Principled Exploration via Optimistic Bootstrapping and Backward Induction

Chenjia Bai [1] **Lingxiao Wang** [2] **Lei Han** [3] **Jianye Hao** [4] **Animesh Garg** [5] **Peng Liu** [1] **Zhaoran Wang** [2]

Abstract

One principled approach for provably efficient exploration is incorporating the upper confidence bound (UCB) into the value function as a bonus. However, UCB is specified to deal with linear and tabular settings and is incompatible with Deep Reinforcement Learning (DRL). In this paper, we propose a principled exploration method for DRL through Optimistic Bootstrapping and Backward Induction (OB2I). OB2I constructs a general-purpose UCB-bonus through non-parametric bootstrap in DRL. The UCB-bonus estimates the epistemic uncertainty of state-action pairs for optimistic exploration. We build theoretical connections between the proposed UCB-bonus and the LSVI-UCB in a linear setting. We propagate future uncertainty in a time-consistent manner through episodic backward update, which exploits the theoretical advantage and empirically improves the sample-efficiency. Our experiments in the MNIST maze and Atari suite suggest that OB2I outperforms several state-of-the-art exploration approaches.

1. Introduction

In Reinforcement learning (RL) (Sutton & Barto, 2018), an agent aims to maximize the long-term return by interacting with an unknown environment. To find the optimal policy, the agent is required to sufficiently explore the unknown environment and exploit in depth along the optimal trajectory. Devising efficient exploration algorithms thus becomes an attractive topic in recent years of RL research. The theoretical achievements in RL offer various provably efficient exploration methods in tabular and linear Markov Decision Processes (MDPs) based on the fundamental value iteration algorithm Least-Squares Value Iteration (LSVI). Among these, *optimism in the face of uncertainty* (Auer & Ortner, 2007; Jin et al., 2018) is a principled approach for efficient exploration with well theoretical guarantees. In tabular cases, the optimism-based methods incorporate the Upper Confidence Bound (UCB) into the value function as bonus and attain the optimal worst-case regret (Azar et al., 2017; Jaksch et al., 2010; Dann & Brunskill, 2015). Randomized value function based on posterior sampling chooses actions according to the randomly sampled statistically plausible value function and is known to achieve near-optimal worst-case and Bayesian regrets (Osband & Van Roy, 2017; Russo, 2019). Recently, the theoretical analyses in tabular cases have been extended to linear MDPs where the transition and reward function are assumed to be linear. In linear cases, LSVI-UCB (Jin et al., 2020) has been demonstrated to enjoy a near-optimal worst-case regret using a provably efficient bonus. Randomized LSVI (Zanette et al., 2020) also obtains a near-optimal worst-case regret.

Although the analyses in tabular and linear cases have induced attractive approaches for efficient exploration, it is still challenging in developing a practical exploration algorithm that is essentially suitable for Deep Reinforcement Learning (DRL) (Mnih et al., 2015), which is necessary to achieve human-level performance in large-scale tasks such as Atari games and robotic tasks. A simple evidence is that, in linear case, the bonus in LSVI-UCB (Jin et al., 2020) and nontrivial noise in randomized LSVI (Zanette et al., 2020) are specifically designed for linear models (Abbasi-Yadkori et al., 2011), without generalizations to fit powerful function approximations such as neural networks.

In this paper, we propose a principled exploration method for DRL through Optimistic Bootstrapping and Backward Induction (OB2I). OB2I is an instantiation of LSVI-UCB (Jin et al., 2020) in DRL by using a general-purpose UCB-bonus to provide an optimistic Q-value and a randomized value function to perform temporally-extended exploration. This general-purpose UCB-bonus represents the disagreement of bootstrapped Q-functions (Osband et al., 2016) to measure the epistemic uncertainty of the unknown optimal value function. Importantly, this proposed UCB-bonus can also be theoretically demonstrated to be equivalent to the bonus-term in LSVI-UCB (Jin et al., 2020), when moving back in linear MDPs. In our case, the Q-value plus the general-purpose UCB-bonus is shown to be an optimistic Q^+ function that is higher than the Q-value for

[1]Harbin Institute of Technology, Harbin, China [2]Northwestern University, Evanston, USA [3]Tencent Robotics X [4]Tianjin University [5]University of Toronto, Vector Institute. Correspondence to: Chenjia Bai <bai_chenjia@stu.hit.edu.cn>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

scarcely visited state-action pairs and remains close to the Q-value for frequently visited pairs. Furthermore, we propose an extension of the Episodic Backward Update (EBU) technique (Lee et al., 2019), which we refer to as Backward Induction, to propagate future uncertainties to the estimated action-value function consistently within an episode. The Backward Induction exploits the theoretical advantage of LSVI-UCB and empirically improves the sample-efficiency of exploration significantly.

Compared to existing count-based and curiosity-driven exploration methods (Taiga et al., 2020), OB2I enjoys the following benefits: (i) we utilize intrinsic rewards to produce optimistic value function and also take advantage of bootstrapped Q-learning to perform temporally-consistent exploration, while existing methods do not combine these two principles; (ii) the generalized UCB-bonus measures the disagreement of bootstrapped Q-values, considering long-term uncertainty in an episode rather than the single-step uncertainty used in most bonus-based methods (Pathak et al., 2019; Burda et al., 2019b); (iii) we provide theoretical analysis showing that OB2I is consistent to LSVI-UCB in linear case; (iv) extensive evaluations show that OB2I outperforms several strong exploration approaches in the MNIST maze game and 49 Atari games.

2. Background

In this section, we review bootstrapped DQN (Osband et al., 2016) and LSVI-UCB (Jin et al., 2020) that are closely related to the proposed OB2I method.

2.1. Bootstrapped DQN

Considering an MDP represented as $(\mathcal{S}, \mathcal{A}, T, \mathbb{P}, r)$, where $T \in \mathbb{Z}_+$ is the episode length, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, r is the reward function, and $\mathbb{P}$ is the unknown dynamics. In each timestep, the agent observes the current state s_t and takes an action a_t, and then it receives a reward r_t and the next state s_{t+1}. The action-value function $Q^\pi(s_t, a_t) := \mathbb{E}_\pi\big[\sum_{i=t}^{T-1} \gamma^{i-t} r_i\big]$ represents the expected cumulative reward starting from state s_t by taking action a_t and following policy $\pi(a_t|s_t)$ until the end of the episode. $\gamma \in [0, 1)$ is the discount factor. The optimal value function $Q^* = \max_\pi Q^\pi$, and the optimal action $a^* = \text{argmax}_{a\in\mathcal{A}} Q^*(s, a)$.

