Wasserstein Actor-Critic: Directed Exploration via Optimism for Continuous-Actions Control

Amarildo Likmeta[1,2]**, Matteo Sacco**[2]**, Alberto Maria Metelli**[2]**, Marcello Restelli**[2]

[1]University of Bologna
[2]Politecnico di Milano

amarildo.likmeta2@unibo.it, matteo3.sacco@mail.polimi.it, albertomaria.metelli@polimi.it, marcello.restelli@polimi.it

Abstract

Uncertainty quantification has been extensively used as a means to achieve efficient directed exploration in Reinforcement Learning (RL). However, state-of-the-art methods for continuous actions still suffer from high sample complexity requirements. Indeed, they either completely lack strategies for propagating the epistemic uncertainty throughout the updates, or they mix it with aleatoric uncertainty while learning the full return distribution (e.g., distributional RL). In this paper, we propose Wasserstein Actor-Critic (WAC), an actor-critic architecture inspired by the recent Wasserstein Q-Learning (WQL), that employs approximate Q-posteriors to represent the epistemic uncertainty and Wasserstein barycenters for uncertainty propagation across the state-action space. WAC enforces exploration in a principled way by guiding the policy learning process with the optimization of an upper bound of the Q-value estimates. Furthermore, we study some peculiar issues that arise when using function approximation, coupled with the uncertainty estimation, and propose a regularized loss for the uncertainty estimation. Finally, we evaluate our algorithm on standard MujoCo tasks as well as suite of continuous-actions domains, where exploration is crucial, in comparison with state-of-the-art baselines. Additional details and results can be found in the supplementary material with our Arxiv preprint.

1 Introduction

Reinforcement Learning (RL, Sutton and Barto 2018) is one of the most widely used frameworks for solving sequential decision-making problems. When an agent acts in an uncertain environment, it faces the choice between *exploring* with the hope of discovering more profitable behaviors or *exploiting* the current information about the actions' values. This exploration-exploitation dilemma is particularly challenging in *continuous-state* spaces, where *function approximation* is required to generalize across states, and an accurate estimate of the uncertainty on the value estimates is not available point-wise. *Continuous-action* tasks pose additional challenges since most exploration methods require the maximization of some objective (e.g., upper bound of the Q-value) over the action space. While in the discrete case, this maximization can be performed by enumeration, in the continuous case it requires solving a complex optimization problem, increasing the computational demands.

Actor-Critic (AC) methods (Haarnoja et al. 2018; Ciosek et al. 2019; Schulman et al. 2015) represent the current state of the art for continuous control. Despite their widespread adoption, these methods still suffer from high sample complexity. Efficient exploration strategies have been extensively studied in the literature as a means of reducing sample complexity mainly in tabular domains (Auer, Jaksch, and Ortner 2008; Ian, Benjamin, and Daniel 2013; O'Donoghue et al. 2018; Metelli, Likmeta, and Restelli 2019). Classical exploration strategies, like ϵ-greedy or Boltzmann (Sutton and Barto 2018), inject noise around the current greedy policy to enforce exploration. Although in simple settings this is enough to guarantee convergence (Szepesvári 1997), this exploration strategy is not efficient in the general case.

A common trend in the RL literature consists of endowing existing methods with some form of uncertainty quantification and using it to perform *directed* exploration while focusing on the most promising regions. In particular, a recent extension of Soft Actor-Critic (SAC, Haarnoja et al. 2018), Optimistic Actor-Critic (OAC, Ciosek et al. 2019), proved to improve sample efficiency over the standard SAC. Indeed, uncertainty quantification is a fundamental step to define efficient exploration strategies. Exploration strategies, coming from the Multi-Armed Bandit (MAB, Lattimore 2020) literature, have been extended for the RL settings, starting from tabular domains (Auer, Jaksch, and Ortner 2008; Ian, Benjamin, and Daniel 2013; Metelli, Likmeta, and Restelli 2019), with theoretical guarantees on the sample complexity and/or regret. In turn, they have been extended to the Deep Reinforcement Learning (DRL) settings too, but the guarantees no longer hold up. Ensemble methods allow quantifying the uncertainty but do not *propagate* it across the state action-space when performing the critic updates. Uncertainty propagation is a fundamental tool of any principled uncertainty estimation approach since most AC methods rely on bootstrapping when updating the critics. This results in Q-value estimates that also incorporate uncertainty about the bootstrapped values. Distributional RL (O'Donoghue et al. 2018) allows for uncertainty propagation but considers only aleatoric uncertainty, being aimed at estimating the full return distribution.

In this paper, we address the problem of uncertainty estimation and propagation in the context of continuous-action RL. Starting from the methodology introduced in WQL (Metelli, Likmeta, and Restelli 2019), we devise a novel actor-critic

algorithm, *Wasserstein Actor-Critic* (WAC), which employs Q-posteriors both to quantify uncertainty on the critic estimates to drive exploration, as well as a tool to propagate it across the state-action space (Section 3). Furthermore, we consider some practical problems that arise while quantifying uncertainty by means of Q-posteriors coped with function approximators, especially neural networks. To this end, we propose a regularization approach for the uncertainty networks (Section 4). After reviewing the literature (Section 5), we present a thorough experimental evaluation over some simple 1D navigation domains, as well as some MuJoCo (Todorov, Erez, and Tassa 2012) tasks designed for exploration to assess the effect of uncertainty estimation and propagation on exploration and sample complexity (Section 6).

