Optimistic Approximate Posterior Sampling for Exploration in Distributional Reinforcement Learning

Alessio Russo
alrusso@student.ethz.ch

Cristian Perez Jensen
cjense@student.ethz.ch

Samuel Cestola
scestola@student.ethz.ch

Abstract

Distributional reinforcement learning (DRL) extends the framework of traditional value-based reinforcement learning to learning the distributions of cumulative rewards, rather than their expected values. Compared to a single point-estimate, a distribution encodes more information. Previous work on DRL has not fully exploited this additional information contained in the learned distributions to drive exploration. In fact, many used standard ϵ-greedy exploration, and others mainly utilized the variance of the distributions. We propose two novel exploration approaches which aim at making the most of the estimated distributions to balance exploration and exploitation. The first method takes the expectation over the upper quantiles of the distribution to optimistically drive exploration. The second method extends the first by employing Bayesian neural networks. Our methods have the advantage that they can be adapted to any distributional representation, which we show by adapting it to a categorical, quantile, and implicit quantile representation. We validate our method on simple grid-based environments and see a much faster convergence than typical ϵ-greedy. Furthermore, we experiment with 18 Atari games. On average, we see an improvement in final score, and, generally, we see a lower variance in return over timesteps, indicating more stable exploration. We release our code under the following repository:
https://github.com/cristianpjensen/drl-optimistic-exploration.

1 Introduction

Distributional reinforcement learning (DRL) is a promising framework for modeling uncertainty and capturing rich representations of value functions in reinforcement learning (RL) problems. Traditional RL algorithms typically focus on estimating scalar value functions, which provide limited information about the uncertainty associated with each action. In contrast, DRL methods seek to model the entire distribution of possible returns for each state-action pair, offering a more nuanced understanding of the environment dynamics.

Exploration in RL is a fundamental challenge, crucial for efficiently discovering optimal policies in unknown environments. Given our interest in DRL, our project aims at designing and evaluating advanced exploration strategies, which leverage the rich information encoded in value distribution. Standard exploration strategies, such as ϵ-greedy, which are typically used [Bellemare et al., 2017, Dabney et al., 2018b,a], lack this possibility. More specifically, we decided to incorporate a notion of "optimism in the face of uncertainty" in the ϵ-greedy framework. The goal is to leverage the value distributions to guide action selection and encourage exploration in regions of the state-action

Foundations of Reinforcement Learning, FS2024. Department of Computer Science, ETH Zürich.

space that offer potentially high rewards, while also discouraging exploration in regions with clearly low returns, thereby accelerating learning and enhancing overall performance. Specifically, we only sample the upper quantiles of the value distributions for estimating the expectation. Furthermore, we investigate another exploration approach, based on approximate posterior sampling using Bayesian neural networks (BNNs), popularly used to tackle multi-armed bandits problems – also conceptually known as Thompson sampling [Thompson, 2012].

2 Related Work

2.1 Deep Q-Network

Q-learning keeps track of a table that maps state-action pairs $(s, a) \in \mathcal{S} \times \mathcal{A}$ to their expected return when performing action a in state s,

$$Q(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \;\middle|\; s_0 = s, a_0 = a\right].$$

Due to state spaces being very large for complex environments, it is not feasible to store the value of every state-action pair in a table. Thus, we need a way to compute the value function using only a finite number of parameters. The DQN was pivotal in advancing reinforcement learning by integrating deep neural networks with Q-learning Mnih et al. [2013, 2015]. It parametrizes a deep neural network to take as input a state s and outputs the value for every action $a \in \mathcal{A}$.

A DQN typically consists of two components: a feature extraction function $\phi_{\boldsymbol{\xi}} : \mathcal{S} \to \mathbb{R}^d$, which outputs d-dimensional features, and a final predictor $f_{\boldsymbol{\zeta}} : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{A}|}$. The final network then computes the Q-value of a state-action pair by $Q_{\boldsymbol{\theta}}(s,a) = f_{\boldsymbol{\zeta}}(\phi_{\boldsymbol{\xi}}(s))_a$ with parameters $\boldsymbol{\theta} = \{\boldsymbol{\xi}, \boldsymbol{\zeta}\}$. Given a transition (s, a, r, s'), these networks learn by minimizing the difference between the current Q-value and the target,

$$Q_{\boldsymbol{\theta}}(s,a) - (r + \gamma Q_{\bar{\boldsymbol{\theta}}}(s', a^\star)), \quad a^\star \in \operatorname*{arg\,max}_{a \in \mathcal{A}} Q(s', a),$$

where $\bar{\boldsymbol{\theta}}$ is only updated periodically to stabilize training.

2.2 Distributional reinforcement learning

DRL strictly generalizes Q-learning by learning the value distribution for each state-action pair, rather than only the expectation. It has gained significant attention lately for its ability to capture uncertainty and learn rich representations of the value distribution, which Q-learning cannot. Bellemare et al. [2017] presented the foundational work in DRL, introducing the concept of value distributional functions and proposing an algorithm to learn a categorical value distribution in an off-policy fashion. In this paper, they introduce the Categorical Temporal Difference Learning (CTDL) algorithm, which generalizes tabular Q-learning. The authors also generalize the DQN to output categorical distributions, called Categorical 51 (C51). It augments the model of DQN by defining the final component to be $f : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{A}| \times N}$, where N is the number of categories. As the name suggests, the number of categories is typically set to $N = 51$. The categorical policy acts greedily according to the expected value of the distributions,

$$a_t \in \operatorname*{arg\,max}_{a \in \mathcal{A}} \sum_{i=1}^{N} p_{\boldsymbol{\theta}}^{(i)}(s_t, a) z^{(i)},$$

where $z^{(i)} = V_{\min} + (i-1) \cdot {}^{V_{\max} - V_{\min}}/_{N-1}$. The model learns by minimizing the cross entropy between the current distribution $Z_{\boldsymbol{\theta}}(s,a)$ and the target distribution $\Phi(r + \gamma Z_{\bar{\boldsymbol{\theta}}}(x', a^\star))$, where Φ is the projection operator that projects probability distributions to their closest categorical distribution.

Given that CTDL has a bounded support $[V_{\min}, V_{\max}]$, this poses a problem. In fact, it could be the case that the environment gives returns that are unsupported by this representation. Quantile Temporal Difference Learning (QTDL) solves this problem by learning the locations of the quantiles, making the support learnable [Dabney et al., 2018b]. Refer to Appendix A for a visual difference between the categorical and quantile representations. The authors generalize the DQN to output quantile distributions, called Quantile Regression Deep Q-Network (QR-DQN). Just like C51, it augments the final component of DQN to be $f : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{A}| \times M}$, where M represents the number of quantiles, with $M = 32$ being the default. The quantile policy also acts greedily according to the expected value of its distribution,

$$a_t \in \arg\max_{a \in \mathcal{A}} \sum_{i=1}^{M} p^{(i)} z_{\boldsymbol{\theta}}^{(i)}(s_t, a),$$

where $p^{(i)} = {}^1\!/_M$. This network minimizes the quantile Huber loss between the current and target distributions.

Implicit Q-Network (IQN) further generalizes QR-DQN by adding the desired quantile as an input of the network [Dabney et al., 2018a]. Specifically, it introduces a component $\psi_{\boldsymbol{\nu}} : [0, 1] \to \mathbb{R}^d$ that takes a quantile and outputs a feature vector. This feature vector is then combined with the feature vector computed by $\phi_{\boldsymbol{\xi}}$ by element-wise multiplication, making the final network compute the quantile value τ of state-action pair (s, a) as $Z_{\boldsymbol{\theta}}(s, a)_\tau = f_{\boldsymbol{\zeta}}(\phi_{\boldsymbol{\xi}}(s) \odot \psi_{\boldsymbol{\nu}}(\tau))_a$. This approach allows the agent to be as fine-grained as required, since it can sample more or less quantiles as needed.

There have also been many recent advancements in the field of DRL algorithms, of which we have done an extensive analysis in Appendix B.

2.3 Existing distributional reinforcement learning Exploration Schemes

The exploration-exploitation trade-off represents a fundamental challenge in RL, wherein an agent must balance exploring new actions to gather information and exploiting known actions to maximize rewards.