Bootstrapped DQN (Osband et al., 2016; 2018) is a non-parametric posterior sampling method, which maintains K estimations of Q-values to represent the posterior distribution of the randomized value function. Bootstrapped DQN uses a multi-head network with a shared representation and K heads. Each head defines a Q^k-function. Bootstrapped DQN diversifies different Q^k by using different random initialization and individual target networks. The loss for training Q^k is

$$L(\theta^k) = \mathbb{E}\Big[\big(r_t + \gamma \max_{a'} Q^k(s_{t+1}, a'; \theta^{k-}) - Q^k(s_t, a_t; \theta^k)\big)^2\Big].$$

The k-th head $Q^k(s, a; \theta^k)$ is trained with its own target network $Q^k(s, a; \theta^{k-})$ with slow-moving parameter θ^{k-}. The agent follows a sampled head Q^k to choose actions in an entire episode, which provides temporally-consistent exploration for DRL.

Algorithm 1 LSVI-UCB in linear MDP

1: **Initialize:** $\Lambda_t \leftarrow \lambda \cdot \mathbf{I}$ and $w_h \leftarrow 0$
2: **for** episode $m = 0$ **to** $M - 1$ **do**
3: Receive the initial state s_0
4: **for** step $t = 0$ **to** $T - 1$ **do**
5: Take action $a_t = \arg\max_a Q_t(s_t, a)$ and observe s_{t+1}
6: **end for**
7: **for** step $t = T - 1$ **to** 0 **do**
8: $\Lambda_t \leftarrow \sum_{\tau=0}^m \phi(x_t^\tau, a_t^\tau)\phi(x_t^\tau, a_t^\tau)^\top + \lambda \cdot \mathbf{I}$
9: $w_t \leftarrow \Lambda_t^{-1} \sum_{\tau=0}^m \phi(x_t^\tau, a_t^\tau)[r_t(x_t^\tau, a_t^\tau) + \max_a Q_{t+1}(x_{t+1}^\tau, a)]$
10: $Q_t(\cdot,\cdot) = \min\{w_t^\top \phi(\cdot,\cdot) + \alpha[\phi(\cdot,\cdot)^\top \Lambda_t^{-1} \phi(\cdot,\cdot)]^{1/2}, T\}$
11: **end for**
12: **end for**

2.2. LSVI-UCB

LSVI-UCB (Jin et al., 2020) uses an optimistic Q-value with LSVI in linear MDP. We denote the feature map of the state-action pair as $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$. Furthermore, the transition kernel and reward function are assumed to be linear in ϕ. The LSVI-UCB algorithm is shown in Algorithm 1. For lines 3-6, the agent executes the policy to collect data in an episode. For lines 7-11, the parameter w_t of Q-function is updated in closed-form by following the regularized least-squares problem as $w_t \leftarrow \text{argmin}_{w\in\mathbb{R}^d} \sum_{\tau=0}^m \big[r_t(s_t^\tau, a_t^\tau) + \max_{a\in\mathcal{A}} Q_{t+1}(s_{t+1}^\tau, a) - w^\top \phi(s_t^\tau, a_t^\tau)\big]^2 + \lambda\|w\|^2$, where m is the total number of episodes, and τ is the episodic index. The least-squares problem has the explicit solution $w_t = \Lambda_t^{-1} \sum_{\tau=0}^m \phi(x_t^\tau, a_t^\tau)\big[r_t(x_t^\tau, a_t^\tau) + \max_a Q_{t+1}(x_{t+1}^\tau, a)\big]$ (line 9), where Λ_t is the Gram matrix. The value function is estimated by $Q_t(s, a) \approx w_t^\top \phi(s, a)$. LSVI-UCB uses an UCB-bonus (Abbasi-Yadkori et al., 2011) in line 10

$$r^{\text{ucb}} = [\phi(s, a)^\top \Lambda_t^{-1} \phi(s, a)]^{1/2} \tag{1}$$

to measure the uncertainty of state-action pairs. The term $u := (\phi^\top \Lambda_t^{-1} \phi)^{-1}$ can be intuitively considered as a pseudo count of the state-action pair in the representation space of ϕ. Thus, the bonus $r^{\text{ucb}} = 1/\sqrt{u}$ represents the uncertainty along the direction of ϕ. By adding the bonus to the Q-value, we obtain an optimistic value function Q^+, which serves as an upper bound of Q to encourage exploration. The bonus in each step is propagated from the end of the episode by the backward update of the Q-value (lines 7-11), which

follows the principle of dynamic programming. Theoretical analysis shows that LSVI-UCB achieves a near-optimal worst-case regret of $\tilde{\mathcal{O}}(\sqrt{d^3T^3L^3})$ with proper selection of α and λ, where L is the total number of steps.

LSVI-UCB (Jin et al., 2020) has been demonstrated to be effective in principled exploration. Nevertheless, developing a practical exploration algorithm for DRL is challenging, since (i) the UCB-bonus utilized by LSVI-UCB is specifically defined for linear MDPs, and (ii) LSVI-UCB utilizes backward update of Q-functions (lines 7-11 in Alg. 1) to aggregate uncertainty. Although the backward update is a standard approach in theoretical analysis of sample-efficient exploration (Shani et al., 2020; Cai et al., 2020; Wang et al., 2019), such an approach is scarcely studied in developing practical exploration algorithm for DRL.

3. Proposed Method

OB2I solves the efficient exploration problem for DRL in the following directions:

- we propose a general-purpose UCB-bonus for optimistic exploration. More specifically, we utilize bootstrapped DQN to construct a general-purpose UCB-bonus, which is theoretically consistent with LSVI-UCB for linear MDPs. We refer to § 3.1 for the details;
- we integrate bootstrapped Q-functions and UCB-bonus into the backward update, which follows the principle of dynamic programming. More specifically, we extend Episodic Backward Update (EBU) (Lee et al., 2019) from standard Q-learning to bootstrapped Q-learning, and we refer this extension to as Bootstrapped EBU (BEBU). We refer to § 3.2 for the details.

3.1. General-Purpose UCB-Bonus

Optimistic exploration uses an optimistic action-value function Q^+ to encourage exploration by adding a bonus term to the standard Q-value. Thus Q^+ serves as an upper bound of the standard Q. The bonus term represents the epistemic uncertainty that results from lacking experiences of the corresponding states and actions. For DRL with deep Q network, it is impractical to derive a closed-form optimistic bonus like (1). Instead, we propose a general-purpose UCB-bonus $\mathcal{B}(s_t, a_t)$ by measuring the disagreement of multiple bootstrapped Q-values $\{Q^k(s_t, a_t)\}_{k=1}^K$ of the state-action pair (s_t, a_t) in a bootstrapped DQN. That is,

$$\mathcal{B}(s_t, a_t) := \sqrt{\frac{1}{K}\sum_{k=1}^{K}\Big(Q^k(s_t, a_t) - \bar{Q}(s_t, a_t)\Big)^2}, \quad (2)$$

where $\bar{Q}(s_t, a_t)$ is the mean of the bootstrapped Q-values. A similar uncertainty measurement was used in Chen et al. (2017). We discuss the difference between Chen et al. (2017) and our algorithm in §4. We surprisingly find that this simple form in (2) is also provably efficient for linear MDPs. Indeed, the following theorem establishes the connection between the general-purpose UCB-bonus defined in (2) and the bonus in LSVI-UCB defined in (1).

