Epistemic Bellman Operators

van der Vaart, Pascal R.; Spaan, Matthijs T.J.; Yorke-Smith, Neil

DOI
10.1609/aaai.v39i20.35393

Publication date
2025

Document Version
Final published version

Published in
Proceedings of the AAAI Conference on Artificial Intelligence

Citation (APA)
van der Vaart, P. R., Spaan, M. T. J., & Yorke-Smith, N. (2025). Epistemic Bellman Operators. *Proceedings of the AAAI Conference on Artificial Intelligence*, *39*(20), 20973-20981.
https://doi.org/10.1609/aaai.v39i20.35393

Epistemic Bellman Operators

Pascal R. van der Vaart, Matthijs T. J. Spaan, Neil Yorke-Smith

Delft University of Technology, Delft, Netherlands
{p.r.vandervaart-1, m.t.j.spaan, n.yorke-smith}@tudelft.nl

Abstract

Uncertainty quantification remains a difficult challenge in reinforcement learning. Several algorithms exist that successfully quantify uncertainty in a practical setting. However it is unclear whether these algorithms are theoretically sound and can be expected to converge. Furthermore, they seem to treat the uncertainty in the target parameters in different ways. In this work, we unify several practical algorithms into one theoretical framework by defining a new Bellman operator on distributions, and show that this Bellman operator is a contraction. We highlight use cases of our framework by analyzing an existing Bayesian Q-learning algorithm, and also introduce a novel uncertainty-aware variant of PPO that adaptively sets its clipping hyperparameter.

Introduction

Reinforcement learning (RL) algorithms have surpassed humans' ability in many games (Mnih et al. 2015; Schrittwieser et al. 2020), and have now also found success in real world problems such as controlling plasma in a nuclear fusion reactor (Degrave et al. 2022), video compression (Mandhane et al. 2022), large language models (Ouyang et al. 2022) and algorithm design (Fawzi et al. 2022; Mankowitz et al. 2023). However, even for relatively simple tasks, algorithms still require many simulations or real interactions to learn a strong policy, making them inefficient. One approach to attack this problem is by making algorithms aware of their epistemic uncertainty, which is uncertainty caused by a lack of data. This allows them to explore only parts of the problem that are still uncertain, decreasing the total amount of interactions required.

However, proper uncertainty quantification is still an open problem in reinforcement learning. Many techniques from supervised learning, such as ensembles (Dietterich 2000; Lakshminarayanan, Pritzel, and Blundell 2017) and Bayesian methods (Chen, Fox, and Guestrin 2014; Liu and Wang 2016; D'Angelo and Fortuin 2021; Wenzel et al. 2020), have found success in practice when applied to supervised learning tasks with labelled data. However, in reinforcement learning data is not labelled with a ground truth, and instead the label for the current state is a self-supervised bootstrap from the label of the next state, known as the target value. Uncertainty quantification in RL must consider this sequential nature. At the heart of this problem is the fact that uncertainty in the current state should include the uncertainty in the target values, which is the uncertainty in the future states.

Adaptations of uncertainty quantification methods from supervised learning have been applied to reinforcement learning settings (Osband et al. 2016; Osband, Aslanides, and Cassirer 2018; Fortunato et al. 2017; Azizzadenesheli, Brunskill, and Anandkumar 2018; Burda et al. 2018; Dwaracherla and Roy 2021; Schmitt, Shawe-Taylor, and van Hasselt 2023; Van der Vaart, Yorke-Smith, and Spaan 2024) with good practical results, but there is no guarantee that the way these algorithms treat the uncertainty in the successor state leads to a theoretically sound algorithm, in the sense that the uncertainty quantification aspect can be expected to converge to a solution at all. At least guaranteeing that these methods work in potentially simplified scenarios is essential for the adoption of uncertainty quantification in algorithms in the real world. Furthermore, some algorithms seemingly disagree in their decisions on how to treat the uncertainty in the target values.

When adapting Deep Q-learning (DQN)-style algorithms to uncertainty aware algorithms like BootDQN (Osband et al. 2016), EVE (Schmitt, Shawe-Taylor, and van Hasselt 2023), Langevin-DQN (Dwaracherla and Roy 2021), LMCDQN (Ishfaq et al. 2023), SMC-DQN (Van der Vaart, Yorke-Smith, and Spaan 2024) and BDQN (Azizzadenesheli, Brunskill, and Anandkumar 2018), there are decisions to be made about how to use and update the target parameters. Generally, these algorithms condition their posterior on a posterior of the target parameters. As a main problem, we highlight that there is no guarantee that the process of repeatedly updating the current distribution, conditioned on the distribution over target parameters, and copying it to the target parameters will converge to a limiting distribution.

Recently, Fellows, Hartikainen, and Whiteson (2021) studied this problem theoretically and contended that Bayesian model-free reinforcement learning algorithms create a posterior over Bellman operators. They showed that the posterior converges to the true Bellman operator in the limit of infinite data. We instead take an arguably more natural and direct approach, and show that the problem can be formulated as a generic Bellman operator that works on distributions.

Specifically, our contributions are as follows:
1) We introduce Epistemic Bellman operators as a tool to analyze existing algorithms and develop theoretically sound uncertainty aware RL algorithms. Our unified framework formalizes the process of conditioning on distributions over target parameters.
2) We prove that Epistemic Bellman operators are contractions, implying that the process of interleaving posterior inference and target updates converges to a consistent fixed point for a general class of distributions and return estimators. Furthermore, we show that the mean of the fixed point of an Epistemic Bellman operator for policy evaluation is the fixed point of its non-epistemic counterpart.
3) We highlight the utility of Epistemic Bellman operators by analyzing an existing Bayesian Q-learning algorithm, alleviating an overestimation problem and experimentally verify our theory. Furthermore, we develop a novel uncertainty aware version of Proximal Policy Optimization that clips less aggressively whenever it is certain about its advantages, and show improved performance in several environments.

