting is characterized by abrupt degradation of performance. In such cases the learning curve drops to the minimum and stays their indefinitely. A more detailed discussion of this point is presented in [21].

[4] QR-DQN with $\kappa = 1$ is denoted as QR-DQN-1



(a) Environment with Symmetric distributions.



(b) Environment with Asymmetric distributions.

Figure 4: Environments with symmetric and asymmetric rewards distributions.

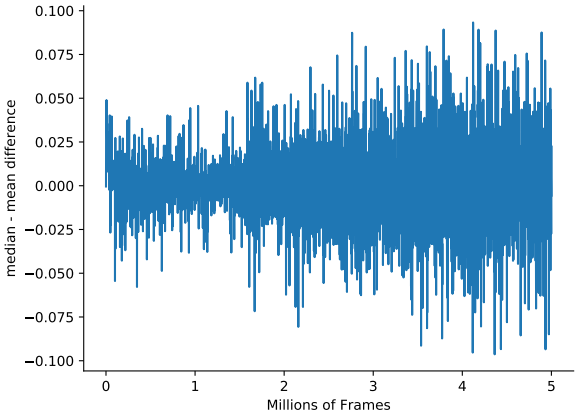


Figure 5: Pong. Empirical distributions of the Q function for a single action obtained from QR-DQN-1 during training.

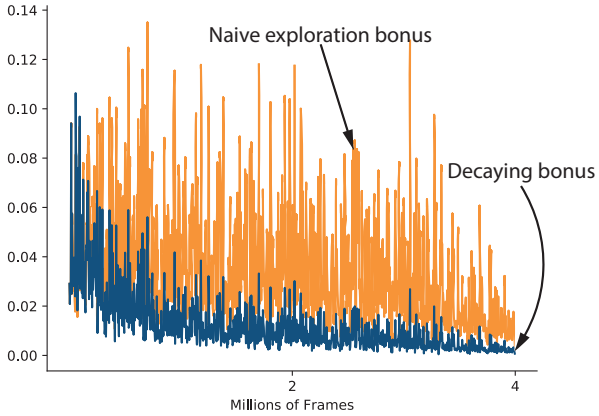


Figure 6: Naive exploration bonus and decaying bonus (as used in DLTV) for Atari 2600 Pong game.

is better suited for gradient descent methods). We followed closely [10] in setting the hyper parameters, except for the learning rate of the Adam optimizer which we set to $\alpha = 0.0001$.

The most significant distinction of our DLTV is the way the exploration is performed. *As opposed to QR-DQN there is no epsilon greedy exploration schedule in DLTV.* The exploration is performed via the σ_+^2 term only (line 2 of Algorithm 1).

An important hyper parameter which is introduced by DLTV is the schedule, i.e. the sequence of multipliers for σ_+^2, $\{c_t\}_t$. In our experiments we used the following schedule $c_t = 50\sqrt{\frac{\log t}{t}}$.

We studied the effect of schedule in the Atari 2600 game Venture. Figure 8 show that constant schedule for DLTV significantly degenerates the performance. This empirical results show that the decaying schedule in DLTV is very important.

5 CARLA EXPERIMENTS

A particularly interesting application of the Distributional RL approach is driving safety. In the classical RL setting the agent only cares about the mean. In Distributional RL the estimate of the whole distribution allows for the construction of the risk-sensitive policies. For that reason we further validate DLTV in CARLA environment which is a 3D self driving simulator.

5.1 Sample efficiency

It should be noted that CARLA is a more visually complex environment than Atari 2600, since it is based on a modern Unreal Engine 4 with realistic physics and visual effects. For the purpose of this study we picked the task in which the ego car has to reach a goal position following predefined paths. In each episode the start

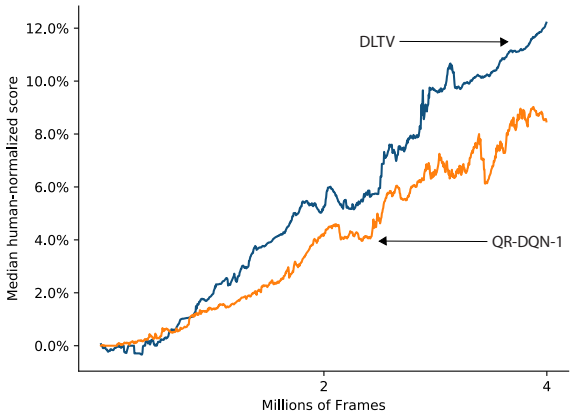


Figure 7: Median human-normalized performance across 49 Atari 2600 games.