Typically, research introducing novel DRL methods neglects the potential of incorporating advanced exploration strategies that leverage the information provided by distributions. Indeed, empirical results usually rely on an ϵ-greedy strategy for exploration [Dabney et al., 2018b, Lin et al., 2019, Nguyen-Tang et al., 2021], with only occasional brief suggestions for potential improvements [Bellemare et al., 2017, Yang et al., 2019, Bellemare et al., 2023].

Nevertheless, there have been developments on more complex exploration strategies. Specifically, these novel solutions often focus on the idea of exploiting the variance of the learned distributions to drive exploration. For instance, Mavrin et al. [2019] propose to implement QR-DQN with a decaying exploration bonus to actions based on the variance of the upper half of the distributions. Similarly, Clements et al. [2019] propose a strategy based on unbiased estimates of the epistemic and aleatoric uncertainty. Moreover, Nikolov et al. [2018] propose a more complex solution, however, this solution is still based solely on using an estimate of the variance. Despite the progress made, all these methods show limitations and opportunities to improve. In fact, they all focus mainly on the idea of variance, glazing over any other information which could be extracted from the distributions.

In contrast, there have also been proposals of strategies that exploit information provided by the full distribution. For example, Zhang and Yao [2019] present an approach based on the concept of "options"', which are groupings of consecutive quantiles. In this method, quantiles are split into disjoint options, each governed by an independent policy. At each timestep, we either follow the previous option, or a different one with a small probability. The main argument for this approach is that solely following the mean of the distribution may lead to suboptimal solutions. Sometimes acting optimistically or pessimistically can result in finding large positive reward events after a period of small negative ones. The greatest drawback of this method is that, at any given timestep, it never exploits the full distribution altogether, and it requires many separate policies.

Additionally, Keramati et al. [2020] propose an exploration strategy similar to the method we will present in further sections, applied to the C51 algorithm. This method employs the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality [Dvoretzky et al., 1956] to provide a bound on the worst-case distance of an empirically determined cumulative distribution function (CDF). This method aligns with the "optimism in the face of uncertainty"-principle and benefits from theoretical guarantees provided by the DKW inequality.

Another proposed general exploration scheme for DRL consists in performing posterior sampling at each step and acting greedily with respect to the average of the sampled value distributions [Tang and Agrawal, 2018]. This approach encourages exploration by adding a term in the objective which pushes towards a high-entropy distribution. In order to make this method analytically tractable, one needs to make the strong assumption that the value distribution can be modeled as a Gaussian distribution. This assumption is questionable, given that it empirically appears that the value distributions – especially for actions that lead to the highest rewards [Bellemare et al., 2023] – are often multi-modal, therefore representing them using simple Gaussians could lead to losing the edge that DRL has over traditional RL methods.

Therefore, based on these examples, we will explore the possibility of developing an exploration scheme for DRL that harnesses all the information contained in the value distributions, does not make strong assumptions on the value distributions, and could benefit from both the "optimism in the face of uncertainty"-principle and the idea of posterior sampling.

2.4 Bayesian neural networks

BNNs provide a probabilistic interpretation of deep learning models by incorporating Bayesian inference principles. Unlike traditional neural networks that generate point estimates for weights, BNNs treat weights as distributions, thereby capturing model uncertainty more effectively. The concept of BNNs was initially proposed by MacKay [1992], who demonstrated the potential of combining Bayesian inference with neural networks to quantify uncertainty in model predictions.

In particular, a BNN typically involves defining a prior distribution over the network weights and updating this prior based on observed data to obtain a posterior distribution. Consequently, this approach allows BNNs to naturally incorporate uncertainty in predictions, making them well-suited for tasks where quantifying uncertainty is crucial. Notably, early work by Neal [1995] laid the theoretical groundwork for BNNs, but practical implementations remained computationally challenging until recent advancements in variational inference methods and Monte Carlo sampling techniques.

One popular method for training BNNs is variational inference, which approximates the true posterior distribution with a simpler, tractable distribution. For example, Graves [2011] applied this approach to neural networks, and Blundell et al. [2015] further improved it by introducing Bayes by Backprop, which enables efficient training of BNNs using stochastic gradient descent. Additionally, Gal and Ghahramani [2016] proposed the use of dropout as a Bayesian approximation, providing a simple yet effective means of estimating model uncertainty in deep networks.

In the context of RL, BNNs have been used to enhance model-based approaches by incorporating uncertainty into the model of the environment. Deisenroth and Rasmussen [2011] employed Gaussian processes to model dynamics and policy uncertainty, and subsequent works have extended these ideas using BNNs for more scalable solutions. For instance, Bayesian Deep Q-Networks integrate BNNs with deep Q-learning to account for uncertainty in Q-value estimates [Azizzadenesheli et al., 2018].

2.5 Bayesian neural networks and Exploration

BNNs offer a promising avenue for addressing the exploration-exploitation trade-off by leveraging their capacity to quantify uncertainty in action-value estimates.

Incorporating BNNs into exploration strategies enables more informed decision-making by guiding exploration based on model uncertainty. Thompson Sampling, a well-known method for exploration in multi-armed bandit problems, can be naturally extended to BNNs. By sampling from the posterior distribution of the network weights, an agent can select actions based on sampled Q-values, effectively incorporating uncertainty into the exploration process [Thompson, 1933, Russo and Van Roy, 2014].

Recent work has explored the integration of BNNs with various exploration strategies in the context of deep reinforcement learning. For example, Osband et al. [2016] proposed Bootstrapped DQN, which uses multiple Q-network heads with randomized priors to approximate posterior samples and enhance exploration. This method can be viewed as a form of approximate posterior sampling, leveraging the uncertainty captured by multiple network heads to drive exploration.

Moreover, recent advances have employed BNNs directly within the DRL framework to improve exploration efficiency. For instance, Riquelme et al. [2018] demonstrated the efficacy of BNNs for contextual bandit problems, showing that BNNs could effectively balance exploration and exploitation by providing a measure of uncertainty for each action. Similarly, Liu et al. [2020] applied BNNs to model-based RL, where uncertainty in the learned dynamics model guides exploration, leading to more efficient policy learning.

Additionally, the principle of "optimism in the face of uncertainty" can also be effectively implemented using BNNs. By leveraging the posterior distribution over Q-values, agents can select actions that maximize the upper confidence bound, thereby encouraging exploration in regions with high uncertainty and potential high rewards. Clements et al. [2019] applied this idea to DRL, where BNNs were used to model the uncertainty in value distributions, guiding exploration based on the epistemic uncertainty.

We note that Tiapkin et al. [2022] explored approximate optimistic posterior sampling in model-based methods using Dirichlet distributions in the model-based setting. While the application context is different, the idea of combining optimism and posterior sampling is similar to what we want to investigate, in spirit.

3 Method

Traditionally, in value-based RL, actions are chosen to maximize the expected discounted future return,

$$a_t \in \operatorname*{arg\,max}_{a \in \mathcal{A}} V_{\boldsymbol{\theta}}(s,a) := \mathbb{E}[Z_{\boldsymbol{\theta}}(s,a)],$$

where Z is the probability density function (PDF) of V, which are both parametrized by parameters $\boldsymbol{\theta}$. By making use of inverse transform sampling [Devroye, 2006], we can rewrite this expectation as

$$a_t \in \operatorname*{arg\,max}_{a \in \mathcal{A}} \mathbb{E}_{\tau \sim \mathrm{Unif}([0,1])} \left[F_{\boldsymbol{\theta}}(s,a)_{\tau}^{-1} \right],$$

where F is the CDF and F^{-1} is its inverse. In contrast, our exploration method consists of choosing the next action based on only the quantiles $\tau \geq \xi$ for some threshold $\xi \in [0,1)$,