Background

Markov Decision Processes

We focus on Markov Decision Processes (MDP) with infinite horizon in the discounted reward setting. Formally, a Markov Decision Process is a tuple $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$ of a state space $\mathcal{S}$, action space $\mathcal{A}$, transition function $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and discount factor $0 \leq \gamma < 1$. At each time step t, an agent observes the current state s_t, chooses an action $a_t \sim \pi(s_t)$ according to its policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, and receives reward $r_t = R(s_t, a_t)$. The goal of reinforcement learning is to find a policy π that maximizes the discounted cumulative reward $\mathbb{E}_{T,\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$. Of central importance is the Q-function

$$Q^\pi(s, a) = R(s, a) + \mathbb{E}_{T,\pi}\left[\sum_{t=1}^{\infty} \gamma^t r_t \mid s_0 = s, a_0 = a\right],$$

denoting the expected discounted future reward if the agent executes action a in state s and then follows the policy π.

In a tabular setting, we represent the reward function, transition function and policy as vectors and matrices $R \in \mathbb{R}^{|S||A|}$, $T \in \mathbb{R}^{|S||\mathcal{A}| \times |\mathcal{S}|}$, $\pi \in \mathcal{R}^{|S||A|}$. The Bellman operator for a policy π can then be written as

$$B^\pi_{T,R} Q = R + \gamma T^\pi Q,$$

where $T^\pi \in \mathbb{R}^{|S||A| \times |S||A|}$ is the transition function from state-action to state-action induced by the transition function T and the policy π, defined by

$$\begin{aligned}(T^\pi)_{sas'a'} &= \mathbb{P}(s_{t+1}, a_{t+1} = s', a' \mid s_t, a_t = s, a) \\ &= T_{sas'} \pi_{s'a'}.\end{aligned} \tag{1}$$

Since the transition function T and reward function R are assumed to be unknown to the agent, computing a strong policy requires exploration of the environment to learn which actions result in optimal return.

Model-free Reinforcement Learning

Typically interesting problems have large states and action spaces, making it difficult to learn the transition and reward functions. Model-free algorithms such as actor-critics (Mnih et al. 2016; Schulman et al. 2017; Haarnoja et al. 2018) and Q-learning (Mnih et al. 2015) bypass this step and instead aim to learn a good policy or the values of a good policy directly, without estimating T and R.

A common component is to learn the values or Q-values by representing them by a neural network and minimizing the squared temporal difference loss on a dataset $\mathcal{D}$:

$$\begin{aligned}L_{TD}(\theta, \theta', \mathcal{D}) &= \sum_{(s,a,r,s') \in \mathcal{D}} TD(\theta, \theta', (s, a, r, s'))^2 \\ &= \sum_{(s,a,r,s') \in \mathcal{D}} \left[Q_\theta(s, a) - r - \gamma G(\theta', s')\right]^2,\end{aligned} \tag{2}$$

where $G(\theta', s')$ is some return estimator usually depending on a bootstrap from a target network θ' (Mnih et al. 2015). Examples are $G(\theta', s') = \max_{a'} Q_{\theta'}(s', a')$ in the case of one step Q-learning, or $G(\theta', a') = \sum_{a' \in A} \pi(a'|s') Q_{\theta'}(s', a')$ in the case of policy evaluation in actor-critics.

Agents use empirically observed transitions (s, a, r, s') to learn these models, requiring exploration to sufficiently cover the environment to achieve accurate values. Quantifying uncertainty in the value models can greatly improve the exploration capability of reinforcement learning algorithms through Thompson Sampling (Osband et al. 2016; Osband, Aslanides, and Cassirer 2018; O'Donoghue et al. 2018; Fortunato et al. 2017; Schmitt, Shawe-Taylor, and van Hasselt 2023; Azizzadenesheli, Brunskill, and Anandkumar 2018; Dwaracherla and Roy 2021) or exploration bonuses (Ostrovski et al. 2017; Bellemare et al. 2016; Burda et al. 2018). Furthermore, uncertainty quantification can also aid in general stability of algorithms by reweighting Bellman errors (Lee et al. 2021).

Bayesian Value Learning

One method to quantify uncertainty is through Bayesian algorithms. Generally, a Bayesian neural network is any neural network parameterized by $\theta \in \Theta$ where one attempts to model the posterior distribution

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{\int p(\mathcal{D}|\theta)p(\theta)d(\theta)},$$

where $p(\mathcal{D}|\theta)$ is the likelihood, $p(\theta)$ is a prior and $\mathcal{D}$ is some data set. The posterior density $p(\theta|\mathcal{D})$ signifies how likely values of θ are, and is a natural method to model uncertainty as a distribution.

To equip an agent with uncertainty quantification, a posterior distribution over the parameters of a Q-function can be constructed $p(\theta|\mathcal{D}, \theta') \propto p(\mathcal{D}|\theta, \theta')p(\theta)$. Since the squared error loss is proportional to the log-density of a normal distribution, defining

$$p(\mathcal{D}|\theta, \theta') = \exp\left(-\sum_{(s,a,r,s') \in \mathcal{D}} \left[Q_\theta(s, a) - r - \gamma G(\theta', s')\right]^2\right) \tag{3}$$

is a natural candidate for the likelihood when extending value learning algorithms to a Bayesian paradigm. This corresponds to the assumption that the temporal difference errors are normally distributed:

$$TD(\theta, \theta', (s, a, r, s')) \sim \mathcal{N}(0, \sigma). \tag{4}$$

While this assumption is in general not correct for every MDP, it is a convenient design choice and it should come as no surprise that several previous works have used this likelihood before (Osband, Aslanides, and Cassirer 2018; Schmitt, Shawe-Taylor, and van Hasselt 2023; Dwaracherla and Roy 2021; Azizzadenesheli, Brunskill, and Anandkumar 2018; Ishfaq et al. 2023).

The likelihood $p(\mathcal{D}|\theta, \theta')$ and therefore also the posterior density $p(\theta|\mathcal{D}, \theta')$ does not only depend on the data, i.e., the observed transitions, it is also conditioned on the target values θ'. Handling this dependency is crucial for a theoretically sound algorithm that handles the sequential nature of uncertainty in this setting. Furthermore, posterior distributions are generally difficult to compute in practice, requiring approximate models. For example, BootDQN (Osband et al. 2016; Osband, Aslanides, and Cassirer 2018) uses ensembles, Langevin-DQN, LMCDQN and SMC-DQN (Dwaracherla and Roy 2021; Ishfaq et al. 2023; Van der Vaart, Yorke-Smith, and Spaan 2024) use Monte Carlo methods, EVE (Schmitt, Shawe-Taylor, and van Hasselt 2023) uses a Laplace approximation and BDQN (Azizzadenesheli, Brunskill, and Anandkumar 2018) performs inference over only the final layer of the Q-network.