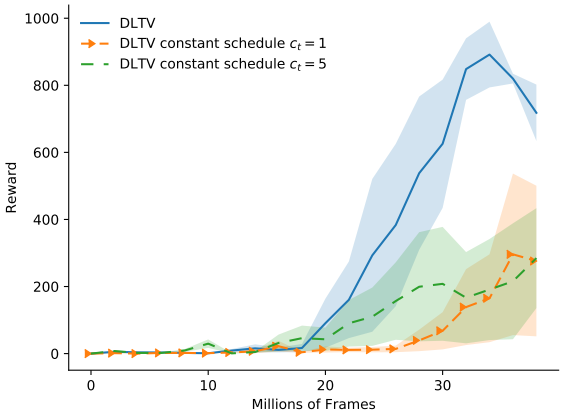


Figure 8: Atari 2600 Venture game. Online training curves for DLTV with decaying schedule and DLTV with constant schedule.

and goal positions are sampled uniformly from a predefined set of locations (around 20). We conducted our experiments in Town 2. We simplified the reward signal provided in the original paper

[11]. We assign reward of -1.0 for any type of infraction and a a small positive reward for travelling in the correct direction without any infractions, i.e. $0.001(distance_t - distance_{t+1})$. The infractions we consider are: collisions with cars, collisions with humans, collisions with static objects, driving on the opposite lane and driving on a sidewalk. The continuous action space was discretized in a coarse grain fashion. We defined 7 actions: 6 actions for going in different directions using fixed values for steering angle and throttle and a no op action. The training learning curves are presented in Figure 10. DLTV significantly outperforms QR-DQN-1 and DQN. Interestingly QR-DQN-1 performs on par with DQN.

Average distance between infractions	$VaR_{90\%}$ or $q_{0.1}$	**Mean**
Opposite lane	**4.55**	1.35
Sidewalk	**None**	None
Collision-static	**None**	3.54
Collision-car	0.70	**1.53**
Collision-pedestrian	**52.33**	16.41
Average collision impact		
Collision-static	**None**	509.81
Collision-car	**497.22**	1078.76
Collision-pedestrian	40.79	**40.70**
Distance, km	**104.69**	98.66
# of evaluation episodes	1000	1000

Table 1: Safety performance in CARLA.

5.2 Driving Safety

A byproduct of Distributional RL is the estimated distribution of $Q(s, a)$. The access to this density allows for different approaches to control. For example Morimura et al. [23] derive risk-sensitive policies based on the quantiles rather than the mean. The reasoning behind such approach is to view quantile as a risk metric. For instance, one particularly interesting risk metric is Value-at-Risk (VaR) which has been in use for a few decades in Financial Industry [27]. Artzner et al. [1] define $VaR_\alpha(X)$ as $Prob(X \leq -VaR_\alpha(X)) = 1 - \alpha$, that is $VaR_\alpha(X) = (1 - \alpha)th$ quantile of X.

It might be easier to understand the idea behind VaR in financial setting. Consider two investments: first investment will lose 1 dollar of its value or more with 10% probability ($VaR_{10\%} = 1$) and second investment will lose 2 dollars or more of its value with 5 percent probability ($VaR_{10\%} = 2$). Second investment is riskier than the first one, that is a risk-sensitive investor will pick an investment with the higher VaR. This same reasoning applies directly to RL setting. Here, instead of investments we deal with actions. risk-sensitive policy will pick the action that has highest VaR. For instance Morimura et al. [23] showed in a simple environment of Cliff Walk the policy maximizing low quantiles yields paths further away from the dangerous cliff.