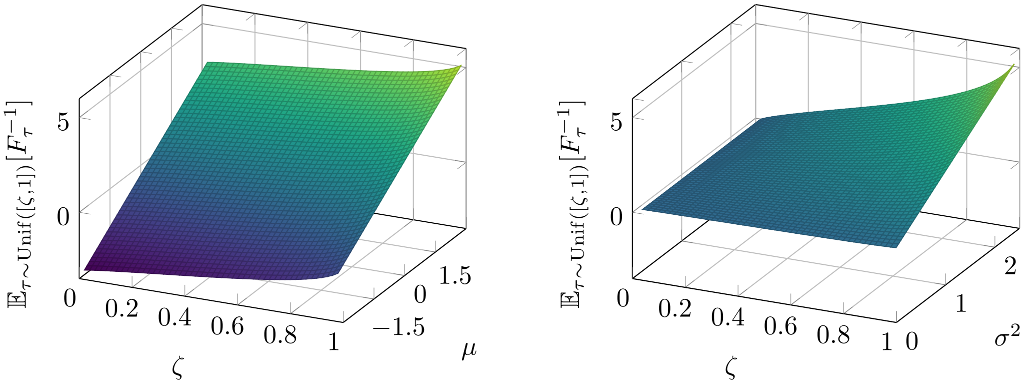
$$a_t \in \operatorname*{arg\,max}_{a \in \mathcal{A}} \mathbb{E}_{\tau \sim \mathrm{Unif}([\xi,1])} \left[F_{\boldsymbol{\theta}}(s,a)_{\tau}^{-1} \right].$$

This puts more emphasis on *potentially* high value events. As shown in Figure 1, as the mean of the distribution increases, the optimistic estimates uniformly increase for all thresholds. Thus, if the model is certain that an action will give a high return, it is more likely to be chosen for any threshold, leading to exploitation. However, as the variance of the distribution increases (and thus the uncertainty), a higher threshold will give markedly more weight to these actions, leading to exploration. Thus, we achieve a natural trade-off between exploration and exploitation, which is governed by a threshold $\xi \in [0,1)$.

Our method can also be seen as discarding information about low-return events. In particular, this is the information in the bottom $1 - \xi$ quantiles. As the threshold ξ increases, more of this information is discarded. Thus, we act more optimistically by acting less pessimistically. In this way, we have a trade-off between exploration and exploitation, where a high ξ_t drives high exploration and a low ξ_t drives high exploitation. However, unlike ϵ-greedy exploration, our method performs informed exploration, where it favors actions with high uncertainty in its return. If there are no actions with high uncertainty or they all have equal uncertainty, it will act greedily according to the mean, as can be seen in Figure 1b.

Considering that our method is defined for the general notion of return distributions, we can apply this method to any distributional representation, such as the ones employed by C51, QR-DQN, and IQN, leading to the models denoted by O-C51, O-QR-DQN, and O-IQN. Since C51 employs a categorical distribution, it is easily turned into its CDF by a cumulative sum. To index its inverse CDF by τ, one only needs to locate the greatest number in the CDF whose value is less than or equal to τ, which can be done in $\mathcal{O}(\log(N))$ time, where N is the number of categories. Moreover, QR-DQN directly represents the distribution by storing the value of specified quantiles, thus we take the mean over values of quantiles greater than τ. Lastly, IQN directly parametrizes the inverse CDF. Therefore, we can sample as many values as desired for quantiles in $[\xi, 1]$.

In more detail, we define a schedule $\{\xi_t \in [0,1)\}_{t=0}^{T}$, where $T \in \mathbb{N}$ is the total number of timesteps. In particular, we use a linear interpolation between the points $[{}^{1}\!/_{2}, {}^{1}\!/_{10}, {}^{1}\!/_{100}]$ at timesteps $[0, 5\mathrm{M}, 20\mathrm{M}]$. We found that such a long schedule is necessary, because each step contains less exploration than a step of ϵ-greedy. We leave fine tuning of this parameter to future work.



(a) Optimistic values under different means with $\sigma^2 = 1$. As the mean increases, the optimistic estimates uniformly increase.

(b) Optimistic values under different variances with $\mu = 0$. As the variance (uncertainty) grows, the more optimistic estimates are proportionally larger than the less optimistic estimates.

Figure 1: Optimistic values under various distribution conditions.

In practice, we found that an initial period of ϵ-greedy exploration is necessary. We believe this is the case because the distributions are not correctly initialized to form well-behaved distributions, which results in the optimistic scheme not acting as intended. In particular, we scheduled the ϵ-greedy exploration as a linear interpolation for the first 100K steps, decaying from 1 to 0. This is a short period, relative to the default period of 1M steps.

To eliminate reliance on ϵ-greedy, we introduce a BNN to the architecture as a replacement, given its intrinsic capability of capturing model uncertainty. We add this only to the final linear layers, not the convolutional layers. BNNs specifies a posterior $q_{\boldsymbol{\phi}}$ over parametrizations of the model. We then choose actions according to the following rule,

$$\begin{aligned}\boldsymbol{\theta} &\sim q_{\boldsymbol{\phi}}\\ a_t &\in \arg\max_{a\in\mathcal{A}} \mathbb{E}_{\tau\sim\text{Unif}([\xi,1])}\left[F_{\boldsymbol{\theta}}(s,a)_\tau^{-1}\right].\end{aligned}$$

Specifically, we parametrize $\boldsymbol{\theta}$ by a normal distribution with a diagonal covariance matrix. Thus, instead of learning $\boldsymbol{\theta}$, the model learns $\boldsymbol{\phi} = \{\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\sigma}^2_{\boldsymbol{\theta}}\}$. We train the model using Bayes by backprop [Blundell et al., 2015].

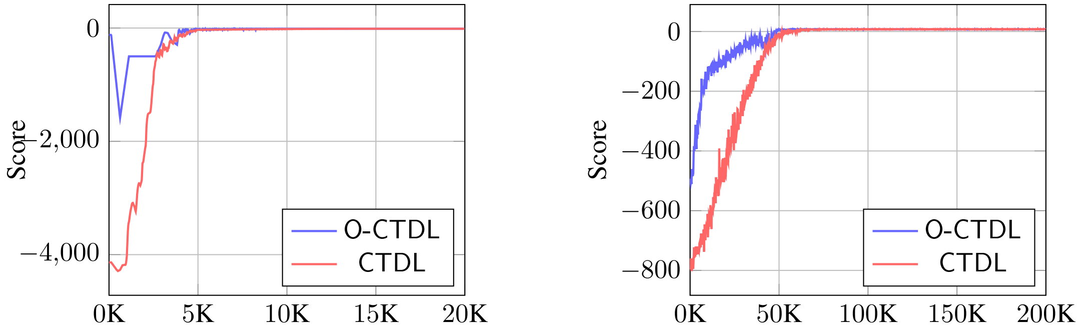
This method is not necessarily limited to DRL and would work for DQNs as well. It adds a layer of exploration because actions with high certainty will have similar outputs in all parametrizations, while actions with high uncertainty will have varied outputs, leading to a natural trade-off between exploration and exploitation. Our hope was that this approach would make the initial period of ϵ-greedy exploration unnecessary.

4 Results

4.1 Gridworlds

We began our experiments with simple, tabular environments, composed of small, discrete state and action spaces. Specifically, we used the Toy Text environments Cliff Walking and Taxi, described in detail in Appendix C, from the Gymnasium library [Towers et al., 2023]. For both environments, we conducted 20 training runs, adjusting the number of steps based on the environments' complexity and the algorithms' convergence rates to the optimal policy.

We evaluated both the CTDL and Optimistic CTDL (O-CTDL) algorithms. The primary difference between these algorithms is the exploration scheme: O-CTDL employs an optimistic approach, whereas CTDL uses ϵ-greedy exploration. The results indicate that optimistic exploration significantly accelerates the achievement of higher returns and stable policies, as shown in Figure 2.



(a) CLIFF WALKING, 5K linear decaying schedule.

(b) TAXI, 50K linear decaying schedule.

Figure 2: Mean return of 20 training runs of Categorical Temporal Difference Learning with ϵ-greedy and optimistic exploration schemes on various environments. We used a learning rate at timestep t equal to $\alpha_t = {}^{1}/_{(1+\#(s,a,t))^{0.6}}$, where $\#(s, a, t)$ is equal to the number of times, until time t, that we visited state s and performed action a.