2 Preliminaries

Markov Decision Processes We consider infinite-horizon discounted Markov Decision Processes (MDP, Puterman 2014). An MDP is a 5-tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, defined by the state space $\mathcal{S}$, the action space $\mathcal{A}$, a transition kernel $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, a reward $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathbb{R})$ and the discount factor $\gamma \in [0, 1)$.[1] Let $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be the expectation of the reward $\mathcal{R}$ that we assume bounded in $[r_{\min}, r_{\max}]$. The behavior of an agent is described by a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$. The performance of a policy π is measured by its action-value function defined as $Q^\pi(s, a) = \mathbb{E}[\sum_{t=0}^\infty \gamma^t r_t | s_0 = s, a_0 = a]$, where we fix the first action and follow policy π for the next steps. The value functions satisfy the Bellman equations, $Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}[Q^\pi(s', a')]$ for every $(s, a) \in \mathcal{S} \times \mathcal{A}$. The optimal action-value function Q^* is the maximum, over all policies, of the Q function, for all state-action pairs, $Q^*(s, a) = \sup_\pi \{Q^\pi(s, a)\}$. Q^* satisfies the Bellman optimality equation $Q^*(s, a) = r(s, a) + \gamma \mathbb{E}[\sup_{a' \in \mathcal{A}} \{Q^*(s', a')\}]$. The Bellman equations form the basis of Temporal Difference (TD) learning, which updates the estimation of the Q function in the current state using estimates of the next-states Q function. The goal of learning algorithms in this setting is to find the optimal policy π^*, which is defined as the policy that acts greedily w.r.t. Q^*, $\pi^*(\cdot|s) \in \Delta(\arg\max_{a \in \mathcal{A}} \{Q^*(s, a)\}), \forall s \in \mathcal{S}$.

Actor-Critic Methods AC methods maintain a parameterized value-function Q_ω (*critic*) to estimate the value of the current (or a given target) policy, and a parameterized policy π_θ (*actor*), trained through gradient descent. In particular, SAC, employs an *entropy-regularized* architecture, i.e., the agents optimize a modified objective regularized with the entropy of the policy favoring stochastic policies over deterministic ones, shown in Equation 2. Specifically, it maintains two parameterized action-value functions $\{Q_{\omega_1}, Q_{\omega_2}\}$ to estimate the entropy-regularized value function of policy π_θ. They are trained on the same samples and differ only on the initialization of ω_1 and ω_2. The actor optimizes a "lower bound" of the action-value function, $Q_{LB}(s, a) = \min\{Q_{\omega_1}(s, a), Q_{\omega_2}(s, a)\}$. To update the critic, given a sample (s, a, r, s'), SAC uses the SARSA (Sutton and Barto

2018) update rule, $Q_{\{\omega_1, \omega_2\}}(s, a) \leftarrow r + \gamma Q_{LB}(s', a')$, where $a' \sim \pi_\theta(s')$. Specifically, SAC maintains experience collected with previous policies π_θ in a *replay buffer* $\mathcal{D}$ (Sutton and Barto 2018). The critic is trained to minimize the (entropy regularized) Bellman error over this replay buffer, as follows:

$$J_C(\{\omega_1, \omega_2\}) = \mathbb{E}_{s,a,r,s' \sim \mathcal{D}} \Big[(Q_{\{\omega_1, \omega_2\}}(s, a) - \\ (r + \gamma \widetilde{Q}(s', a')))^2 \Big], \quad (1)$$

where $\widetilde{Q}(s, a) = \overline{Q}_{LB}(s, a) - \alpha \log \pi_\theta(s', a')$, $\overline{Q}_{LB}(s, a) = \min\{Q_{\overline{\omega}_1}(s, a), Q_{\overline{\omega}_2}(s, a)\}$ is the lower bound of the Q-values given by two target networks which are updated slowly to improve stability (Mnih et al. 2015), $a' \sim \pi_\theta(s')$, and $\alpha > 0$ specifies the level of entropy regularization. The actor network is trained to optimize an entropy-regularized objective. Since the target Q-function is a parameterized function approximator, the policy can directly follow the gradient of the critic:

$$J_A(\theta) = \mathbb{E}_{\substack{s_t \sim \mathcal{D} \\ a_t \sim \pi_\theta(s_t)}} [\log \pi_\theta(s_t, a_t) - Q_{LB}(s_t, a_t)]. \quad (2)$$

Wasserstein TD-Learning Bayesian approaches to RL (Dearden, Friedman, and Russell 1998; Metelli, Likmeta, and Restelli 2019) maintain, for each state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, a probability distribution $\mathcal{Q}(s, a)$, called a *Q-posterior*, used to represent the *epistemic* uncertainty over the value estimates. In practice, $\mathcal{Q}$ is an approximate distribution in a class $\mathcal{Q}$ (e.g., Gaussians). Using the concept of barycenter, we can also propagate the uncertainty of the value function estimates across the state-action space. As in (Metelli, Likmeta, and Restelli 2019), we employ the notion of barycenter defined in terms of Wasserstein (Villani 2008) divergence since the variance of our Q-posteriors vanishes as the number of samples grows to infinity, and the Wasserstein divergence allows computing the distance between distributions with disjoint support. Given two probability distributions, μ and ν, the L_p-Wasserstein distance between μ and ν is defined as: $W_p(\mu, \nu) = (\inf_{\rho \in \Gamma(\mu, \nu)} \mathbb{E}_{X, Y \sim \rho}[d(X, Y)^p])^{1/p}$, where $\Gamma(\mu, \nu)$ is the set of all joint distributions with marginals μ and ν, and d is a metric (i.e., we use the L_2-norm). Given a class of probability distributions $\mathcal{N}$, a set of probability distributions $\{\mu_i\}_{i=1}^n$, $\mu_i \in \mathcal{N}$ and a set of weights $\{\xi_i\}_{i=1}^n$, $\sum_{i=1}^n \xi_i = 1$ and $\xi_i \geq 0$, the L_2-Wasserstein barycenter is defined as (Agueh and Carlier 2011): $\overline{\mu} \in \arg\min_{\mu \in \mathcal{N}} \{\sum_{i=1}^n \xi_i W_2(\mu, \mu_i)^2\}$. From this, having observed a transition (s, a, r, s'), the *Wasserstein Temporal Difference* (WTD, Metelli, Likmeta, and Restelli 2019) update rule is defined via the computation of the barycenter of the current Q-posterior and the *TD- target posterior*, defined as $\mathcal{T}_t = r + \gamma \mathcal{Q}_t(s', a')$:

$$\mathcal{Q}_{t+1}(s, a) \in \arg\min_{\mathcal{Q} \in \mathcal{Q}} \Big\{ (1 - \alpha_t) W_2(\mathcal{Q}, \mathcal{Q}_t(s, a))^2 + \\ \alpha_t W_2(\mathcal{Q}, \mathcal{T}_t)^2 \Big\}, \quad (3)$$

---

[1]$\Delta(\mathcal{X})$ denotes the set of probability distributions over $\mathcal{X}$.