Problem Statement

In this section we identify a key problem with model-free Bayesian reinforcement learning algorithms and motivate the value of our main contribution.

Problems with Target Updates

Roughly speaking, algorithms such as BootDQN, Langevin-DQN, LMCDQN, SMC-DQN, BDQN and EVE operate by interleaving steps

1. Infer a posterior given the current targets, $p_{\texttt{main}}(\theta|\mathcal{D}) = p(\theta|\mathcal{D}, \theta')$, where the targets are drawn or assumed to be from some distribution over targets $p_{\texttt{target}}(\theta')$.
2. Update the distribution over targets: $p_{\texttt{target}}(\theta) \leftarrow p_{\texttt{main}}(\theta|\mathcal{D}) = p(\theta|\mathcal{D}, \theta')$ to the current distribution over the main parameters θ.

This is analogous to the target update in many non-probabilistic algorithms that use temporal difference learning, and may seem like a reasonable adaptation to the Bayesian setting. However, for distributions there is no guarantee that this scheme converges, or is in fact well defined, since setting $p_{\texttt{target}}(\theta) \leftarrow p_{\texttt{main}}(\theta)$ is mathematically unsupported when $p_{\texttt{main}}(\theta)$ is a distribution that was conditioned on the target parameters. Furthermore, if this scheme does not converge to the same $p_{\texttt{main}}(\theta|\mathcal{D})$ for a fixed data set $\mathcal{D}$ and every starting distribution, it is not sensible to define a posterior $p_{\texttt{main}}(\theta|\mathcal{D})$ that is only conditioned on $\mathcal{D}$.

Fellows, Hartikainen, and Whiteson (2021) propose interpreting the problem as inferring a posterior distribution over Bellman operators, and show convergence of the posterior to the true Bellman operator as more data is collected.

Instead, we propose a new Bellman operator that operates on posterior-like distributions, and prove that this operator is a contraction and has a fixed point. Roughly speaking, we show that an algorithm that alternates between updating a distribution conditioned on the targets, and updating the distribution over targets converges to a limiting distribution, proving that several common Bayesian algorithms which are special cases of our operator can be expected to converge, independent of the starting distribution.

Visualizing the Distributions

Before we introduce Epistemic Bellman Operators, we analyze which distributions Bayesian Q-learning algorithms actually attempt to approximate. To this end, we study BootDQN and EVE in a tabular setting, and assume there exists some idealized distribution over targets $Q' \sim p_{\texttt{target}}(Q)$ that our agent currently has. Furthermore, as in BootDQN and EVE, we are equipped with a likelihood

$$p(\mathcal{D}|Q, Q') \propto \exp\left(-\frac{1}{2\sigma^2}\sum_{(s,a,r,s')\in\mathcal{D}} \mathrm{TD}(Q, Q', s, a, r, s')^2\right),$$

also conditioned on a set of target values Q'. This results in a posterior distribution

$$p(Q|\mathcal{D}, Q') \propto p(\mathcal{D}|Q, Q')p(Q).$$

However, this distribution is conditioned on a single value for the targets and does not yet incorporate the fact that $Q' \sim p_{\texttt{target}}(Q)$, i.e., the uncertainty over the targets.

In the case of BootDQN, $p_{\texttt{target}}(Q)$ is modelled by the ensemble of target networks $\theta'_1, \dots, \theta'_n$, and to approximate the posterior each ensemble member optimizes for its own loss $Q_i^* = \arg\max p(Q_i|\mathcal{D}, Q'_i)$. On the other hand, EVE has a Laplace approximation for $p_{\texttt{target}}(Q)$, and updates the main distribution by sampling one $\tilde{Q}' \sim p_{\texttt{target}}(Q)$, maximizing $Q = \arg\max p(Q|\mathcal{D}, \tilde{Q}')$ and also updating the Fisher information.

In our idealized setting, we can directly consider the marginalization of the conditioned posterior over targets Q':

$$p_{\texttt{main}}(Q|\mathcal{D}) = \int p(Q|\mathcal{D}, q')dp_{\texttt{target}}(q').$$

Figure 1 shows a graphical presentation of this marginalization, together with BootDQN and EVE, in a simplified setting with an MDP with one state and one action. The top row is the idealized version of Bayesian model-free reinforcement learning algorithms. A distribution over the targets defines a distribution over the main values, which can exactly be inferred by a fully expressive model class. The second row contains a sketch of the situation with ensembles. The distribution $p_{\texttt{target}}$ is an ensemble, which together with the normal distribution likelihoods makes a mixture distribution for the main values. Estimating this distribution with an ensemble ideally returns an ensemble containing the modes of the new distribution.

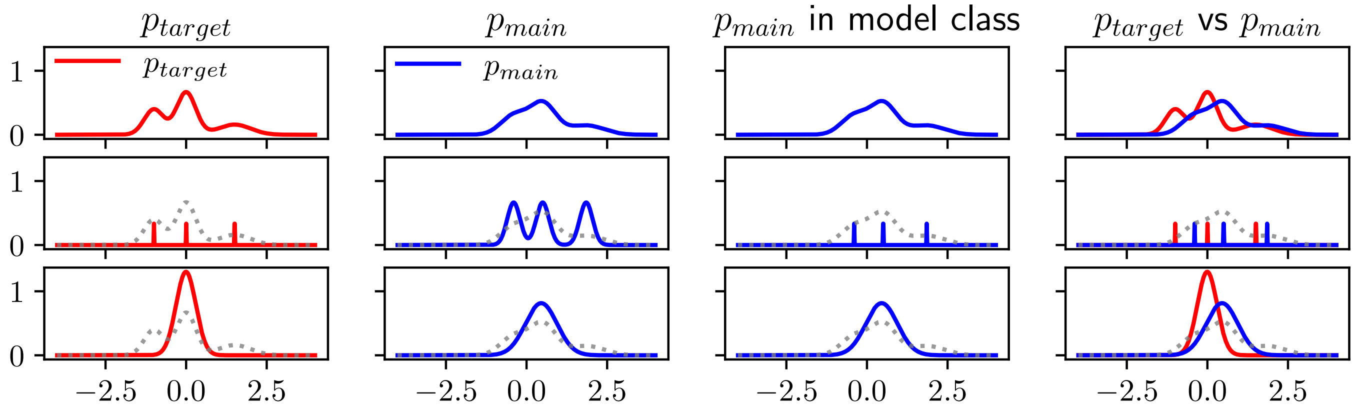

Figure 1: Plots of the distribution over the Q-value of a single state-action. The final column shows the difference between the target distribution (red) and the current distribution (blue). Rows are (1) idealized model class, (2) ensemble approximation (BootDQN), (3) Laplace approximation (EVE).