Theorem 1. *In linear MDPs, the UCB-bonus $\mathcal{B}(s_t, a_t)$ in OB2I is equivalent to the bonus-term $[\phi_t^\top \Lambda_t^{-1} \phi_t]^{1/2}$ in LSVI-UCB, where $\Lambda_t \leftarrow \sum_{\tau=0}^{m} \phi(x_t^\tau, a_t^\tau)\phi(x_t^\tau, a_t^\tau)^\top + \lambda \cdot \mathbf{I}$, and m is the current episode.*

In Theorem 1, we cast the variance that defines the UCB-bonus of OB2I as the posterior variance of value functions under the Bayesian learning regime. We remark that the bootstrapped distribution of value functions coincides with the posterior under a Bayesian setting where the prior is uninformative (Friedman et al., 2001). We refer to Appendix A for the details and complete statement. Theorem 1 shows that the general-purpose UCB-bonus in (2) is provably efficient and equivalent to bonus-term in LSVI-UCB for linear cases. Importantly, (2) is a general form for arbitrary Q functions such as deep neural networks.

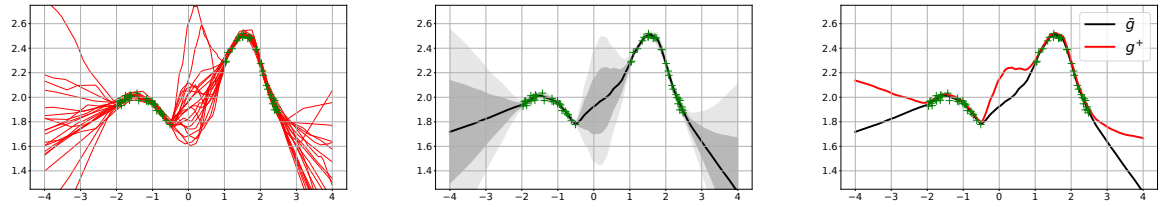
Overall, for general DRL problem, using the UCB-bonus $\mathcal{B}(s_t, a_t)$ in (2) is desirable for the following reasons.

- Bootstrapped DQN is a non-parametric posterior sampling method, that is naturally compatible with deep neural networks (Osband et al., 2019).
- $\mathcal{B}(s_t, a_t)$ quantifies the epistemic uncertainty of (s_t, a_t). Due to the non-convexity nature of optimizing neural network and independency of random initialization, if (s_t, a_t) is scarcely visited, $\mathcal{B}(s_t, a_t)$ obtained via bootstrapped Q-values will tend to be large. Moreover, $\mathcal{B}(s_t, a_t)$ converges to zero asymptotically as the samples increases to infinity.
- $\mathcal{B}(s_t, a_t)$ is computed for batch data sampled from experience replay. This is more efficient than other optimistic methods that change the action-selection scheme in each timestep (Chen et al., 2017; Nikolov et al., 2019) to choose optimistic actions based on uncertainty estimation or information-directed sampling.

The optimistic Q^+ is obtained by summing up $\mathcal{B}(s_t, a_t)$ and the estimated Q-function, which takes the form as

$$Q^+(s_t, a_t) := Q(s_t, a_t) + \alpha\mathcal{B}(s_t, a_t), \quad (3)$$

where α is a tuning parameter. We use a simple regression task with neural networks to illustrate the proposed UCB-bonus, as shown in Figure 1. We use 20 neural networks with the same network architecture to solve the same regression problem. According to Osband et al. (2016), the differences among the outcomes of fitting the 20 neural networks is a result of random initializations. For a given input x, the networks yield different estimations $\{g_i(x)\}_{i=1}^{20}$. It



(a) (b) (c)

Figure 1. Illustration of the general-purpose UCB-bonus in a simple regression task. Green markers indicate there are 60 data points. (a) Regression curves of 20 neural networks. (b) Mean estimation (black curve) and uncertainty measurement (shadow region). (c) The optimistic value (red) and mean value (black).

follows from Figure 1(a) that the estimations $\{g_i(x)\}_{i=1}^{20}$ behave similar in the region with large amount of observations, resulting in small disagreement of the estimations. However, for regions with less observations, the disagreement of the estimations inflates a lot. In Figure 1(b), we illustrate the confidence bound of the regression results $\bar{g}(x) \pm \tilde{\sigma}(g_i(x))$ and $\bar{g}(x) \pm 2\tilde{\sigma}(g_i(x))$, where $\bar{g}(x)$ and $\tilde{\sigma}(g_i(x))$ are the mean and standard deviation of the estimations. The standard deviation $\tilde{\sigma}(g_i(x))$ captures the epistemic uncertainty of regression results. Figure 1(c) shows the optimistic estimation $g^+(x) = \bar{g}(x) + \tilde{\sigma}(g_i(x))$ plus the standard deviation. Clearly, the optimistic estimation g^+ is close to $\bar{g}$ in the region with dense observations, and it is larger than $\bar{g}$ in the region with fewer observations.

In DRL, the bootstrapped Q-functions $\{Q^k(s_t, a_t)\}_{k=1}^K$, estimated by fitting the target Q-function, perform similarly as $\{g_i(x)\}_{i=1}^{20}$ in the above regression task. A higher UCB-bonus $\mathcal{B}(s_t, a_t) := \tilde{\sigma}(Q^k(s_t, a_t))$ indicates a higher epistemic uncertainty of the action-value function with (s_t, a_t). Therefore, Q^+ produces optimistic estimation for novel state-action pairs and behaves similar to the Q-function in areas that are well explored by the agent. Hence, the optimistic estimation Q^+ encourages the agent to explore the potentially informative state-action pairs efficiently.

3.2. Backward Induction of Uncertainty

OB2I adopts BEBU for backward induction when updating the action-value function. BEBU collects a complete trajectory from the replay buffer for each update. Such an approach allows OB2I to infer the long-term effect in an episode for decision making. In contrast, DQN and Bootstrapped DQN sample one-step transitions, which loses the information containing long-term effects.

It has to be mentioned that BEBU is required to propagate future uncertainty to the estimated action-value function consistently via UCB-bonus. For instance, let $t_2 > t_1$ be indices of two steps in an episode. If Q_{t_2} updates after that of Q_{t_1}, then the uncertainty propagated to Q_{t_1} is inconsistent with that propagated to Q_{t_2}.