4.2 Atari

We conducted extensive experiments on 18 Atari games to evaluate the performance of our optimistic exploration methods compared to their ϵ-greedy counterparts. The hyperparameters used in our experiments are detailed in Appendix G. We chose these parameters based on commonly used values in the literature and preliminary experiments to ensure stable training. We evaluate our algorithms by making use of an average return metric at predetermined timesteps, as outlined by Machado et al. [2018]. These values can be found in Table 2. Furthermore, we evaluate the models based on their convergence plots, which can be found in Figures 4 to 6. Specifically, we train each model with 5 different seeds per game and average over them. The convergence plots also show the empirical standard deviation from the mean.

4.2.1 Optimistic Algorithm

We conducted extensive experiments on 18 Atari games to evaluate the performance of our methods. Table 1 summarizes the performance of our optimistic exploration schemes, relative to their ϵ-greedy counterparts.

More specifically, from Table 1 we can see that for some games employing an optimistic approach yielded better relative increase in the mean score, particularly for GRAVITAR and FREEWAY. Interestingly, some games exhibited a relative decrease in score with optimistic exploration, and even when there was a mean score increase across models, individual models did not always show consistent improvements.

First of all, one possible explanation for the observed decrease in some games is that overly optimistic exploration can be detrimental in high-risk scenarios. For instance, in the game TENNIS, an optimistic strategy might encourage aggressive maneuvers without adequate caution, leading to frequent mistakes and lost points. In contrast, a more cautious (or pessimistic) exploration strategy might foster safer and more reliable strategies, resulting in better performance over time.

Additionally, our initial assumption that optimism would show greater improvements with richer return distribution representations, such as IQN, was contradicted by the results in Table 1. Instead, O-QR-DQN showed the best mean relative increase in performance with optimism. Furthermore, considering the number of games where each optimistic model provided a relative increase in mean score (9 for O-C51, 11 for O-QR-DQN, and 8 for O-IQN), O-QR-DQN outperformed O-C51 and O-IQN.

Lastly, the convergence plots in Figures 4 to 6 illustrate that the optimistic exploration scheme sometimes improved convergence time, average performance, and return stability, as evidenced by reduced variance in games like DEMON ATTACK for O-C51, FREEWAY for O-QR-DQN, and MS.

Table 1: Performance increase of optimistic models, relative to their ϵ-greedy versions, on Atari games. We report the relative increase in the mean score of the last 100 episodes of training.

	O-C51	O-QR-DQN	O-IQN	**Mean**
Air Raid	1.4%	−7.2%	−2.8%	**−2.9%**
Alien	−5.9%	36.8%	−0.9%	**10.0%**
Amidar	−3.5%	10.2%	−20.2%	**−4.5%**
Assault	5.4%	−0.2%	5.0%	**3.3%**
Asterix	7.0%	−7.0%	−20.4%	**−6.8%**
Asteroids	6.4%	15.3%	−5.6%	**5.4%**
Boxing	−0.2%	−0.7%	0.0%	**−0.3%**
Breakout	0.2%	0.2%	5.6%	**2.0%**
Demon Attack	22.9%	−7.3%	9.0%	**8.2%**
Freeway	0.0%	37.8%	0.0%	**12.6%**
Gopher	−8.7%	2.9%	−7.2%	**−4.4%**
Gravitar	−6.3%	27.1%	62.8%	**27.9%**
Ice Hockey	−3.0%	42.0%	16.3%	**18.4%**
Ms. Pac-Man	3.7%	−8.5%	29.5%	**8.2%**
Pong	−5.7%	2.1%	−4.1%	**−2.6%**
Space Invaders	−5.5%	3.1%	6.9%	**1.5%**
Tennis	16.0%	−51.7%	−31.1%	**−22.3%**
Tutankham	1.6%	16.6%	0.6%	**6.3%**
Mean	**1.4%**	**6.2%**	**2.4%**	**3.3%**

Pac-Man for O-IQN. However, as shown in the tabular summary, optimism did not consistently improve performance across all games or models.

In summary, the use of optimistic exploration produced mixed results. While it generally improved convergence times and stabilized returns in certain games, its effectiveness varied across different environments and models. The most significant improvements were seen in the O-QR-DQN model. Furthermore, for every model, a mean relative increase in performance was observed, suggesting that the interplay between optimistic exploration and the underlying DRL algorithm warrants further investigation by experimenting with additional environments.

4.2.2 Optimistic Bayesian Algorithm

The initial experiments utilizing BNNs for optimistic exploration yielded poor results. Table 3 and fig. 7 display the average returns at different timesteps for Ms. Pac-Man and Tennis. Although the results fell short of our expectations, further experimentation is necessary to enhance the method.

Moreover, integrating Bayesian methods with deep RL frameworks can introduce stability issues. The stochastic nature of Bayesian inference can lead to high variance in gradient estimates, which might destabilize training. Employing techniques like gradient clipping, adaptive learning rates, or more robust optimization algorithms could help mitigate these issues.

5 Conclusion

In this work, we introduced two novel exploration strategies within the DRL framework. The first method utilizes the "optimism in the face of uncertainty"-principle, while the second method incorporates posterior sampling. Our results demonstrate that the optimistic exploration strategy exhibits promising performance, particularly in environments that require extensive exploration. Furthermore, increased performance can be extracted from the method by further tuning the schedule of the optimism threshold. We leave this for future work. Conversely, the Bayesian algorithms performed below expectations. Despite this, we believe that the exploration method is valid and warrants further investigation to enhance its effectiveness.

Additionally, we discussed recent advancements in DRL. Integrating our exploration strategies into many of the algorithms presented could be a fruitful direction for future research, which is possible for any distributional representation. This potential integration presents an opportunity to further advance the state-of-the-art in DRL exploration techniques.

References

Kamyar Azizzadenesheli, Emma Brunskill, and Anima Anandkumar. Efficient exploration through bayesian deep q-networks. arXiv preprint arXiv:1802.04412, 2018.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In International conference on machine learning, pages 449–458. PMLR, 2017.

Marc G Bellemare, Will Dabney, and Mark Rowland. Distributional reinforcement learning. MIT Press, 2023.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In International conference on machine learning, pages 1613–1622. PMLR, 2015.

William R Clements, Bastien Van Delft, Benoît-Marie Robaglia, Reda Bahi Slaoui, and Sébastien Toth. Estimating risk and uncertainty in deep reinforcement learning. arXiv preprint arXiv:1905.09638, 2019.

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In International conference on machine learning, 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.

Marc Peter Deisenroth and Carl Edward Rasmussen. Pilco: A model-based and data-efficient approach to policy search. Proceedings of the 28th International Conference on Machine Learning (ICML-11), pages 465–472, 2011.

Luc Devroye. Nonuniform random variate generation. Handbooks in operations research and management science, 13:83–121, 2006.

Thang Doan, Bogdan Mazoure, and Clare Lyle. Gan q-learning. arXiv preprint arXiv:1805.04874, 2018.

Jingliang Duan, Yang Guan, Shengbo Eben Li, Yangang Ren, Qi Sun, and Bo Cheng. Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors. IEEE transactions on neural networks and learning systems, 33(11):6584–6598, 2021.

Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. The Annals of Mathematical Statistics, pages 642–669, 1956.

Dror Freirich, Tzahi Shimkin, Ron Meir, and Aviv Tamar. Distributional multivariate policy evaluation and exploration with the bellman gan. In International Conference on Machine Learning, pages 1983–1992. PMLR, 2019.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In International Conference on Machine Learning, pages 1050–1059. PMLR, 2016.

Alex Graves. Practical variational inference for neural networks. arXiv preprint arXiv:1112.6355, 2011.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.

Ramtin Keramati, Christoph Dann, Alex Tamkin, and Emma Brunskill. Being optimistic to be conservative: Quickly learning a cvar policy. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pages 4436–4443, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In International Conference on Machine Learning, pages 5556–5566. PMLR, 2020.

Zichuan Lin, Li Zhao, Derek Yang, Tao Qin, Tie-Yan Liu, and Guangwen Yang. Distributional reward decomposition for reinforcement learning. Advances in neural information processing systems, 32, 2019.

Yang Liu, Zhiyuan Zhang, Xiangyang Li, and Yu Gong. Uncertainty-aware self-training for text classification with few labels. arXiv preprint arXiv:2006.15315, 2020.

