Propagating Uncertainty in Reinforcement Learning via Wasserstein Barycenters

Alberto Maria Metelli[*]
DEIB
Politecnico di Milano
Milan, Italy
albertomaria.metelli@polimi.it

Amarildo Likmeta[*]
DEIB
Politecnico di Milano
Milan, Italy
amarildo.likmeta@polimi.it

Marcello Restelli
DEIB
Politecnico di Milano
Milan, Italy
marcello.restelli@polimi.it

Abstract

How does the uncertainty of the value function propagate when performing temporal difference learning? In this paper, we address this question by proposing a Bayesian framework in which we employ approximate posterior distributions to model the uncertainty of the value function and Wasserstein barycenters to propagate it across state-action pairs. Leveraging on these tools, we present an algorithm, *Wasserstein Q-Learning* (WQL), starting in the tabular case and then, we show how it can be extended to deal with continuous domains. Furthermore, we prove that, under mild assumptions, a slight variation of WQL enjoys desirable theoretical properties in the tabular setting. Finally, we present an experimental campaign to show the effectiveness of WQL on finite problems, compared to several RL algorithms, some of which are specifically designed for exploration, along with some preliminary results on Atari games.

1 Introduction

Effectively balancing exploration and exploitation is a key challenge in Reinforcement Learning [RL, 43]. When an agent takes decisions under uncertainty, it faces the dilemma between *exploiting* the information collected so far to execute what is believed to be the best action or to choose a possibly suboptimal action to *explore* new portions of the environment and gather new information, leading to more profitable behaviors in the future. Traditional exploration strategies, such as ϵ-greedy and Boltzmann exploration [43], inject random noise into the action-selection process, i.e., the policy, to guarantee that each action is tried often enough. Although these methods allow RL algorithms to learn the optimal value function under mild assumptions [39], they are not *efficient*, since exploration is random and not driven by confidence on the value function estimate. Therefore, they might converge towards the optimal behavior after an exponential number of steps [24].

The exploration-exploitation dilemma has been extensively analyzed in the RL community, focusing on the definition of proper indices for *provably-efficient* exploration and devising algorithms with strong theoretical guarantees [25, 11, 21, 30]. Most of these algorithms are inherently model-based, i.e., they need to maintain and update estimates of the environment dynamics and the reward function during the learning process. For this reason, model-based methods are rather unsuited to problems with large state spaces and inapplicable to continuous environments. Apart from rare exceptions [41], the RL community has only recently focused on devising efficient model-free exploration strategies. Some works have succeeded in obtaining provably-efficient algorithms [35, 31, 23]; whereas others are more empirically-oriented [30, 29, 6].

[*]Equal contribution.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

A fundamental step towards efficient exploration is the quantification of the *uncertainty* of the value function. The notion of uncertainty is formalized in Bayesian statistics by means of a *posterior* distribution. Bayesian Reinforcement Learning incorporates the Bayesian inference tools to provide a principled way to address the exploration-exploitation dilemma [20]. However, these methods rarely exploit the specific way in which the uncertainty propagates through the Bellman equation. Recently, in [28] a partial answer has been provided, proposing an uncertainty Bellman inequality; although no posterior distribution is explicitly considered.

In this paper, we propose a novel Bayesian framework to address the problem of exploration using posterior distributions over the value function. Specifically, we focus on how to *model* and *propagate* uncertainty when performing temporal-difference learning (Section 3). Moreover, we show how to use this uncertainty information to effectively explore the environment. Finally, we combine these elements to build our algorithm: *Wasserstein Q-Learning* (WQL, Section 4). Similarly to Bayesian Q-Learning [15], we equip each state-action pair with an approximate posterior distribution (named *Q-posterior*), whose goal is to quantify the uncertainty of the value function. Whenever a transition occurs, we update our distribution, in a temporal difference [TD, 43] fashion, in order to incorporate all sources of uncertainty: i) the one due to the sample estimate of the reward function and environment dynamics; ii) the uncertainty injected using the estimate of the next-state value function. Rather than employing a standard Bayesian update, we resort to a variational approach to approximate the posterior distribution, based on Wasserstein barycenters [2]. Recently, several works have embedded into RL algorithms notions coming from *Optimal Transport* [OT, 51], especially the Wasserstein metric, to improve the learning abilities of policy search algorithms [34] or in the filed of robust RL [1]. Furthermore, we prove in Section 5, that a slight modification of WQL, in tabular domains, is *PAC-MDP in the average loss setting* [42]. After examining the related literature (Section 6), we present an experimental evaluation on tabular environments to show the effectiveness of WQL, compared to the classic RL algorithms, some of which specifically designed for exploration (Section 7.1). Finally, we provide some preliminary results on the application of WQL to deep architectures (Section 7.2). The proofs of all results are reported in Appendix B. The implementation of the proposed algorithms can be found at `https://github.com/albertometelli/wql`.

2 Preliminaries

In this section, we provide the notation and the basic notions we will use in the following. Given a set $\mathcal{X}$, we denote with $\mathscr{P}(\mathcal{X})$ the set of all probability measures over $\mathcal{X}$.

Markov Decision Processes A discrete-time Markov Decision Process [MDP, 36] is defined as a 5-tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the (finite) action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathscr{P}(\mathcal{S})$ is a Markovian transition model, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathscr{P}(\mathbb{R})$ is a Markovian reward model, $\gamma \in [0, 1)$ is the discount factor. The behavior of an agent is defined by means of a Markovian policy $\pi : \mathcal{S} \to \mathscr{P}(\mathcal{A})$. Whenever the environment is in state $s \in \mathcal{S}$, the agent performs action $A \sim \pi(\cdot|s)$ and the environment transitions to the next state $S' \sim \mathcal{P}(\cdot|s, A)$ providing the agent with the reward $R \sim \mathcal{R}(\cdot|s, A)$. We assume $|R| \leq r_{\max} < +\infty$ almost surely. We indicate with $r(s, a) = \mathbb{E}_{R\sim\mathcal{R}(\cdot|s,a)}[R]$ the expected reward obtained by taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. Given a policy π we define the state-value function, or V-function, as $v_\pi(s) = \mathbb{E}_{A\sim\pi(\cdot|s),S'\sim\mathcal{P}(\cdot|s,A)}\left[r(s, A) + \gamma v_\pi(S')\right]$. The action-value function, or Q-function, is given by $q_\pi(s, a) = r(s, a) + \gamma\, \mathbb{E}_{S'\sim\mathcal{P}(\cdot|s,a),A'\sim\pi(\cdot|S')}\left[q_\pi(S', A')\right]$. The optimal action-value function is defined as $q^*(s, a) = \sup_{\pi\in\Pi}\{q_\pi(s, a)\}$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ and it satisfies the optimal Bellman equation: $q^*(s, a) = r(s, a) + \gamma\, \mathbb{E}_{S'\sim\mathcal{P}(\cdot|s,a)}\left[\max_{a'\in\mathcal{A}}\{q^*(S', a')\}\right]$. The boundedness of the reward function implies that the Q-function is uniformly bounded, i.e., $|q^*(s, a)| \leq q_{\max} \leq r_{\max}/(1 - \gamma)$. Then, an optimal policy π^* is any policy that plays only greedy actions w.r.t. q^*, i.e., for all $s \in \mathcal{S}$ we have $\pi^*(\cdot|s) \in \mathscr{P}\left(\arg\max_{a\in\mathcal{A}}\{q^*(s, a)\}\right)$.

Temporal Difference Learning Temporal-difference methods update the estimate of the optimal Q-function using the estimates of the next states V-functions [43]. For TD(0), we have that whenever a $(S_t, A_t, S_{t+1}, R_{t+1})$ tuple is collected, the *temporal difference update rule* is executed:

$$q_{t+1}(S_t, A_t) = (1 - \alpha_t)\, q_t(S_t, A_t) + \alpha_t \left(R_{t+1} + \gamma v_t(S_{t+1})\right), \tag{1}$$

where q_t is the estimated Q-function at time t, $\alpha_t \geq 0$ is a learning rate, and v_t is an estimate of the V-function at time t. Different choices for v_t generate different learning algorithms. If $v_t(S_{t+1}) =$

$q_t(S_{t+1}, A_{t+1})$ we get the SARSA update [38], if $v_t(S_{t+1}) = \mathbb{E}_{A\sim\pi_t(\cdot|S_{t+1})}[q_t(S_{t+1}, A)]$ we get the Expected SARSA update [50], being π_t the exploration policy played at time t, and if $v_t(S_{t+1}) = \max_{a\in\mathcal{A}}\{q_t(S_{t+1}, a)\}$ we are performing Q-learning [52].

Wasserstein Barycenters Let $(\mathcal{X}, d)$ be a complete separable metric (Polish) space and $x_0 \in \mathcal{X}$ be an arbitrary point. For each $p \in [1, +\infty)$ we define $\mathscr{P}_p(\mathcal{X})$ as the set of all probability measures μ over $(\mathcal{X}, \mathscr{F})$ such that $\mathbb{E}_{X\sim\mu}[d(X, x_0)^p] < +\infty$. Let $\mu, \nu \in \mathscr{P}_p(\mathcal{X})$, the L^p-Wasserstein distance between μ and ν is defined as [51]:

$$W_p(\mu, \nu) = \left(\inf_{\rho\in\Gamma(\mu,\nu)} \mathop{\mathbb{E}}_{X,Y\sim\rho}[d(X, Y)^p]\right)^{1/p}, \tag{2}$$

where $\Gamma(\mu, \nu)$ is the set of all probability measures on $\mathcal{X} \times \mathcal{X}$ (couplings) with marginals μ and ν. With little abuse of notation, we will indicate with $W_p(X, Y) = W_p(\mu, \nu)$, whenever clear from the context. The Wasserstein distance comes from the optimal transport community. Intuitively, it represents the "cost" to move the probability mass to turn one distribution into the other. Given a set of probability measures $\{\nu_i\}_{i=1}^n$, belonging to the class $\mathscr{N}$, 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 [2]:

$$\overline{\nu} = \operatorname*{arg\,inf}_{\nu\in\mathscr{N}} \left\{\sum_{i=1}^n \xi_i W_2(\nu, \nu_i)^2\right\}. \tag{3}$$

3 How to Model and Propagate Uncertainty?

In this section, we introduce a unifying Bayesian framework for exploration in RL that employs (approximate) posterior distributions to *model* uncertainty of value functions (Section 3.1) and Wasserstein barycenters to *propagate* uncertainty when performing TD updates (Section 3.2). Furthermore, we discuss how to leverage on the Q-posteriors to estimate the action that attains the maximum return in each state (Section 3.3) and to effectively explore the environment (Section 3.4).

3.1 Modeling Uncertainty via Q-Posteriors

Taking inspiration from Bayesian approaches to RL [15, 20], for each state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$ we maintain a probability distribution $\mathcal{Q}(s, a)$, which we call *Q-posterior*, representing a (possibly approximate) posterior distribution of the Q-function estimate. This distribution will depend on the underlying MDP, in particular, the environment dynamics $\mathcal{P}$ and reward model $\mathcal{R}$, and on the updates of the Q-function estimates performed. As in a model-free scenario we cannot represent such distribution exactly, we employ a class of approximating probability distributions $\mathscr{Q} \subseteq \mathscr{P}(\mathbb{R})$. Similarly to usual value functions, we introduce the *V-posterior* $\mathcal{V}(s)$ which represents the (possibly approximate) posterior distribution of V-function, that combines the uncertainties modeled by the Q-posteriors $\mathcal{Q}(s, a)$. Furthermore, being the V-function defined, in the usual framework, as the expectation of the Q-function over the action space, i.e., $v_\pi(s) = \mathbb{E}_{A\sim\pi}[q_\pi(s, a)]$, it is natural to define, in our setting, the V-posterior $\mathcal{V}(s)$ as the Wasserstein barycenter of the Q-posteriors $\mathcal{Q}(s, a)$.[2]

Definition 3.1 (V-posterior)**.** *Given a policy $\overline{\pi}$ and a state $s \in \mathcal{S}$, we define the* V-posterior $\mathcal{V}(s)$ *induced by the Q-posteriors $\mathcal{Q}(s, a)$ with $a \in \mathcal{A}$ as the Wassertein barycenter of the $\mathcal{Q}(s, a)$:*

$$\mathcal{V}(s) \in \operatorname*{arg\,inf}_{\mathcal{V}\in\mathscr{Q}} \left\{\mathop{\mathbb{E}}_{A\sim\overline{\pi}(\cdot|s)} \left[W_2\left(\mathcal{V}, \mathcal{Q}(s, A)\right)^2\right]\right\}. \tag{4}$$

When the policy $\overline{\pi}$ is known, the expectation over the action space can be computed as we are assuming that $\mathcal{A}$ is finite. In a prediction problem, policy $\overline{\pi}$ is a fixed policy, whereas, in a control problem, $\overline{\pi}$ is a policy aimed at properly selecting the best action in state s accounting for the uncertainty modeled by the Q-posterior (see Section 3.3). Moreover, when $\mathcal{Q}(s, a)$ are deterministic distributions, $\mathcal{V}(s)$ is a deterministic distribution too centered in the mean of the $\mathcal{Q}(s, a)$. In this way, we obtain the usual V-function definition (see Proposition A.3).

[2]The Wasserstein barycenter can be regarded as a way of averaging distributions [2].

It is important to stress that our approach is rather different from *Distributional Reinforcement Learning* [9, 13, 12, 37]. Indeed, we employ a distribution to represent the *uncertainty* of the Q-function estimate and not the *intrinsic randomness* of the return. The two distributions are clearly related and both depend on the stochasticity of the reward and of the transition model. However, in our approach the stochasticity refers to the uncertainty on the Q-function estimate which reduces as the number of updates increases, being a sample mean.[3]

3.2 Propagating Uncertainty via Wasserstein Barycenters

In this section, we discuss the problem of uncertainty propagation, i.e., how to deal with the update of the Q-posteriors when experiencing a transition $(S_t, A_t, S_{t+1}, R_{t+1})$. Whenever a TD update (Equation (1)) is performed, there are two sources of uncertainty involved. First, we implicitly estimate the environment dynamics $\mathcal{P}(\cdot|S_t, A_t)$ and the reward model $\mathcal{R}(\cdot|S_t, A_t)$ using a set of sampled transitions $(S_t, A_t, S_{t+1}, R_{t+1})$. Second, when using the V-function estimates of the next states $v_t(S_{t+1})$ we bring into $q_{t+1}(S_t, A_t)$ part of the uncertainty of $v_t(S_{t+1})$ and they become correlated. For this reason, the standard Bayesian posterior update, used for instance in Bayesian Q-learning [15], becomes rather inappropriate as it assumes that the samples are independent, which is clearly not true. We argue that, rather than using a Bayesian update, when we have a Q-posterior $\mathcal{Q}_t(S_t, A_t)$ and a V-posterior $\mathcal{V}_t(S_{t+1})$ we can combine them using a notion of barycenter, which does not require the independence assumption. We formalize this idea in the following update rule.