Algorithm 1: Wasserstein Actor-Critic.

Input: critic parameters $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2$, policy parameters $\boldsymbol{\theta}, \boldsymbol{\theta}^T$
Initialize $\mathcal{Q}_{\{1,2\}}(s,a)$ with the prior $\mathcal{Q}_0$
Initialize replay buffer $\mathcal{D} \leftarrow \emptyset$
for epoch $= 1, 2, ...$ **do**
   for $t = 1, 2, ...$ **do**
      Take action $a_t \sim \pi_{\boldsymbol{\theta}}(\cdot | s_t)$
      Observe s_{t+1} and r_{t+1}
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r_{t+1}, s_{t+1})\}$
   end for
   $\sigma_{\text{old}}^{\{1,2\}} \leftarrow \sigma_{\boldsymbol{\omega}_{\{1,2\}}}$
   for iteration $= 1, 2, ...$ **do**
      Update critic weights $\boldsymbol{\omega}_{\{1,2\}}$ using Equation (8)
      Update actor (resp. target) weights $\boldsymbol{\theta}$ (resp. $\boldsymbol{\theta}^T$) using Equation (6) (resp. Equation (7))
   end for
end for

---

where α_t is the learning rate, and a' is the action taken in the next step. Depending on the policy used to select action a', the update can be either on-policy or off-policy. The presence of γ in the definition of $\mathcal{T}_t$ shrinks the posteriors, vanishing the uncertainty when the number of samples grows to infinity. When the Q-posteriors become point estimates, the update rule reduces to the classic TD update rule.

3 Wasserstein Actor-Critic

In this section, we introduce Wasserstein Actor-Critic (WAC), which extends WQL to handle environments with continuous-action spaces. We present the algorithm, define the update rules, and a regularization for the uncertainty estimates.

Distributional Critic For each state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, we maintain an approximate distribution $Q(s,a) \sim \mathcal{Q}(s,a)$, to model the uncertainty estimate on the value function. While these distributions will generally depend on the aleatoric uncertainty of the environment (state transition and reward), our updates will vanish the variance as we collect samples. This represents our main difference w.r.t. Distributional RL (Bellemare, Dabney, and Munos 2017), as we do not require learning the whole return distribution, while still propagating uncertainty across the state-action space. More specifically, given a replay buffer of past behavior $\mathcal{D}$, our critic minimizes the L_2-Wasserstein distance between the Q-posterior $\mathcal{Q}_{\boldsymbol{\omega}}$ and the target posterior $r + \gamma \mathcal{Q}_{\overline{\boldsymbol{\omega}}}$, defined through the target parameters $\overline{\boldsymbol{\omega}}$ and target policy $\pi_{\boldsymbol{\theta}^T}$:

$$J_C(\boldsymbol{\omega}) = \mathbb{E}_{s,a,s',r\sim\mathcal{D}}\Big[W_2\Big(\mathcal{Q}_{\boldsymbol{\omega}}(s,a), \\ r + \gamma \mathcal{Q}_{\overline{\boldsymbol{\omega}}}\big(s', \pi_{\boldsymbol{\theta}^T}(s')\big)\Big)^2\Big]. \quad (4)$$

Different flavors of the algorithm can be proposed, based on the combination of: (i) distribution classes $\mathscr{Q}$, (ii) behavioral policy $\pi_{\boldsymbol{\theta}}$, and (iii) target policy $\pi_{\boldsymbol{\theta}^T}$. We focus on optimistic exploration, that requires optimizing upper bounds. Moreover, although other distribution classes, like particle-models, could be employed, we limit our discussion to Gaussian posteriors, as their parametrization allows for direct control over the distribution variance.

Similar to WQL, we maintain a parameterized distributional critic using a function approximator (e.g., neural network) that outputs the parameters of the distribution. For the Gaussian case, $Q(s,a) \sim \mathcal{N}(\mu_{\boldsymbol{\omega}}(s,a), \sigma_{\boldsymbol{\omega}}(s,a))$, the Wasserstein distance has a closed form, and the critic objective becomes:

$$J_C(\boldsymbol{\omega}) = \mathbb{E}_{s,a,s',r\sim\mathcal{D}}\Big[\big(\mu_{\boldsymbol{\omega}}(s,a) - (r + \gamma\widetilde{\mu}_{\overline{\boldsymbol{\omega}}}(s', \pi_{\boldsymbol{\theta}^T}(s')))\big)^2 \\ + \big(\sigma_{\boldsymbol{\omega}}(s,a) - \gamma\sigma_{\overline{\boldsymbol{\omega}}}(s', \pi_{\boldsymbol{\theta}^T}(s'))\big)^2\Big], \quad (5)$$

where $\widetilde{\mu}_{\overline{\boldsymbol{\omega}}}(s,a) = \mu_{\overline{\boldsymbol{\omega}}}(s,a) - \alpha \log \pi_{\boldsymbol{\theta}^T}(s,a)$. In practice, $\mu_{\boldsymbol{\omega}}$ and $\sigma_{\boldsymbol{\omega}}$ can use either a shared network architecture or two different networks. We initialize the posterior networks using the bias of the last layer of the network. If the reward function is limited in the interval $[r_{\min}, r_{\max}]$, the Q values will be in the range $[q_{\min}, q_{\max}]$ with $q_{\min} = r_{\min}/(1-\gamma)$ and $q_{\max} = r_{\max}/(1-\gamma)$. We therefore initialize the uncertainty networks to $\sigma_0 = (q_{\max} - q_{\min})/\sqrt{12}$, i.e. variance of the Gaussian minimizing the KL divergence with the uniform distribution in $[q_{\min}, q_{\max}]$ (Metelli, Likmeta, and Restelli 2019).