Risk-sensitive policies are not only applicable to toy domains. In fact risk sensitive policies is a very important research question in self-driving. In that respect CARLA is a non trivial domain where risk-sensitive policies can be thoroughly tested. In [11] authors introduce simple safety performance metric such as average distance travelled between infractions. In addition to this metric we also consider the collision impact. This metric allows one to differentiate policies with the same average distance between infractions. Given the impact is not avoidable, a good policy should minimize the impact.

We trained our agent using DLTV approach and during evaluation we used risk-sensitive policy derived from $VaR(Q(s, a)_{90\%})$ instead of the usual mean. Interestingly, this approach does employ mean-centered RL at all. We benchmark this approach against the agent that uses mean for control. The safety results for the risk-sensitive and the mean agents are presented in Table 1. It can be

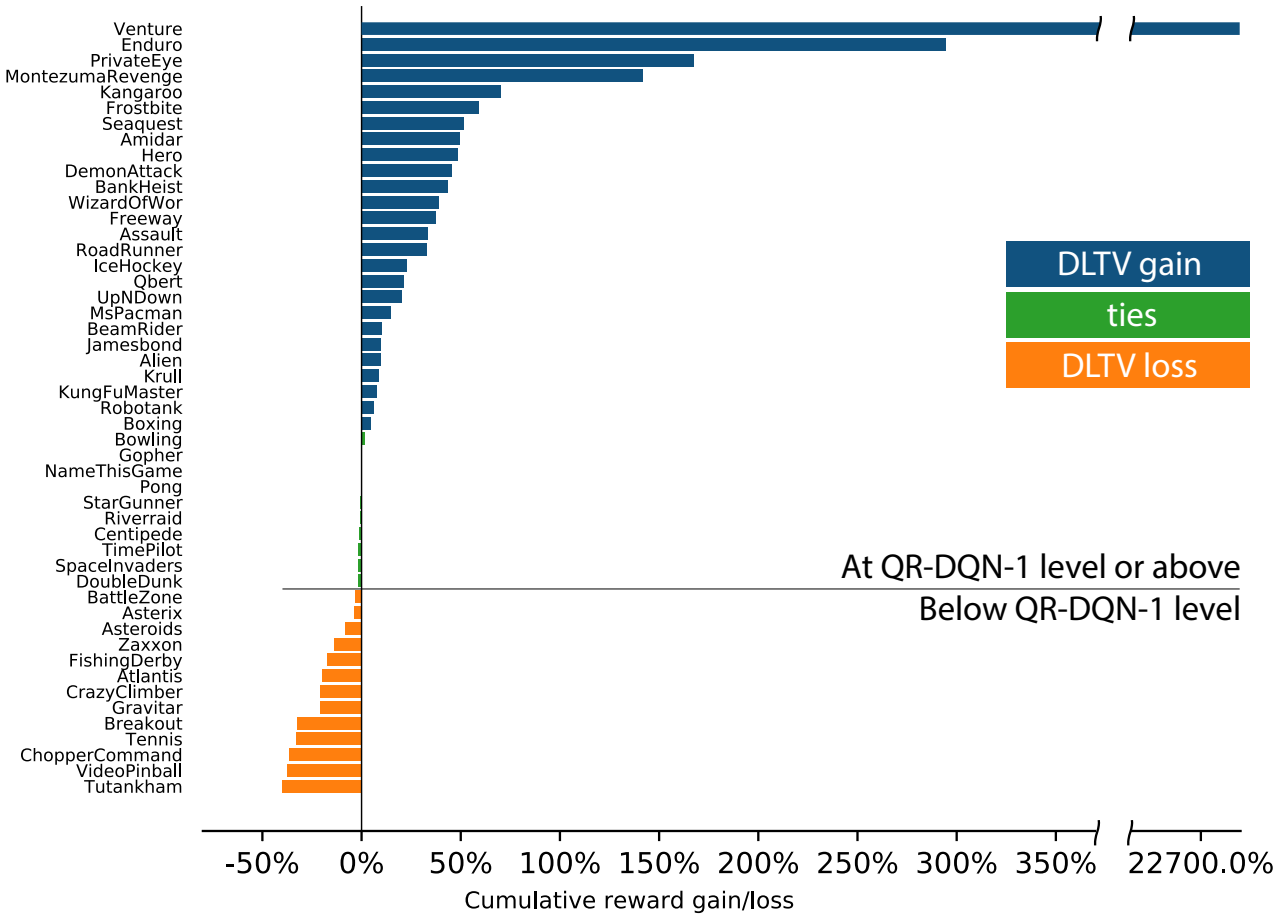


Figure 9: Cumulative rewards performance comparison of DLTV and QR-DQN-1. The bars represent relative gain/loss of DLTV over QR-DQN-1.