Definition 3.2 (Wasserstein Temporal Difference)**.** *Let $\mathcal{Q}_t$ be the current Q-posterior, given a transition $(S_t, A_t, S_{t+1}, R_{t+1})$, we define the* TD-target-posterior *as $\mathcal{T}_t = R_{t+1} + \gamma\mathcal{V}_t(S_{t+1})$. Let $\alpha_t \geq 0$ be the learning rate, we define the* Wasserstein Temporal Difference *(WTD) update rule as:*

$$\mathcal{Q}_{t+1}(S_t, A_t) \in \operatorname*{arg\,inf}_{\mathcal{Q}\in\mathscr{Q}} \left\{(1-\alpha_t)W_2\left(\mathcal{Q}, \mathcal{Q}_t(S_t, A_t)\right)^2 + \alpha_t W_2\left(\mathcal{Q}, \mathcal{T}_t\right)^2\right\}. \tag{5}$$

Therefore, the new Q-posterior $\mathcal{Q}_{t+1}(S_t, A_t)$ is the Wasserstein barycenter between the current Q-posterior $\mathcal{Q}_t(S_t, A_t)$ and the TD-target posterior $\mathcal{T}_t = R_{t+1} + \gamma\mathcal{V}_t(S_{t+1})$, which in turn embeds information of the current transition (i.e., the reward R_{t+1} and the next state S_{t+1}) and the next-state V-posterior $\mathcal{V}_t(S_{t+1})$. It is worth noting that the two terms appearing in Equation (5) account for all sources of uncertainty. Indeed, the first term $W_2\left(\mathcal{Q}, \mathcal{Q}_t(S_t, A_t)\right)$ avoids moving too far from the current estimation $\mathcal{Q}_t(S_t, A_t)$, as we are performing the update experimenting a single transition, whereas $W_2\left(\mathcal{Q}, \mathcal{T}_t\right)$ allows bringing in the new Q-posterior the V-posterior of the next-state $\mathcal{V}_t(S_{t+1})$ (including its uncertainty). We stress the analogy with the standard TD update in the following result.

Proposition 3.1. *If $\mathscr{Q}$ is the set of deterministic distributions over $\mathbb{R}$, then the WTD update rule (Equation* (5)*) has a unique solution that corresponds to the TD update rule (Equation* (1)*).*

Supporting deterministic distributions, as the Q-posteriors, is fundamental for our method that models a sample mean, whose variance reduces as the number of samples increases, moving towards a deterministic distribution. This justifies the choice of the Wasserstein metric over other distributional distances (e.g., α-divergences). The choice of the prior for $\mathcal{Q}_0$ plays an important role, along with the learning rate schedule α_t. We will show in Section 5 that specific choices of $\mathcal{Q}_0$ and α_t, for a particular class of distributions $\mathscr{Q}$, allow achieving PAC-MDP property in the average loss setting.

3.3 Estimating the Maximum Expected Value

The TD-target-posterior $\mathcal{T}_t = R_{t+1} + \gamma\mathcal{V}_t(S_{t+1})$ is defined in terms of the next state V-posterior $\mathcal{V}_t(S_{t+1})$. In a control problem, we aim at learning the optimal Q-function q^* and, thus, we are interested in propagating back to $\mathcal{Q}_{t+1}(S_t, A_t)$ a V-posterior $\mathcal{V}_t(S_{t+1})$ related to the optimal action to be taken in the next state.[4] This can be performed by a suitable choice of the policy $\overline{\pi}$, as in Definition 3.1. A straightforward approach consists in propagating the Q-posterior $\mathcal{Q}(S_{t+1}, a)$ of the action with the highest estimated mean, i.e., $\overline{\pi}^M(\cdot|s) \in \mathscr{P}\left(\arg\max_{a\in\mathcal{A}}\{\mathbb{E}_{Q\sim\mathcal{Q}(s,a)}[Q]\}\right)$.

[3] A notable difference w.r.t. the distributional RL is that the variance of our posterior distribution $\mathbb{V}\mathrm{ar}_{Q\sim\mathcal{Q}(s,a)}[Q]$ vanishes as the number of updates grows to infinity.

[4] We stress that we are uninterested in modeling the distribution $\max_{a\in\mathcal{A}}\{\mathcal{Q}(s, a)\}$, but rather in exploiting the uncertainty modeled by $\mathcal{Q}(s, a)$ to properly perform the computation of the optimal action.

Table 1: Probability density function (pdf), Wasserstein Temporal Difference (WTD) update rule and computation of the V-posterior for Gaussian and Particle posterior distributions.

$\mathscr{Q}$	pdf	WTD and V-posterior
Gaussian	$\frac{\exp\left\{-\frac{1}{2}\left(\frac{x-m(s,a)}{\sigma(s,a)}\right)^2\right\}}{\sqrt{2\pi\sigma^2(s,a)}}$	$m_{t+1}(S_t,A_t) = \alpha_t m_t(S_t,A_t) + (1-\alpha_t)\left(R_{t+1} + \gamma m_t(S_{t+1})\right)$ $\sigma_{t+1}(S_t,A_t) = \alpha_t \sigma_t(S_t,A_t) + (1-\alpha_t)\gamma\sigma_t(S_{t+1})$ $m(s) = \mathbb{E}_{A\sim\overline{\pi}(\cdot\|s)}\left[m(s,A)\right]$ $\sigma(s) = \mathbb{E}_{A\sim\overline{\pi}(\cdot\|s)}\left[\sigma(s,A)\right]$
Particle	$\sum_{j=1}^{M} w_j \delta(x - x_j(s,a))$ $x_1(s,a) \leq ... \leq x_M(s,a)$ $\sum_{i=j}^{M} w_j = 1$ and $w_j \geq 0$	$x_{j,t+1}(S_t,A_t) = \alpha_t x_{j,t}(S_t,A_t) + (1-\alpha_t)\left(R_{t+1} + \gamma x_{j,t}(S_{t+1})\right)$ $x_j(s) = \mathbb{E}_{A\sim\overline{\pi}(\cdot\|s)}\left[x_j(s,A)\right],\ j = 1,2,...,M$

We refer to this approach as *Mean Estimator* (ME) for the maximum. However, when posterior distributions are available, we can use them to define a wiser way to estimate the V-posterior of the next state.[5] A first method based on Optimism in the Face of Uncertainty [OFU, 3] consists in selecting the action that maximizes a statistical upper bound $u^\delta(s,a)$ of the Q-posterior, i.e., $\overline{\pi}^O(\cdot|s) \in \mathscr{P}\left(\arg\max_{a\in\mathcal{A}}\{u^\delta(s,a)\}\right)$. We will refer to this method as *Optimistic Estimator* (OE). However, if we want to make full usage of the Q-posteriors, we can resort to the *Posterior Estimator* (PE) of the maximum, based on Posterior Sampling [PS, 47]. In this case, each action contributes to the update rule weighted by the probability of being the optimal action, i.e., $\overline{\pi}^P(a|s) = \Pr_{Q_{s,a}\sim\mathcal{Q}(s,a)}\left(a \in \arg\max_{a'\in\mathcal{A}}\{Q_{s,a'}\}\right)$.

3.4 Exploring using the Q-posteriors

In the previous section, we have introduced two approaches that exploit the Q-posterior to properly define the V-posterior of the next state, using specific policies $\overline{\pi}$. These policies can also be used to implement effective exploration strategies aware of the uncertainty. Using the optimistic policy $\overline{\pi}^O$ in each state, we play (deterministically) the action that maximizes the statistical upper bound on the estimated Q-function $u^\delta(s,a)$, we call this strategy *Optimistic Exploration* (OX). Instead, we can directly use the posterior policy $\overline{\pi}^P$ to sample the action from the Q-posterior $\mathcal{Q}(s,a)$. Thus, in *Posterior Exploration* (PX), each action is played with the probability of being optimal.

4 Wasserstein Q-Learning

The ideas presented so far can be combined in an algorithm, *Wasserstein Q-Learning* (WQL), whose pseudocode is reported in Algorithm 1.

Input: a prior distribution $\mathcal{Q}_0$, a step size schedule $(\alpha_t)_{t\geq 0}$, an exploration policy schedule $(\pi_t)_{t\geq 0}$
1: Initialize $\mathcal{Q}(s,a)$ with the prior $\mathcal{Q}_0$
2: **for** $t = 1, 2, ...$ **do**
3: Take action $A_t \sim \pi_t(\cdot|S_t)$
4: Observe S_{t+1} and R_{t+1}
5: Compute $\mathcal{V}_t(S_{t+1})$ using Equation (4)
6: Update $\mathcal{Q}_{t+1}(S_t,A_t)$ using Equation (5)
7: **end for**

Algorithm 1: Wasserstein Q-Learning.

We developed our approach for a generic class of distributions $\mathscr{Q}$, however, in practice, we focus on two specific classes: *Gaussian* posteriors (G-WQL) and *Particle* posteriors (P-WQL), i.e., a mixture of $M > 1$ Dirac deltas. For both classes the Wasserstein Barycenter is unique and can be computed in closed form (see Appendix A.3).[6] In Table 1, we summarize the main relevant features of these distributions classes. WQL simply needs to store the parameters of the Q-posterior for every state-action pair ($m(s,a)$ and $\sigma(s,a)$ for G-WQL and $x_j(s,a)$ for P-WQL). Therefore, unlike the majority of provably-efficient algorithms, it can be extended straightforwardly to continuous state spaces as long as we adopt a function approximator for the parameters of the posterior. For instance, we could approximate $m(s,a)$ and $\sigma(s,a)$ or the particles $x_j(s,a)$ using a neural network with multiple heads. For this reason, our method easily

[5] This problem was treated in RL, without distributions, proposing several estimators, such as the double estimator [48] and the weighted estimator [17, 16].

[6] It is worth noting that, even for the Gaussian case, using the standard Bayesian posterior update is inappropriate, as the independence of the Q-function estimates across state-action pairs cannot be assumed.

applies to deep architecture by adopting a network that directly outputs the posterior parameters, instead of the value function (see Section 7.2).

5 Theoretical Analysis

In this section, we show that WQL, with some modifications, enjoys desirable theoretical properties in the tabular setting. We start providing a modification of the WTD update rule that will be used for the analysis; then we prove that with such modification our algorithm, under certain assumptions, is PAC-MDP in the average loss setting [42].

Definition 5.1 (Modified Wasserstein Temporal Difference)**.** *Let $\mathcal{Q}_t$ be the current Q-posterior and $\mathcal{Q}_b$ be a zero-mean distribution, given a transition $(S_t, A_t, S_{t+1}, R_{t+1})$, we define the* TD-target-posterior *as $\mathcal{T}_t = R_{t+1} + \gamma \mathcal{V}_t(S_{t+1})$. Let $\alpha_t\,, \beta_t \geq 0$ be the learning rates, we define the* Modified Wasserstein Temporal Difference *(MWTD) update rule as:*

$$\widetilde{\mathcal{Q}}_{t+1}(S_t, A_t) \in \operatorname*{arg\,inf}_{\mathcal{Q} \in \mathscr{Q}} \left\{ (1-\alpha_t) W_2 \left(\mathcal{Q}, \widetilde{\mathcal{Q}}_t(S_t, A_t) \right)^2 + \alpha_t W_2 \left(\mathcal{Q}, \mathcal{T}_t \right)^2 \right\},$$

$$\mathcal{Q}_{t+1}(S_t, A_t) \in \operatorname*{arg\,inf}_{\mathcal{Q} \in \mathscr{Q}} \left\{ W_2 \left(\mathcal{Q}, \widetilde{\mathcal{Q}}_{t+1}(S_t, A_t) + \beta_t \mathcal{Q}_b \right)^2 \right\},$$

We will denote the algorithm employing this update rule as *Modified Wasserstein Q-Learning* (MWQL). The reason why we need to change the WTD lies in the fact that the uncertainty on the Q-function value (the Q-posterior) is, as already mentioned, the contribution of two terms: i) the uncertainty on the reward and transition model; ii) the uncertainty on the next-state Q-function. These terms need to be averaged into the Q-posterior at different speeds. If $n_t(s,a)$ is the number of times (s,a) is visited up to time t, (i) has to reduce proportionally to $1/\sqrt{n_t(s,a)}$ being a sample mean, while (ii) is averaged with coefficients proportional to $1/n_t(s,a)$. Therefore, we should keep the two sources of uncertainty separated. To this end, we use an additional distribution $\mathcal{Q}_b$ to prevent the uncertainty from reducing too fast.

The notion of *PAC-MDP in the average loss setting* [42] is a relaxation of the classical PAC-MDP notion introduced in [24], in which we consider the actual reward received by the algorithm while learning, instead of the expected values over future policies. We recall the definitions given in [42].

Definition 5.2 (Definition 4 of [42])**.** *Suppose a learning algorithm $\mathfrak{A}$ is run for T steps. Consider partial sequence $S_0, R_1, ..., S_{T-1}, R_T, S_T$ visited by $\mathfrak{A}$. The* instantaneous loss *of the agent at time t is $il_{\mathfrak{A}}(t) = v^*(S_t) - \sum_{i=t}^{T} \gamma^{i-t} R_{i+1}$. The quantity $\mathcal{L}_{\mathfrak{A}} = \frac{1}{T} \sum_{t=1}^{T} il_{\mathfrak{A}}(t)$ is called the* average loss*.*

Then, a learning algorithm $\mathfrak{A}$ is PAC-MDP in the average loss setting if for any $\epsilon \geq 0$ and $\delta \in [0,1]$, we can choose a value T, polynomial in the relevant quantities $(1/\epsilon, 1/\delta, |\mathcal{S}|, |\mathcal{A}|, 1/(1-\gamma))$, such that the average loss $\mathcal{L}_{\mathfrak{A}}$ of the agent (following the learning algorithm $\mathfrak{A}$) on a trial of T steps is guaranteed to be less than ϵ with probability at least $1-\delta$.

In the following, we will restrict our attention to MWQL with Gaussian posterior, optimistic estimator (OE) and optimistic exploration policy (OX). We leave the analysis of the posterior sampling exploration (PX) as future work. To prove the main result we need an intermediate result.

Theorem 5.1. *Let $S_0, ..., S_{T-1}, S_T$ be the sequence of states and actions visited by MWQL with Gaussian posterior, OE and OX. Then, there exists a prior $\mathcal{Q}_0$ and a zero-mean distribution $\mathcal{Q}_b$ and a learning rate schedule for $(\alpha_t, \beta_t)_{t \geq 0}$ (whose values are reported in Appendix B.1), such that for any $\delta \in [0,1]$, with probability at least $1-\delta$ it holds that:*[7]

$$\sum_{t=1}^{T} [v^*(S_t) - v_{\mathfrak{A}}(S_t)] \leq \mathcal{O}\left(\frac{q_{\max}}{(1-\gamma)^{\frac{3}{2}}} \sqrt{|\mathcal{S}||\mathcal{A}| T \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta}} \right), \tag{6}$$

where $v_{\mathfrak{A}}$ is the value function induced by the (non-stationary) policy played by algorithm $\mathfrak{A}$.

From this result, we can exploit an analysis similar to [42] to prove that MWQL with Gaussian posterior, OE and OX is PAC-MDP in the average loss setting.

[7] This performance index resembles the *regret* [21]. However, it is a weaker notion, being defined in terms of the trajectory generated by algorithm $\mathfrak{A}$, instead of the trajectories of an optimal policy.

Theorem 5.2. *Under the hypothesis of Theorem 5.1, MWQL with Gaussian posterior, OE and OX is PAC-MDP in the average loss setting, i.e., for any $\epsilon \geq 0$ and $\delta \in [0, 1]$, after*

$$T = \mathcal{O}\left(\frac{q_{\max}^2 |\mathcal{S}||\mathcal{A}|}{\epsilon^2(1-\gamma)^3} \log \frac{q_{\max}^2 |\mathcal{S}|^2 |\mathcal{A}|^2}{\delta \epsilon^2 (1-\gamma)^3}\right)$$

steps we have that the average loss $\mathcal{L}_{\mathfrak{A}} \leq \epsilon$ with probability at least $1 - \delta$.