Yudong Luo, Guiliang Liu, Haonan Duan, Oliver Schulte, and Pascal Poupart. Distributional reinforcement learning with monotonic splines. In International Conference on Learning Representations, 2021.

Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew 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.

David JC MacKay. A practical bayesian framework for backpropagation networks. Neural computation, 4(3):448–472, 1992.

Borislav Mavrin, Hengshuai Yao, Linglong Kong, Kaiwen Wu, and Yaoliang Yu. Distributional reinforcement learning for efficient exploration. In International conference on machine learning, 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.

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. Human-level control through deep reinforcement learning. nature, 518(7540):529–533, 2015.

Daniel W Nam, Younghoon Kim, and Chan Y Park. Gmac: A distributional perspective on actor-critic framework. In International Conference on Machine Learning, pages 7927–7936. PMLR, 2021.

Radford M Neal. Bayesian Learning for Neural Networks, volume 118. Springer, 1995.

Thanh Nguyen-Tang, Sunil Gupta, and Svetha Venkatesh. Distributional reinforcement learning via moment matching. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 9144–9152, 2021.

Nikolay Nikolov, Johannes Kirschner, Felix Berkenkamp, and Andreas Krause. Information-directed exploration for deep reinforcement learning. arXiv preprint arXiv:1812.07544, 2018.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. arXiv preprint arXiv:1602.04621, 2016.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. arXiv preprint arXiv:1802.09127, 2018.

Daniel J Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. Mathematics of Operations Research, 39(4):1221–1243, 2014.

Yunhao Tang and Shipra Agrawal. Exploration by distributional reinforcement learning. arXiv preprint arXiv:1805.01907, 2018.

Chen Tessler, Guy Tennenholtz, and Shie Mannor. Distributional policy optimization: An alternative approach for continuous control. Advances in Neural Information Processing Systems, 32, 2019.

Steven K Thompson. Sampling, volume 755. John Wiley & Sons, 2012.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika, 25(3-4):285–294, 1933.

Daniil Tiapkin, Denis Belomestny, Daniele Calandriello, Eric Moulines, Remi Munos, Alexey Naumov, Mark Rowland, Michal Valko, and Pierre Ménard. Optimistic posterior sampling for reinforcement learning with few samples and tight guarantees. Advances in Neural Information Processing Systems, 35:10737–10751, 2022.

Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. Gymnasium, March 2023. URL `https://zenodo.org/record/8127025`.

Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu. Fully parameterized quantile function for distributional reinforcement learning. Advances in neural information processing systems, 32, 2019.

Yuguang Yue, Zhendong Wang, and Mingyuan Zhou. Implicit distributional reinforcement learning. Advances in Neural Information Processing Systems, 33:7135–7147, 2020.

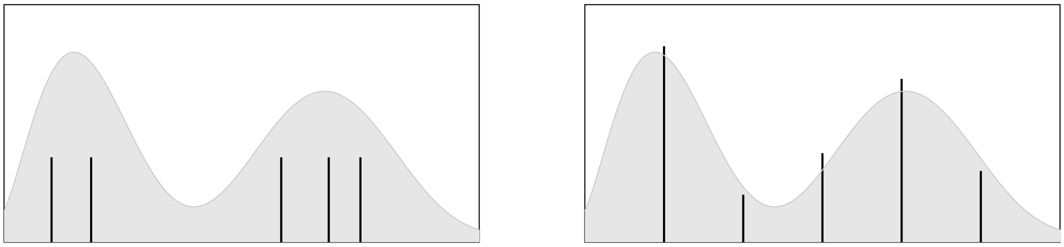
Shangtong Zhang and Hengshuai Yao. Quota: The quantile option architecture for reinforcement learning. In Proceedings of the AAAI conference on artificial intelligence, volume 33, pages 5797–5804, 2019.

Fan Zhou, Zhoufan Zhu, Qi Kuang, and Liwen Zhang. Non-decreasing quantile function network with efficient exploration for distributional reinforcement learning. arXiv preprint arXiv:2105.06696, 2021.

Appendices

A Categorical and Quantile Representations

To further illustrate the difference between the categorical and quantile distributions, we have plotted the difference in Figure 3. As can be seen, the points in the quantile representation all have the same probability, but the locations vary, which make the support of this representation $\mathbb{R}$. The categorical distribution can only vary the probabilities assigned to the fixed locations, thus the support is $[V_{\min}, V_{\max}]$.



(a) Quantile representation, where the probabilities are equally spaced, but the locations vary.

(b) Categorical, where all the values are equally spaced and the probabilities vary.

Figure 3: Comparison of the quantile and the categorical representations.

B Recent Advancements in DRL Algorithms

Recently, many DRL algorithms have emerged. For example, the Fully-parameterized Quantile Function further extends the idea of IQN by parametrizing both the quantile fraction axis and the value axis [Yang et al., 2019]. Furthermore, Nguyen-Tang et al. [2021] present a moment-matching algorithm, which can be interpreted as implicitly matching all orders of moments between a return distribution and its Bellman target. In both papers, they made use of the ϵ-greedy exploration method and did not investigate alternatives.

Moreover, Zhou et al. [2021] propose parametrizing, and thus learning, the difference between successive quantiles, rather than the quantile locations directly. Also, they propose a novel exploration solution, which consists of using three neural networks instead of the two used in DQN: an online network, a target network, and a predictor network. The online and target networks are used as in DQN, while the predictor network is trained on the sampled data, using the quantile Huber loss, as done in QR-DQN. The proposed algorithm empirically approximates the 1-Wasserstein metric between target and predictor networks, which is larger for an unobserved state-action pair than for a frequently visited one. Therefore, this value can be used as a bonus to guide the exploration of unknown states.

A different distribution parametrization has also been presented by Luo et al. [2021]. Their proposal is to learn a distribution as a smooth continuous quantile function represented by monotonic rational-quadratic splines, rather than as a step function or a piece-wise linear function.

Additionally, there has also been an increased interest in the utilization of Generative Adversarial Networks in the DRL framework [Doan et al., 2018, Freirich et al., 2019]. Ideas on how these particular architectures could be exploited for exploration have also been presented by the authors.

DRL is also one of the six extensions of DQN implemented in the Rainbow agent [Hessel et al., 2018], which shows remarkable results. In fact, the algorithm learns the distribution of returns as categoricals, like C51.

Lastly, there have been proposals of implementing existing methods defined in the traditional RL framework to make them compatible with the DRL framework [Lin et al., 2019, Yue et al., 2020]. In addition, there has been much interest in combining the DRL framework with the actor-critic one, typically by modifying the critic to include distributional predictions [Tessler et al., 2019, Kuznetsov et al., 2020, Duan et al., 2021, Nam et al., 2021].

C Gridworld Environments

Cliff Walking is a 4×12 grid world in which the player has to reach the goal in the shortest amount of steps. Both the player starting point and the goal are always located in the same spots. The agent can move in 4 directions (up, down, left, or right), and gets a reward of -1 for each step. The player can also fall off the cliff, which makes the game terminate and gives a reward of -100. The maximum undiscounted return an optimal policy achieves is -13.

Taxi is a 5×5 grid world in which the player has to navigate to passengers, pick them up, and drop them off in a specified location. The taxi starts off at a random square and the passenger at one of four designated locations, which are also the same possible goal positions. The taxi can move in 4 directions (up, down, left, or right), as well as pickup or drop off the passenger (in any location). The agent receives a reward of -1 for each step, -10 for executing "pickup" and "drop-off" actions illegally, and $+20$ points for correctly delivering a passenger. Since the starting locations for the taxi and the passenger are stochastic, the maximum undiscounted return an optimal policy can achieve varies from episode to episode, ranging between 6.5 and 7.5.

