Efficient Exploration through Bayesian Deep Q-Networks

Kamyar Azizzadenesheli [1] **Animashree Anandkumar** [2]

Abstract

We propose Bayesian Deep Q-Networks (BDQN), a Thompson sampling approach for Deep Reinforcement Learning (DRL) in Markov decision processes (MDP). BDQN is an efficient exploration-exploitation algorithm which combines Thompson sampling with deep-Q networks (DQN) and directly incorporates uncertainty over the Q-value in the last layer of the DQN, on the feature representation layer. This allows us to efficiently carry out Thompson sampling through Gaussian sampling and Bayesian Linear Regression (BLR), which has fast closed-form updates. We apply our method to a wide range of Atari games and compare BDQN to a powerful baseline: the double deep Q-network (DDQN). Since BDQN carries out more efficient exploration, it is able to reach higher rewards substantially faster: in less than 5M$\pm$1M interactions for almost half of the games to reach DDQN scores. We also establish theoretical guarantees for the special case when the feature representation is d-dimensional and fixed. We provide the Bayesian regret of posterior sampling RL (PSRL) and frequentist regret of the optimism in the face of uncertainty (OFU) for episodic MDPs.

1. Introduction

One of the central challenges in reinforcement learning (RL) is to design efficient exploration-exploitation trade-off that also scales to high-dimensional state and action spaces. Recently deep RL has shown good promise in being able to scale to high-dimensional (continuous) spaces. These successes are mainly demonstrated in simulated domains where exploration is considered to be inexpensive and simple exploration strategies are deployed, e.g. ε-greedy which uniformly explores over all the actions with ε probability. Such exploration strategies inherently inefficient for complex high-dimensional environments. On the other hand, more sophisticated strategies have mostly been limited to low dimensional MDPs. For example, OFU is only practical when the domain is small enough to be represented with lookup tables for the Q-values (Jaksch et al., 2010; Brafman & Tennenholtz, 2003).

An alternative to optimism-under-uncertainty is Thompson Sampling (TS), a general sampling and randomization approach (in both frequentist and Bayesian settings) (Thompson, 1933). Under the Bayesian framework, Thompson sampling maintains a posterior distribution over the environment model and updates it as more observation is experienced. Thompson sampling has been observed to provide compelling performance compared to optimistic approaches in many low dimensional settings such as contextual bandits (Chapelle & Li, 2011), small MDPs (Osband et al., 2013) and also has strong theoretical bounds (Russo & Van Roy, 2014a;b; Agrawal & Goyal, 2012; Osband et al., 2013; Abbasi-Yadkori & Szepesvári, 2015).

In the MDP setting, (model-based) Thompson Sampling involves sampling the parameters of the reward and dynamics model, it then performs MDP planning using the sampled model and deploys the corresponding policy for exploration-exploitation (Strens, 2000; Osband et al., 2013; Osband & Van Roy, 2014b;a). However, the posterior sampling and planning computational costs becomes intractable as the problem dimension grows. To mitigate the computation bottleneck, function approximation methods on either the model, the Q-value, or the policy are proposed to scale Thompson Sampling to high dimensional domains. To address this, Osband et al. (2014) introduces randomized least-squares value iteration (RLSVI) which combines linear value function approximation with Bayesian regression to directly sample the value-function weights from a distribution. The authors prove a regret bound for this approach in tabular MDPs. This has been extended to continuous spaces by Osband et al. (2016), where deep networks are used to approximate the Q function. Through a bootstrapped-ensemble approach, several deep-Q network (DQN) models are trained in parallel to approximate the posterior distribution. Other works use the posterior distribution over the parameters of each node in the network and employ variational approximation (Lipton et al., 2016b) or

[1]University of California, Irvine, *kazizzad@uci.edu* [2]Caltech, *anima@caltech.edu*.

noisy networks (Fortunato et al., 2017). These approaches significantly increase the computation cost over the standard DQN. For instance, the bootstrapped-ensemble incurs a computation overhead that is linear in the number of bootstrap models. Moreover, despite principled design of these methods, they do not provide performance beyond the modest gain of DQN in empirical studies.

Contribution 1 – Design of BDQN: We introduce Bayesian Deep Q-Network (BDQN), a Thompson-sampling algorithm for deep RL. It is a simple approach that extends randomized least-squares value iteration (Osband et al., 2014) to deep neural networks. We introduce stochasticity only in the last layer of the Q-network using independent Gaussian priors on the weights. This allows us to efficiently approximate Thompson sampling[1] using Bayesian linear regression (BLR), which has fast closed-form updates and sampling from the resulting Gaussian posterior distribution is inexpensive. The rest of the Q-network is trained through standard back propagation.