The per-step computational complexity of MWQL is $\mathcal{O}(\log |\mathcal{A}|)$ as we can maintain the upper bounds of the Q-function as a max-priority queue [40] and the space complexity is $\mathcal{O}(|\mathcal{S}||\mathcal{A}|)$.

Despite the theoretical guarantees, MWQL turns out to be often impractical for two main reasons. First, MWQL cannot be extended to continuous MDPs, as α_t and β_t are defined in terms of number of visits $n(s, a)$ (Equation (20)), which can only be computed for finite MDPs. Second, as many provably efficient RL algorithms, MWQL is extremely conservative, leading to very slow convergence. This is why most provably efficient RL algorithms, when used in practice, are run with non-theoretical values of hyperparameters. In this sense, WQL can be seen as a "practical" version of MWQL in which α_t is treated as a normal hyper-parameter and $\beta_t = 0$.

6 Related Works

A variety of approaches has been proposed in the RL literature to tackle the exploration-exploitation trade-off [44]. We consider only those that do not assume the availability of a simulator of the environment [26]. A first dimension of classification is the RL setting they consider: *finite-horizon*, *discounted* or *undiscounted*. Finite-horizon MDPs are a convenient framework to devise provably-efficient exploration algorithms with theoretical guarantees on the *regret* [32, 14, 5]. Recently, in [23] it was shown that Q-learning, in the finite-horizon setting, can be made efficient by resorting to suitable exploration bonuses. Similar results have been proposed in the infinite-horizon undiscounted case. The main challenge of this class of problems is the connection structure of the MDP [7]. Early approaches [25, 4, 46, 21] impose restrictive requirements on either mixing/hitting times or diameter, which have been progressively relaxed [19]. A significant part of the early provably-efficient algorithms considers the discounted setting [25, 11, 41, 45, 27]. However, their theoretical guarantees are based on the notion of PAC-MDP [24] rather than on regret.

Another relevant dimension is the kind of policy used for exploration. Taking inspiration from the Multi Armed Bandit [MAB, 10] framework, two main approaches have been proposed: Optimism in the Face of Uncertainty [3] and Thompson Sampling [47]. Most exploration algorithms employ the optimistic technique, selecting actions from the optimal policy of an optimistic approximation of the MDP [21] or of the value function directly [41, 23]. Some methods, instead, use a posterior sampling approach in which either the entire MDP or a value function is sampled from a (possibly approximate) posterior distribution.

Inspired by these methods, numerous practical variants have been devised. Exploration bonuses, based on pseudo-counts [8, 33], mimicking optimism, have been applied with positive results to deep architectures. Likewise, with the idea of approximating a posterior distribution, *Bootstrapped DQN* [30] and *Bayesian DQN* [6] succeeded in solving challenging Atari games. Recently, new results of sample-efficiency beyond tabular domains have been derived [22].

7 Experiments

In this section, we provide an experimental evaluation of WQL on tabular domains along with some preliminary results on Atari games (implementation details are reported in Appendix C).

7.1 Tabular Domains

We evaluate WQL on a set of RL tasks designed to emphasize exploration: the Taxi problem [15], the Chain [15], the River Swim [42], and the Six Arms [42]. We extensively test several WQL variants that differ on: i) the Q-posterior model (Gaussian G-WQL vs particle P-WQL); ii) the exploration strategy (optimistic OX vs posterior sampling PX), iii) the estimator of the maximum (ME, OE, and PE).

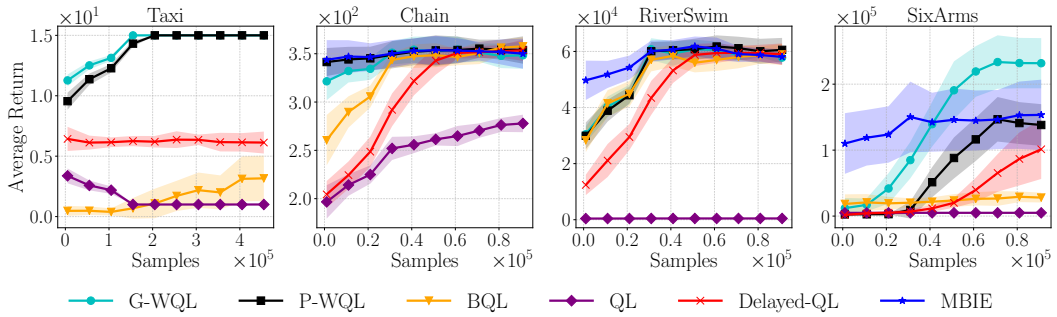

Figure 1: Online average return as a function of the number of samples, comparison of P-WQL and G-WQL with QL, BQL, Delayed-QL, and MBIE-EB. 10 runs, 95% c.i.

We compare these combinations with the classic Q-learning [QL, 52] (Boltzmann exploration), Bootstrapped Q-learning [BQL, 30] both with the double estimator [48], Delayed Q-learning [Delayed-QL, 41] and MBIE-EB [42].[8]

Figure 1 shows the *online* performance on the considered tabular tasks. While we tried all the WQL variants, due to space constraints, we show the best combination of exploration strategy and maximum estimator for both Gaussian and particle models (complete results are reported in Appendix C). We can see that WQL learns substantially faster than classical approaches, like QL, in tasks that require significant exploration, such as Taxi, Six Arms, or River Swim. Our algorithm also outperforms BQL in most tasks, except in the River Swim, where performances are not substantially different. Finally, we can see that across all the tasks WQL displays a faster learning curve w.r.t. to Delayed-QL. MBIE-EB outperforms WQL in small domains like Chain and RiverSwim, but not in SixArms. MBIE-EB was not tested on the Taxi domain as the number of states (~ 200) makes the computational time demands prohibitive. We cross-validate the hyperparameter of Delayed Q-Learning and MBIE-EB.

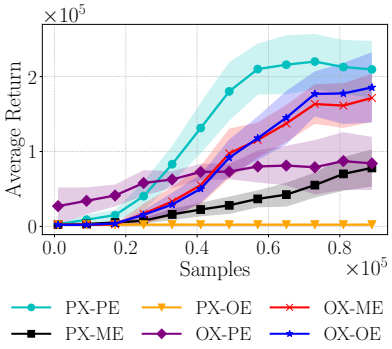

Figure 2: Online average return as a function of the number of samples for the different versions of G-WQL algorithm. 10 runs, 95% c.i.

Among the variants of WQL, we discovered that the choice of the exploration strategy and the maximum estimator are highly task dependent. However, we can see a general pattern across the tasks. As intuition suggests, being the exploration strategy and the maximum estimator closely related, the best combinations are: OX exploration with OE estimator and PX exploration with PE estimator. We illustrate in Figure 2 all the possible combinations of G-WQL on Six Arms, a domain in which exploration is essential. We can notice that the "hybrid" combinations, such as OX with PE and PX with OE are significantly outperformed by the more "coherent" ones.

7.2 Atari Games

We adapted WQL with the particle model to be used paired with deep architectures. For this purpose, similarly to Bootstrapped DQN [BDQN, 30], we use a network architecture with a head for each particle while the convolutional layers are shared among them. We compare the resulting algorithm, which we call *Particle DQN* (PDQN), with Double DQN [DDQN, 49], a classic benchmark in Deep-RL, and Bootstrapped DQN, specifically designed for deep exploration using Q-posteriors. To compare algorithms we consider *offline* scores, i.e., the scores collected using the current greedy policy. The goal of this experiment, conducted on three Atari games, is to prove that WQL, although designed to work in finite environments, can easily be extended to deep networks with potentially good results. In Figure 3, we can see that PDQN, compared to BDQN and DDQN, manages to

[8] We are considering a discounted setting, thus, several provably efficient algorithms, like UCRL2 [21], PSRL [32], RLSVI [30], optimistic Q-learning [23] and UCBVI [5], cannot be compared as they consider either average reward or finite-horizon setting.

Input: a prior distribution $\{x_i\}_{i=1}^M$, a step size schedule $(\alpha_t)_{t\geq 0}$, an exploration policy schedule $(\pi_t)_{t\geq 0}$

1: Initialize a Q-function network with M outputs $\{Q_j\}_{j=1}^M$ and parameters $\boldsymbol{\theta}$ and the target network with parameters $\boldsymbol{\theta}^- = \boldsymbol{\theta}$
2: **for** $t = 1, 2, \ldots$ **do**
3: Take action $A_t \sim \pi_t(\cdot|S_t; \boldsymbol{\theta})$
4: Store transition $(S_t, A_t, S_{t+1}, R_{t+1})$ in the replay buffer
5: Sample random a batch of transitions $(S_l, A_l, S_{l+1}, R_{l+1})$ from the replay buffer
6: Compute targets $y_j(S_{l+1}) = \mathbb{E}_{A\sim\overline{\pi}(\cdot|S_{l+1})}[Q_j(S_{l+1}, A; \boldsymbol{\theta}^-)]$ for each output Q_j where $\overline{\pi} \in \{\overline{\pi}^M, \overline{\pi}^O, \overline{\pi}^P\}$ as in Section 3.3
7: Perform a gradient descent step w.r.t. $\boldsymbol{\theta}$ on the objective $\sum_{j=1}^M (y_j(S_{t+1}) - Q_j(S_l, A_l; \boldsymbol{\theta}))^2$ and using the step size α_t
8: Periodically update target network $\boldsymbol{\theta}^- = \boldsymbol{\theta}$
9: **end for**

Algorithm 2: Particle DQN.

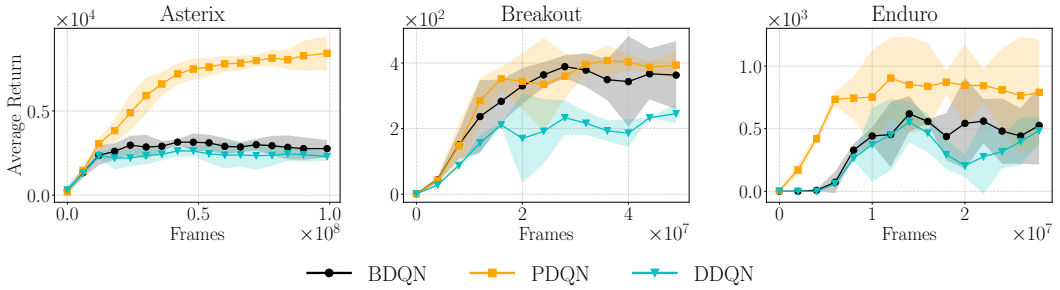

Figure 3: Offline average return of the greedy policy as a function of the number of collected frames, comparing PDQN, DDQN and BDQN on Asterix, Enduro and Breakout games. 5 runs, 95% c.i.

achieve higher scores in Asterix and Enduro, where exploration is needed, while achieving similar scores in Breakout. A relevant feature of PDQN is the particle initialization interval. Indeed, a narrower initial interval causes faster learning but might lead to premature convergence. In this sense, the initial interval becomes a hyperparameter of PDQN, which influences the amount of exploration and it is likely task-dependent. The pseudocode of PDQN is shown in Algorithm 2.

8 Discussion and Conclusions

In this paper, we presented a novel RL algorithm, Wasserstein Q-Learning (WQL), which addresses several issues related to efficient exploration in model-free RL. We discussed how to model uncertainty of the estimated Q-function by means of approximate posterior distributions (Q-posteriors). Then, we devised a variational method to propagate uncertainty across state-action pairs when performing TD learning, based on Wasserstein barycenters. The experimental evaluation allowed us to appreciate the properties of WQL. In tabular domains, whenever exploration is really necessary, our approach is able to significantly outperform TD methods even if designed specifically for exploration (e.g., Bootstrapped Q-Learning and Delayed Q-Learning). Although preliminary, the results on the Atari games are promising and need to be further investigated as future work in order to make WQL scale on complex environments. We believe that our algorithm contributes to bridging the gap between theory and practice of exploration in RL. WQL is a theoretically grounded method, equipped with guarantees in the average loss setting, but, at the same time, it is a very simple algorithm, easily extensible to deal with continuous domains.

References

[1] Mohammed Amin Abdullah, Hang Ren, Haitham Bou Ammar, Vladimir Milenkovic, Rui Luo, Mingtian Zhang, and Jun Wang. Wasserstein robust reinforcement learning. *arXiv preprint arXiv:1907.13196*, 2019.

[2] Martial Agueh and Guillaume Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

[3] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[4] Peter Auer and Ronald Ortner. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 49–56, 2007.

[5] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 263–272, 2017.

[6] Kamyar Azizzadenesheli, Emma Brunskill, and Animashree Anandkumar. Efficient exploration through bayesian deep q-networks. *arXiv preprint arXiv:1802.04412*, 2018.

[7] Peter L Bartlett and Ambuj Tewari. Regal: A regularization based algorithm for reinforcement learning in weakly communicating mdps. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 35–42. AUAI Press, 2009.

[8] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016.

[9] Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 449–458. PMLR, 2017.

[10] Donald A Berry and Bert Fristedt. Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability). *London: Chapman and Hall*, 5:71–87, 1985.

[11] Ronen I. Brafman and Moshe Tennenholtz. R-MAX - A general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, 2002.

[12] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1104–1113. PMLR, 2018.

[13] Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 2892–2901. AAAI Press, 2018.

[14] Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2818–2826, 2015.

[15] Richard Dearden, Nir Friedman, and Stuart J. Russell. Bayesian q-learning. In Jack Mostow and Chuck Rich, editors, *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.*, pages 761–768. AAAI Press / The MIT Press, 1998.

[16] Carlo D'Eramo, Alessandro Nuara, Matteo Pirotta, and Marcello Restelli. Estimating the maximum expected value in continuous reinforcement learning problems. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 1840–1846. AAAI Press, 2017.

[17] Carlo D'Eramo, Marcello Restelli, and Alessandro Nuara. Estimating maximum expected value through gaussian approximation. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1032–1040. JMLR.org, 2016.

[18] DC Dowson and BV Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

[19] Ronan Fruit, Matteo Pirotta, Alessandro Lazaric, and Ronald Ortner. Efficient bias-span-constrained exploration-exploitation in reinforcement learning. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1573–1581. PMLR, 2018.

[20] Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, Aviv Tamar, et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.

[21] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

[22] Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1704–1713. JMLR. org, 2017.

[23] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4868–4878. Curran Associates, Inc., 2018.

[24] Sham Machandranath Kakade et al. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.

[25] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002.

[26] Sven Koenig and Reid G Simmons. Complexity analysis of real-time reinforcement learning applied to finding shortest paths in deterministic domains. Technical report, Carnegie Mellon University Pittsburgh PA School of Computer Science, 1992.

[27] Tor Lattimore and Marcus Hutter. Near-optimal PAC bounds for discounted mdps. *Theor. Comput. Sci.*, 558:125–143, 2014.

[28] Brendan O'Donoghue, Ian Osband, Remi Munos, and Vlad Mnih. The uncertainty Bellman equation and exploration. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3839–3848, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[29] Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 8626–8638, 2018.

[30] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4026–4034, 2016.

[31] Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2377–2386. JMLR.org, 2016.

[32] Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3003–3011, 2013.

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

[34] Aldo Pacchiano, Jack Parker-Holder, Yunhao Tang, Anna Choromanska, Krzysztof Choromanski, and Michael I. Jordan. Behavior-guided reinforcement learning. *arXiv preprint arXiv:1906.04349*, 2019.

[35] Jason Pazis, Ronald Parr, and Jonathan P. How. Improving PAC exploration using the median of means. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3891–3899, 2016.