Actor The actor in WAC is updated by optimizing an *upper bound* $U_{\boldsymbol{\omega}}^{\delta}$ of the estimated Q-value, which we can efficiently compute using Gaussian posterior: $U_{\boldsymbol{\omega}}^{\delta}(s,a) = \mu_{\boldsymbol{\omega}}(s,a) + \sigma_{\boldsymbol{\omega}}(s,a)\Phi^{-1}(\delta)$, where Φ^{-1} is the quantile function of the standard normal and $\delta \in (0,1)$. When actions are finite, no actor is needed, as we can compute the maximum by enumeration. However, in the continuous-action case, we need an actor that follows $U_{\boldsymbol{\omega}}^{\delta}(s,a)$, which is differentiable in $\boldsymbol{\omega}$, leading to the minimization of the objective:

$$J_A(\boldsymbol{\theta}) = \mathbb{E}_{\substack{s_t\sim\mathcal{D}\\a_t\sim\pi_{\boldsymbol{\theta}}(s_t)}}\big[\log\pi_{\boldsymbol{\theta}}(s_t, a_t) - U_{\boldsymbol{\omega}}^{\delta}(s_t, a_t)\big], \quad (6)$$

where $\boldsymbol{\theta}$ are the parameters of the behavioral policy.

Target Policy We propose two alternatives for the target policy $\pi_{\boldsymbol{\theta}^T}$, corresponding to different estimators for the target posterior $\mathcal{T}_t$. First, we can use the same policy we use for exploration, i.e., $\boldsymbol{\theta} = \boldsymbol{\theta}^T$, like SAC. This has the advantage of not requiring a second parameterized policy. We call this version *Optimistic Estimator-WAC* (OE-WAC), which represents an on-policy algorithm. Alternatively, we can use a *greedy* policy that optimizes the expected value of the Q-posteriors (the mean critic $\mu_{\boldsymbol{\omega}}(s,a)$ in the Gaussian case). In this case, the target policy minimizes:

$$J_T(\boldsymbol{\theta}_T) = \mathbb{E}_{\substack{s_t\sim\mathcal{D}\\a_t\sim\pi_{\boldsymbol{\theta}^T}(s_t)}}\big[\log\pi_{\boldsymbol{\theta}^T}(s_t, a_t) - \mu_{\boldsymbol{\omega}}(s_t, a_t)\big]. \quad (7)$$

We call this version *Mean Estimator-WAC* (ME-WAC). The best version to use between the two is task-dependent. Generally, OE-WAC is more suitable for environments that require large exploration, whereas ME-WAC is more suitable for simpler environments where OE-WAC might over-explore and might suffer from some instability.

Remark 1 (What is our Critic Estimating?)**.** *We underline that our distributional critic maintains uncertainty about the*
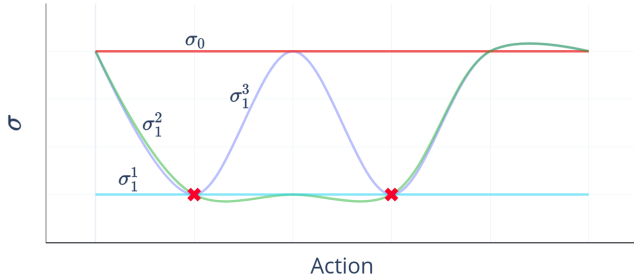
Figure 1: Example of uncertainty estimates. σ_0 shows the initial constant high value.

Q^* and not about the Q-function of the current policy Q^{π_θ}. Indeed, the method starts with an initial high-uncertainty estimate and updates it as we collect samples from the environment. In the tabular case, it can be proven that these upper-bounds on the value of Q^* are valid with high probability for every timestep t, under some conditions on the learning rate α_t (Metelli, Likmeta, and Restelli 2019). When extending the method to DeepRL, these guarantees are no longer valid, since the uncertainty estimates are outputs of general function approximators and local updates are no longer possible.

4 Regularized Uncertainty Estimation

Our Q-posteriors are initialized to high uncertainty at the beginning of the learning process. Since they represent epistemic uncertainty, their variance will shrink as we observe more samples. This is apparent in Equation (5), where the targets σ_ω are multiplied with γ. In tabular settings, the updates are localized, i.e., they affect a single state-action pair, without interfering with the others. However, when function approximators are involved, generalizing uncertainty in an uncontrolled way might cause non-visited areas of the state-action space to take low uncertainty values, which might be undesired.[2] Consider the example in Figure 1 showing the uncertainty estimate as a function of the action, in a fixed state s. Starting from an initial high constant estimate of σ_0, at the beginning of the learning process, we will observe samples like the red crosses in the figure, i.e., with lower uncertainty since it gets shrunk with γ. Among all the possible fitting lines, we would prefer an estimate like σ_1^3, which keeps high uncertainty in unseen regions, and would like to avoid failures like σ_1^1. This requires controlling the "smoothness" properties of the approximator. To avoid the additional computational burden, we propose a simple scheme based on *synthetic* samples. Specifically, we periodically save the weights of the uncertainty network σ_{old} and use it as the target for state-action pairs drawn *uniformly* from the state-action

---

[2]This generalization phenomenon happens for the mean too, but, as visible in Equation (5), is particularly critical for the variance that gets updated with the next-state-action variance scaled by $\gamma < 1$.

space. More formally, our distributional critic minimizes:

$$J'_C(\boldsymbol{\omega}_{\{1,2\}}) = J_C(\boldsymbol{\omega}_{\{1,2\}}) + $$
$$\lambda \mathop{\mathbb{E}}_{s,a \sim \mathcal{U}(\mathcal{S} \times \mathcal{A})} \left[\left(\sigma_{\boldsymbol{\omega}_{\{1,2\}}}(s,a) - \sigma_{old}(s,a)^2 \right)^2 \right], \quad (8)$$

where $J_C(\boldsymbol{\omega}_{\{1,2\}})$ is defined in Equation (5) and $\lambda \geq 0$ defines the relative weight of the regularization. Furthermore, in practice we add a second parameter, $\rho \in [0,1]$ which represents the fraction of *fake* samples (w.r.t. the samples used for $J_C(\boldsymbol{\omega}_{\{1,2\}})$) drawn for regularization. Specifically, if we estimate $J_C(\boldsymbol{\omega}_{\{1,2\}})$ using N samples from replay buffer $\mathcal{D}$, we will estimate the expectation in Equation (8) with $M = \rho N$ samples from $\mathcal{U}(\mathcal{S} \times \mathcal{A})$. Algorithm 1 reports the pseudocode of WAC, embedding the regularized uncertainty estimation.