To integrate the general-purpose UCB-bonus into bootstrapped Q-learning, we propose a novel Q-target by adding the bonus in both the immediate reward and the next-Q value. The proposed Q-target needs to be suitable for BEBU in training. Formally, the Q-target for updating Q^k is defined as

$$y_t^k := \big[r(s_t, a_t) + \alpha_1 \mathcal{B}(s_t, a_t; \theta)\big] + \gamma \big[Q^k(s_{t+1}, a'; \theta^{k-}) + \alpha_2 \mathbb{1}_{a' \neq a_{t+1}} \tilde{\mathcal{B}}^k(s_{t+1}, a'; \theta^-)\big], \tag{4}$$

where $a' = \operatorname{argmax}_a Q^k(s_{t+1}, a; \theta^{k-})$. The choice of a' is determined by the target Q-value without considering the bonus. The immediate reward is added by $\mathcal{B}(s_t, a_t; \theta)$ with a factor α_1, where the bonus $\mathcal{B}$ is computed by bootstrapped Q-network with parameter θ. The next-Q value is added by $\mathbb{1}_{a' \neq a_{t+1}} \tilde{\mathcal{B}}^k(s_{t+1}, a'; \theta^-)$ with factor α_2, where the bonus $\tilde{\mathcal{B}}^k$ is computed by the target network with parameter θ^-. We assign different bonus $\tilde{\mathcal{B}}^k$ of next-Q value to different heads, since the choices of a' are different among the heads. Meanwhile, we assign the same bonus $\mathcal{B}$ of immediate reward for all the heads. We introduce an indicator function $\mathbb{1}_{a' \neq a_{t+1}}$ to control backward update of Q-values. More specifically, in the t-th step, the action-value function Q^k is updated optimistically at the state-action pair (s_{t+1}, a_{t+1}) due to the backward update. Thus, we ignore the bonus of next-Q value in the update of Q^k when a' is equal to a_{t+1}.

We use an example to illustrate the process of backward update. We store and sample the episodic experiences in a replay buffer. Considering an episode containing three time steps, $(s_0, a_0) \to (s_1, a_1) \to (s_2, a_2)$. We thus update the Q-value in the head k in the backward manner, namely $Q(s_2, a_2) \to Q(s_1, a_1) \to Q(s_0, a_0)$ from the end of the episode. We describe the process as follows,

- first, we update $Q(s_2, a_2) \leftarrow r(s_2, a_2) + \alpha_1 \mathcal{B}(s_2, a_2)$. Note that in the last time step, we do not need to consider the next-Q value;
- then, we have $Q(s_1, a_1) \leftarrow [r(s_1, a_1) + \alpha_1 \mathcal{B}(s_1, a_1)] + [Q(s_2, a') + \alpha_2 \mathbb{1}_{a' \neq a_2} \tilde{\mathcal{B}}(s_2, a')]$ by following (4), where $a' = \operatorname{argmax}_a Q(s_2, a)$. Since $Q(s_2, a_2)$ is updated optimistically in the first step, we ignore the bonus-term

$\tilde{\mathcal{B}}$ in next-Q value when $a' = a_2$. The UCB-bonus is augmented by adding $\mathcal{B}$ and $\tilde{\mathcal{B}}$ to the immediate reward and next-Q value, respectively;

- as for $Q(s_0, a_0)$, its update follows the same principle. The optimistic Q-value is $Q(s_0, a_0) \leftarrow [r(s_0, a_0) + \alpha_1 \mathcal{B}(s_0, a_0)] + [Q(s_1, a') + \alpha_2 \mathbb{1}_{a' \neq a_1} \tilde{\mathcal{B}}(s_1, a')]$, where $a' = \operatorname{argmax}_a Q(s_1, a)$.

In practice, the episodic update typically leads to instability in DRL due to strong correlation in consecutive transitions. Hence, we propose a diffusion factor $\beta \in [0, 1]$ in BEBU to prevent such instability as that used in Lee et al. (2019). The Q-value is therefore computed as the weighted sum of the current value and the back-propagated estimation scaled with factor β. We consider an episodic experience that contains T transitions, denoted by $E = \{\mathbf{S}, \mathbf{A}, \mathbf{R}, \mathbf{S}'\}$, where $\mathbf{S} = \{s_0, \dots, s_{T-1}\}$, $\mathbf{A} = \{a_0, \dots, a_{T-1}\}$, $\mathbf{R} = \{r_0, \dots, r_{T-1}\}$ and $\mathbf{S}' = \{s_1, \dots, s_T\}$. We initialize a Q-table $\tilde{\mathbf{Q}} \in \mathbb{R}^{K \times |\mathcal{A}| \times T}$ by $Q(\cdot; \theta^-)$ to store the next-Q values of all the next states $\mathbf{S}'$ and valid actions for K heads. We initialize $\mathbf{y} \in \mathbb{R}^{K \times T}$ to store the Q-target for K heads and T steps. We use bootstrapped Q-network with parameters θ to compute the bonus $\mathbf{B} = [\mathcal{B}(s_0, a_0), \dots, \mathcal{B}(s_{T-1}, a_{T-1})]$ for immediate reward, and use the target network with parameters θ^- to compute bonus $\tilde{\mathbf{B}}^k = [\tilde{\mathcal{B}}^k(s_1, a'_1), \dots, \tilde{\mathcal{B}}^k(s_T, a'_T)]$ for next-Q value in each head, where $a'_t = \operatorname{argmax}_a Q^k(s_t, a; \theta^{k-})$. The bonus vector $\mathbf{B} \in \mathbb{R}^T$ is the same for all Q-heads, while $\tilde{\mathbf{B}} \in \mathbb{R}^{K \times T}$ contains different values for different heads because the choices of a'_t are different.