[36] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[37] Mark Rowland, Marc G. Bellemare, Will Dabney, Rémi Munos, and Yee Whye Teh. An analysis of categorical distributional reinforcement learning. In Amos J. Storkey and Fernando Pérez-Cruz, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pages 29–37. PMLR, 2018.

[38] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, England, 1994.

[39] Satinder P. Singh, Tommi S. Jaakkola, Michael L. Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38(3):287–308, 2000.

[40] Alexander L. Strehl, Lihong Li, and Michael L. Littman. Reinforcement learning in finite mdps: PAC analysis. *Journal of Machine Learning Research*, 10:2413–2444, 2009.

[41] Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman. PAC model-free reinforcement learning. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, pages 881–888, 2006.

[42] Alexander L. Strehl and Michael L. Littman. An analysis of model-based interval estimation for markov decision processes. *J. Comput. Syst. Sci.*, 74(8):1309–1331, 2008.

[43] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[44] Csaba Szepesvári. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103, 2010.

[45] Istvan Szita and Csaba Szepesvári. Model-based reinforcement learning with nearly tight exploration complexity bounds. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 1031–1038, 2010.

[46] Ambuj Tewari and Peter L Bartlett. Optimistic linear programming gives logarithmic regret for irreducible mdps. In *Advances in Neural Information Processing Systems*, pages 1505–1512, 2008.

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

[48] Hado Van Hasselt. Double q-learning. In *Advances in Neural Information Processing Systems*, pages 2613–2621, 2010.

[49] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2094–2100. AAAI Press, 2016.

[50] Harm Van Seijen, Hado Van Hasselt, Shimon Whiteson, and Marco Wiering. A theoretical and empirical analysis of expected sarsa. In *Adaptive Dynamic Programming and Reinforcement Learning, 2009. ADPRL'09. IEEE Symposium on*, pages 177–184. IEEE, 2009.

[51] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[52] Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards*. PhD thesis, King's College, Cambridge, 1989.

Index of the Appendix

In the following, we briefly recap the contents of the Appendix.

- Appendix A provides additional details about the properties of the Wasserstein metric and the Wasserstein barycenters.
- Appendix B reports all proofs and derivations.
- Appendix C reports additional experiments in the tabular domain along with some implementation details.

A Details on Wasserstein distance

In this appendix, we provide some additional details on Wasserstein distance and some properties of our approximate posterior distribution models. It can be proved [51] that the functions W_p are metrics on the sets $\mathscr{P}_p(\mathcal{X})$. Moreover, the following monotonicity property holds: $W_p(\mu,\nu) \leq W_q(\mu,\nu)$ if $p \leq q$. We will assume that all the involved probability measures μ admit cumulative distribution function (c.d.f.) F_μ and probability density function (p.d.f.) f_μ w.r.t. the Lebesgue measure. When $\mathcal{X} = \mathbb{R}$ and $d(x,y) = |x-y|$ is the Euclidean distance, the Wasserstein distance can be rephrased in terms of the quantile functions:

$$W_p(\mu,\nu) = \left(\int_0^1 \left| F_\mu^{-1}(t) - F_\nu^{-1}(t) \right|^p \mathrm{d}t \right)^{1/p}, \tag{7}$$

where F^{-1} is the quantile function, i.e., $F_\mu^{-1}(t) = \inf\{x \in \mathbb{R} : t \leq F_\mu(x)\}$. The L^2-Wasserstein distance admits a closed form for both the Gaussian and particle models. For two Gaussian distributions, considering just the univariate case, $\mu = \mathcal{N}(m_1, \sigma_1^2)$ and $\nu = \mathcal{N}(m_2, \sigma_2^2)$, we have [18]:

$$W_2(\mu,\nu)^2 = (m_1 - m_2)^2 + (\sigma_1 - \sigma_2)^2. \tag{8}$$

For the particle models, with the same weighting w_j we have the following result.

Proposition A.1. *Let μ and ν be two mixture of M Dirac deltas with the same weighting w_j for $j = 1, 2, ..., M$ having $f_\mu(x) = \sum_{j=1}^M w_j \delta(x - x_j)$ and $f_\nu(x) = \sum_{j=1}^M w_j \delta(x - y_j)$ as p.d.f.s with $x_1 \leq x_2 \leq ... \leq x_M$ and $y_1 \leq y_2 \leq ... \leq y_M$. Then, the L^2-Wasserstein distance between μ and ν is given by:*

$$W_2(\mu,\nu)^2 = \sum_{j=1}^M w_j (x_j - y_j)^2. \tag{9}$$

Proof. We use Equation (7), thus we need to compute the quantile function of a mixture of M Dirac deltas. Let us introduce the intervals $I_j = [t_{j-1}, t_j)$, $t_0 = 0$, $t_j = \sum_{k=1}^j w_k$ and $|I_j| = t_j - t_{j-1}$ for $j = 1, 2, ...N$. It is clear that:

$$F_\mu^{-1}(t) = \sum_{j=1}^M x_j \mathbb{1}_{I_j}(t), \quad F_\mu^{-1}(t) = \sum_{j=1}^M y_j \mathbb{1}_{I_j}(t),$$

where $\mathbb{1}_{\mathcal{X}}$ is the indicator function of the set $\mathcal{X}$. Thus we can compute the Wasserstein distance by employing Equation (7):

$$\begin{aligned}
W_2(\mu,\nu)^2 &= \int_0^1 \left| F_\mu^{-1}(t) - F_\nu^{-1}(t) \right|^2 \mathrm{d}t \\
&= \sum_{j=1}^M \int_{I_j} \left| F_\mu^{-1}(t) - F_\nu^{-1}(t) \right|^2 \mathrm{d}t \\
&= \sum_{j=1}^M \int_{I_j} (x_j - y_j)^2 \mathrm{d}t \\
&= \sum_{j=1}^M (x_j - y_j)^2 \int_{I_j} \mathrm{d}t
\end{aligned}$$

$$= \sum_{j=1}^{M} w_j (x_j - y_j)^2,$$

where we observed that $\int_{I_j} \mathrm{d}t = t_j - t_{j-1} = w_j$. □

A.1 Approximation of an arbitrary distribution with a mixture of Deltas

In this section, we show that we are able to approximate an arbitrary distribution with a mixture of Deltas, provided that we consider a sufficiently large number of components.

Proposition A.2. *Let μ be an arbitrary probability measure over the interval $[a, b] \subset \mathbb{R}$ admitting F_μ as c.d.f. and let ν be a mixture of M Dirac deltas $\widehat{x}_1 \leq \widehat{x}_2 \leq ... \leq \widehat{x}_M$ weighted by w_j for $j = 1, 2, ..., M$ fixed. Then, the L^2-Wasserstein $W_2(\mu, \nu)$ has a unique minimizer:*

$$\widehat{x}_j = \frac{1}{|I_j|} \int_{I_j} F_\mu^{-1}(t) \mathrm{d}t, \quad j = 1, 2, \ldots, M,$$

where $I_j = [t_{j-1}, t_j)$, $t_0 = 0$, $t_j = \sum_{k=1}^{j} w_k$ and $|I_j| = t_j - t_{j-1}$ for $j = 1, 2, ...N$. In this case, the L^2-Wasserstain distance can be bounded as:

$$W_2(\mu, \nu)^2 \leq \frac{(b-a)^2}{4} \max_{j=1,2,\ldots,n} |I_j|.$$

Proof. Let us first compute the quantile function of ν, i.e., F_ν^{-1}:

$$F_\nu^{-1}(t) = \sum_{j=1}^{M} x_j \mathbb{1}_{I_j}(t). \tag{10}$$

Using Equation (7), the L^2-Wasserstein distance can be written as:

$$\begin{aligned} W_2(\mu, \nu)^2 &= \int_0^1 \left(F_\mu^{-1}(t) - F_\nu^{-1}(t)\right)^2 \mathrm{d}t \\ &= \sum_{j=1}^{M} \int_{I_j} \left(F_\mu^{-1}(t) - x_j\right)^2 \mathrm{d}t. \end{aligned}$$

The objective is clearly convex, thus, we take the derivative w.r.t. x_j and we get:

$$\frac{\partial W_2^2}{\partial x_j} = -2 \int_{I_j} \left(F_\mu^{-1}(t) - x_j\right) \mathrm{d}t = 0, \tag{11}$$

from which the first result follows, by observing that $\int_{I_j} \mathrm{d}t = |I_j|$. Let us now observe, since F_μ^{-1} is monotonically increasing, that for every j:

$$\begin{aligned} \int_{I_j} \left(F_\mu^{-1}(t) - \widehat{x}_j\right)^2 \mathrm{d}t &\leq \int_{I_j} \left(F_\mu^{-1}(t) - \frac{F_\mu^{-1}(t_j) + F_\mu^{-1}(t_{j-1})}{2}\right)^2 \mathrm{d}t && (12) \\ &\leq \frac{1}{4} \int_{I_j} \left[(F_\mu^{-1}(t) - F_\mu^{-1}(t_j)) + (F_\mu^{-1}(t) - F_\mu^{-1}(t_{j-1}))\right]^2 \mathrm{d}t \\ &\leq \frac{1}{4} \int_{I_j} \left[(F_\mu^{-1}(t) - F_\mu^{-1}(t_{j-1})) - (F_\mu^{-1}(t_j) - F_\mu^{-1}(x))\right]^2 \mathrm{d}t && (13) \\ &\leq \frac{1}{4} \int_{I_j} \left(F_\mu^{-1}(t_j) - F_\mu^{-1}(t_{j-1})\right)^2 \mathrm{d}t \\ &\leq \frac{1}{4} |I_j| \left(F_\mu^{-1}(t_j) - F_\mu^{-1}(t_{j-1})\right)^2, \end{aligned}$$

where (12) follows from the fact that $\widehat{x}_j$ is the minimizer and (13) derives from observing that $F_\mu^{-1}(t) - F_\mu^{-1}(t_{j-1}) \leq 0$ by definition of $F_\mu^{-1}(t_{j-1})$ whereas $F_\mu^{-1}(t) - F_\mu^{-1}(t_{j-1}) \geq 0$. Let us rename $\Delta_j = F_\mu^{-1}(t_j) - F_\mu^{-1}(t_{j-1})$, the error can be bounded overall as:

$$W_2(\mu, \nu)^2 \leq \frac{1}{4} \sum_{j=1}^{M} |I_j| \Delta_j^2$$

$$
\begin{aligned}
&\leq \frac{1}{4} \max_{j=1,2,\ldots,M} |I_j| \sum_{j=1}^{n} \Delta_j^2 \\
&\leq \frac{1}{4} \max_{j=1,2,\ldots,M} |I_j| \left(\sum_{j=1}^{M} \Delta_j \right)^2 \\
&\leq \frac{1}{4} \max_{j=1,2,\ldots,M} |I_j| \, (b-a)^2 ,
\end{aligned}
$$

where we used Cauchy–Schwarz inequality in the last passage, observing that $\Delta_j \geq 0$ from the monotonicity of F_μ^{-1}. □

When we consider a uniform particle model, i.e., $w_j = 1/M$, the result reduces to:

$$
W_2(\mu, \nu)^2 \leq \frac{(b-a)^2}{4M}. \tag{14}
$$

The result tells us that when $M \to \infty$ the error vanishes as expected. Up to now, we considered the error introduced by representing a given distribution with a mixture of deltas. It is interesting to investigate the properties of the approximating distribution ν.

Lemma A.1. *Let ν be the best mixture of deltas L^2-Wasserstein approximation of an arbitrary distribution μ, as defined in Proposition A.2. If μ admits expectation, then it holds that:*

$$
\mathop{\mathbb{E}}_{X \sim \mu}[X] = \mathop{\mathbb{E}}_{X \sim \nu}[X]. \tag{15}
$$

Proof. We are going to assume that μ admits a p.d.f. f_μ and we indicate with f_ν the p.d.f. of ν. We first observe that by making the substitution $x = F_\mu^{-1}(t)$, we have the identity:

$$
\int_{I_j} F_\mu^{-1}(t) \mathrm{d}t = \int_{F_\mu^{-1}(t_{j-1})}^{F_\mu^{-1}(t_j)} x f_\mu(x) \mathrm{d}x. \tag{16}
$$

Therefore:

$$
\begin{aligned}
\mathop{\mathbb{E}}_{X \sim \nu}[X] &= \int_a^b x f_\nu(x) \mathrm{d}x \\
&= \sum_{j=1}^{M} w_j \widehat{x}_j \\
&= \sum_{j=1}^{M} w_j \frac{1}{|I_j|} \int_{I_j} F_\mu^{-1}(t) \mathrm{d}t \\
&= \sum_{j=1}^{M} \int_{I_j} F_\mu^{-1}(t) \mathrm{d}t \\
&= \sum_{j=1}^{M} \int_{F_\mu^{-1}(t_{j-1})}^{F_\mu^{-1}(t_j)} x f(x) \mathrm{d}x \\
&= \int_a^b x f_\mu(x) \mathrm{d}x = \mathop{\mathbb{E}}_{X \sim \mu}[X].
\end{aligned}
$$

□

Thus, our approximation preserves the mean, but unfortunately, this does not hold for the higher order moments.

A.2 Combining Posteriors via Wasserstein Barycenters

In this section, we introduce the notion of Fréchet mean, we show how it reduces to the Wasserstein barycenter when selecting the Wasserstein distance as a metric.

Definition A.1. *Let $(\mathcal{X}, d)$ be a complete separable metric (Polish) space and $\mu \in \mathscr{P}_2(\mathcal{X})$ a probability measure over $\mathcal{X}$. The Fréchet mean is defined as:*

$$
\overline{x} = \arg\inf_{x \in \mathcal{X}} \mathop{\mathbb{E}}_{Y \sim \mu} \left[d(x, Y)^2 \right]. \tag{17}
$$

The Fréchet mean generalizes the notion of mean by minimizing the expected value of a metric. In the particular case in which $\mathcal{X} = \mathbb{R}$, and $d(x, y) = |x - y|$ is the Euclidean distance, the Fréchet mean is the expectation of X under μ, i.e., $\overline{x} = \mathbb{E}_{X\sim\mu}[X]$.[9]

In our scenario, we have that $\mathcal{X} = \mathscr{Q}$ is a set of probability measures (the Q-posteriors) and we select as metric $d = W_2$, i.e., the L^2-Wasserstein distance. Therefore, Equation (17) defines the *Wasserstein barycenter* [2]. Typically, the notion of barycenter is defined in terms of a finite set of distributions [2]. However, we can also consider infinite (possibly continuous) sets of distributions. For our posterior distribution models, the Wasserstein barycenter is unique and we have a closed form expression.

A.3 Closed-Forms for Wasserstein Barycenters of Gaussians and Particle models

We show in the following that the standard close forms for finite sets, when using Gaussian and particle models, naturally extends for continuous spaces.