To investigate the effectiveness of the regularized uncertainty loss, on an illustrative example, we trained two different agents, in a one-dimensional Linear Quadratic Regulator (LQG, Dorato, Cerone, and Abdallah 2000). This task has a one-dimensional state and action spaces, which allows us to visualize the uncertainty estimates. Figure 2 shows the resulting uncertainty estimates. On the left, we show the empirical state-action visitation distribution. The agent starts in one of the borders of the state space and has to reach the center in a few steps while calibrating the actions. This is apparent in the histogram, with the highest densities in the borders and the center. We consider it desirable to obtain uncertainty estimates that mirror these state-action densities, as the epistemic uncertainty is inversely proportional to the state-action visitation. While in both cases, the state-action densities are similar, the uncertainty estimates are completely different. In Figure 2b, we see that without regularization, the critic completely fails to represent the uncertainty. In Figure 2c, we can see that the regularized uncertainty critic, almost perfectly matches the state-action densities. In Section 6 and Appendix B, we show a more thorough investigation of the effect of the regularized uncertainty loss.

Remark 2 (Where does WAC differ from WTD?). *While WAC is a direct extension of Wasserstein TD-learning to the actor-critic architecture, it does come with some modifications (mostly to regularize the learning process). Indeed, the entropy setting we employ in the critic and actor fit is not a part of the base framework. Wasserstein TD-learning, in it's pure form, would employ deterministic policies. In practice, we observed that this, coupled with optimistic exploration, caused instability in the learning process. Adding entropy regularization improves stability while not decreasing significantly the exploration capabilities of the method. Moreover, the regularization term in Equation (8) is added due to the use of general function approximators for generalizing uncertainty, and was not found in the main Wasserstein TD-framework. We use the regularization term as a way to control the "smoothness" of the function approximator. The synthetic samples do not necessarily need to be "valid" state-action pairs. They need to be in the input space of the approximator. Indeed, in our implementation we simply uniformly sample in $[-1,1]^{nS+nA}$ (where nS and nA are the dimension of the state and action space, each normalized in $[-1,1]$).*

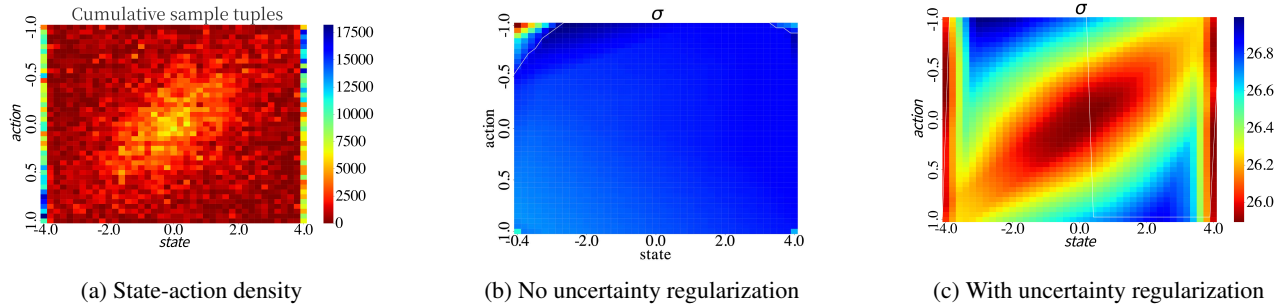| (a) State-action density | (b) No uncertainty regularization | (c) With uncertainty regularization |

Figure 2: Comparison on the uncertainty estimates after training with and without uncertainty regularization in the LQG illustrative example (action on the y axis and state on the x axis).

5 Related Works

There exists a large body of literature studying efficient exploration techniques in RL. In the tabular settings, *provably efficient* methods have been devised, both in the model-based (Jaksch, Ortner, and Auer 2010; Ian, Benjamin, and Daniel 2013) and model-free (Strehl et al. 2006; Jin et al. 2018) settings. These methods cannot be easily extended to the Deep RL setting, or when extensions are proposed, they lose their theoretical guarantees. In this section, we focus on tractable exploration methods proposed for Deep RL for continuous action spaces. Two main exploration frameworks exist: *uncertainty-based* methods and *intrinsic motivation* methods (Houthooft et al. 2016; Zhang et al. 2021; Mutti, Pratissoli, and Restelli 2021). For space reasons, we will focus our discussion on uncertainty-based methods.

Classical value-based methods (including ACs) maintain a point estimate of the value functions for each state (or state-action pairs). Exploration policies, like ϵ-greedy or Boltzmann (Sutton and Barto 2018), add noise around the greedy action derived from these point estimates. These methods are not efficient, mainly because the exploration is not *directed* towards unvisited regions of the state space. The entropy regularization of SAC is a form of undirected exploration too, as the policies are trained to sacrifice some returns to preserve stochastic behavior. In recent years, several methods that move away from point estimates have been proposed. Ensemble methods (Chen et al. 2021; Wang, Lin, and Zhang 2021) implicitly model the epistemic uncertainty of the Q-value estimates by maintaining multiple Q-function approximators. OAC (Ciosek et al. 2019) explicitly models the uncertainty on the value estimates by computing the variance of two critics, and it uses it to compute an exploration policy that optimizes an upper bound of the Q-values. Unfortunately, this uncertainty estimate is just heuristic and only stems from the disagreement between the two Q-networks with different initialization. Indeed, the networks are also trained with the same samples, and same target Q-values, so any disagreement is purely due to the random initialization only. Recently, SUNRISE (Lee et al. 2021) proposes a framework to unify ensemble methods for epistemic uncertainty estimation and shows considerable performance improvements in discrete and continuous action spaces. Distributional RL, on the other hand, models the *aleatoric* uncertainty, as its goal is

to estimate the whole return distribution. First proposed for problems with a discrete action space (Bellemare, Dabney, and Munos 2017; Dabney et al. 2018; Mavrin et al. 2019), it has been successfully extended also to the AC setting in TOP (Moskovitz et al. 2021). TOP models both aleatoric and epistemic uncertainty and adapts the level of optimism/pessimism by means of a MAB approach. While TOP deals with uncertainty propagation, it mixes the epistemic and aleatoric uncertainty while estimating the return distribution.