For EVE, the target distribution is a normal distribution. The distribution for the current state is therefore also a normal distribution, and representing it in the model class of normal distributions returns a normal distribution.

Both BootDQN and EVE can be considered as approximations to this marginalization, approximating the integral with an ensemble in the case of BootDQN and a single sample from $p_{\mathtt{target}}(q')$ in the case of EVE. After constructing an approximate $\tilde{p}_{\mathtt{main}}(Q|\mathcal{D})$ each method then attempts to represent this distribution in their model class.

Considering this marginalization process, we can now define what it means for a well-defined posterior to exist. If the process of

$$\begin{aligned} p^{(k)}_{\mathtt{main}}(Q|\mathcal{D}) &= \int p(Q|\mathcal{D}, q')dp^{(k)}_{\mathtt{target}}(q') && (5)\\ p^{(k+1)}_{\mathtt{target}}(q'|\mathcal{D}) &= p^{(k)}_{\mathtt{main}}(Q|\mathcal{D}) && (6)\\ k &= k+1 && (7) \end{aligned}$$

converges to the same limiting distribution $p(Q|\mathcal{D})^*$ for every starting $p^{(0)}_{\mathtt{target}}(q')$, the posterior-like distribution $p(Q|\mathcal{D})$ is well-defined. We formalize this process with the Epistemic Bellman Operator.

Epistemic Bellman Operators

For any Bellman operator or contraction $B_{\mathcal{D}}$, perhaps depending on some data set $\mathcal{D}$, we can define a pushforward distribution with additive noise as

$$p(Q|\mathcal{D}, Q') = \text{Law}\left(B_{\mathcal{D}}(Q') + \epsilon_{\mathcal{D}}\right), \tag{8}$$

where Law (X) denotes the probability density of X. This is equivalent to the notion that the Q-values are distributed around the target values Q' with some *local* uncertainty $\epsilon_{\mathcal{D}}$, independent of Q'. This is a naturally occurring distribution in literature, since the posterior distribution of a normal likelihood with a normal prior takes this shape, which is commonly used in model-free deep RL literature (Osband et al. 2016; Osband, Aslanides, and Cassirer 2018; Schmitt, Shawe-Taylor, and van Hasselt 2023; Fortunato et al. 2017; Azizzadenesheli, Brunskill, and Anandkumar 2018; Dwaracherla and Roy 2021; Ishfaq et al. 2023). The Epistemic Bellman Operator for this distribution marginalizes the distribution over Q', and returns a new distribution.

Definition 1 (EBO)**.** For any measurable set A, let $\mathcal{P}(A)$ denote the set of probability distributions over A. Let $p(q|q')$ be a distribution over Q-values conditioned on target Q-values, e.g., Equation 8. We define the corresponding Epistemic Bellman Operator (EBO), as an operator $\mathcal{B}_p : \mathcal{P}(\mathbb{R}^{|S||A|}) \to \mathcal{P}(\mathbb{R}^{|S||A|})$, mapping distributions over Q-values to another distribution over Q-values by

$$\mathcal{B}_p P_Q(q) = \int p(q|q')dP_Q(q'). \tag{9}$$

When $p(q|q')$ is of the form Law $(B_{\mathcal{D}}(q') + \epsilon_{\mathcal{D}})$, we can equivalently write Equation 9 as

$$\mathcal{B}_p P_Q = \text{Law}\left(B_{\mathcal{D}}(Q) + \epsilon_{\mathcal{D}}, Q \sim P_Q\right). \tag{10}$$

If the distribution $p(q|q') = \text{Law}\left(B_{\mathcal{D}}(Q) + \epsilon_{\mathcal{D}}\right)$ has contracting properties, for example when $B_{\mathcal{D}}$ is a Bellman operator, it can be shown that the respective EBO is also a contraction. This is formalized in Theorem 1, whose proof is provided in Appendix A (Van der Vaart, Spaan, and Yorke-Smith 2025).

Theorem 1 (Contraction)**.** *Let $\mathcal{Q} = (\mathbb{R}^{|S||A|}, \|.\|_\infty)$ be a metric space, $B_{\mathcal{D}}$ be a contraction on $\mathcal{Q}$, and let $p_B(q|q') =$ Law $(B_{\mathcal{D}}(q') + \epsilon_{\mathcal{D}})$ be a distribution over $\mathcal{Q}$ conditioned on target values in $\mathcal{Q}$.*

Then the corresponding Epistemic Bellman Operator $\mathcal{B}_p : \mathcal{P}(\mathcal{Q}) \to \mathcal{P}(\mathcal{Q})$ defined by Equation 10, where ϵ_D is independent of Q, is a W_ℓ-contraction on $\mathcal{P}(\mathcal{Q})$ for any $\ell \in [1, \infty)$.

This theorem implies that for any dataset $\mathcal{D}$, and any contractive return estimator, repeatedly applying an EBO to any starting distribution will converge to a fixed point. A consequence is that algorithms which interleave posterior inference with target distribution updates are theoretically sound in the sense that they converge to a unique solution $p(Q|\mathcal{D})$.

Theorem 1 does not characterize the optimality of this solution, because this depends on the inner non-epistemic Bellman operator, which is typically the decisive factor for the functioning of an algorithm. For example, in the next section we will apply EBOs to the Optimal Bellman operator as well as Proximal Policy Optimization's return estimator, yielding two very different algorithms.

In the case of policy evaluation with a one-step Bellman Operator

$$BQ = R + \gamma T^{\pi} Q,$$

the fixed point of the EBO $\mathcal{B}$ is simple to characterize, and can be theoretically verified to be consistent with its non-epistemic counterpart. This can be extended to any affine B.