D Training Plots

D.1 C51

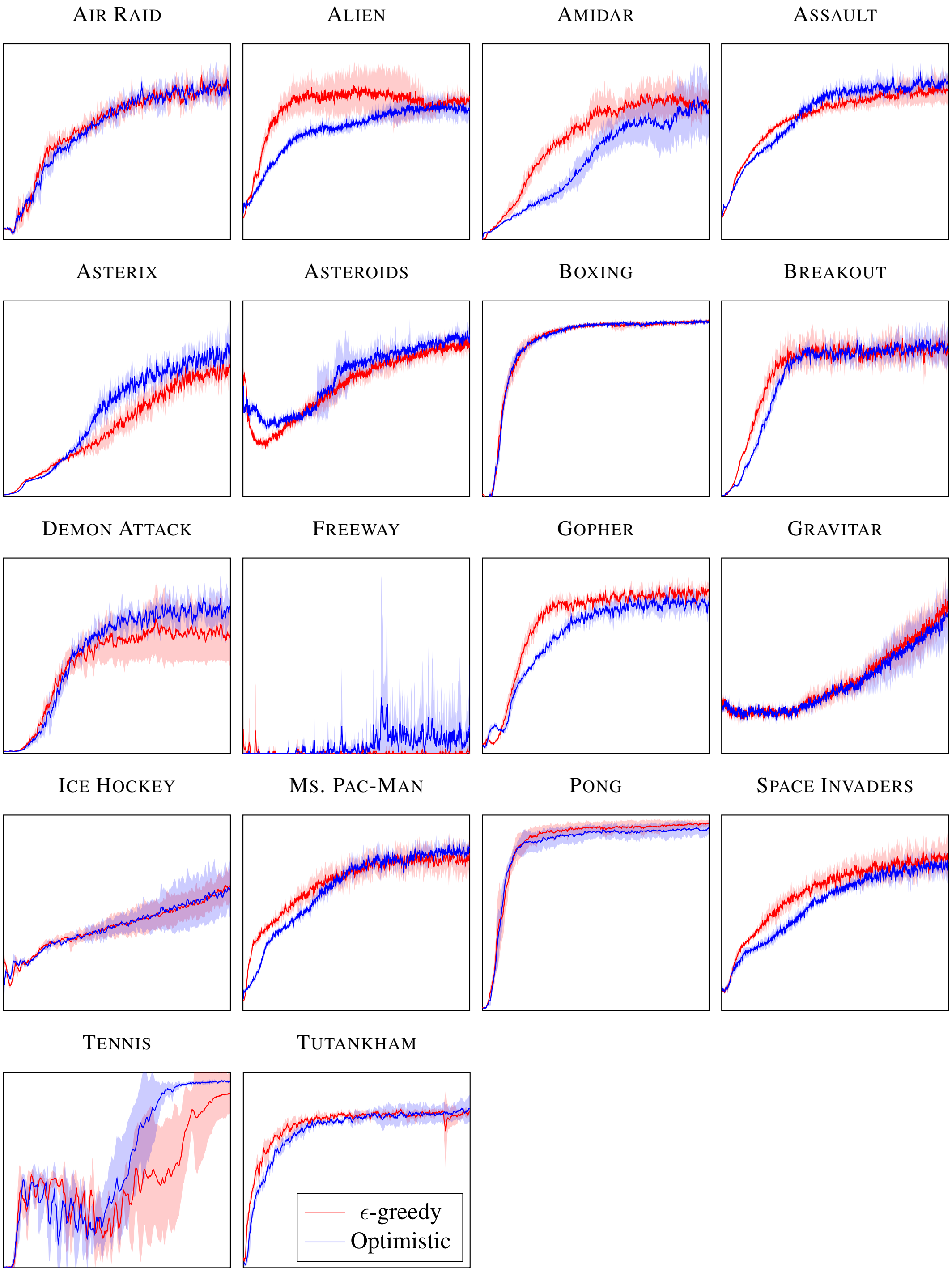


Figure 4: 100-episode moving average of scores during training of Categorical 51 (C51) with ϵ-greedy and optimistic exploration schemes on Atari games.

D.2 QR-DQN

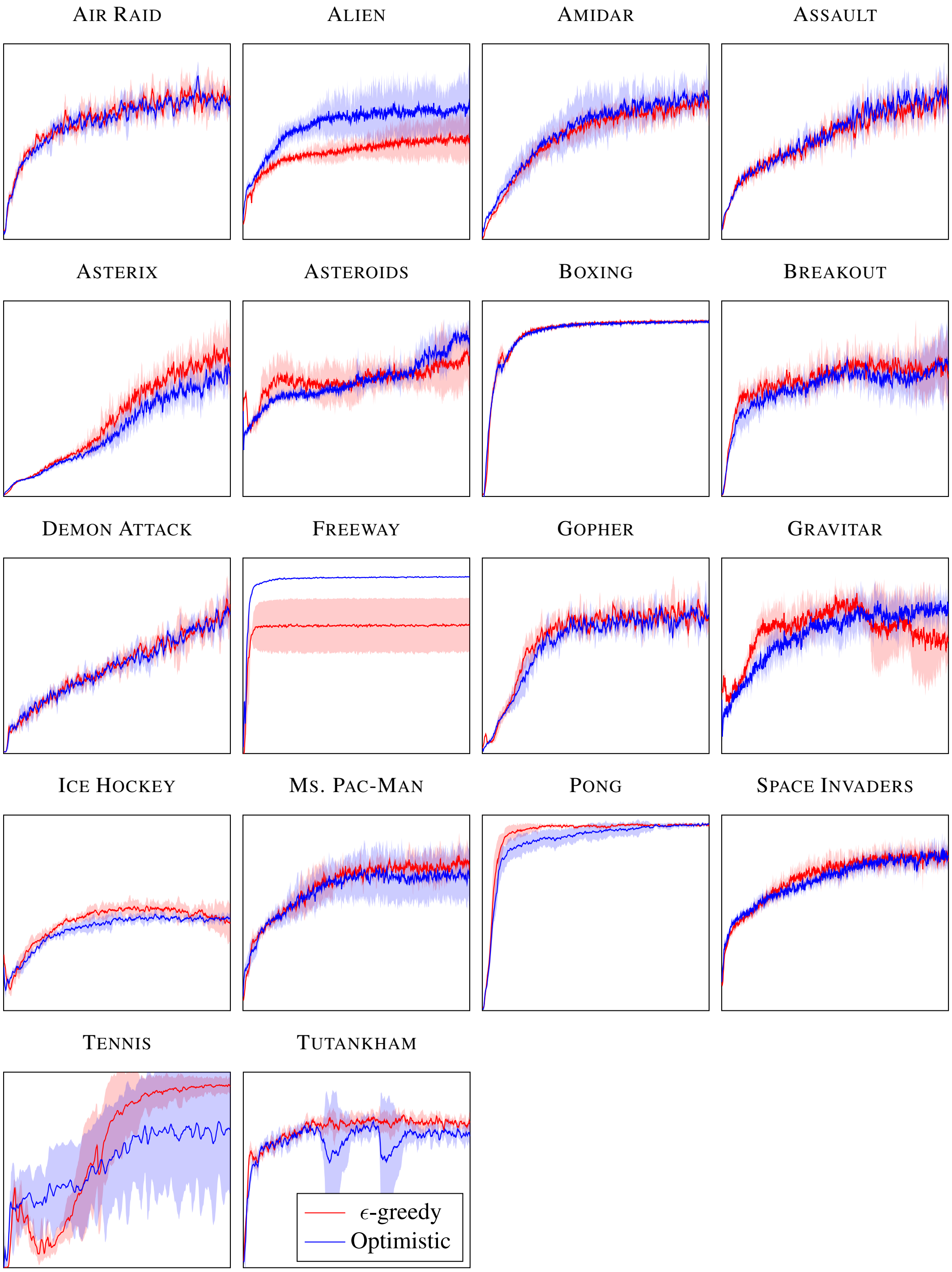


Figure 5: 100-episode moving average of scores during training of Quantile Regression Deep Q-Network (QR-DQN) with ϵ-greedy and optimistic exploration schemes on Atari games.