6 Experiments

In this section, we present the empirical evaluation of WAC in various continuous control domains. We start from simple 1D-navigation, where we can better visualize the effects of the Q-posteriors in the learning and exploration process. In Appendix B, we show an evaluation on several standard MuJoCo tasks, which show that this suite of environments does not pose significant exploration challenges. Hence, we focus our evaluation of WAC on a set of MuJoCo tasks specifically designed for exploration. Our results can be reproduced using the source code in https://github.com/amarildolikmeta/wac_explore.

1D Navigation To measure the effect of uncertainty estimation on exploration we keep track of the cumulative *coverage* of the state-action space, i.e., the portion of the total volume visited with relative frequency larger than $\epsilon > 0$. We consider a one-dimensional LQG, an environment with no particular exploration challenges, and a more challenging continuous-action version of the Riverswim (Strehl and Littman 2008), where long sequences of rewardless actions are needed to reach high reward states. A full description of the environments is reported in Appendix A. Figure 3 shows the results of these experiments. For each environment, we train WAC, varying the parameters λ and ρ of the regularized uncertainty loss in Equation (8). For each value, we report the coverage averaged over all training epochs. Firstly, we observe that the coverage is monotonically increasing with both λ and ρ. As expected, low values of ρ cause higher variance, as fewer samples are employed to estimate the uncertainty regularization. This can be seen in all the curves of the leftmost plot, as well as in the third plot, where the black curve corresponding to $\rho = 0.25$ suffers from a high variance. In Appendix B,
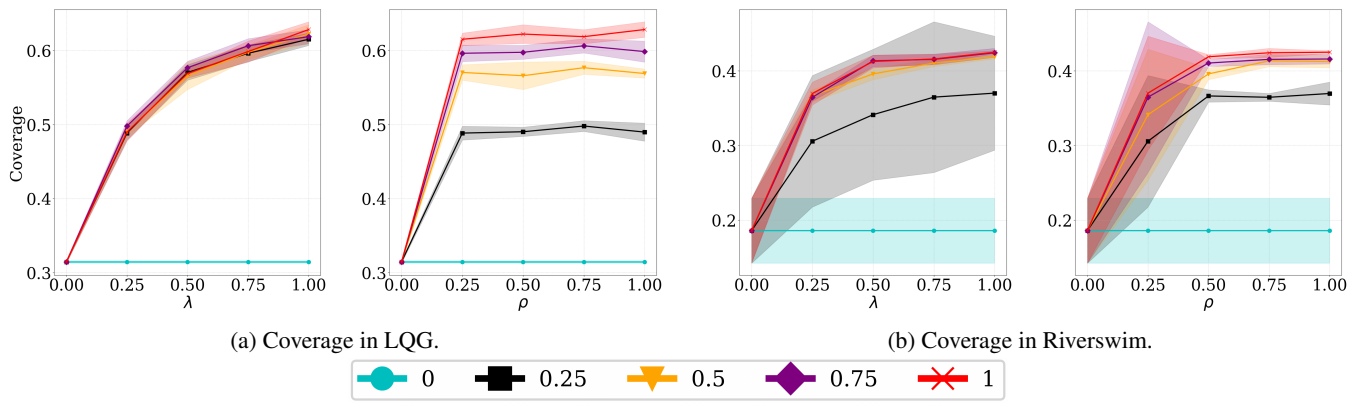
(a) Coverage in LQG.

(b) Coverage in Riverswim.

| 0 | 0.25 | 0.5 | 0.75 | 1 |

Figure 3: Coverage in LQG and Riverswim as function of λ and ρ; average of 5 seeds, 95% c.i..

(a) Average return in 4 2D navigation tasks.

(b) Number of episodes completed in 3000 steps in 4 2D navigation tasks.

(c) Number of episodes completed in 3000 steps in 4 2D navigation tasks with sparse reward function.