In the training of head k, we initialize the Q-target in the last step by $\mathbf{y}[k, T-1] = \mathbf{R}_{T-1} + \alpha_1 \mathbf{B}_{T-1}$. We then perform a recursive backward update to get all Q-target values. The elements of $\tilde{\mathbf{Q}}[k, a_{t+1}, t]$ for step t in head k is updated by using its corresponding Q-target $\mathbf{y}[k, t+1]$ with the diffusion factor as follows,

$$\tilde{\mathbf{Q}}[k, a_{t+1}, t] \leftarrow \beta \mathbf{y}[k, t+1] + (1 - \beta) \tilde{\mathbf{Q}}[k, a_{t+1}, t]. \quad (5)$$

Then, we update $\mathbf{y}[k, t]$ in the previous time step based on the newly updated t-th column of $\tilde{\mathbf{Q}}[k]$ as follows,

$$\mathbf{y}[k, t] \leftarrow \big(\mathbf{R}_t + \alpha_1 \mathbf{B}_t\big) + \gamma\big(\tilde{\mathbf{Q}}[k, a', t] + \alpha_2 \mathbb{1}_{a' \neq a_{t+1}} \tilde{\mathbf{B}}[k, t]\big), \quad (6)$$

where $a' = \operatorname{argmax}_a \tilde{\mathbf{Q}}[k, a, t]$. In practice, we construct a matrix $\tilde{\mathbf{A}} = \operatorname{argmax}_a \tilde{\mathbf{Q}}[\cdot, a, \cdot] \in \mathbb{R}^{K \times T}$ to gather all the actions a' that correspond to the next-Q, and then construct a mask matrix $\mathbf{M} \in \mathbb{R}^{K \times T}$ to store the information whether $\tilde{\mathbf{A}}$ is identical to the executed action in the corresponding timestep or not. The bonus of next-Q is the element-wise product of $\mathbf{M}$ and $\tilde{\mathbf{B}}$ with factor α_2. After the backward update, we compute the Q-value of $(\mathbf{S}, \mathbf{A})$ as $\mathbf{Q} = Q(\mathbf{S}, \mathbf{A}; \theta) \in \mathbb{R}^{K \times T}$. The loss function takes the form of $L(\theta) = \mathbb{E}\big[(\mathbf{y} - \mathbf{Q})^2 | (s_t, a_t, r_t, s_{t+1}) \in E,\ E \sim \mathcal{D}\big]$, where the episodic experience E is sampled from replay buffer to perform gradient descent. The gradients of all heads can be computed simultaneously via BEBU. We refer the full algorithm of OB2I to Appendix B.

To summarize, we use BEBU to propagate the future uncertainty in an episode, which is an extension of EBU (Lee et al., 2019). Compared to EBU, BEBU requires extra tensors to store the UCB-bonus for immediate reward and next-Q value, which are integrated to propagate uncertainties. Meanwhile, integrating uncertainty into BEBU needs special design by using the mask. The previous works (Chen et al., 2017; Lee et al., 2020) do not propagate the future uncertainty and, therefore, does not capture the core benefit of utilizing UCB-bonus for the exploration of MDPs. We highlight that OB2I propagates future uncertainty in a time-consistent manner based on BEBU, which exploits the theoretical analysis established by Jin et al. (2020). Only in this way, Q^+ incorporates the epistemic uncertainty across *multiple steps*, so that the greedy action with respect to Q^+ (at the decision stage) performs deep exploration. In contrast, separating the bonus function from the bootstrapping process (i.e., only using it at the decision stage) fails to propagate uncertainty. The backward update also empirically improves the sample-efficiency significantly by allowing bonuses and delayed rewards to propagate through transitions of a complete episode.

3.3. Comparison with LSVI-UCB

We remark that both LSVI-UCB and OB2I constructs the confidence interval of value functions based on the frequentist approaches. Specifically, LSVI-UCB constructs the confidence intervals explicitly based on the linear model, whereas OB2I constructs the confidence interval based on the non-parametric bootstrapped approach. In OB2I, we adopt Bootstrapped Q-values to calculate the standard deviation of Q-functions with neural network parameterization, which coincides with the bonus in LSVI-UCB on linear MDPs. When the sample size increases, the distribution of bootstrapped Q-values converges asymptotically to the posterior under a Bayesian setting where the prior is uninformative (Friedman et al., 2001). Hence, in Theorem 1, we use the Bayesian setting as a simplification to motivate our algorithm while this is not necessary. A recent approach also uses a similar way to motivate the worst-case regret of randomized value functions (Russo, 2019).

From an empirical perspective, LSVI-UCB requires strict linear assumption in the transition dynamics and value function. To the opposite, OB2I uses a non-parametric form and the general UCB-bonus works for arbitrary Q-function types such as deep neural networks. In OB2I, the neural networks can be updated by gradient descent using batch episodic trajectories sampled from the replay buffer in each training step. However, in LSVI-UCB, all historical samples

have to be used to update the Q-function and calculate the confidence bonus in each training step, since the posterior matrix Λ relies on the update-to-date representation ϕ which is varies as the training proceeds. As a consequence, OB2I is much more sample-efficient empirically. Moreover, in LSVI-UCB, the target Q-function is updated in each iteration, whereas in OB2I, the target-network is updated less frequent. Similar empirical tricks are commonly used in most existing off-policy DRL algorithms (Osband et al., 2016; Lillicrap et al., 2015; Fujimoto et al., 2018).

4. Related Work

We discuss a number of closely related approaches in this section and choose the most important ones to compare in our experiment. One practical principle for exploration in DRL is maintaining the epistemic uncertainty. Epistemic uncertainty comes from the unawareness of the environments, and it decreases as the exploration proceeds. Bootstrapped DQN (Osband et al., 2016; 2018) samples Q-values from the randomized value functions to encourage exploration through Thompson sampling. Chen et al. (2017) proposes to use the standard-deviation of bootstrapped Q-functions to measure the uncertainty. Although the uncertainty measurement is similar to that of OB2I, our method is different from Chen et al. (2017) in the following aspects: (i) our approach propagates the uncertainty through backward update; (ii) Chen et al. (2017) does not use the bonus in the update of Q-functions and their bonus is computed when taking the actions; (iii) we establish theoretical connections between the proposed UCB-bonus and LSVI-UCB. SUNRISE (Lee et al., 2020) extends Chen et al. (2017) to continuous control through confidence reward and weighted Bellman backup. Information-Directed Sampling (IDS) (Nikolov et al., 2019) is based on bootstrapped DQN, and chooses actions by balancing the instantaneous regret and information gain. OAC (Ciosek et al., 2019) uses two Q-networks to get lower and upper bounds of the Q-value to perform exploration in continuous control tasks. These methods seek to estimate the epistemic uncertainty and choose the optimistic actions. In contrast, we use the uncertainty of value function to construct intrinsic rewards and perform backward update, which propagates future uncertainty to the estimated Q-value.

Uncertainty Bellman Equation (UBE) (O'Donoghue et al., 2018) proposes an upper bound on the variance of the posterior of Q-values, which is further utilized for optimism in exploration. Bayesian-DQN (Azizzadenesheli et al., 2018) replaces the last layer in deep Q-network with Bayesian Linear Regression (BLR) that estimates a posterior of the Q-function. These methods use parametric distributions to describe the posterior while OB2I uses the bootstrapped method to construct the confidence bonus. UBE and BLR also require inverting a large matrix in training and hence is computational expensive. Previous methods also utilize the epistemic uncertainty of dynamics through Bayesian posterior (Ratzlaff et al., 2020) and ensembles (Pathak et al., 2019). Nevertheless, they consider single-step uncertainty, while we consider the long-term uncertainty in an episode.