Proposition A.3. *Let $(\mathcal{T}, \mathscr{F})$ be a measurable space and let $\mu \in \mathscr{P}(\mathcal{T})$ be a probability measure over $\mathcal{T}$. Let $\{\nu(t)\}_{t\in\mathcal{T}}$ be a family of probability measures. Then,*

$$\inf_{\nu\in\mathcal{N}} \mathop{\mathbb{E}}_{T\sim\mu}\left[W_2\left(\nu, \nu(T)\right)^2\right]$$

has a unique solution both for Gaussian and uniform particle models. In particular, for the Gaussian model, the parameters of the L^2-Wasserstein barycenter are:

$$\overline{m} = \mathop{\mathbb{E}}_{T\sim\mu}\left[\mu(T)\right], \quad \overline{\sigma} = \mathop{\mathbb{E}}_{T\sim\mu}\left[\sigma(T)\right],$$

and for equally weighted particle models:

$$\overline{x}_j = \mathop{\mathbb{E}}_{T\sim\mu}\left[x_j(T)\right], \quad j = 1, 2, ..., M.$$

Proof. All it takes is to write down the objective function and compute its minimizer. Let us start with the Gaussian model. The L^2-Wasserstein distance between two Gaussians is given in Equation (8), therefore our objective becomes:

$$\mathcal{L}(\mu, \sigma) = \mathop{\mathbb{E}}_{T\sim\mu}\left[W_2\left(\nu, \nu(T)\right)^2\right] = \mathop{\mathbb{E}}_{T\sim\mu}\left[(\overline{m} - m(T))^2 + (\overline{\sigma} - \sigma(T))^2\right],$$

where $\nu(T) = \mathcal{N}(m(T), \sigma^2(T))$ and $\overline{\nu} = \mathcal{N}(\overline{m}, \overline{\sigma}^2)$. The objective is clearly convex, therefore we just need to take the derivatives w.r.t. μ and σ:

$$\frac{\partial\mathcal{L}}{\partial\overline{\mu}} = 2\mathop{\mathbb{E}}_{T\sim\mu}\left[\overline{m} - m(T)\right] = 0 \implies \overline{m} = \mathop{\mathbb{E}}_{T\sim\mu}\left[m(T)\right],$$
$$\frac{\partial\mathcal{L}}{\partial\overline{\sigma}} = 2\mathop{\mathbb{E}}_{T\sim\mu}\left[\overline{\sigma} - \sigma(T)\right] = 0 \implies \overline{\sigma} = \mathop{\mathbb{E}}_{T\sim\mu}\left[\sigma(T)\right],$$

from which the result follows. Similarly, for the equally weighted particle models, using Proposition A.1, we have:

$$\begin{aligned}\mathcal{L}(x_1, x_2, ..., x_M) &= \mathop{\mathbb{E}}_{T\sim\mu}\left[W_2\left(\nu, \nu(T)\right)^2\right] \\ &= \mathop{\mathbb{E}}_{T\sim\mu}\left[\sum_{j=1}^{M} w_j(\overline{x}_j - x_j(T))^2\right] \\ &= \sum_{j=1}^{M} w_j \mathop{\mathbb{E}}_{T\sim\mu}\left[(\overline{x}_j - x_j(T))^2\right].\end{aligned}$$

We take the derivatives again and we obtain for every $j = 1, 2, ..., M$:

$$\frac{\partial\mathcal{L}}{\partial\overline{x}_j} = 2w_j \mathop{\mathbb{E}}_{T\sim\mu}\left[\overline{x}_j - x_j(T)\right] = 0 \implies \overline{x}_j = \mathop{\mathbb{E}}_{T\sim\mu}\left[x_j(T)\right].$$

□

[9] Different choices of exponent to which $d(x, Y)$ is raised generate different indexes of central tendency, like median for exponent 1 and mode for the exponent going to 0 in the limit.

In other words, the L^2-Wasserstein barycenter of a set of Gaussians is a Gaussian distribution having as mean the expectations of the mean and standard deviation the expectations of the standard deviations. Similarly, the L^2-Wasserstein barycenter of a set of uniform mixtures of deltas is a uniform mixture of deltas where each particle is located at the expectation of the locations.

B Proofs and Derivations

Proposition 3.1. *If $\mathscr{Q}$ is the set of deterministic distributions over $\mathbb{R}$, then the WTD update rule (Equation* (5)*) has a unique solution that corresponds to the TD update rule (Equation* (1)*).*

Proof. It is a simple application of Proposition A.3 for the particle model setting $M = 1$, $\mu = (1 - \alpha_t, \alpha_t)^T$, $\nu_1 = q_t(S_t, A_t)$ and $\nu_2 = R_{t+1} + \gamma v_t(S_{t+1})$. □

B.1 Provable Efficiency

In this section, we provide a series of results about the provable efficiency of a slightly modified version of WQL. We restrict our attention to the Gaussian model and we consider the optimistic estimator (OE) for the maximum and optimistic exploration (OX). We will consider w.l.o.g. the following assumption.

Assumption B.1. *The reward function is deterministic and positive.*

As a consequence we have that $0 \leq q_\pi(s, a) \leq q_{\max}$. Several parts of the proofs we are going to present are inspired to [23]. We now illustrate how to modify WQL in order to have the desired theoretical guarantees.

B.1.1 Modified Gaussian WQL

First of all, we need to particularize the MWTD for the Gaussian case:

$$\begin{aligned}
&\widetilde{m}_{t+1}(s, a) = (1 - \alpha_t)\widetilde{m}_t(s, a) + \alpha_t \left(R_{t+1} + \gamma m_t(S_{t+1})\right), \\
&\widetilde{\sigma}_{t+1}(s, a) = (1 - \alpha_t)\widetilde{\sigma}_t(s, a) + \alpha_t \gamma \sigma_t(S_{t+1}), \\
&m_{t+1}(s, a) = \widetilde{m}_{t+1}(s, a) + \beta_t m_b, \\
&\sigma_{t+1}(s, a) = \widetilde{\sigma}_{t+1}(s, a) + \beta_t \sigma_b,
\end{aligned}$$

with $m_b = 0$ by definition. We also define the auxiliary quantities that account for the accumulated effect of the learning rate:

$$\alpha_t^0 = \prod_{i=0}^{t} (1 - \alpha_i), \qquad \alpha_t^i = \alpha_i \prod_{j=i+1}^{t} (1 - \alpha_j)$$

It is clear that, by definition, for any $t = 0, 1, ...$ we have that $\alpha_0^t + \sum_{i=1}^{t} \alpha_i^t = 1$. Using these quantities we can rewrite the update rules for the mean and the standard deviation:

$$\begin{aligned}
m_t(s, a) &= \alpha_{n_t(s,a)} m_0 + \sum_{i=1}^{n_t(s,a)} \alpha_{n_t(s,a)}^i \left[R_{t_i+1} + \gamma m_{t_i}(S_{t_i+1})\right], \\
\sigma_t(s, a) &= \alpha_{n_t(s,a)} \sigma_0 + \gamma \sum_{i=1}^{n_t(s,a)} \alpha_{n_t(s,a)}^i \sigma_{t_i}(S_{t_i+1}) + \beta_t \sigma_b,
\end{aligned}$$

where $n_t(s, a)$ is the number of times pair (s, a) was visited up to time t, t_i is the time at which pair (s, a) was visited for the i-th time, $m_{t_i}(S_{t_i+1}) = m_{t_i}(S_{t_i+1}, \overline{a})$ and $\sigma_{t_i}(S_{t_i+1}, \overline{a})$, where $\overline{a} = \arg\max_{a \in \mathcal{A}} u_{t_i}^\delta(S_{t_i+1}, a)$ is the action that maximizes the upper bound of the Q-function, defined as:

$$\overline{u}_t^\delta(s, a) = m_t(s, a) + z_{1-\delta}\sigma_t(s, a), \quad u_t^\delta(s, a) = \min\left\{\overline{u}_t^\delta(s, a), q_{\max}\right\} \tag{18}$$

Notice that we can define an update rule for the upper bound $\overline{u}_t^\delta(s, a)$ too:

$$\overline{u}_t^\delta(s, a) = \alpha_{n_t(s,a)} \left[m_0 + z_{1-\delta}\sigma_0\right]$$

Input: m_0, σ_0, σ_b

1: **for** $t = 1, 2, ...$ **do**

2: Take action $A_t \in \arg\max_{a \in \mathcal{A}} u_t^\delta(S_t, A_t)$

3: Observe S_{t+1} and R_{t+1}

4: Update the posterior distribution

$$
\begin{aligned}
m_{t+1}(S_t, A_t) &= (1 - \alpha_t)\widetilde{m}_t(S_t, A_t) + \alpha_t \left(R_{t+1} + \gamma m_t(S_{t+1})\right) \\
\widetilde{\sigma}_{t+1}(S_t, A_t) &= (1 - \alpha_t)\widetilde{\sigma}_t(S_t, A_t) + \alpha_t \gamma \sigma_t(S_{t+1}) \\
\sigma_{t+1}(S_t, A_t) &= \widetilde{\sigma}_{t+1}(S_t, A_t) + \beta_t \sigma_b \\
n_{t+1}(S_t, A_t) &= n_t(S_t, A_t) + 1
\end{aligned}
$$

5: **end for**

Algorithm 3: Modified Gaussian Wasserstein Q-Learning

$$
\begin{aligned}
&+ \sum_{i=1}^{n_t(s,a)} \alpha^i_{n_t(s,a)} \left[R_{t_i+1} + \gamma \left(m_{t_i}(S_{t_i+1}) + z_{1-\delta}\sigma_{t_i}(S_{t_i+1})\right)\right] + \beta_t \sigma_b \\
&= \alpha_{n_t(s,a)} \left[m_0 + z_{1-\delta}\sigma_0\right] + \sum_{i=1}^{n_t(s,a)} \alpha^i_{n_t(s,a)} \left[R_{t_i+1} + \gamma u_t^\delta(S_{t_i+1})\right] + \beta_t \sigma_b
\end{aligned}
$$

where $u_t(S_{t_i+1})$ is the upper bound of the V-function defined as:

$$
u_t^\delta(s) = \max_{a \in \mathcal{A}} \left\{u_t^\delta(s, a)\right\}. \tag{19}
$$

In Algorithm 3 we provide the pseudocode of *Modified Wasserstein Q-Learning* (MWQL). The learning rates and the initialization values will be specified later in the analysis.

B.1.2 Learning Rate

In this section, we introduce the learning rates α_t and β_t we will use to prove our theoretical results and we will present some properties we are going to exploit for the subsequent proofs:

$$
\alpha_t = \frac{a}{b + t}, \qquad \beta_t = \frac{c}{\sqrt{d + t}}. \tag{20}
$$

with $0 < a \leq b + 1$, $b \geq 1$, $0 < c^2 \leq d$, $d \geq 1$ and $d \leq b$, whose values will be specified later. This choice of learning rates allows us to prove the following properties.

Lemma B.1. *If $a > 1$ and $b > 1$, the following relations hold for any $t = 1, 2, ...$ and $i = 1, 2, ...$:*

1. $\sum_{t=1}^{+\infty} \alpha_t^i \leq \frac{a}{a-1}$;
2. $\sum_{t=1}^{+\infty} \alpha_t^0 = \frac{b}{a-1} - 1$;
3. $\sum_{i=1}^{t} (\alpha_t^i)^2 \leq \frac{a}{b+t}$.

Proof. Let us start with 1. By using the properties of the Gamma function, we have that:

$$
\sum_{t=1}^{M} \alpha_t^i = \frac{a}{a-1} \frac{\Gamma(b+i)}{\Gamma(1-a+b+i)} \left[\frac{\Gamma(2-a+b)}{\Gamma(1+b)} - \frac{\Gamma(2-a+b+M)}{\Gamma(1+b+M)}\right].
$$

Since $a > 1$ we have that $2 - a + b + M < 1 + b + M$, thus for $M \to +\infty$ the second addendum goes to zero. Moreover, for the same reason $\frac{\Gamma(b+i)}{\Gamma(1-a+b+i)} \leq 1$ and $\frac{\Gamma(2-a+b)}{\Gamma(1+b)} \leq 1$. The result follows immediately. A similar argument can be stated for 2. By using the properties of the Gamma function, we have the following equality:

$$
\sum_{t=1}^{M} \alpha_t^0 = \frac{b}{a-1} - 1 - \frac{\Gamma(1+b)\Gamma(2-a+b+M)}{(a-1)\Gamma(1-a+b)\Gamma(1+b+M)}.
$$

Since $a > 1$ we have that $2 - a + b + M < 1 + b + M$ and therefore for $M \to +\infty$ the ratio of Gamma functions goes to zero, from which the result follows. Finally, for 3 we employ an argument similar to that of Lemma 4.1 (b) of [23]:

$$\begin{aligned}\alpha_t^i &= \frac{a}{b+i}\left(\frac{b+i+1-a}{b+i+1}\cdot\frac{b+i+2-a}{b+i+2}\cdots\frac{b+t-a}{b+t}\right)\\ &= \frac{a}{b+t}\left(\frac{b+i+1-a}{b+i}\cdot\frac{b+i+2-a}{b+i+1}\cdots\frac{b+t-a}{b+t-1}\right)\\ &\leq \frac{a}{b+t},\end{aligned}$$

where we exploited the fact that $\frac{b+i+j-a}{b+i+j-1} \leq 1$ for all $j = 1, ...t-i$ being $a > 1$. Now, observing that $\sum_{i=1}^t \alpha_t^i \leq 1$, we have:

$$\sum_{i=1}^t (\alpha_t^i)^2 \leq \frac{a}{b+t}\sum_{i=1}^t \alpha_t^i \leq \frac{a}{b+t}.$$

□

B.1.3 Optimism

We now prove that with a suitable choice of m_0, σ_0 and σ_b we are able to guarantee that $u_t^\delta(s,a)$ is optimistic w.r.t. $q^*(s,a)$ with high probability. We start proving the following intermediate result.

Lemma B.2. *For any $\delta \in [0,1]$, with probability at least $1-\delta$, we have simultaneously for all $s \in \mathcal{S}$, $a \in \mathcal{A}$ and $t \in \{1,2,...,T\}$:*

$$\sum_{i=1}^{n_t(s,a)} \alpha_{n_t(s,a)}^i \left[u_{t_i}^\delta(S_{t_i+1}) - \mathop{\mathbb{E}}_{S'\sim\mathcal{P}(\cdot|s,a)}\left[u_{t_i}^\delta(S')\right]\right] \leq q_{\max}\sqrt{\sum_{i=1}^{n_t(s,a)}\left(\alpha_{n_t(s,a)}^i\right)^2 \log\frac{|\mathcal{S}||\mathcal{A}|T}{\delta}}.$$

Proof. Let us provide a formal definition of t_i:

$$t_i = \min\left(\{t \in \{1,2,...,T\} : t > t_{i-1} \wedge (s_t,a_t) = (s,a)\} \cup \{T+1\}\right), \tag{21}$$

where we have assigned the value $T+1$ if (s,a) is experienced less than i times. Consider the filtration $\mathcal{F}_i = \sigma(S_0, A_0, R_1, ..., S_{t_i-1}, A_{t_i-1}, R_{t_i})$ generated by all the random variables realized until time t_i. The random variable $X_{t_i} = \mathbb{1}_{\{t_i \leq T\}}\left[u_{t_i}^\delta(S_{t_i+1}) - \mathbb{E}_{S'\sim\mathcal{P}(\cdot|s,a)}\left[u_{t_i}^\delta(S')\right]\right]$ is a martingale difference sequence (MDS) w.r.t. the filtration $\{\mathcal{F}_i\}_{i=1,2,...}$, as $\mathbb{E}[X_{t_i}|\mathcal{F}_i] = 0$ and $|X_{t_i}| < q_{\max}$ a.s.. Using Azuma-Hoeffding inequality and a union bound over the time $\{1,2,...,T\}$, the states $\mathcal{S}$ and the actions $\mathcal{A}$ we have that w.p. at least $1-\delta$ the statement holds. □

We are now ready to prove that $u_t^\delta(s,a)$ are optimistic with high probability.