seen that risk-sensitive agent significantly improves safety performance across almost all metrics, except for collisions with cars. However, the impact of colliding with cars is twice lower for the risk-sensitive agent.

6 RELATED WORK

Tang and Agrawal [34] combined Bayesian parameter updates with distributional RL for efficient exploration. However, they demonstrated improvement in only simple domains. Zhang et al. [37] generated risk-seeking and risk-averse policies via distributional RL for exploration, making use of both optimism and pessimism of intrinsic uncertainty. To our best knowledge, we are the first to use the parametric uncertainty in the estimated distributions learned by distributional RL algorithms for exploration.

For optimism in the face of uncertainty in deep RL setting, Bellemare et al. [4] and Ostrovski et al. [26] exploited a generative model to enable pseudo-count. Tang et al. [33] combined task-specific features from an auto-encoder with similarity hashing to count high dimensional states. Chen et al. [8] used Q-ensemble to compute variance-based exploration bonus. O'Donoghue et al. [25] used uncertainty Bellman equation to propagate the uncertainty through time steps. Most of those approaches bring in non-negligible computation overhead. In contrast, our DLTV achieves this optimism via distributional RL (QR-DQN in particular) and requires very little extra computation.

7 CONCLUSIONS

Recent advancements in distributional RL, not only established new theoretically sound principles but also achieved state-of-the-art performance in challenging high dimensional environments like Atari 2600. We take a step further by studying the learned distributions by QR-DQN, and discovered the composite effect of intrinsic and parametric uncertainties is challenging for efficient exploration. In addition, the distribution estimated by distributional RL can be asymmetric. We proposed a novel decaying scheduling to suppress the intrinsic uncertainty, and a truncated variance for calculating

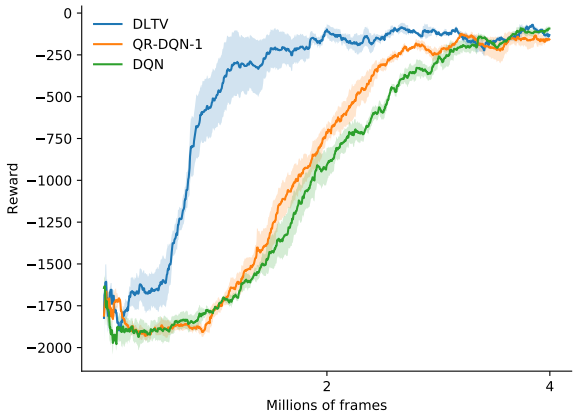


Figure 10: Naive exploration bonus and decaying bonus (as used in DLTV) for CARLA.

exploration bonus, resulting in a new exploration strategy for QR-DQN. Empirical results showed that the our method outperforms QR-DQN (with epsilon-greedy strategy) significantly in Atari 2600. Our method can be combined with other advancements in deep RL, e.g. Rainbow [13], to yield yet better results.

REFERENCES

[1] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. 1999. Coherent measures of risk. *Mathematical finance* 9, 3 (1999), 203–228.

[2] Peter Auer. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* (2002).

[3] Gabriel Barth-Maron, Matthew W Hoffman, David Budden, Will Dabney, Dan Horgan, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. 2018. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617* (2018).

[4] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. 2016. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*.

[5] Marc G Bellemare, Will Dabney, and Rémi Munos. 2017. A distributional perspective on reinforcement learning. *arXiv preprint arXiv:1707.06887* (2017).

[6] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research* (2013).

[7] Richard Bellman. 2013. *Dynamic programming*. Courier Corporation.