Notably, the following theorem states that the mean of the fixed point is equal to the fixed point of the non-epistemic Bellman operator in $p(q|q')$ when it is affine and ϵ has mean zero. We refer to Appendix A (Van der Vaart, Spaan, and Yorke-Smith 2025) for the proof.

Theorem 2 (Mean of $\mathcal{B}$)**.** *Let $\mathcal{B}$ be the EBO corresponding to $p_B(q|q') = Law\,(B(q') + \epsilon)$ with $\mathbb{E}[\epsilon] = 0$. Let $P_B(Q)$ be the fixed point of $\mathcal{B}$, and Q_B be the fixed point of B. If B is an affine contraction, then $\mathbb{E}_{P_B}[Q] = Q_B$. Furthermore, writing $B(Q) = AQ + b$, the covariance $\Sigma_Q = \mathbb{E}_{P_B}\left[QQ^\top - Q_B Q_B^\top\right]$ is given by*

$$Vec(\Sigma_Q) = (I - A \otimes A)^{-1} Vec(\Sigma_\epsilon)$$

where $Vec(X)$ denotes the vectorization of X and $\otimes$ is the Kronecker product.

To showcase what our theorems state, we conduct an experiment in an MDP with one state and two actions so that the distributions are easy to visualize. We initialize a multivariate normal distribution, and iteratively apply the EBO. Figure 2 displays the density of the distribution over time, with the fixed point of the non-epistemic Bellman equation Q^{π} marked in red. It can be seen that the distributions converge to a normal distribution centered around Q^{π}, where the Q-values are strongly correlated. This correlation is expected, since both actions transition to the same state. Furthermore, Figure 6 in the appendix shows that the Wasserstein distance to the fixed point matches the theoretical contraction rate of γ.

Use Cases of Epistemic Bellman Operators

In this section we highlight two main use cases for Epistemic Bellman Operators: gaining theoretical insight into existing methods by interpreting them with EBOs, and creating new methods using EBOs to guide the model updates.

Thompson Sampling with EBOs

Thompson sampling is a popular exploration algorithm (Azizzadenesheli, Brunskill, and Anandkumar 2018; Dwaracherla and Roy 2021; Ishfaq et al. 2023), making use of approximate sampling from a posterior distribution. More precisely, given a distribution P_Q of likely models, Thompson sampling samples a candidate model $Q \sim P_Q$ and acts greedily with respect to Q. We can model this behaviour with Epistemic Bellman Operators by taking the standard Optimal Bellman Operator as inner operator for our EBO:

$$p(q|q') = \text{Law}\left(R + \gamma T^{\pi^*_{q'}} q' + \epsilon\right),$$

where $\pi^*_{q'}$ denotes the greedy policy with respect to q'. The corresponding Epistemic Bellman Operator reads

$$\mathcal{B}P_Q = \text{Law}\left(R + \gamma T^{\pi^*_Q} Q + \epsilon_{\mathcal{D}}, Q \sim P_Q\right).$$

As a result of Theorem 1, it is known that this operator is a contraction and has a fixed point. However, since

$$\begin{aligned} &\mathbb{E}_{P_Q, P_\epsilon}[\max_a (Q(s,a) + \epsilon(s,a))] \\ &\geq \mathbb{E}_{P_\epsilon}[\max_a \mathbb{E}_{P_Q}[Q(s,a) + \epsilon(s,a)]] \\ &> \max_a \mathbb{E}_{P_Q}[Q(s,a)] \end{aligned} \tag{11}$$

when the event $\epsilon > 0$ has positive probability, it can be predicted that the mean of the fixed point of $\mathcal{B}$ will overestimate the true values of the Thompson sampling policy, similar to the overestimation bias in Q-learning (Van Hasselt 2010; Van Hasselt, Guez, and Silver 2016). Epistemic Bellman Operators can remedy the overestimation in the same manner as in Q-learning, through sampling two independent samples from P_Q. This leads to the operator

$$\mathcal{B}_2 P_Q = \text{Law}\left(R + \gamma T^{\pi^*_{Q'}} Q + \epsilon_{\mathcal{D}}, Q, Q' \sim P_Q\right), \tag{12}$$

which reduces the estimation bias by selecting actions from an independent sample. We conjecture that this is operator is also a contraction under the same assumptions as Theorem 1, and we see in experiments that the values do converge.

Experiments To showcase this result, we run Thompson Sampling (TS) policies in a tabular environment, using Hamiltonian Monte Carlo, a standard MCMC algorithm, to approximately sample from the posterior, and using EBOs to directly sample from the exact distribution. Both methods are provided with unbiased estimators for T and R, so that any errors in the value models are purely due to bias in the algorithms. We then compare the mean of the sampled values to the true values achieved by the TS policy. The results are shown in Figure 3, with implementation details in Appendix B. It can be seen that both the approximate sampler (MCMC) and exact sampler (EBO) overestimate the values with the same linear scaling in ϵ. However, using the double-sampling EBO, eliminates the bias. Furthermore, approximately sampling the fixed point of the double-sampling EBO with an MCMC algorithm also eliminates the bias. In agreement, Ishfaq et al. (2023) report that the double-Q sampling trick also helps MCMC methods in deep RL settings.

Epistemic Clipping PPO

Since Theorem 1 holds for any contraction, it is applicable to a wide range of return estimators used in practice. To showcase the generality of our results, we modify Proximal Policy Optimization (PPO) (Schulman et al. 2017) into Epistemic Clipping PPO (ECPPO) by replacing the value models

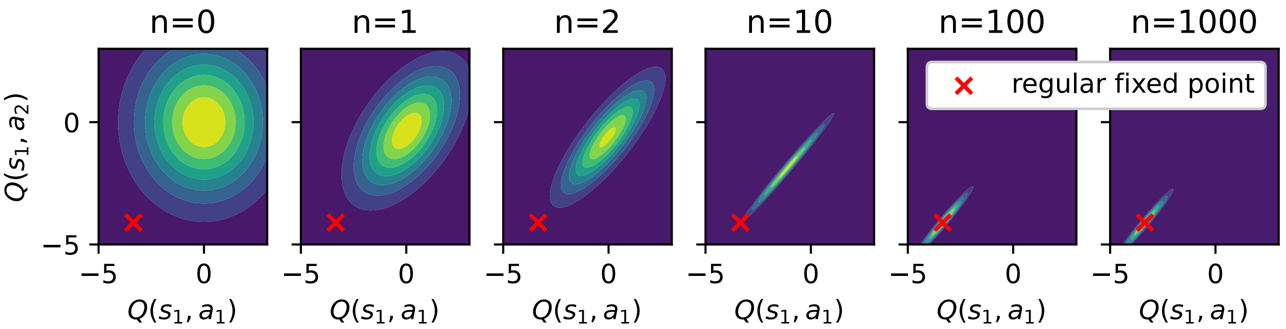

Figure 2: The Epistemic Bellman Operator applied iteratively to an initial distribution with a fixed policy in a single-state, two-actions MDP. The fixed point of the regular Bellman operator is in red.