D.3 IQN

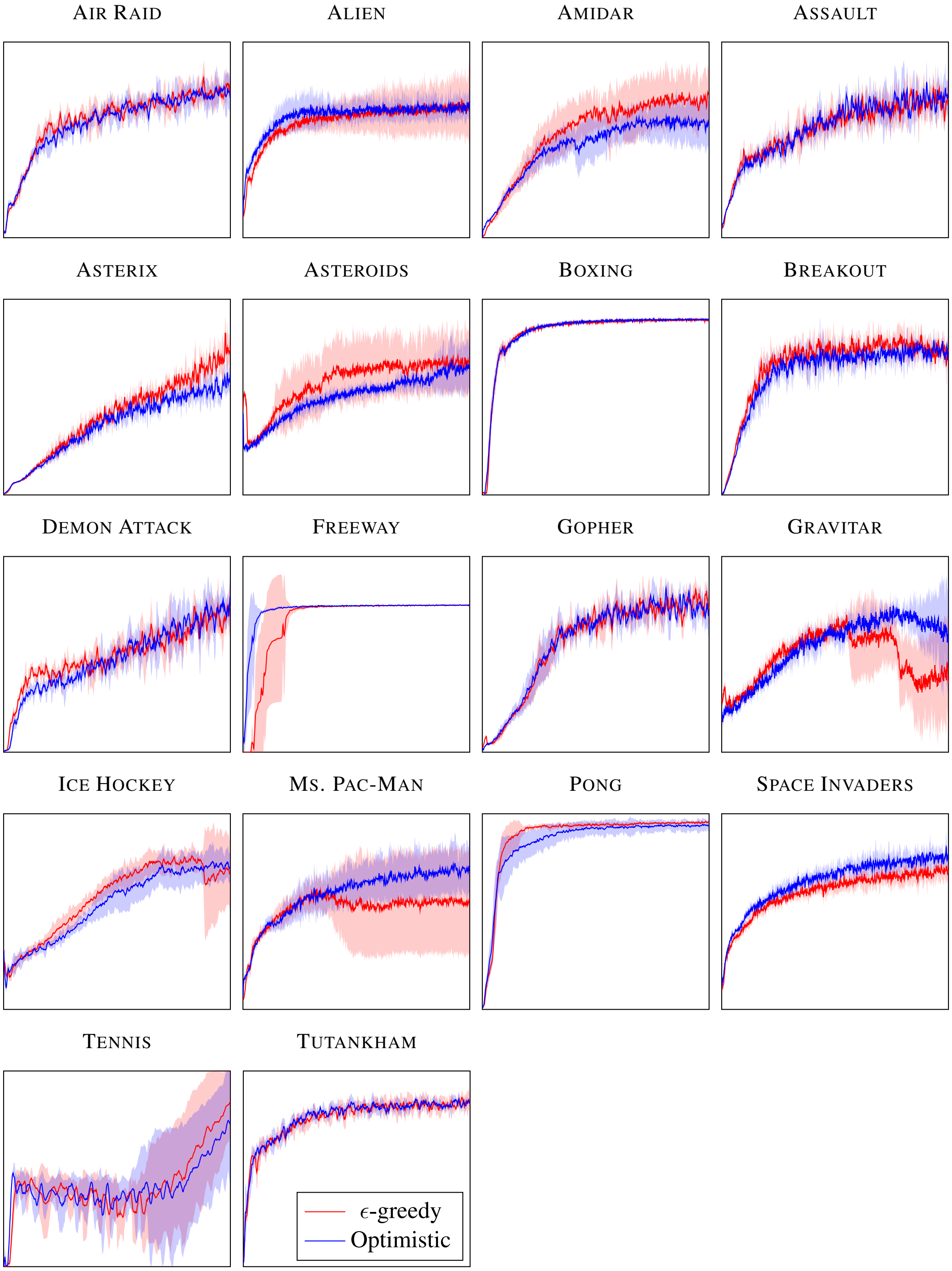


Figure 6: 100-episode moving average of scores during training of Implicit Q-Network (IQN) with ϵ-greedy and optimistic exploration schemes on Atari games.

E Results

Table 2: Results of the models with ϵ-greedy and optimistic exploration schemes. The 100-episode average return at the specified timesteps are reported. The optimistic exploration scheme is denoted by the prefix O-. The values are the mean of 5 training runs per model-game pair. Results reported as outlined in [Machado et al., 2018].

	10M frames		50M frames		100M frames		200M frames	
	ϵ-greedy	Optimistic	ϵ-greedy	Optimistic	ϵ-greedy	Optimistic	ϵ-greedy	Optimistic
Air Raid								
C51	687.0	592.0	5 779.8	5 448.4	7 618.2	7 364.7	8 703.2	8 827.6
QR-DQN	3 938.1	3 654.4	7 026.0	7 335.0	8 404.8	7 862.2	9 465.4	8 781.6
IQN	3 326.4	3 469.4	11 453.8	9 698.1	11 411.4	11 517.5	13 288.9	12 922.6
Alien								
C51	553.6	429.9	1 320.4	974.0	1 398.2	1 105.5	1 285.9	1 209.9
QR-DQN	670.4	781.4	1 088.4	1 448.3	1 190.8	1 566.2	1 266.6	1 732.7
IQN	675.3	823.7	1 106.0	1 190.7	1 178.3	1 236.1	1 272.9	1 261.1
Amidar								
C51	80.5	70.8	529.6	258.2	912.8	651.3	967.9	934.4
QR-DQN	144.5	176.6	487.0	501.3	630.6	656.5	705.5	777.7
IQN	131.3	160.9	620.1	580.1	645.0	715.5	1 000.8	798.4
Assault								
C51	503.3	506.0	1 180.6	1 050.9	1 358.4	1 513.4	1 538.0	1 620.7
QR-DQN	1 137.0	1 096.6	2 184.2	2 118.1	2 766.1	3 006.2	3 485.0	3 477.4
IQN	1 311.4	1 109.0	2 442.4	2 456.1	2 859.7	3 049.6	3 501.9	3 676.2
Asterix								
C51	631.9	504.4	4 626.8	4 427.7	9 175.2	13 110.3	17 833.5	19 081.3
QR-DQN	1 701.7	1 733.1	4 839.4	4 628.0	10 456.5	8 625.3	17 062.0	15 689.6
IQN	2 344.4	2 259.2	9 655.3	8 858.0	16 694.5	13 446.2	28 576.0	22 735.0
Asteroids								
C51	436.2	657.3	604.4	591.9	883.9	985.0	1 126.9	1 199.2
QR-DQN	726.2	712.1	1 125.0	962.0	1 106.2	1 072.8	1 287.6	1 484.8
IQN	506.3	475.0	901.2	858.2	1 240.1	986.0	1 289.9	1 217.3
Boxing								
C51	6.3	3.7	86.4	85.9	93.4	93.1	94.8	94.6
QR-DQN	58.5	56.8	91.2	89.5	93.4	93.4	94.6	93.9
IQN	54.5	55.0	90.2	91.4	93.4	94.4	94.8	94.8
Breakout								
C51	15.7	12.9	176.1	153.7	200.7	191.2	203.5	203.9
QR-DQN	75.5	59.7	118.5	106.0	125.0	124.9	126.2	126.5
IQN	52.8	48.5	205.8	192.6	225.4	220.1	208.4	220.1
Demon Attack								