[8] Richard Y Chen, Szymon Sidor, Pieter Abbeel, and John Schulman. 2017. UCB Exploration via Q-Ensembles. *arXiv preprint arXiv:1706.01502* (2017).

[9] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. 2018. Implicit Quantile Networks for Distributional Reinforcement Learning. *arXiv preprint arXiv:1806.06923* (2018).

[10] Will Dabney, Mark Rowland, Marc G Bellemare, and Rémi Munos. 2017. Distributional reinforcement learning with quantile regression. *arXiv preprint arXiv:1710.10044* (2017).

[11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. *arXiv preprint arXiv:1711.03938* (2017).

[12] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. 2011. *Robust statistics: the approach based on influence functions*. Vol. 196. John Wiley & Sons.

[13] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. 2017. Rainbow: Combining improvements in deep reinforcement learning. *arXiv preprint arXiv:1710.02298* (2017).

[14] Peter J Huber. 2011. Robust statistics. In *International Encyclopedia of Statistical Science*. Springer, 1248–1251.

[15] Peter J Huber et al. 1964. Robust estimation of a location parameter. *The Annals of Mathematical Statistics* (1964).

[16] Stratton C Jaquette. 1973. Markov decision processes with a new optimality criterion: Discrete time. *The Annals of Statistics* (1973).

[17] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. 2012. On Bayesian upper confidence bounds for bandit problems. In *Artificial Intelligence and Statistics*. 592–600.

[18] Roger Koenker and Gilbert Bassett Jr. 1978. Regression quantiles. *Econometrica: journal of the Econometric Society* (1978), 33–50.

[19] R. Koenker, A. Chesher, and M. Jackson. 2005. *Quantile Regression*. Cambridge University Press. https://books.google.ca/books?id=hdkt7V4NXsgC

[20] Tze Leung Lai and Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics* (1985).

[21] Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. 2017. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *arXiv preprint arXiv:1709.06009* (2017).

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

[23] Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. 2012. Parametric return density estimation for reinforcement learning. *arXiv preprint arXiv:1203.3497* (2012).

[24] John P Mullooly. 1988. The variance of left-truncated continuous nonnegative distributions. *The American Statistician* 42, 3 (1988), 208–210.

[25] Brendan O'Donoghue, Ian Osband, Remi Munos, and Volodymyr Mnih. 2017. The Uncertainty Bellman Equation and Exploration. *arXiv preprint arXiv:1709.05380* (2017).

[26] Georg Ostrovski, Marc G Bellemare, Aaron van den Oord, and Rémi Munos. 2017. Count-based exploration with neural density models. *arXiv preprint arXiv:1703.01310* (2017).

[27] Jorion Philippe. 2001. Value at risk: the new benchmark for managing financial risk. *NY: McGraw-Hill Professional* (2001).

[28] Mark Rowland, Marc G Bellemare, Will Dabney, Rémi Munos, and Yee Whye Teh. 2018. An Analysis of Categorical Distributional Reinforcement Learning. *arXiv preprint arXiv:1802.08163* (2018).

[29] Alexander L Strehl and Michael L Littman. 2005. A theoretical analysis of model-based interval estimation. In *Proceedings of the 22nd International Conference on Machine Learning*.

[30] Richard S Sutton. 1988. Learning to predict by the methods of temporal differences. *Machine Learning* (1988).

[31] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction (2nd Edition)*. MIT press.

[32] Csaba Szepesvári. 2010. *Algorithms for reinforcement learning*. Morgan and Claypool.

[33] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. 2017. # Exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*.

[34] Yunhao Tang and Shipra Agrawal. 2018. Exploration by distributional reinforcement learning. *arXiv preprint arXiv:1805.01907* (2018).

[35] Emiliano A Valdez. 2005. Tail conditional variance for elliptically contoured distributions. *Belgian Actuarial Bulletin* 5, 1 (2005), 26–36.

[36] Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine Learning* (1992).

[37] Shangtong Zhang, Borislav Mavrin, Hengshuai Yao, Linglong Kong, and Bo Liu. 2019. QUOTA: The Quantile Option Architecture for Reinforcement Learning. *In Proceedings of the 33rd AAAI Conference* (2019).