Theorem B.1. *(Optimism) Let $m_0 = q_{\max}$, $\sigma_0 = 0$ and $\sigma_b = \frac{\gamma q_{\max}}{c z_{1-\delta}}\sqrt{a\log\frac{|\mathcal{S}||\mathcal{A}|T}{\delta}}$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$. Then, for any $\delta \in [0,1]$, with probability at least $1-\delta$, we have simultaneously for all $s \in \mathcal{S}$, $a \in \mathcal{A}$ and $t \in \{1,2,...,T\}$:*

$$u_t^\delta(s,a) \geq q^*(s,a). \tag{22}$$

Proof. The proof is by induction on t. For $t = 0$, we have that $u_0^\delta(s,a) \geq q_{\max} \geq q^*(s,a)$. Let us assume the statement hold up to time $t-1$, we prove that it holds for t. Recall that $u_t^\delta(s,a) = \min\{\overline{u}_t^\delta(s,a), q_{\max}\}$. If $u_t^\delta(s,a) = q_{\max}$ then the statement hold. Otherwise we have $u_t^\delta(s,a) = \overline{u}_t^\delta(s,a)$. Let us write explicitly the expression of the upper bound $\overline{u}_t^\delta(s,a)$. With probability at least $1-\delta$ we have:

$$\begin{aligned}\overline{u}_t^\delta(s,a) &= \alpha_{n_t(s,a)}\left[m_0 + z_{1-\delta}\sigma_0\right] + \sum_{i=1}^{n_t(s,a)} \alpha_{n_t(s,a)}^i\left[R_{t_i+1} + \gamma u_t^\delta(S_{t_i+1})\right] + \beta_{n_t(s,a)} z_{1-\delta}\sigma_b\\ &= \alpha_{n_t(s,a)}\left[m_0 + z_{1-\delta}\sigma_0\right] + \beta_{n_t(s,a)} z_{1-\delta}\sigma_b\\ &\quad + \sum_{i=1}^{n_t(s,a)} \alpha_{n_t(s,a)}^i\left[r(s,a) + \gamma \mathop{\mathbb{E}}_{S'\sim\mathcal{P}(\cdot|s,a)}\left[u_{t_i}^\delta(S')\right]\right]\\ &\quad + \gamma\sum_{i=1}^{n_t(s,a)} \alpha_{n_t(s,a)}^i\left[u_{t_i}^\delta(S_{t_i+1}) - \mathop{\mathbb{E}}_{S'\sim\mathcal{P}(\cdot|s,a)}\left[u_{t_i}^\delta(S')\right]\right]\end{aligned} \tag{23}$$

$$\geq \alpha^0_{n_t(s,a)} q^*(s,a) + \sum_{i=1}^{n_t(s,a)} \alpha^i_{n_t(s,a)} \left[r(s,a) + \gamma \mathop{\mathbb{E}}_{S' \sim \mathcal{P}(\cdot|s,a)} \left[v^*(S') \right] \right] \tag{24}$$

$$+ \beta_{n_t(s,a)} z_{1-\delta} \sigma_b - \gamma q_{\max} \sqrt{\sum_{i=1}^{n_t(s,a)} \left(\alpha^i_{n_t(s,a)} \right)^2 \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta}} \tag{25}$$

$$\geq q^*(s,a) + \beta_{n_t(s,a)} z_{1-\delta} \sigma_b - \gamma q_{\max} \sqrt{\frac{a}{b + n_t(s,a)} \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta}}, \tag{26}$$

where line (23) derives from the fact that the reward is deterministic ($R_{t_i+1} = r(s,a)$), line (24) is an application of the inductive hypothesis being all $t_i < t$ and observing that $u^\delta_{t_i}(S') = \max_{a \in \mathcal{A}} \{u^\delta_{t_i}(S', a)\} \geq \max_{a \in \mathcal{A}} \{q^*(S', a)\} = v^*(S')$ and we applied Lemma B.2 at line (25). Line (26) follows from the application of Bellman equation and using Lemma B.1 to bound the summation in the square root. In order to guarantee that this expression is non-negative for all $t \in \{1, 2, ..., T\}$ we need to satisfy:

$$\beta_{n_t(s,a)} z_{1-\delta} \sigma_b \geq \gamma q_{\max} \sqrt{\frac{a}{b + n_t(s,a)} \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta}}$$

$$\implies \sigma_b \geq \gamma q_{\max} \frac{\sqrt{d + n_t(s,a)}}{c z_{1-\delta}} \sqrt{\frac{a}{b + n_t(s,a)} \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta}}.$$

The term depending on t can be bounded recalling that $d \leq b$ and that $n_t(s,a) \leq T$:

$$\sqrt{\frac{d + n_t(s,a)}{b + n_t(s,a)}} \leq \sqrt{\frac{d+T}{b+T}} \leq 1.$$

Therefore, we choose:

$$\sigma_b = \frac{\gamma q_{\max}}{c z_{1-\delta}} \sqrt{a \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta}}.$$

□

B.1.4 Main Result

We now provide this central lemma that we will use to state a bound on the sample complexity considering our running algorithm as a non stationary policy $\mathfrak{A}$.

Lemma B.3. *Let $s \in \mathcal{S}$ be any state and let $\Delta_t(s) = v^*(s) - v_{\mathfrak{A}}(s)$ be the instantaneous regret of state s at time t and define $\Psi_t(s) = u^\delta_t(s) - v_{\mathfrak{A}}(s)$. Let $\delta \in [0,1]$, then with probability at least $1 - \delta$ the following chain inequalities holds simultaneously for all $s \in \mathcal{S}$, $a \in \mathcal{A}$ and $t \in \{1, 2, ..., T\}$: $\Delta_t(s) \leq \Psi_t(s)$ and*

$$\Psi_t(s) \leq \alpha^0_{n_t(s,a)} q_{\max} + \gamma \sum_{i=1}^{n_t(s,a)} \alpha^i_{n_t(s,a)} \Psi_{t_i}(s_{t_i+1}) + 2\gamma q_{\max} \sqrt{\frac{a}{d + n_t(s,a)} \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta}}, \tag{27}$$

where $a \in \arg\max_{a' \in \mathcal{A}} \{u^\delta_t(s, a')\}$.

Proof. Consider a state $s \in \mathcal{S}$, we decompose the instantaneous regret at time t, i.e., $\Delta_t(s)$. It is important to notice that s is not necessarily the state visited by our algorithm at time t, i.e., s_t. With probability at least $1 - \delta$ we have simultaneously for all t:

$$\begin{aligned}
\Delta_t(s) &= v^*(s) - v_{\mathfrak{A}}(s) \\
&= \max_{a \in \mathcal{A}} \{q^*(s,a)\} - v_{\mathfrak{A}}(s) \\
&\leq \max_{a' \in \mathcal{A}} \{u^\delta_t(s, a')\} - v_{\mathfrak{A}}(s) && (28) \\
&= u^\delta_t(s) - v_{\mathfrak{A}}(s) = \Psi_t(s) && (29) \\
&\leq u^\delta_t(s,a) - q_{\mathfrak{A}}(s,a), && (30)
\end{aligned}$$

were $a \in \arg\max_{a' \in \mathcal{A}} u^\delta_t(s, a')$. Line (28) follows from the optimism (Theorem B.1) and line (30) is obtained by observing that $\max_{a' \in \mathcal{A}} \{q_{\mathfrak{A}}(s, a')\} \geq q_{\mathfrak{A}}(s,a)$ for any $a \in \mathcal{A}$. We now apply the Bellman equation on the upper confidence bound:

$$u^\delta_t(s,a) - q_{\mathfrak{A}}(s,a) \leq \overline{u}^\delta_t(s,a) - q_{\mathfrak{A}}(s,a) \tag{31}$$

$$
\begin{aligned}
&= \alpha^0_{n_t(s,a)}\left(m_0 - q_{\mathfrak{A}}(s,a)\right) + \beta_{n_t(s,a)} z_{1-\delta}\sigma_b \\
&\quad + \sum_{i=1}^{n_t(s,a)} \gamma \alpha^i_{n_t(s,a)} \left(u^\delta_{t_i}(s_{t_i+1}) - \mathop{\mathbb{E}}_{S'\sim\mathcal{P}(\cdot|s,a)}\left[v_{\mathfrak{A}}(S')\right]\right) \\
&= \alpha^0_{n_t(s,a)}\left(m_0 - q_{\mathfrak{A}}(s,a)\right) + \beta_{n_t(s,a)} z_{1-\delta}\sigma_b \\
&\quad + \gamma \sum_{i=1}^{n_t(s,a)} \alpha^i_{n_t(s,a)} \left(u^\delta_{t_i}(s_{t_i+1}) - v_{\mathfrak{A}}(s_{t_i+1})\right) \\
&\quad + \gamma \sum_{i=1}^{n_t(s,a)} \alpha^i_{n_t(s,a)} \left(v_{\mathfrak{A}}(s_{t_i+1}) - \mathop{\mathbb{E}}_{S'\sim\mathcal{P}(\cdot|s,a)}\left[v_{\pi_t}(S')\right]\right) \\
&\leq \alpha^0_{n_t(s,a)} q_{\max} + \gamma q_{\max}\sqrt{\frac{a}{d+n_t(s,a)}\log\frac{|\mathcal{S}||\mathcal{A}|T}{\delta}} \qquad (32) \\
&\quad + \gamma \sum_{i=1}^{n_t(s,a)} \alpha^i_{n_t(s,a)} \left(u^\delta_{t_i}(S_{t_i+1}) - v_{\mathfrak{A}}(S_{t_i+1})\right) \\
&\quad + \gamma q_{\max}\sqrt{\sum_{i=1}^{n_t(s,a)} \left(\alpha^i_{n_t(s,a)}\right)^2 \log\frac{|\mathcal{S}||\mathcal{A}|T}{\delta}} \qquad (33) \\
&\leq \alpha^0_{n_t(s,a)} q_{\max} + 2\gamma q_{\max}\sqrt{\frac{a}{d+n_t(s,a)}\log\frac{|\mathcal{S}||\mathcal{A}|T}{\delta}} + \gamma \sum_{i=1}^{n_t(s,a)} \alpha^i_{n_t(s,a)} \Psi_{t_i}(S_{t_i+1}),
\end{aligned}
$$

where line (32) follows from observing that $m_0 - q_{\mathfrak{A}}(s,a) = q_{\max} - q_{\mathfrak{A}}(s,a) \leq q_{\max}$ and by substitution of the value of $\sigma^{(2)}_0(s,a)$ and line (33) is obtained by applying Azuma-Hoeffding inequality, like in Lemma B.2 (all it takes it to consider $v_{\mathfrak{A}}$ instead of $u^\delta_{t_i}$) and recalling that $d \leq b$. □

Using the previous lemma we are able to state the following theorem that represents the core of our analysis. In this case, we are going to evaluate how well is our algorithm performing (in terms of value function) over the states visited by the algorithm itself. This will allow us to derive immediately a guarantee on the sample complexity of PE-WQL.

Theorem B.2. *Let $S_0, S_1, ..., S_T$ be the sequence of states and actions visited by the algorithm. Let $a = \frac{2+\gamma}{2(1-\gamma)}$ and $b = a - 1$. Then, under the same assumptions as Lemma B.1, for any $\delta \in [0,1]$, with probability at least $1-\delta$ it holds that:*

$$
\sum_{t=1}^{T} \Delta_t(S_t) \leq \mathcal{O}\left(\frac{q_{\max}}{(1-\gamma)^{3/2}}\sqrt{|\mathcal{S}||\mathcal{A}|T\log\frac{|\mathcal{S}||\mathcal{A}|T}{\delta}}\right). \qquad (34)
$$

Proof. We are now going do deal with the summation $\sum_{t=1}^{T}\Psi_t(s_{t+1})$:

$$
\begin{aligned}
\sum_{t=1}^{T}\Psi_t(s_{t+1}) = q_{\max}\underbrace{\sum_{t=1}^{T}\alpha^0_{n_t(s_{t+1},a_{t+1})}}_{\text{(i)}} + \gamma q_{\max}\sqrt{a\log\frac{|\mathcal{S}||\mathcal{A}|T}{\delta}}\underbrace{\sum_{t=1}^{T}\frac{1}{\sqrt{d+n_t(s_{t+1},a_{t+1})}}}_{\text{(ii)}} \\
+ \underbrace{\gamma\sum_{t=1}^{T}\sum_{i=1}^{n_t(s_{t+1},a_{t+1})}\alpha^i_{n_t(s_{t+1},a_{t+1})}\Psi_{t_i(s_{t+1},a_{t+1})}(s_{t_i(s_{t+1},a_{t+1})+1})}_{\text{(iii)}}
\end{aligned}
$$

We make the following observation we are going to use throughout the proof

$$
n_t(s_{t+1},a_{t+1}) = n_{t+1}(s_{t+1},a_{t+1}) - 1. \qquad (35)
$$

Let us start with (i):

$$
\begin{aligned}
\sum_{t=1}^{T}\alpha^0_{n_t(s_{t+1},a_{t+1})} &= \sum_{t=1}^{T}\alpha^0_{n_{t+1}(s_{t+1},a_{t+1})-1} \\
&= \sum_{h=2}^{T+1}\alpha^0_{n_h(s_h,a_h)-1} \qquad (36)
\end{aligned}
$$

$$
\begin{aligned}
&\leq \sum_{h=1}^{T+1} \alpha^0_{n_h(s_h,a_h)-1} \\
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \sum_{i=1}^{n_{T+1}(s,a)} \alpha^0_{i-1} && (37)\\
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \sum_{i=0}^{n_{T+1}(s,a)-1} \alpha^0_{i} \\
&\leq \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \sum_{i=0}^{+\infty} \alpha^0_{i} = \left(\frac{b}{a-1}+1\right)|\mathcal{S}||\mathcal{A}|, && (38)
\end{aligned}
$$

where we made the change of variable $h = t + 1$ to get line (36), we decomposed the summation over the state action pairs obseving that each of them appears $n_{T+1}(s, a)$ times to get line (37) and we used Lemma B.1 to get line (38).

Let us now consider (ii); using Equation (35) we get:

$$
\begin{aligned}
\sum_{t=1}^{T} \frac{1}{\sqrt{d+n_t(s_{t+1},a_{t+1})}} &= \sum_{t=1}^{T} \frac{1}{\sqrt{d+n_{t+1}(s_{t+1},a_{t+1})-1}} \\
&= \sum_{h=2}^{T+1} \frac{1}{\sqrt{d+n_h(s_h,a_h)-1}} \\
&\leq \sum_{h=1}^{T+1} \frac{1}{\sqrt{d+n_h(s_h,a_h)-1}} \\
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \sum_{i=1}^{n_{T+1}(s,a)} \frac{1}{\sqrt{d+i-1}} && (39)\\
&\leq 2\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \sqrt{n_{T+1}(s,a)} && (40)\\
&\leq 2\sqrt{|\mathcal{S}||\mathcal{A}|(T+1)}, && (41)
\end{aligned}
$$

where line (39) derives from decomposing the summation over state-action pairs and observing that each state-action pair appears $n_{T+1}(s, a)$ times. Line (40) is obtained by bounding the summation with the integral: $\int_1^{n_{T+1}(s,a)+1} \frac{1}{\sqrt{b+x-1}}\mathrm{d}x = 2\sqrt{d+n_{T+1}(s,a)} - 2\sqrt{d} \leq 2\sqrt{n_{T+1}(s,a)}$, where the last inequality derives from the subadditivity of the square root. Finally, line (41) is obtained by observing that $\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} n_{T+1}(s,a) = T+1$ and the expression is maximized by taking $n_{T+1}(s,a) = \frac{T+1}{|\mathcal{S}||\mathcal{A}|}$.