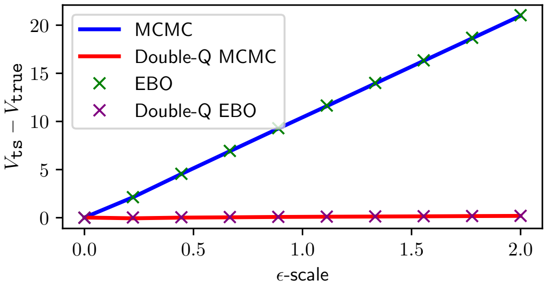

Figure 3: The gap between predicted values and true values of Thompson Sampling policies on a tabular MDP, with various local noise scales. Each line is the mean of 10 independent experiments.

with a distributional model. PPO estimates the advantages of its policy with the following return estimator:

$$A_t = \delta_t + \gamma\lambda\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1}, \quad (13)$$

$$\text{where } \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t). \quad (14)$$

An approximation of the posterior over $V(s)$ provides the agent with uncertainty quantification on the advantages, which we use to clip less aggressively in the policy loss whenever we are certain about the advantages. To this end, the typical policy loss in PPO

$$L^{PPO}(\theta) = \mathbb{E}[\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-c, 1+c)A_t)], \quad (15)$$

is modified to

$$L^{ECPPO}(\theta) = \mathbb{E}[\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-c\phi(U_t), 1+c\phi(U_t))A_t)], \quad (16)$$

where U_t is an estimate of the uncertainty in A_t and ϕ is a monotonically decreasing function, such that the clipping range expands whenever U_t is low.

To approximate the distributions defined by the Epistemic Bellman Operator, we present two options: ensembles and a Laplace approximation. In the ensemble implementation of ECPPO, the value network of PPO is replaced by an ensemble $V_1(s), \ldots, V_n(s)$, and the advantages are computed according to each ensemble member k independently:

$$A_t^{(k)} = \delta_t^{(k)} + \gamma\lambda\delta_{t+1}^{(k)} + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1}^{(k)}, \quad (17)$$

$$\delta_t^{(k)} = r_t + \gamma V_k(s_{t+1}) - V_k(s_t). \quad (18)$$

As in standard PPO, the advantages are then normalized $\tilde{A}_t^{(k)} = \frac{A_t^{(k)}-\mu}{\sigma}$ using statistics μ, σ estimated from the minibatch, and the uncertainty is defined as $U_t = \sqrt{\frac{1}{n}\sum_{k=1}^n (\tilde{A}_t^{(k)})^2 - (\frac{1}{n}\sum_{k=1}^n \tilde{A}_t^{(k)})^2}$, which is the empirical standard deviation of the ensemble. The clipping range is modified by a function $\phi(U_t)$ such that $0.5 \le \phi(U_t) \le 2$. For exact specifications we refer to Appendix B.

The Laplace-based version of ECPPO uses a Laplace approximation with diagonal covariance for the value network $V(s)$. To approximate the uncertainty U_t, it is important to keep in mind the covariance between the values within the same trajectory $V(s_t), V(s_{t+1}), \ldots, V(s_T)$. To this end, the advantages A_t are computed with a set of candidate models $V_1(s), \ldots V_n(s)$ drawn from the approximate posterior $\mathcal{N}(\theta^{MLE}, \frac{1}{n}\mathcal{I}(\theta^{MLE})^{-1})$, where θ^{MLE} is the MLE estimator and $\mathcal{I}(\theta^{MLE})$ is the Fisher Information. We refer to Daxberger et al. (2021) for a more in-depth overview of Laplace approximations. The advantages and uncertainty are computed from $V_1(s), \ldots V_n(s)$ analogously to the ensemble-based ECPPO. However, unlike the ensemble-based version, no gradients are computed for these candidate models as they are only used to compute targets. This makes the Laplace version more scalable.

Experiments We test the RL agent with base clipping hyperparameter of $c = 0.2$ on all discrete state environments in Gymnax (Lange 2022), which includes environments from OpenAI Gym (Brockman et al. 2016), BSuite (Osband et al. 2020), MinAtar (Young and Tian 2019), and several miscellaneous environments (Lange and Sprekeler 2022; Miconi et al. 2018; Sutton, Precup, and Singh 1999; Wang et al. 2016), but excluding SimpleBandit-bsuite, which is non-sequential and trivial, and MemoryChain-bsuite, which is non-Markovian.

We compare results against the baseline version of PPO in PureJaxRL (Lu et al. 2022), which also has clipping ratio $c = 0.2$, and is tuned for these environments by the authors. Furthermore, since ECPPO is a modification to the

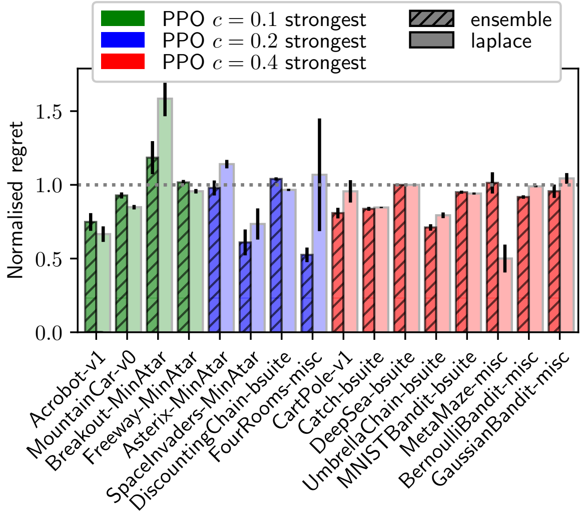

Figure 4: Regret of ECPPO with $c = 0.2$ relative to the regret of the baseline with $c = 0.2$ (lower is better). Environments are grouped and colour-coded by optimal baseline clipping parameter. Average of 20 seeds, with error bars denoting one standard error.

clipping behaviour, we group environments by whether PPO improves with $c = 0.1$ and $c = 0.4$, which are the smallest and highest clipping ratio achievable by ECPPO.