To measure the novelty of states for constructing count-based intrinsic rewards, previous methods have attempted to use density model (Bellemare et al., 2016; Ostrovski et al., 2017), static hashing (Tang et al., 2017; Choi et al., 2019; Rashid et al., 2020), episodic curiosity (Savinov et al., 2019; Badia et al., 2020), curiosity-bottleneck (Kim et al., 2019b), information gain (Houthooft et al., 2016) and prediction error from random networks (Burda et al., 2019b) for novelty evaluation. The curiosity-driven exploration based on prediction-error of environment models such as ICM (Pathak et al., 2017; Burda et al., 2019a), EMI (Kim et al., 2019a), and variational dynamics (Bai et al., 2020) enable the agents to explore in a self-supervised manner. According to Taiga et al. (2020), although bonus-based methods show promising results in hard exploration tasks like Montezuma's Revenge, they do not perform well on other Atari games. Meanwhile, NoisyNet (Fortunato et al., 2018) performs significantly better than bonus-based methods evaluated by the entire Atari suite. Overall, Taiga et al. (2020) suggests that the pace of the exploration progress might have been obfuscated by some promising results only on a few selected hard exploration games. We follow this principle and evaluate OB2I on the Atari suite with 49 games.

Beyond model-free methods, model-based RL also uses optimism for planning and exploration (Nix & Weigend, 1994). Model-assisted RL (Kalweit & Boedecker, 2017) uses ensembles to make use of artificial data with high uncertainty. Buckman et al. (2018) uses ensemble dynamics and Q-functions to use model rollouts when they do not cause large errors. Planning to explore (Sekar et al., 2020) seeks out future uncertainty by integrating uncertainty to Dreamer (Hafner et al., 2020). Ready Policy One (Ball et al., 2020) optimizes policies for both reward and model uncertainty reduction. Noise-Augmented RL (Pacchiano et al., 2020) uses statistical bootstrap to generalize the optimistic posterior sampling (Agrawal & Jia, 2017) to DRL. Hallucinated UCRL (Curi et al., 2020) reduces optimistic exploration to exploitation by enlarging the control space. The model-based RL needs to estimate the posterior of dynamics, while OB2I relies on the posterior of Q-functions.

5. Experimental Results

5.1. Environmental Baselines

We evaluate the algorithms in high-dimensional image-based tasks, including MNIST Maze (Lee et al., 2019) and 49 Atari games. We refer Appendix C for the experiments

on MNIST Maze, and discuss the experiments on Atari games in this section. Directly comparing OB2I with baselines using Bootstrapped DQN is not fair, since OB2I uses backward update for training. To achieve fair comparison, we reimplement all Bootstrapped DQN-based baselines with BEBU. We compare the following methods in experiments:

- **OB2I**: the proposed principled exploration method.
- **BEBU**: a reimplementation of Bootstrapped DQN (Osband et al., 2016) with BEBU.
- **BEBU-UCB**: BEBU with optimistic actions selected by the upper bound of Q (Chen et al., 2017; Lee et al., 2020).
- **BEBU-IDS**: integrating homoscedastic IDS (Nikolov et al., 2019) into BEBU without distributional RL.

We refer to Appendix B for the algorithmic comparison between all methods. According to EBU (Lee et al., 2019), the backward update is significantly more sample-efficient than standard Q-learning by using only 20M training frames to achieve the mean human-normalized score of standard DQN, which requires 200M training frames. We follow this setting by training all BEBU-based methods with 20M frames. In our experiments, 20M frames in OB2I is sufficient to produce strong empirical results and achieve competitive results with several baselines using 200M frames.

We additionally compare the performance of DQN (Mnih et al., 2015), NoisyNet (Fortunato et al., 2018), Bootstrapped DQN (BootDQN) (Osband et al., 2016), BootDQN-IDS (Nikolov et al., 2019), UBE (O'Donoghue et al., 2018) in 200M training frames, and Bayesian DQN (Azizzadenesheli et al., 2018) in 20M training frames. We choose NoisyNet as a baseline since it has been evaluated on the entire Atari suite (instead of several hard exploration games) such that it performs substantially better than existing bonus-based methods (Taiga et al., 2020), including CTS-counts (Bellemare et al., 2016), PixelCNN-counts (Ostrovski et al., 2017), RND (Burda et al., 2019b), and ICM (Pathak et al., 2017). UBE and Bayesian-DQN are selected as baselines because they use parametric functions to approximate the posterior of Q-values, while OB2I uses a non-parametric bootstrap. BootDQN-IDS has been demonstrated to be a strong baseline (Nikolov et al., 2019) based information-directed sampling and BootDQN.

5.2. Evaluation Metric and Hyperparameters

An ensemble policy by a majority vote of Q-heads is used for 30 no-op evaluation. The no-op evaluation indicates a setting that 30 no-op actions are first executed in each evaluation episode to provide diversity for the agent (Mnih et al., 2015). The majority-vote combines all the heads into a single ensemble policy, which follows the same evaluation method as in Osband et al. (2016). We use the popular

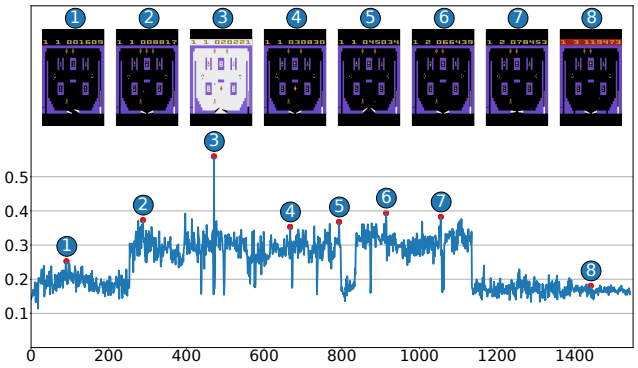


Figure 2. Visualizing UCB-bonus in VideoPinball. See video at https://rb.gy/xmzw4g.

human-normalized score $\frac{\text{Score}_{\text{Agent}}-\text{Score}_{\text{Random}}}{|\text{Score}_{\text{human}}-\text{Score}_{\text{random}}|}$ as a baseline score. In Atari games, Osband et al. (2016) observes that the bootstrapping does not contribute much in performance. Empirically, Bootstrapped DQN uses the same samples to train all Q-heads in each training step. This empirical simplification is also adopted by Chen et al. (2017); Osband et al. (2018); Nikolov et al. (2019). We use such a simplification for OB2I and all bootstrapped-based methods.

For OB2I, we set both α_1 and α_2 as 0.5×10^{-4} by tuning over five popular tasks, including Breakout, Freeway, Qbert, Seaquest, and SpaceInvaders. Generally, small α_1 and α_2 yield better performance empirically since the bonus accumulates along the episode that usually contains thousands of steps in Atari. We use diffusion factor $\beta = 0.5$ for all methods by following Lee et al. (2019). We refer to Appendix D for the detailed specifications. The code is available at https://github.com/Baichenjia/OB2I.