Now we consider the term (iii). First observe that $n_t(s_{t+1}, a_{t+1}) \geq 1$ in order to appear in the inner summation. Consider the derivation:

$$
\begin{aligned}
&\sum_{t=1}^{T} \sum_{i=1}^{n_t(s_{t+1},a_{t+1})} \alpha^i_{n_t(s_{t+1},a_{t+1})} \Psi_{t_i(s_{t+1},a_{t+1})}(s_{t_i(s_{t+1},a_{t+1})+1}) \\
&\quad= \sum_{t=1}^{T} \sum_{i=1}^{n_{t+1}(s_{t+1},a_{t+1})-1} \alpha^i_{n_{t+1}(s_{t+1},a_{t+1})-1} \Psi_{t_i(s_{t+1},a_{t+1})}(s_{t_i(s_{t+1},a_{t+1})+1}) \\
&\quad= \sum_{h=2}^{T+1} \sum_{i=1}^{n_h(s_h,a_h)-1} \alpha^i_{n_h(s_h,a_h)-1} \Psi_{t_i(s_h,a_h)}(s_{t_i(s_h,a_h)+1}) \\
&\quad\leq \sum_{h=1}^{T+1} \sum_{i=1}^{n_h(s_h,a_h)-1} \alpha^i_{n_h(s_h,a_h)-1} \Psi_{t_i(s_h,a_h)}(s_{t_i(s_h,a_h)+1}) && (42)\\
&\quad= \sum_{t=1}^{T+1} \Psi_t(s_{t+1}) \sum_{i=n_t(s_t,a_t)}^{n_{T+1}(s_t,a_t)} \alpha^{n_t(s_t,a_t)}_{i-1} && (43)\\
&\quad\leq \sum_{t=1}^{T+1} \Psi_t(s_{t+1}) \sum_{i=2}^{+\infty} \alpha^{n_t(s_t,a_t)}_{i-1} && (44)
\end{aligned}
$$

$$\leq \sum_{t=1}^{T+1} \Psi_t(s_{t+1}) \sum_{i=1}^{+\infty} \alpha_i^{n_t(s_t,a_t)}$$

$$\leq \frac{a}{a-1} \sum_{t=1}^{T+1} \Psi_t(s_{t+1}) \tag{45}$$

$$\leq \frac{a}{a-1} \sum_{t=1}^{T} \Psi_t(s_{t+1}) + \frac{a}{a-1} q_{\max}. \tag{46}$$

where line (43) is obtained by observing that given $\Psi_t(s_{t+1})$ it is going to appear in the summation for all $t \geq t_i(s_t, a_t)$. The first time it will appear multiplied by $\alpha_{n_t(s_t,a_t)-1}^{n_t(s_t,a_t)}$, the second time with $\alpha_{n_t(s_t,a_t)}^{n_t(s_t,a_t)}$ and so on. Line (44) derives from observing that $n_t(s_{t+1}, a_{t+1}) \geq 1$, thus $n_{t+1}(s_{t+1}, a_{t+1}) \geq 2$. Line (45) is obtained by applying Lemma B.1.

Now, we put all together into the bound on the summation:

$$\sum_{t=1}^{T} \Psi_t(s_{t+1}) \leq q_{\max} \left(\frac{b}{a-1} + 1 \right) |\mathcal{S}||\mathcal{A}| + 2\gamma q_{\max} \sqrt{a \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta}} \sqrt{|\mathcal{S}||\mathcal{A}|(T+1)}$$
$$+ \frac{a\gamma}{a-1} \sum_{t=1}^{T} \Psi_t(s_{t+1}) + \frac{a\gamma}{a-1} q_{\max}.$$

In order to solve the inequality, we must require $\frac{a\gamma}{a-1} < 1$, i.e., $a > \frac{1}{1-\gamma}$. In such a case, we obtain:

$$\sum_{t=1}^{T} \Psi_t(s_{t+1}) \leq \left(1 - \frac{a\gamma}{1-a}\right)^{-1} \left[q_{\max} \left(\frac{b}{a-1} + 1 \right) |\mathcal{S}||\mathcal{A}| \right.$$
$$\left. + 2\gamma q_{\max} \sqrt{a \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta}} \sqrt{|\mathcal{S}||\mathcal{A}|(T+1)} + \frac{a\gamma}{a-1} q_{\max} \right].$$

Now, we propose a value for a and b that fulfills all the conditions. In particular, among all possible values we select:[10] $a = \frac{2+\gamma}{2(1-\gamma)}$. Then we take the smallest possible value for b, i.e., $b = a - 1$. From which we get:

$$\sum_{t=1}^{T} \Psi_t(s_{t+1}) \leq \frac{3}{1-\gamma} \left[2q_{\max}|\mathcal{S}||\mathcal{A}| + 2\gamma q_{\max} \sqrt{\frac{2+\gamma}{2(1-\gamma)} \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta}} \sqrt{|\mathcal{S}||\mathcal{A}|(T+1)} + \frac{2+\gamma}{3} q_{\max} \right]$$
$$\leq \frac{3}{1-\gamma} \left[q_{\max}(|\mathcal{S}||\mathcal{A}| + 1) + 2\gamma q_{\max} \sqrt{\frac{3}{2(1-\gamma)} \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta}} \sqrt{|\mathcal{S}||\mathcal{A}|(T+1)} \right]$$
$$\leq \mathcal{O}\left(\frac{q_{\max}}{(1-\gamma)^{3/2}} \sqrt{|\mathcal{S}||\mathcal{A}|T \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta}} \right),$$

where the last passage is obtained by observing that: if $T + 1 \geq \sqrt{|\mathcal{S}||\mathcal{A}|(T+1)}$ then $\sqrt{|\mathcal{S}||\mathcal{A}|(T+1)} \geq |\mathcal{S}||\mathcal{A}|$; on the contrary $\sum_{t=1}^{T} \Psi_t(s_{t+1}) \leq T \leq \sqrt{|\mathcal{S}||\mathcal{A}|(T+1)} - 1$. Thus, we can discard the first term. □

Theorem B.2 allows us to bound the per-step regret over the trajectories visited by the algorithm.

Corollary B.1. *Let $a = \frac{2+\gamma}{2(1-\gamma)}$ and $b = a - 1$. Then, under the same assumptions as Lemma B.1, for any $\delta \in [0, 1]$, with probability at least $1 - \delta$ for $T \geq T_0$:*

$$T_0 = \mathcal{O}\left(\frac{q_{\max}^2 |\mathcal{S}||\mathcal{A}|}{\epsilon^2 (1-\gamma)^3} \log \frac{q_{\max}^2 |\mathcal{S}|^2 |\mathcal{A}|^2}{\delta \epsilon^2 (1-\gamma)^3} \right)$$

we have that:

$$\frac{1}{T} \sum_{t=1}^{T} \Delta_t(S_t) \leq \epsilon.$$

[10] It can be easily proved that taking the value of a that minimizes $\left(1 - \frac{a\gamma}{1-a}\right)^{-1} \sqrt{a}$ just changes the bound by a constant and does not modify the dependence on $(1 - \gamma)$.

Proof. Following the reasoning of [21], we just need to find a sufficiently large T_0 such that for all $T \geq T_0$ the per step regret is smaller than ϵ:

$$\frac{1}{T}\mathcal{O}\left(\frac{q_{\max}}{(1-\gamma)^{3/2}}\sqrt{|\mathcal{S}||\mathcal{A}|T\log\frac{|\mathcal{S}||\mathcal{A}|T}{\delta}}\right) < \epsilon \implies T \geq \mathcal{O}\left(\frac{q_{\max}^2|\mathcal{S}||\mathcal{A}|}{\epsilon^2(1-\gamma)^3}\log\frac{|\mathcal{S}||\mathcal{A}|T}{\delta}\right)$$

We rename $\tau = \frac{q_{\max}^2|\mathcal{S}||\mathcal{A}|}{\epsilon^2(1-\gamma)^3}$. We select $T_0 = \mathcal{O}\left(2\tau\log\frac{\tau|\mathcal{S}||\mathcal{A}|}{\delta}\right)$. Therefore, we have:

$$\begin{aligned}
T_0 &= \mathcal{O}\left(2\tau\log\frac{\tau|\mathcal{S}||\mathcal{A}|}{\delta}\right) \\
&= \mathcal{O}\left(\tau\log\left(\frac{\tau|\mathcal{S}||\mathcal{A}|}{\delta}\frac{\tau|\mathcal{S}||\mathcal{A}|}{\delta}\right)\right) \\
&\geq \mathcal{O}\left(\tau\log\left(2\frac{\tau|\mathcal{S}||\mathcal{A}|}{\delta}\log\frac{\tau|\mathcal{S}||\mathcal{A}|}{\delta}\right)\right) \\
&= \mathcal{O}\left(\tau\log\frac{T_0|\mathcal{S}||\mathcal{A}|}{\delta}\right) \\
&= \mathcal{O}\left(\frac{q_{\max}^2|\mathcal{S}||\mathcal{A}|}{\epsilon^2(1-\gamma)^3}\log\frac{|\mathcal{S}||\mathcal{A}|T_0}{\delta}\right),
\end{aligned}$$

where we exploited the inequality $x > 2\log x$ for $x > 0$. □

Finally, we prove that MWQL is PAC-MDP in the average loss setting. We import several ideas from [41]. First of all, we recall the notion of adjusted loss.

Definition B.1 (Definition 5 of [42])**.** *Suppose a learning algorithm $\mathfrak{A}$ is run for one sequence of $T_1 + T_2 - 1$ steps. Consider partial sequence $S_0, R_1, ..., S_{t-1}, R_t, S_t$ visited by $\mathfrak{A}$. For any policy π and integer t such that $t \leq T_1$, let $R_{T_2}^{\pi}(t) = \sum_{t'=t}^{t+T_2-1}\gamma^{t'-t}R_{t'+1} + \gamma^{T_2}v_{\pi}(S_{t+T_2})$ be the* adjusted return*. Let $I_{T_2}^{\pi}(t) = v_{\pi}(S_t) - R_{T_2}^{\pi}(t)$ be the* adjusted instantaneous loss*. Let $L_{T_1,T_2}^{\pi} = \frac{1}{T_1}\sum_{t=1}^{T_1} I_{T_2}^{\pi}(t)$ be the* adjusted average loss*.*

We are now ready to prove the result.

Theorem 5.2. *Under the hypothesis of Theorem 5.1, MWQL with Gaussian posterior, OE and OX is PAC-MDP in the average loss setting, i.e., for any $\epsilon \geq 0$ and $\delta \in [0, 1]$, after*

$$T = \mathcal{O}\left(\frac{q_{\max}^2|\mathcal{S}||\mathcal{A}|}{\epsilon^2(1-\gamma)^3}\log\frac{q_{\max}^2|\mathcal{S}|^2|\mathcal{A}|^2}{\delta\epsilon^2(1-\gamma)^3}\right)$$

steps we have that the average loss $\mathcal{L}_{\mathfrak{A}} \leq \epsilon$ with probability at least $1 - \delta$.

Proof. A running algorithm can be viewed as a non stationary policy $\mathfrak{A}$. Define $T = T_1 + T_2 - 1$ and define the adjusted average loss of $\mathfrak{A}_t$ w.r.t. itself.

$$L_{T_1,T_2}^{\mathfrak{A}} = \frac{1}{T_1}\sum_{t=1}^{T_1} I_{T_2}^{\mathfrak{A}}(t). \tag{47}$$

In [42] it is proven that for any algorithm $\mathfrak{A}$ and for

$$T_1 \geq \max\left\{\frac{1 + 2\log(1/\delta)q_{\max}^2}{\epsilon^2(1-\gamma)^2}, 2\log(1/\delta)q_{\max}^2(T_2 - 1)\right\}, \tag{48}$$

we have that $L_{T_1,T_2}^{\mathfrak{A}} \leq \epsilon$ with probability at least $1 - \delta$. We now consider the adjusted average loss w.r.t. to the optimal policy π^* and we decompose it:

$$\begin{aligned}
L_{T_1,T_2}^{\pi^*} &= \frac{1}{T_1}\sum_{t=1}^{T_1} I_{T_2}^{\pi^*}(t) &(49)\\
&= \frac{1}{T_1}\sum_{t=1}^{T_1} I_{T_2}^{\pi^*}(t) \pm \frac{1}{T_1}\sum_{t=1}^{T_1} I_{T_2}^{\mathfrak{A}}(t) \\
&\leq \epsilon + \frac{1}{T_1}\sum_{t=1}^{T_1}\left(I_{T_2}^{\pi^*}(t) - I_{T_2}^{\mathfrak{A}}(t)\right) &(50)
\end{aligned}$$

$$= \epsilon + \frac{1}{T_1}\sum_{t=1}^{T_1}\left(v_{\pi^*}(S_t) - R^{\pi^*}_{T_2}(t) - \left(v_{\mathfrak{A}}(S_t) - R^{\mathfrak{A}}_{T_2}(t)\right)\right) \tag{51}$$

$$= \epsilon + \frac{1}{T_1}\sum_{t=1}^{T_1}\left(v_{\pi^*}(S_t) - v_{\mathfrak{A}}(S_t)\right) + \frac{1}{T_1}\sum_{t=1}^{T_1}\gamma^{T_2}\left(v_{\mathfrak{A}}(S_{t+T_2}) - v_{\pi^*}(S_{t+T_2})\right) \tag{52}$$

$$\leq \epsilon + \frac{1}{T_1}\sum_{t=1}^{T_1}\Delta_t(S_t) \tag{53}$$

$$\leq 2\epsilon, \tag{54}$$

where (50) derives from the fact that $L^{\mathfrak{A}}_{T_1,T_2} \leq \epsilon$, (51) is from the definition of adjusted instantaneous loss, (52) derives from the definition of adjusted return, (53) is obtained by observing that $v_{\mathfrak{A}}(S_{t+T_2}) \leq v_{\pi^*}(S_{t+T_2})$ and the definition of $\Delta_t(S_t)$ and (54) derives from Corollary B.1. Therefore, the inequality hold for T_1 satisfying condition (48) and Corollary B.1, with probability at least $1-2\delta$. Proposition 3 of [42] proves that for $T_1 \geq \frac{2T_2}{\epsilon}$ and $T_2 \geq \frac{\log(\epsilon(1-\gamma))}{\log\gamma}$ we have that the adjusted loss $L^{\pi^*}_{T_1,T_2}$ is ϵ-close to the average loss $\mathcal{L}_{\mathfrak{A}}$. Therefore we have that $\mathcal{L}_{\mathfrak{A}} \leq 3\epsilon$ with probability at least $1-2\delta$ provided that:

$$T_1 = \mathcal{O}\left(\frac{q^2_{\max}|\mathcal{S}||\mathcal{A}|}{9\epsilon^2(1-\gamma)^3}\log\frac{q^2_{\max}|\mathcal{S}|^2|\mathcal{A}|^2}{2\delta\epsilon^2(1-\gamma)^3}\right) = \mathcal{O}\left(\frac{q^2_{\max}|\mathcal{S}||\mathcal{A}|}{\epsilon^2(1-\gamma)^3}\log\frac{q^2_{\max}|\mathcal{S}|^2|\mathcal{A}|^2}{\delta\epsilon^2(1-\gamma)^3}\right). \tag{55}$$

It can be easily proved that among all the conditions T_1 has to satisfy the most restrictive is the one imposed by Corollary B.1. □

C Additional Experimental Results

In this section, we provide the experimental setup we adopted and some additional results we did not include in the main paper.