| SAC | OAC | OE-WAC | ME-WAC |
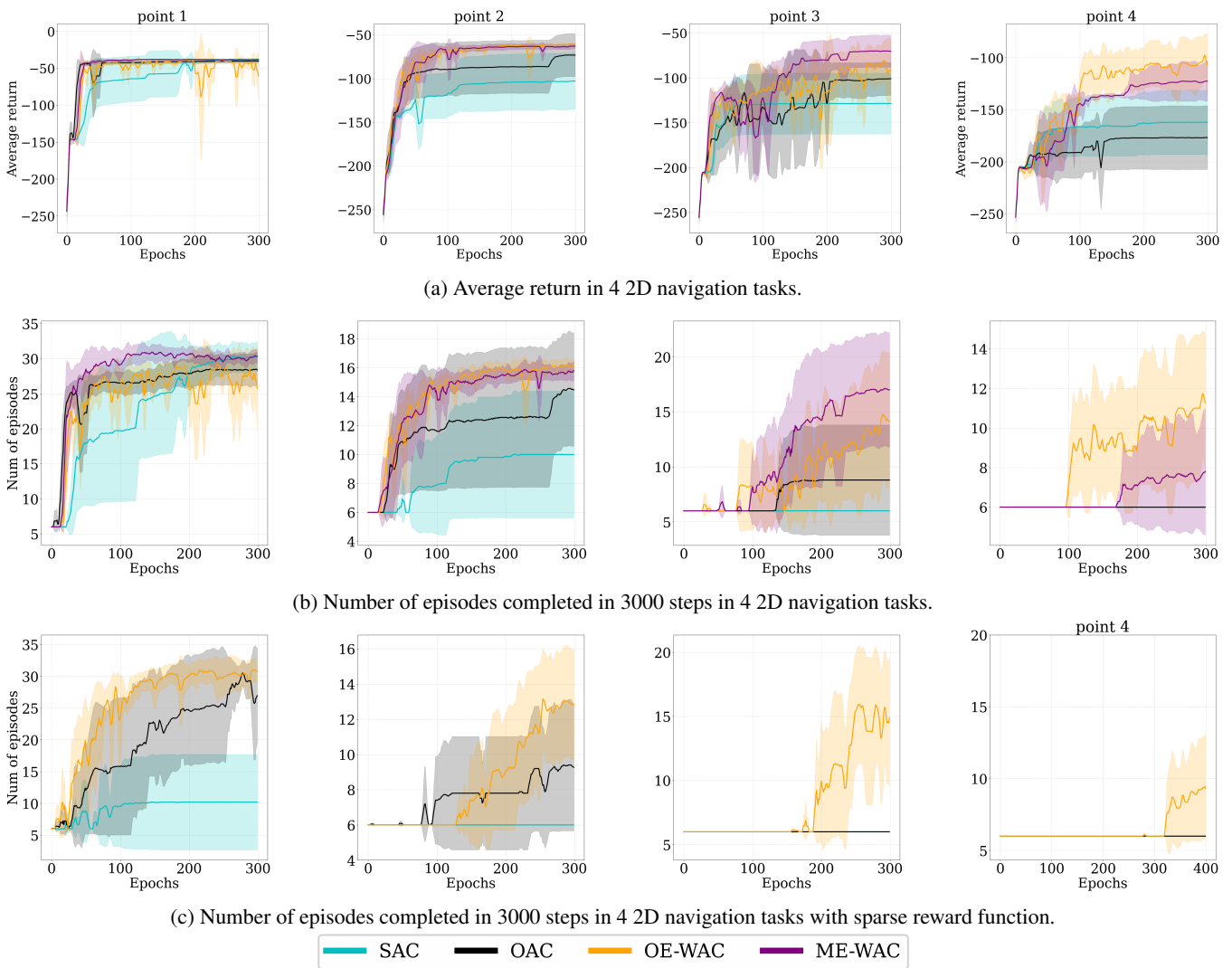
Figure 4: Experimental results in 4 2D navigation tasks starting from the easiest (left) to the hardest (right); average of 5 seeds, 95% c.i..

we perform a similar study for OAC and we observe that the coverage is not so easily controllable. We attribute this to the heuristic nature of the uncertainty estimation of OAC.

2D Navigation To assess whether a principled uncertainty estimation and propagation translate into lower sample complexity, we perform an empirical evaluation in a set of MuJoCo (Todorov, Erez, and Tassa 2012) tasks, where the amount of exploration needed to solve the task can be controlled. We start from the 2D navigation task used in (Moro et al. 2022), where the agent has to reach a goal state in a 2D world, by avoiding obstacles. The reward is the negative Euclidean distance from the goal state. While this is a *dense reward*, the obstacle presence generates local optima which the agent needs to overcome by exploring efficiently. We progressively make the task more challenging by adding additional walls and obstacles. We leave the full description of the environments in Appendix A. We name the tasks as "Point x" with $x \in \{1, 2, 3, 4\}$, where a higher x means a more difficult exploration challenge. We compare the performance of WAC, in both versions defined in Section 3, with SAC and OAC. In each task, we track the cumulative return, as well as the number of episodes completed in a fixed number of steps (higher is better). We use the implementation of SAC and OAC used in (Ciosek et al. 2019), and extend the repository with our implementation, to guarantee comparable results. The same network architectures are used for all algorithms. For the common hyperparameters, we only tune SAC and use the same values for WAC and OAC, by additionally tuning the algorithm specific parameters (δ and β for OAC and λ and ρ for WAC). Details on the hyperparameter tuning are in Appendix A.

In Figure 4a, we present the average return as a function of the training epochs, whereas in Figure 4b we present the number of episodes completed in 3000 steps of interaction. Starting from left to right, we increase the difficulty of the task. We can see that for the easiest task, all algorithms are able to find the optimal policy of quickly avoiding an obstacle in the middle to reach the goal state, even though SAC learns slower compared to the others. Being a simple exploration task, ME-WAC performs better and is more stable than OE-WAC. OAC is also able to quickly solve the task. While the difference in return is negligible, the number of completed episodes shows an advantage for ME-WAC, which completes more episodes faster. Finally, we underline that even though the task does not require particular exploration, WAC does not over-explore, but rather solves with a speed comparable with the other baselines. The clear advantages of WAC in terms of exploration can be seen starting from the second task, where the exploration requirements are increased. Both versions of WAC learn faster and with less variance compared with both SAC and OAC. The difference is even more apparent in the number of episodes completed, where SAC and WAC have disjoint confidence intervals. In the third task, SAC completely fails in learning to reach the goal, while OAC succeeds in some of the seeds only, showing a high variance. WAC, on the other hand, outperforms them both in terms of return and completed episodes. ME-WAC performs better, even though the task requires a good amount of ex-

ploration. Compared to ME-WAC, OE-WAC over-explores and it shows a slower learning curve. The last task is solved by the WAC agents only. SAC and OAC never reach the goal state. WAC outperforms them, in both versions, with statistical significance. We also see the need for larger exploration, apparent from the difference in performance between OE-WAC and ME-WAC.

Finally, Figure 4c presents the number of episodes completed in a *sparse reward* version of the same tasks. In this scenario, we do not show the return as it is proportional to the number of completed episodes. We only trained OE-WAC agents in these tasks, as they present a substantial exploration challenge. The advantage of WAC is extremely evident in these tasks. SAC and OAC are only able to solve the simplest task. In a sparse reward setting, SAC and OAC will only explore randomly so they fully rely on the chance of reaching the goal state with random actions. OAC explores more compared to SAC, but since the exploration does not depend on the state-action visitations, but only on the disagreement between the critics, sparse reward tasks are a great challenge. WAC, instead, will still explore, even when facing sparse rewards since the uncertainty will gradually decline in visited regions, so the upper bounds will favor reaching unvisited ones. Indeed, OE-WAC outperforms both baselines in all the tasks with sparse rewards.