Table continues on next page

Table continued from previous page

C51	150.1	159.4	4 871.2	4 339.3	7 360.6	8 651.8	7 802.3	9 591.7
QR-DQN	2 062.1	2 136.4	5 087.8	4 789.8	7 112.9	7 442.9	11 918.2	11 050.7
IQN	3 708.5	3 054.0	7 544.8	7 590.9	8 875.4	9 939.8	14 094.1	15 361.8
Freeway								
C51	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
QR-DQN	23.3	31.4	24.0	32.9	24.1	33.0	24.1	33.2
IQN	2.4	26.5	32.6	32.8	33.0	33.1	33.2	33.2
Gopher								
C51	400.5	1 058.2	5 440.5	3 821.5	6 209.6	5 622.0	6 519.4	5 952.8
QR-DQN	873.8	1 204.4	7 342.5	6 745.9	9 035.6	8 020.4	8 718.0	8 966.8
IQN	708.2	882.4	6 953.9	6 640.4	10 124.3	10 046.1	11 259.5	10 445.9
Gravitar								
C51	158.3	176.9	165.9	158.4	269.4	248.9	573.4	537.3
QR-DQN	212.9	167.5	415.7	359.6	480.5	409.4	368.5	468.4
IQN	173.7	153.1	358.8	297.0	451.8	440.5	250.0	407.0
Ice Hockey								
C51	−11.0	−10.2	−5.9	−6.1	−2.4	−2.5	5.3	5.2
QR-DQN	−12.9	−12.0	−2.3	−4.4	0.4	−1.6	−1.9	−1.1
IQN	−11.0	−10.7	−3.3	−5.5	6.0	4.0	7.8	9.1
Ms. Pac-Man								
C51	1 845.4	836.7	3 033.4	2 644.8	3 764.4	3 829.3	4 179.2	4 335.9
QR-DQN	1 837.4	1 644.0	2 941.2	2 870.3	3 623.9	3 531.0	3 895.2	3 562.7
IQN	1 747.0	1 633.2	2 961.8	3 000.1	2 958.0	3 456.9	2 994.0	3 877.8
Pong								
C51	−13.9	−14.5	17.0	15.5	18.4	17.3	19.2	18.1
QR-DQN	4.8	−0.5	18.2	15.5	18.7	17.4	18.7	19.1
IQN	−6.7	−2.4	18.2	15.2	18.8	18.0	19.3	18.5
Space Invaders								
C51	319.2	300.3	754.2	577.5	1 018.5	927.8	1 169.4	1 104.9
QR-DQN	478.9	503.5	690.5	671.9	832.2	792.1	853.1	879.2
IQN	478.1	540.7	755.4	828.2	867.0	945.7	1 007.1	1 076.8
Tennis								
C51	−21.5	−21.6	−3.8	−14.9	−9.7	−2.1	18.8	21.8
QR-DQN	−4.3	−9.7	−15.7	−4.4	14.3	4.8	20.5	9.9
IQN	−6.1	−2.6	−4.0	−8.3	−9.8	−6.6	16.4	11.3
Tutankham								
C51	127.7	95.4	210.9	199.6	221.2	221.6	228.1	231.8
QR-DQN	155.7	145.5	190.2	192.8	212.6	204.2	189.9	221.5
IQN	148.7	149.1	195.4	201.7	215.4	215.4	221.5	222.9

F Bayesian Initial Results

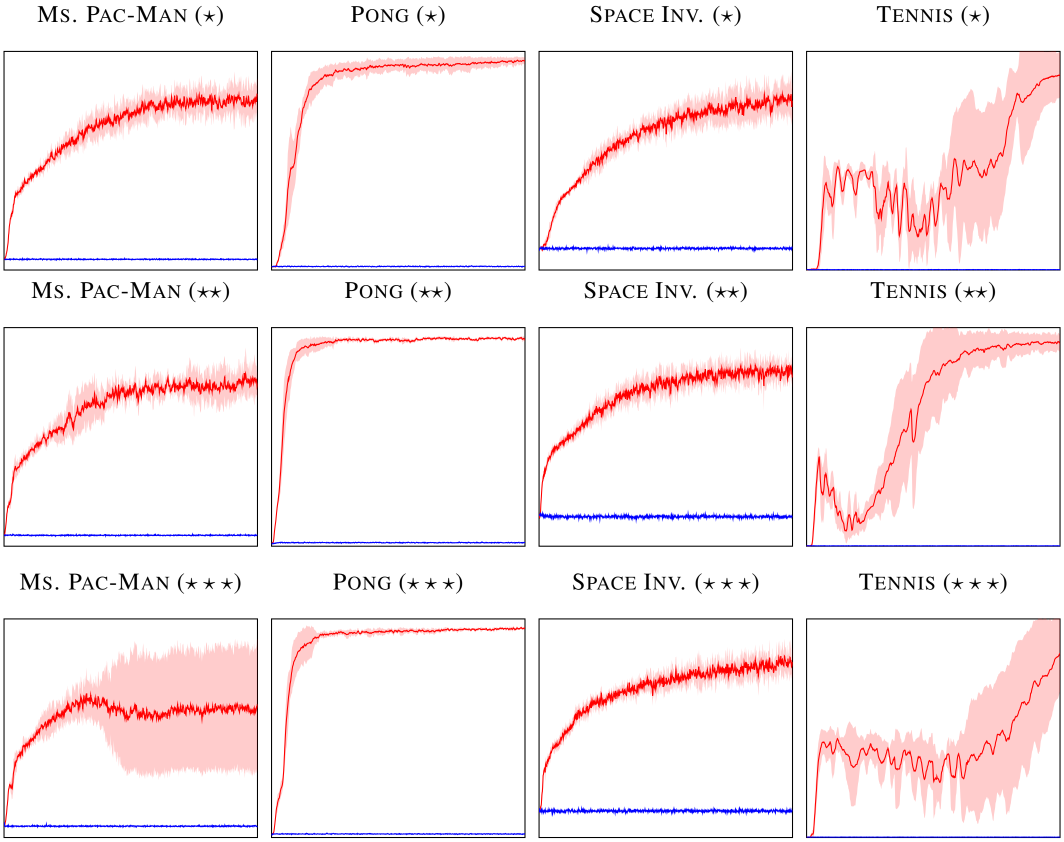


Figure 7: 100-episode moving average of scores during training Bayesian optimistic schemes on Atari games. (⋆): C51, (⋆⋆): QR-DQN, (⋆ ⋆ ⋆): IQN.

Table 3: Results of the models with ϵ-greedy and Bayesian optimistic exploration schemes. The 100-episode average return at the specified timesteps are reported. The optimistic exploration scheme is denoted by the prefix O-. The values are the mean of 5 training runs per model-game pair. Results reported as outlined in [Machado et al., 2018].

	10M frames		50M frames		100M frames		200M frames	
	ϵ-greedy	Bayes Opt	ϵ-greedy	Bayes Opt	ϵ-greedy	Bayes Opt	ϵ-greedy	Bayes Opt
				Ms. Pac-Man				
C51	1 845.4	266.4	3 033.4	254.5	3 764.4	253.5	4 179.2	253.7
QR-DQN	1 837.4	267.7	2 941.2	257.9	3 623.9	248.1	3 895.2	269.3
IQN	1 747.0	242.9	2 961.8	246.8	2 958.0	268.2	2 994.0	240.0
				Pong				
C51	−13.9	−20.3	17.0	−20.3	18.4	−20.4	19.2	−20.3
QR-DQN	4.8	−20.3	18.2	−20.4	18.7	−20.3	18.7	−20.3
IQN	−6.7	−20.3	18.2	−20.4	18.8	−20.4	19.3	−20.3
				Space Invaders				

Table continues on next page

Table continued from previous page

C51	319.2	147.9	754.2	148.2	1 018.5	144.4	1 169.4	145.6
QR-DQN	478.9	146.0	690.5	149.7	832.2	142.7	853.1	144.2
IQN	478.1	150.6	755.4	148.2	867.0	149.7	1 007.1	150.7
				TENNIS				
C51	-21.5	-23.9	-3.8	-23.9	-9.7	-23.9	18.8	-23.9
QR-DQN	-4.3	-23.9	-15.7	-23.9	14.3	-23.9	20.5	-23.9
IQN	-6.1	-23.9	-4.0	-23.9	-9.8	-23.9	16.4	-23.9

G Hyperparameters

Optimizer	ADAM [Kingma and Ba, 2017] with learning rate $^{1}/_{4000}$ and $\epsilon = {}^{0.01}/_{256}$.
Discount factor (γ)	0.99
Minibatch size	256
Replay memory	1 000 000
Replay start size	50 000
Observed history length	4
Update frequency of target network	10 000
Number of parallel environments	32

We chose a batch size of 256 over the usual 32, because it significantly reduces the training time. Furthermore, we decided to use parallel environments with the same goal in mind.

For the ϵ-greedy exploration scheme we used a linear decay during the first 1M transition samples from 1 to $^{1}/_{100}$. For the optimistic exploration scheme, we use a linear decay of ϵ-greedy during the first 100K transition samples from 1 to 0.01, and a linear decay for χ_t from $^{1}/_{2}$ to $^{1}/_{10}$ for the first 5M transition samples and from $^{1}/_{10}$ to $^{1}/_{100}$ for the next 15M.