5.3. Main Results and Visualization

Table 1 reports the overall performance of all the methods on 49 Atari games. According to Table 1, BootDQN-IDS performs better than UBE, BootDQN, and NoisyNet. Thus, BootDQN-IDS outperforms popular bonus-based exploration methods that perform worse than NoisyNet (Taiga et al., 2020). We then reimplement BootDQN-IDS with BEBU, and we refer this version to as BEBU-IDS. We observe that OB2I outperforms BEBU-IDS in both mean and medium scores, as well as outperforming all other bonus-based methods in the backward update setting. We report the detailed raw scores in Appendix F. Moreover, Appendix G shows that OB2I outperforms BEBU, BEBU-UCB, and BEBU-IDS in 36, 34, and 35 games out of all 49 games, respectively.

To understand the general-purpose UCB-bonus, we use a trained OB2I agent to interact with the environment for an

Table 1. Summary of human-normalized scores in 49 Atari games. BEBU, BEBU-UCB, BEBU-IDS and OB2I are trained for 20M frames with RTX-2080Ti GPU for 5 random seeds.

Frames	200M					20M				
	DQN	UBE	BootDQN	NoisyNet	**BootDQN-IDS**	Bayesian-DQN	BEBU	BEBU-UCB	**BEBU-IDS**	**OB2I**
Mean	241%	440%	553%	651%	**757%**	224%	553%	610%	**622%**	**765%**
Median	93%	126%	139%	172%	**187%**	27%	36%	38%	**44%**	**50%**

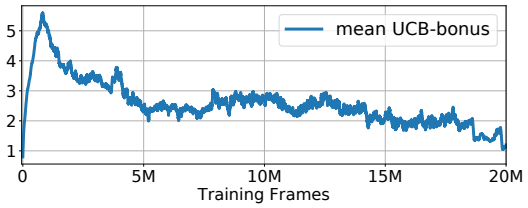


Figure 3. The change of mean UCB-bonus in the learning process.

episode in VideoPinball and record the UCB-bonuses at each step. OB2I improves the performance of VideoPinball significantly and achieves the best score among all baselines. In this task, the pinball moves fast in the playfield to hit bumpers, spinners and rollovers to score points. Our UCB-bonus estimates the uncertainty of interacting with different objects to encourage the pinball to hit less frequently visited objects. The curve in Figure 2 shows the UCB-bonuses of the subsampled steps in the episode. We choose eight spikes and visualize the corresponding frames. The events in spikes correspond to rarely hit objects or crucial events, which are important for the agent to obtain rewards: hitting the rollover (1,4,6), using flippers to send the pinball back into the playfield when it drops to the bottom (2), hitting the specific lit target (3), hitting the bumpers and spinners (7,8), and losing the ball (5). Most obviously, the UCB-bonus increases significantly at spike 3 because the ball hit a specific lit target that causes the screen to flash and the agent scores 1000 points, while hitting other objects gets less than 100 points. In the last stage (including spike 8), the UCB-bonuses are low since the score has reached the upper limit and the flippers are locked. We provide more visualization examples in Appendix E.

We further record the the mean of the UCB-bonus of the training batch in the learning process. The result is shown in Figure 3. The UCB-bonus is low at the beginning since the networks are randomly initialized. When the agent starts to explore the environment, the mean UCB-bonus increases rapidly to award exploration. As more experiences of state-action pairs are gathered, the mean UCB-bonus reduces gradually, indicating that the bootstrapped value functions concentrate around the optimal value and the epistemic uncertainty decreases. Nevertheless, according to Figure 2, the UCB-bonuses are relatively high at scarcely visited areas or crucial events, and therefore the bonuses promote exploration for the corresponding events.

Table 2. Ablation Study

	Backward	Bonus	Qbert	SpaceInvaders	Freeway
OB2I	✓	UCB	4275.0	904.9	32.1
BootDQN-UCB	-	UCB	3284.7	731.8	20.5
BEBU	✓	-	3588.4	814.4	21.5
BootDQN	-	-	2206.8	649.5	18.3
BEBU-RND	✓	RND	3702.5	832.7	22.6

5.4. Ablation Study

We conduct an ablation study to better understand the importance of backward update and bonus term in OB2I. The results of the ablation studies are provided in Table 2. We observe that (i) when we use the ordinary update strategy by sampling transitions instead of episodes, OB2I reduces to BootDQN-UCB with significant performance loss. This is consistent with previous conclusions in (Lee et al., 2019) that backward update is crucial for sample-efficient training; (ii) when the UCB-bonus is set to 0, OB2I reduces to BEBU; (iii) when both the backward update and UCB-bonus are removed, OB2I reduces to standard BootDQN, which performs poorly in 20M training frames; (iv) to illustrate the effect of the proposed UCB-bonus, we substitute it with the popular RND-bonus (Burda et al., 2019b). Specifically, we use an independent RND network to generate RND-bonus for each state in training. The RND-bonus is added to both the immediate reward and next-Q. The result shows that our proposed UCB-bonus outperforms RND-bonus without introducing additional complexities compared to BootDQN.

6. Conclusion

In this work, we have proposed a principled exploration method, i.e., OB2I, that shares nice theoretical properties as LSVI-UCB. By integrating with backward induction, the sample efficiency is further enhanced. We evaluate OB2I empirically by solving MNIST maze and 49 Atari games. Results show that OB2I outperforms several strong baselines. The visualizations suggest that high UCB-bonus corresponds to informative experiences for exploration. As far as we see, our work seems to establish the first empirical attempt of uncertainty propagation in deep RL, which exploits the core benefit of theoretical analysis. Moreover, we

observe that the connection between theoretical analysis and practical algorithm provides strong empirical performance, which hopefully raises insights on combining theory and practice to the community. Future directions include adapting OB2I to continuous control and integrating OB2I with other expressive bonus schemes.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 51935005, in part by the Fundamental Research Program under Grant JCKY20200603C010, and in part by the Science and Technology on Space Intelligent Laboratory under Grant ZDSYS-2018-02. Part of the work was done during internship at Tencent Robotics X lab. The authors thank Tencent Robotics X for the computation resources supported. The authors also thank the anonymous reviewers, whose invaluable comments and suggestions have helped us to improve the paper.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pp. 2312–2320, 2011.

Agrawal, S. and Jia, R. Posterior sampling for reinforcement learning: worst-case regret bounds. In *Advances in Neural Information Processing Systems*, pp. 1184–1194, 2017.

Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pp. 322–332. PMLR, 2019.

Auer, P. and Ortner, R. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 49–56, 2007.

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pp. 263–272, 2017.