C.1 Tabular RL

C.1.1 Experimental Setup

We train each agent for 100 episodes, the length of each episode is domain dependent. During the training periods we collect the rewards and use them to calculate the *online scores*. After each training period, we turn off exploration and evaluate the greedy policies learned by the agent. The length of the evaluation episodes is the same as the training episodes. We use the rewards collected during evaluation to calculate the *offline* scores. We perform this process in each domain, for each algorithm considered and show the mean scores of 10 runs with 95% c.i.. We calculate the undiscounted scores, even though we use a discount factor $\gamma = 0.99$ in each domain during learning.

For our particle algorithms, we initialize the particles equally spaced in an interval $[q_{\min}, q_{\max}]$, for each state action-pair. For the Gaussian model we initialized $\mu_0(s,a) = (q_{\max} + q_{\min})/2$ and $\sigma_0(s,a) = (q_{\max} - q_{\min})/\sqrt{12}$. The range of this interval is problem dependent and we see these hyperparameters as a way to incorporate prior knowledge about the domain. We consider Bootstrapped Q-learning with two policy models, the Bootstrapped policy defined in [30] and the posterior sampling policy. We initialize the Q-tables with values drawn from a Gaussian distribution with parameters $\mu = \frac{q_{\min}+q_{\max}}{2}, \sigma = q_{\max} - q_{\min}$. Furthermore, we consider Q-learning algorithm with ϵ-greedy and Boltzmann exploration. In both Q-learning versions, the Q-table is initialized to 0. We compare our results with Delayed Q-learning [41], a model-free PAC-MDP algorithm. In each of the problems considered we tuned the m parameter, the number of visits necessary to attempt an update for each state-action pair, to find the one that yields better results. We did not employ the theoretical values being too much conservative.

For all algorithms, we use an *exponentially decaying* learning rate given by:

$$\alpha_t(s,a) = \frac{b}{t(s,a)^a}, \tag{56}$$

where $t(s,a)$ is the visit count for state-action pair (s,a), b is the initial value which we set to 1 and a is the *decay exponent*. We cross validated the value of a, which was set to $a = 0.2$, for all our experiments.

For the Q-learning algorithms, we had to chose also the schedules for ϵ and β, for ϵ-greedy and Boltzmann exploration respectively. For ϵ we used an exponentially decaying schedule as in (56) with $b = 1$ and $a = 0.5$ whereas for the Boltzmann policy we used an exponentially decaying β with initial value, $b = 1.5q_{\max}$ and decay exponent, $a = 0.5$.

Here we show the results on more domains: Taxi, Chain and Loop domain from [15], River Swim and Six Arms from [42] and Knight Quest from [19].

C.1.2 Comparison of WQL with State of the Art Algorithms

In Figure 4, we show a comparison between the best version of WQL, using the two models to approximate posteriors, and the considered RL algorithms.

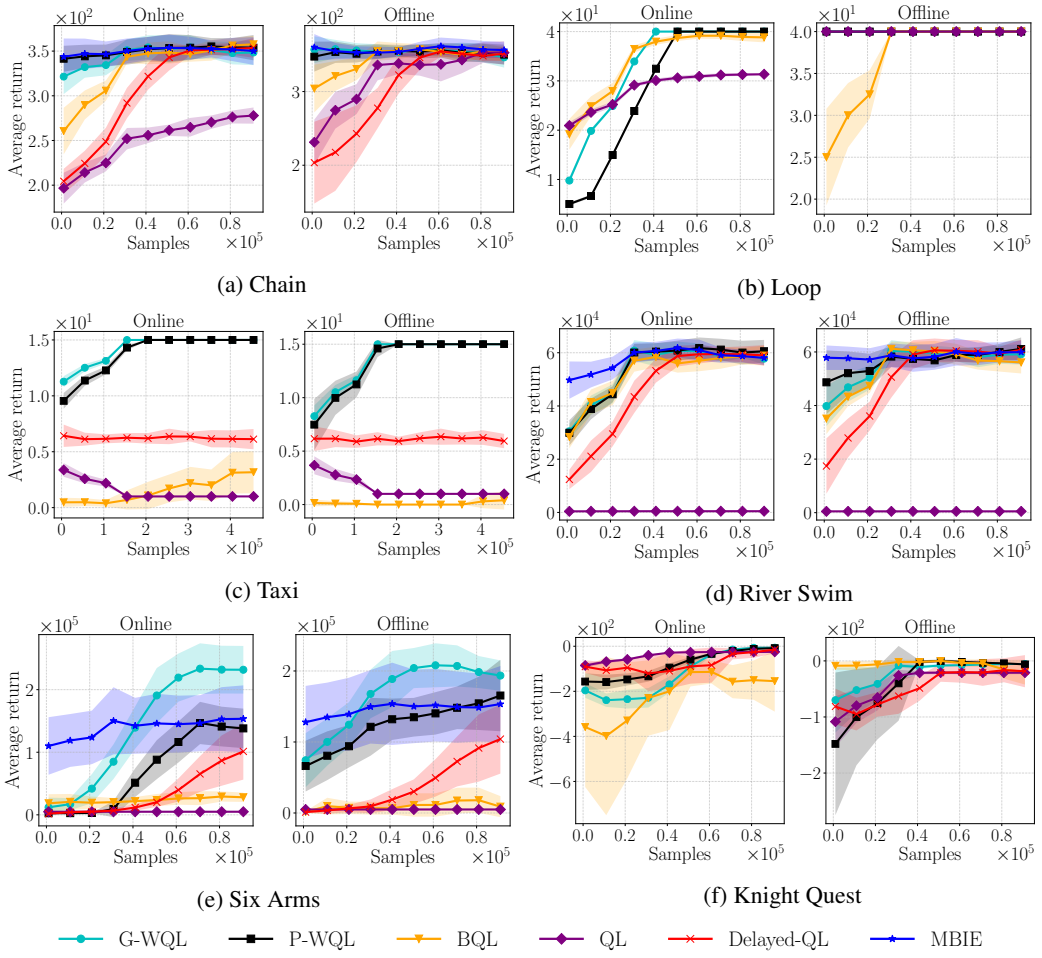

Figure 4: Comparison of G-WQL, P-WQL, QL, BQL and Delayed QL. 10 runs, 95% c.i.

C.1.3 Results of WQL algorithm

Figure 5 provides a full empirical analysis of the different flavors of WQL algorithms in the domains we considered.

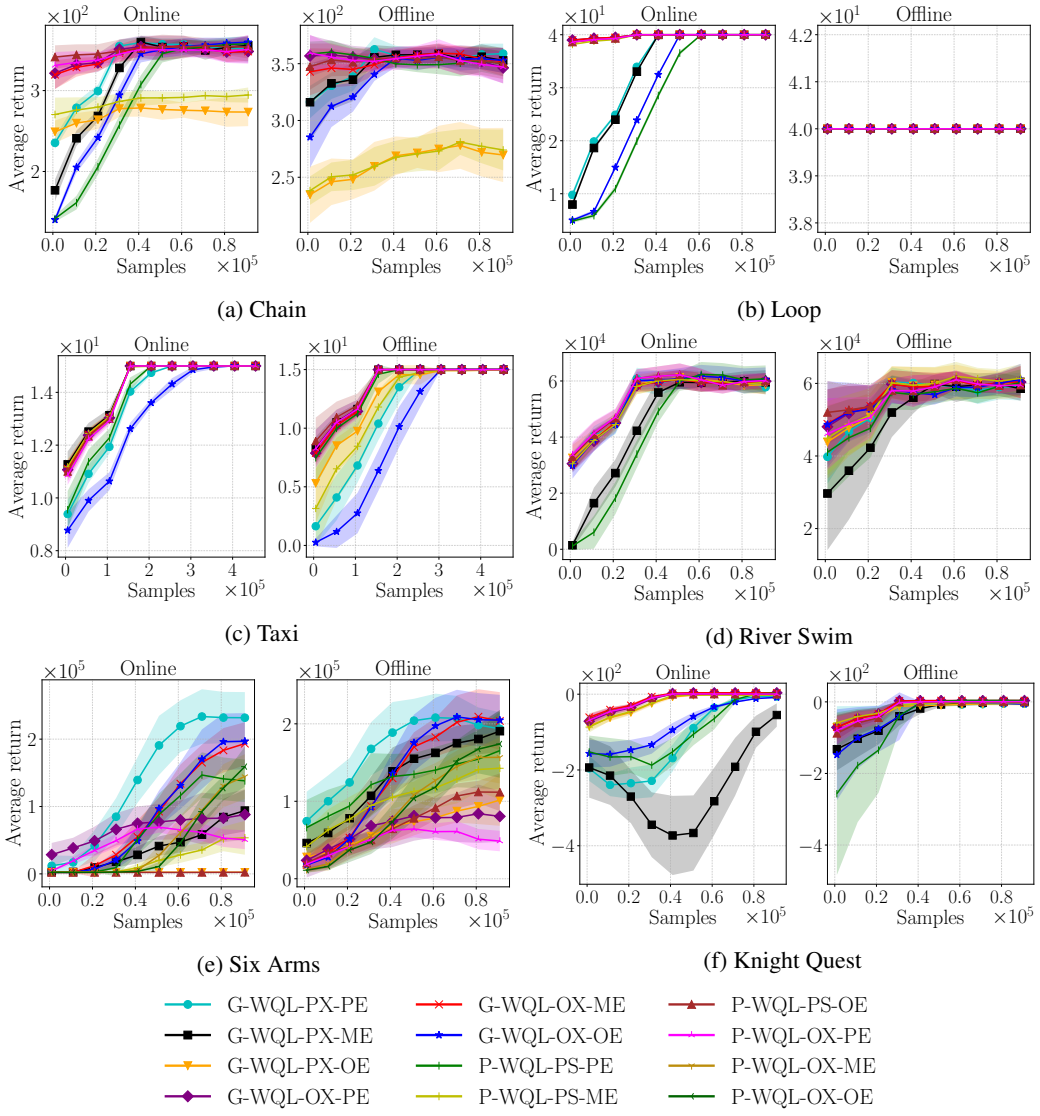

Figure 5: Results of different variations of WQL algorithm. 10 runs, 95% c.i.

C.1.4 Effect of initialization in Particle WQL

We analyzed the effect of the initialization of the prior distributions in the particle case. We argue that the particle algorithm performs better when used with particles equally spaced in a given interval $[q_{\min}, q_{\max}]$. To show this we added noise to these equally spaced particles and ran the learning algorithm in the same domain. Figure 6 shows our results as a function of α, denoting how spaced are the particles between each other. More specifically, $\alpha = 0$ means the particles were drawn uniformly random in the interval $[q_{\min}, q_{\max}]$ and $\alpha = 1$ means the particles are equally spaced in this interval. Any value in between is a combination of the two. For each α, we averaged the learning curves of the agent. It seems clear to us that using equally spaced particles to represent the prior yields better results.

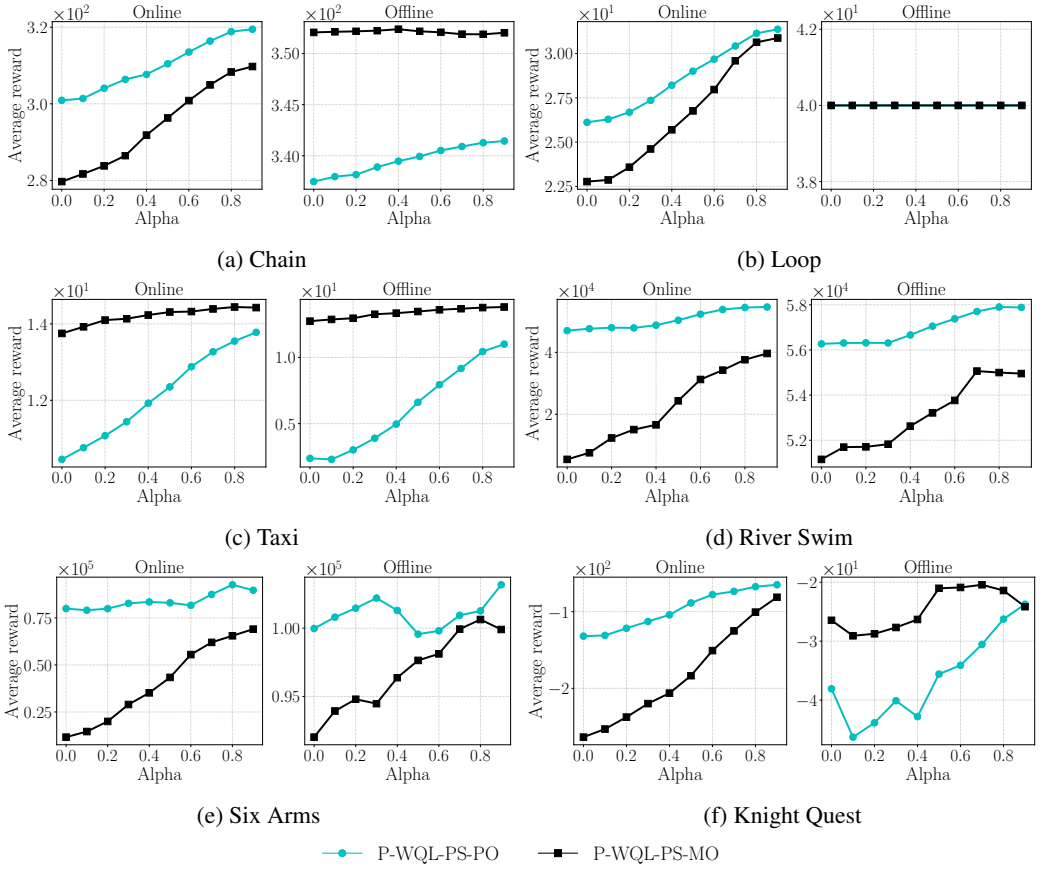

Figure 6: Effects of the initialization of the particles in Particle WQL. 10 runs.

C.2 Deep RL

C.2.1 Experimental Setup

We test the algorithms using the Arcade Learning Environment (ALE). Each step of the agent corresponds to four steps of the emulator, where the same action is repeated. The reward values observed by the agents are clipped between -1 and 1 for stability. We evaluate our agents and report performance based upon the raw scores and not the discounted scores. As it is common in literature, we do not show the online performance of the agent during training. We show the scores collected, when exploiting the greedy policies derived from the Q-function after each training period.

The *convolutional* part of the network used is identical to the one used in [30]. The input to the network is $4 \times 84 \times 84$ tensor with a rescaled, grayscale version of the last four observations. The

first convolutional layer has 32 filters of size 8 with a stride of 4. The second layer has 64 filters of size 4 with stride 2. The last layer has 64 filters of size 3. We split the network beyond the final layer into $M = 10$ distinct heads, each one is fully connected and identical to the network in [30]. This consists of a fully connected layer to 512 units followed by another fully connected layer to the Q-Values for each action. The fully connected layers all use *Rectified Linear Units* (ReLU) as a *non-linearity*. We trained the networks with *RMSProp* optimizer. The discount was set to $\gamma = 0.99$, the number of steps between target updates was set to $\tau = 10000$ steps. The agents were evaluated every 1M frames.

The *experience replay* contains the 1M most recent transitions. We update the network every 4 steps by randomly sampling a minibatch of 32 transitions from the replay buffer to use the exact same minibatch schedule as Bootstrapped DQN.

C.2.2 Network Initialization

An important problem we had to deal with was how to initialize the heads of the deep network. In the tabular case, we initialized the particles equally spaced in the interval $[q_{\min}, q_{\max}]$. We found it is not equally simple to extend this in the deep RL setting. We initialized the networks heads near the same interval by setting the bias of the last layer to the desired values.