Full experiment details and code are in Appendix B (Van der Vaart, Spaan, and Yorke-Smith 2025), and all learning curves are in Appendix C. To highlight how ECPPO improves over the baseline PPO with fixed c, Figure 4 shows the cumulative regret of ECPPO with $c = 0.2$ w.r.t. the strongest PPO baseline, normalised by the regret of the baseline with $c = 0.2$. The environments are grouped by whether decreasing or increasing c improves baseline performance. It is immediately visible that Ensemble-ECPPO dramatically improves performance across several environments, independent of whether high or low c is optimal in the specific environment, and without suffering major performance penalties in other environments. Laplace-ECPPO also improves performance in several independent on the optimal c, but becomes significantly worse on Breakout. Finally, we observe in Figure 8 (provided in Appendix C) that the uncertainty quantification make sense in a qualitative manner in the FourRooms environment, where uncertainty is high where the current policy has low support.

Related Work

There is a large body of research for Bayesian methods in RL. On the practical side, there are algorithms such as BootDQN (Osband et al. 2016; Osband, Aslanides, and Cassirer 2018), EVE (Schmitt, Shawe-Taylor, and van Hasselt 2023), BDQN (Azizzadenesheli, Brunskill, and Anandkumar 2018), Langevin-DQN (Dwaracherla and Roy 2021), LMCDQN (Ishfaq et al. 2023) and SMC-DQN (Van der Vaart, Yorke-Smith, and Spaan 2024). Our main theoretical result aims to theoretically ground these methods within a general framework by interpreting them as special cases of an EBO, which works on distributions, and prove that this is a contraction.

Operators that work on distributions are also a main focus in Distributional RL (Bellemare, Dabney, and Munos 2017). The goal in distributional RL is to model the distribution of returns, as opposed to learning only the mean. Distributional methods model the aleatoric uncertainty, which is the inherent randomness of returns due to the randomness in the policy and MDP. Instead, we focus on learning the mean of the returns, and compute a distribution over possible means given our observations to model the epistemic uncertainty on the mean. Furthermore, our operator naturally takes into account the dependency and covariance of the Q-values.

Dearden, Friedman, and Russell (1998) discuss a similar operator, also providing convergence guarantees with a contraction argument. This result can be interpreted as a special case of our results with a specific return estimator and specific approximation class. Our main theorem instead applies to any return estimator with contractive properties.

Bayesian Bellman Operators (Fellows, Hartikainen, and Whiteson 2021) also focus on the potentially problematic dependence on target values when inferring posterior distributions over Q-functions. In their work, these problems are alleviated by interpreting Bayesian RL methods as inferring posterior distributions over Bellman Operators, while we directly consider distributions over Q-functions. Furthermore, they focus on a standard one-step Bellman operator with parameterized Q-functions, relying on gradient-based optimization theory to prove convergence in the limit of infinite data under assumptions on the data generating distributions. On the other hand, our results hold for any contraction operator and show existence and consistency for any data set.

Conclusion

We have introduced Epistemic Bellman Operators, which are operators that map a distribution over Q-values to the pushforward of regular Bellman operators with additive noise. We have shown that our operator generalizes several probabilistic reinforcement learning algorithms, unifying practical algorithms that appear to have dissimilar architectures. Furthermore, we have proven that Epistemic Bellman Operators are contractions, which implies that interleaving posterior inference and target updates converges to a fixed distribution and motivates these practical algorithms by showing consistency in tabular settings. We showed that the fixed point of an EBO is sensible when doing policy evaluation. Finally, we showcased the generality of our operators by studying an existing Bayesian Q-learning algorithm and modifying PPO into an uncertainty-aware variant that outperforms the original algorithm in several environments.

In future research, the insights from our main theorem can aid in the design of new uncertainty-aware algorithms by guiding practical design choices toward theoretically sound approaches. Another research direction is to study more applications of uncertainty in reinforcement learning, other than exploration and the one presented here. Finally, we aim to investigate the influence of priors and likelihoods and study more suitable distributions than normal distributions.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme, under grant agreements 964505 (E-pi) and 952215 (TAILOR).

References

Azizzadenesheli, K.; Brunskill, E.; and Anandkumar, A. 2018. Efficient exploration through Bayesian Deep Q-Networks. In *2018 Information Theory and Applications Workshop (ITA)*. IEEE.

Bellemare, M.; Srinivasan, S.; Ostrovski, G.; Schaul, T.; Saxton, D.; and Munos, R. 2016. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, volume 29.

Bellemare, M. G.; Dabney, W.; and Munos, R. 2017. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*. PMLR.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym.

Burda, Y.; Edwards, H.; Storkey, A.; and Klimov, O. 2018. Exploration by random Network Distillation. In *International Conference on Learning Representations*.

Chen, T.; Fox, E.; and Guestrin, C. 2014. Stochastic Gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning*.

D'Angelo, F.; and Fortuin, V. 2021. Repulsive Deep Ensembles are Bayesian. In *Advances in Neural Information Processing Systems*, volume 34.

Daxberger, E.; Kristiadi, A.; Immer, A.; Eschenhagen, R.; Bauer, M.; and Hennig, P. 2021. Laplace redux-effortless bayesian deep learning. In *Advances in Neural Information Processing Systems*.

Dearden, R.; Friedman, N.; and Russell, S. 1998. Bayesian Q-Learning. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference*. AAAI Press / The MIT Press.

Degrave, J.; Felici, F.; Buchli, J.; Neunert, M.; Tracey, B.; Carpanese, F.; Ewalds, T.; Hafner, R.; Abdolmaleki, A.; de Las Casas, D.; et al. 2022. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897).

Dietterich, T. G. 2000. Ensemble Methods in Machine Learning. In *Multiple Classifier Systems*. Springer Berlin Heidelberg.

Dwaracherla, V.; and Roy, B. V. 2021. Langevin DQN. arXiv:2002.07282.