7 Conclusions

In this paper, we presented a novel AC algorithm to perform directed exploration. We presented WAC, which extends the recently proposed WQL to the continuous-actions case. Furthermore, we addressed a problem of uncertainty estimation that arises when using function approximation, related to the generalization of the uncertainty estimates. We proposed a simple, yet effective, regularization method, based on synthetic samples that allowed us to better generalize the uncertainty across the state-action space. Finally, we performed a thorough empirical evaluation to investigate the advantages of performing a principled uncertainty estimation and propagation in continuous-action domains. We observed that the uncertainty estimates of WAC can effectively steer exploration towards promising regions of the state-action space, even under sparse rewards, especially when comparing it with heuristic uncertainty estimation based on ensemble methods.

References

Agueh, M.; and Carlier, G. 2011. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2): 904–924.

Auer, P.; Jaksch, T.; and Ortner, R. 2008. Near-optimal Regret Bounds for Reinforcement Learning. In Koller, D.; Schuurmans, D.; Bengio, Y.; and Bottou, L., eds., *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc.

Bellemare, M. G.; Dabney, W.; and Munos, R. 2017. A Distributional Perspective on Reinforcement Learning. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70

of *Proceedings of Machine Learning Research*, 449–458. PMLR.

Chen, X.; Wang, C.; Zhou, Z.; and Ross, K. W. 2021. Randomized Ensembled Double Q-Learning: Learning Fast Without a Model. In *International Conference on Learning Representations*.

Ciosek, K.; Vuong, Q.; Loftin, R.; and Hofmann, K. 2019. Better Exploration with Optimistic Actor Critic. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Dabney, W.; Rowland, M.; Bellemare, M. G.; and Munos, R. 2018. Distributional Reinforcement Learning with Quantile Regression. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press. ISBN 978-1-57735-800-8.

Dearden, R.; Friedman, N.; and Russell, S. J. 1998. Bayesian Q-Learning. In Mostow, J.; and Rich, C., eds., *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference, AAAI 98, IAAI 98, July 26-30, 1998, Madison, Wisconsin, USA.*, 761–768. AAAI Press / The MIT Press.

Dorato, P.; Cerone, V.; and Abdallah, C. 2000. *Linear quadratic control: an introduction*. Krieger Publishing Co., Inc.

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 Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 1861–1870. PMLR.

Houthooft, R.; Chen, X.; Duan, Y.; Schulman, J.; De Turck, F.; and Abbeel, P. 2016. VIME: Variational Information Maximizing Exploration. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, 1117–1125. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510838819.

Ian, O.; Benjamin, V. R.; and Daniel, R. 2013. (More) Efficient Reinforcement Learning via Posterior Sampling. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, 3003–3011. Red Hook, NY, USA: Curran Associates Inc.

Jaksch, T.; Ortner, R.; and Auer, P. 2010. Near-Optimal Regret Bounds for Reinforcement Learning. *J. Mach. Learn. Res.*, 11: 1563–1600.

Jin, C.; Allen-Zhu, Z.; Bubeck, S.; and Jordan, M. I. 2018. Is Q-Learning Provably Efficient? In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 4868–4878. Curran Associates, Inc.

Lattimore, T. 2020. *Bandit algorithms / Tor Lattimore (deepMind) and Csaba Szepesvari (University of Alberta)*. Cambridge, United Kingdom ;: Cambridge University Press. ISBN 9781108486828.

Lee, K.; Laskin, M.; Srinivas, A.; and Abbeel, P. 2021. SUNRISE: A Simple Unified Framework for Ensemble Learning in Deep Reinforcement Learning. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 6131–6141. PMLR.

Mavrin, B.; Yao, H.; Kong, L.; Wu, K.; and Yu, Y. 2019. Distributional Reinforcement Learning for Efficient Exploration. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 4424–4434. PMLR.

Metelli, A. M.; Likmeta, A.; and Restelli, M. 2019. Propagating Uncertainty in Reinforcement Learning via Wasserstein Barycenters. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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): 529–533.

Moro, L.; Likmeta, A.; Prati, E.; and Restelli, M. 2022. Goal-Directed Planning via Hindsight Experience Replay. In *International Conference on Learning Representations*.

Moskovitz, T.; Parker-Holder, J.; Pacchiano, A.; Arbel, M.; and Jordan, M. 2021. Tactical Optimism and Pessimism for Deep Reinforcement Learning. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 12849–12863. Curran Associates, Inc.

Mutti, M.; Pratissoli, L.; and Restelli, M. 2021. Task-agnostic exploration via policy gradient of a non-parametric state entropy estimate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9028–9036.

O'Donoghue, B.; Osband, I.; Munos, R.; and Mnih, V. 2018. The Uncertainty Bellman Equation and Exploration. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 3839–3848. Stockholmsmässan, Stockholm Sweden: PMLR.

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

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust Region Policy Optimization. In Bach, F.; and Blei, D., eds., *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 1889–1897. Lille, France: PMLR.

Strehl, A. L.; Li, L.; Wiewiora, E.; Langford, J.; and Littman, M. L. 2006. PAC model-free reinforcement learning. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, 881–888.

Strehl, A. L.; and Littman, M. L. 2008. An analysis of model-based Interval Estimation for Markov Decision Processes. *J. Comput. Syst. Sci.*, 74(8): 1309–1331.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. The MIT Press, second edition.

Szepesvári, C. 1997. The Asymptotic Convergence-Rate of Q-Learning. In *Proceedings of the 10th International Conference on Neural Information Processing Systems*, NIPS'97, 1064–1070. Cambridge, MA, USA: MIT Press.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo: A physics engine for model-based control. In *IROS*, 5026–5033. IEEE. ISBN 978-1-4673-1737-5.

Villani, C. 2008. *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

Wang, H.; Lin, S.; and Zhang, J. 2021. Adaptive Ensemble Q-learning: Minimizing Estimation Bias via Error Feedback. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 24778–24790. Curran Associates, Inc.

Zhang, T.; Rashidinejad, P.; Jiao, J.; Tian, Y.; Gonzalez, J. E.; and Russell, S. 2021. MADE: Exploration via Maximizing Deviation from Explored Regions. In Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*.