Azizzadenesheli, K., Brunskill, E., and Anandkumar, A. Efficient exploration through bayesian deep q-networks. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–9. IEEE, 2018.

Badia, A. P., Sprechmann, P., Vitvitskyi, A., Guo, D., Piot, B., Kapturowski, S., Tieleman, O., Arjovsky, M., Pritzel, A., Bolt, A., and Blundell, C. Never give up: Learning directed exploration strategies. In *International Conference on Learning Representations*, 2020.

Bai, C., Liu, P., Wang, Z., Liu, K., Wang, L., and Zhao, Y. Variational dynamic for self-supervised exploration in deep reinforcement learning. *arXiv preprint arXiv:2010.08755*, 2020.

Ball, P., Parker-Holder, J., Pacchiano, A., Choromanski, K., and Roberts, S. Ready policy one: World building through active learning. *arXiv preprint arXiv:2002.02693*, 2020.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pp. 1471–1479, 2016.

Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458, 2017.

Buckman, J., Hafner, D., Tucker, G., Brevdo, E., and Lee, H. Sample-efficient reinforcement learning with stochastic ensemble value expansion. In *Advances in Neural Information Processing Systems*, pp. 8224–8234, 2018.

Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., and Efros, A. A. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*, 2019a.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019b.

Cai, Q., Yang, Z., Jin, C., and Wang, Z. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pp. 1283–1294. PMLR, 2020.

Chen, R. Y., Sidor, S., Abbeel, P., and Schulman, J. Ucb exploration via q-ensembles. *arXiv preprint arXiv:1706.01502*, 2017.

Choi, J., Guo, Y., Moczulski, M., Oh, J., Wu, N., Norouzi, M., and Lee, H. Contingency-aware exploration in reinforcement learning. In *International Conference on Learning Representations*, 2019.

Ciosek, K., Vuong, Q., Loftin, R., and Hofmann, K. Better exploration with optimistic actor critic. In *Advances in Neural Information Processing Systems*, pp. 1785–1796, 2019.

Curi, S., Berkenkamp, F., and Krause, A. Efficient model-based reinforcement learning through optimistic policy search and planning. *Advances in Neural Information Processing Systems*, 33, 2020.

Dann, C. and Brunskill, E. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2818–2826, 2015.

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C., and Legg, S. Noisy networks for exploration. In *International Conference on Learning Representations*, 2018.

Friedman, J., Hastie, T., and Tibshirani, R. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596, 2018.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pp. 1109–1117, 2016.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pp. 4863–4873, 2018.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. In *Proceedings of Thirty Third Conference on Learning Theory*, pp. 2137–2143, 2020.

Kalweit, G. and Boedecker, J. Uncertainty-driven imagination for continuous deep reinforcement learning. In *Conference on Robot Learning*, pp. 195–206, 2017.

Kim, H., Kim, J., Jeong, Y., Levine, S., and Song, H. O. Emi: Exploration with mutual information. In *International Conference on Machine Learning*, pp. 3360–3369, 2019a.

Kim, Y., Nam, W., Kim, H., Kim, J.-H., and Kim, G. Curiosity-bottleneck: Exploration by distilling task-specific novelty. In *International Conference on Machine Learning*, pp. 3379–3388, 2019b.

Lee, K., Laskin, M., Srinivas, A., and Abbeel, P. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. *arXiv preprint arXiv:2007.04938*, 2020.

Lee, S. Y., Sungik, C., and Chung, S.-Y. Sample-efficient deep reinforcement learning via episodic backward update. In *Advances in Neural Information Processing Systems*, pp. 2110–2119, 2019.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Nikolov, N., Kirschner, J., Berkenkamp, F., and Krause, A. Information-directed exploration for deep reinforcement learning. In *International Conference on Learning Representations*, 2019.

Nix, D. A. and Weigend, A. S. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, volume 1, pp. 55–60. IEEE, 1994.

Osband, I. and Van Roy, B. Why is posterior sampling better than optimism for reinforcement learning? In *International Conference on Machine Learning*, pp. 2701–2710, 2017.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, pp. 4026–4034, 2016.

Osband, I., Aslanides, J., and Cassirer, A. Randomized prior functions for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 8617–8629, 2018.

Osband, I., Van Roy, B., Russo, D. J., and Wen, Z. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019.

Ostrovski, G., Bellemare, M. G., van den Oord, A., and Munos, R. Count-based exploration with neural density models. In *International Conference on Machine Learning*, pp. 2721–2730, 2017.

O'Donoghue, B., Osband, I., Munos, R., and Mnih, V. The uncertainty bellman equation and exploration. In *International Conference on Machine Learning*, pp. 3836–3845, 2018.

Pacchiano, A., Ball, P., Parker-Holder, J., Choromanski, K., and Roberts, S. On optimism in model-based reinforcement learning. *arXiv preprint arXiv:2006.11911*, 2020.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pp. 2778–2787, 2017.

Pathak, D., Gandhi, D., and Gupta, A. Self-supervised exploration via disagreement. In *International Conference on Machine Learning*, pp. 5062–5071, 2019.

Rashid, T., Peng, B., Boehmer, W., and Whiteson, S. Optimistic exploration even with a pessimistic initialisation. In *International Conference on Learning Representations*, 2020.

Ratzlaff, N., Bai, Q., Fuxin, L., and Xu, W. Implicit generative modeling for efficient exploration. In *International Conference on Machine Learning*, 2020.

Russo, D. Worst-case regret bounds for exploration via randomized value functions. In *Advances in Neural Information Processing Systems*, pp. 14410–14420, 2019.

Savinov, N., Raichuk, A., Marinier, R., Vincent, D., Pollefeys, M., Lillicrap, T., and Gelly, S. Episodic curiosity through reachability. In *International Conference on Learning Representations*, 2019.

Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., and Pathak, D. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, 2020.

Shani, L., Efroni, Y., Rosenberg, A., and Mannor, S. Optimistic policy optimization with bandit feedback. In *International Conference on Machine Learning*, pp. 8604–8613. PMLR, 2020.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Taiga, A. A., Fedus, W., Machado, M. C., Courville, A., and Bellemare, M. G. On bonus based exploration methods in the arcade learning environment. In *International Conference on Learning Representations*, 2020.

Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, O. X., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. Exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in neural information processing systems*, pp. 2753–2762, 2017.

Wang, Y., Wang, R., Du, S. S., and Krishnamurthy, A. Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*, 2019.

West, M. Outlier models and prior distributions in bayesian linear regression. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(3):431–439, 1984.

Zanette, A., Brandfonbrener, D., Brunskill, E., Pirotta, M., and Lazaric, A. Frequentist regret bounds for randomized least-squares value iteration. In *International Conference on Artificial Intelligence and Statistics*, pp. 1954–1964, 2020.