Fawzi, A.; Balog, M.; Huang, A.; Hubert, T.; Romera-Paredes, B.; Barekatain, M.; Novikov, A.; Ruiz, F. J. R.; Schrittwieser, J.; Swirszcz, G.; Silver, D.; Hassabis, D.; and Kohli, P. 2022. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930).

Fellows, M.; Hartikainen, K.; and Whiteson, S. 2021. Bayesian Bellman Operators. In *Advances in Neural Information Processing Systems*, volume 34.

Fortunato, M.; Azar, M. G.; Piot, B.; Menick, J.; Osband, I.; Graves, A.; Mnih, V.; Munos, R.; Hassabis, D.; Pietquin, O.; et al. 2017. Noisy networks for exploration.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*. PMLR.

Ishfaq, H.; Lan, Q.; Xu, P.; Mahmood, A. R.; Precup, D.; Anandkumar, A.; and Azizzadenesheli, K. 2023. Provable and Practical: Efficient Exploration in Reinforcement Learning via Langevin Monte Carlo. arXiv:2305.18246.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems*, volume 30.

Lange, R. T. 2022. gymnax: A JAX-based Reinforcement Learning Environment Library. http://github.com/RobertTLange/gymnax. Accessed: 2024-08-20.

Lange, R. T.; and Sprekeler, H. 2022. Learning not to learn: Nature versus nurture in silico. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI-22)*, volume 36.

Lee, K.; Laskin, M.; Srinivas, A.; and Abbeel, P. 2021. Sunrise: A Simple Unified Framework for Ensemble Learning in Deep Reinforcement Learning. In *International Conference on Machine Learning*.

Liu, Q.; and Wang, D. 2016. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. In *Advances in Neural Information Processing Systems*, volume 29.

Lu, C.; Kuba, J.; Letcher, A.; Metz, L.; Schroeder de Witt, C.; and Foerster, J. 2022. Discovered policy optimisation. In *Advances in Neural Information Processing Systems*, volume 35.

Mandhane, A.; Zhernov, A.; Rauh, M.; Gu, C.; Wang, M.; Xue, F.; Shang, W.; Pang, D.; Claus, R.; Chiang, C.-H.; Chen, C.; Han, J.; Chen, A.; Mankowitz, D. J.; Broshear, J.; Schrittwieser, J.; Hubert, T.; Vinyals, O.; and Mann, T. A. 2022. MuZero with Self-competition for Rate Control in VP9 Video Compression.

Mankowitz, D. J.; Michi, A.; Zhernov, A.; Gelmi, M.; Selvi, M.; Paduraru, C.; Leurent, E.; Iqbal, S.; Lespiau, J.-B.; Ahern, A.; Koppe, T.; Millikin, K.; Gaffney, S.; Elster, S.; Broshear, J.; Gamble, C.; Milan, K.; Tung, R.; Hwang, M.; Cemgil, T.; Barekatain, M.; Li, Y.; Mandhane, A.; Hubert, T.; Schrittwieser, J.; Hassabis, D.; Kohli, P.; Riedmiller, M.; Vinyals, O.; and Silver, D. 2023. Faster sorting algorithms discovered using deep reinforcement learning. *Nature*, 618(7964).

Miconi, T.; Rawal, A.; Clune, J.; and Stanley, K. O. 2018. Backpropamine: training self-modifying neural networks with differentiable neuromodulated plasticity. In *International Conference on Learning Representations*.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*. PMLR.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540).

Osband, I.; Aslanides, J.; and Cassirer, A. 2018. Randomized Prior Functions for Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 31.

Osband, I.; Blundell, C.; Pritzel, A.; and Van Roy, B. 2016. Deep Exploration via Bootstrapped DQN. In *Advances in Neural Information Processing Systems*, volume 29.

Osband, I.; Doron, Y.; Hessel, M.; Aslanides, J.; Sezener, E.; Saraiva, A.; McKinney, K.; Lattimore, T.; Szepesvari, C.; Singh, S.; Roy, B. V.; Sutton, R.; Silver, D.; and Hasselt, H. V. 2020. Behaviour Suite for Reinforcement Learning. In *International Conference on Learning Representations*.

Ostrovski, G.; Bellemare, M. G.; Oord, A.; and Munos, R. 2017. Count-Based Exploration with Neural Density Models. In *International Conference on Machine Learning*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. In *Advances in neural information processing systems*, volume 35.

O'Donoghue, B.; Osband, I.; Munos, R.; and Mnih, V. 2018. The uncertainty Bellman equation and exploration. In *International Conference on Machine Learning*.

Schmitt, S.; Shawe-Taylor, J.; and van Hasselt, H. 2023. Exploration via Epistemic Value Estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37.

Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; Lillicrap, T.; and Silver, D. 2020. Mastering Atari, Go, Chess and Shogi by planning with a learned model. *Nature*, 588(7839).

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2).

Van der Vaart, P. R.; Spaan, M. T. J.; and Yorke-Smith, N. 2025. Code and Supplementary Material for Epistemic Bellman Operators. https://github.com/Pascal314/epistemic-Bellman-operators.

Van der Vaart, P. R.; Yorke-Smith, N.; and Spaan, M. T. J. 2024. Bayesian Ensembles for Exploration in Deep Reinforcement Learning. In *Proceedings of the 2024 International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '24. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Van Hasselt, H. 2010. Double Q-learning. In Lafferty, J.; Williams, C.; Shawe-Taylor, J.; Zemel, R.; and Culotta, A., eds., *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

Van Hasselt, H.; Guez, A.; and Silver, D. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Wang, J. X.; Kurth-Nelson, Z.; Tirumala, D.; Soyer, H.; Leibo, J. Z.; Munos, R.; Blundell, C.; Kumaran, D.; and Botvinick, M. 2016. Learning to reinforcement learn.

Wenzel, F.; Roth, K.; Veeling, B.; Swiatkowski, J.; Tran, L.; Mandt, S.; Snoek, J.; Salimans, T.; Jenatton, R.; and Nowozin, S. 2020. How Good is the Bayes Posterior in Deep Neural Networks Really? In *International Conference on Machine Learning*.

Young, K.; and Tian, T. 2019. MinAtar: An Atari-Inspired Testbed for Thorough and Reproducible Reinforcement Learning Experiments. arXiv:1903.03176.