Contribution 2 – Strong empirical results for BDQN: We test BDQN on a wide range of Atari games (Bellemare et al., 2013; Machado et al., 2017), and compare our results to a powerful baseline: Double DQN (DDQN) (Van Hasselt et al., 2016) a bias-reduced extension of DQN. BDQN and DDQN use the same network architecture, and follow the same target objective, and differ only in the way they select actions: DDQN uses ε-greedy exploration while BDQN performs (approximated) Thompson sampling.

We found that BDQN is able to reach much higher cumulative rewards in fewer interaction with the environment, compared to DDQN on all the tested games. We also found that BDQN can be trained with much higher learning rates compared to DDQN. This is intuitive since BDQN has better exploration strategy. The cumulative reward (score) for BDQN at the end of training improves by a median of 300% with a maximum of $80K\%$ in these games. Also, BDQN has $300\% \pm 40\%$ (mean and standard deviation) improvement over these games on area under the performance measure. This can be considered as a surrogate for sample complexity and regret. Indeed, no single measure of performance provides a complete picture of an algorithm, and we present detailed experiments in Section 5.

In terms of computational cost, BDQN is only slightly more expensive compared to DQN and DDQN. For the DQN in Atari games, this is the cost of inverting a 512×512 matrix every 100,000 time steps, which is negligible. On the other hand, more sophisticated Bayesian RL techniques are significantly more expensive and have not lead to large gains over DQN and DDQN (Osband et al., 2016).

Contribution 3 – Bayesian and frequentist regret upper bounds for continuous MDPs: We establish theoretical guarantees for the special case when the feature representation is fixed (i.e. all layers except the last), and not learnt. We consider episodic MDPs with continuous space of states and actions such that the Q-function is a linear function of a given d-dimensional feature map. We show that when PSRL and OFUare deployed, respectively, the Bayesian regret and frequentist regret after T episode are upper bounded by $\widetilde{\mathcal{O}}(d\sqrt{T})$. Similar to linear regression (Hsu et al., 2012), we consider an upper bound on the spectral deviation of the feature representation to derive an upper bound on the model estimation error. We show how this error for the episodic environments compounds to derive the dependence on the horizon length H. Since linear bandits are a special case of episodic continuous MDPs, with horizon length 1, it implies that for this case our regret bounds are tight in the dimension d and in the number of episodes for horizon 1. The Bayesian bound matches the Bayesian regret bound of linear bandits (Russo & Van Roy, 2014a) and the frequentist bound matches the frequentist regret bound for linear bandits (Abbasi-Yadkori et al., 2011). To the best of our knowledge, these are the first model free theoretical guarantee for continuous MDPs beyond the tabular setting.

Thus, our proposed approach has several desirable features; faster learning and better sample complexity due to targeted exploration, negligible computational overhead due to simplicity, significant improvement in experiments, and theoretical bounds. It is worth noting that BDQN can also be seen as a Gaussian approximation to PSRL where the posterior is approximated using a Gaussian distribution. Furthermore, it can also be seen as OFUwhen we fit a Gaussian distribution to the approximated confidence interval (Abeille & Lazaric, 2017).

2. Thompson Sampling vs ε-greedy and Boltzmann exploration

In value approximation RL algorithms, there are different ways to manage the exploration-exploitation trade-off. DQN uses a naive ε-greedy for exploration, where with ε probability it chooses a random action and with $1-\varepsilon$ probability it chooses the greedy action based on the estimated Q function. Note that there are only point estimates of the Q function in DQN. In contrast, our proposed Bayesian approach BDQN maintains uncertainties over the estimated Q function, and employs it to carry out Thompson Sampling based exploration-exploitation. Here, we demonstrate the fundamental benefits of Thompson Sampling over ε-greedy and Boltzmann exploration strategies using simplified examples. In Table 1, we list the three strategies and their

[1] BDQN approximates the posterior distribution which results in approximated Thompson sample.

Table 1. Characteristics of Thompson Sampling, ε-greedy, and Boltzmann exploration what information they use for exploration

Strategy	Greedy-Action	Estimated Q-values	Estimated uncertainties
ε-greedy	✓	✗	✗
Boltzmann exploration	✓	✓	✗
Thompson Sampling	✓	✓	✓

properties.

ε-greedy is among the simplest exploration-exploitation strategies and it is uniformly random over all the non-greedy actions. Boltzmann exploration is an intermediate strategy since it uses the estimated Q function to sample from action space. However, it does not maintain uncertainties over the Q function estimation. In contrast, Thompson sampling incorporates the Q estimate as well as the uncertainties in the estimation and utilizes the most information for exploration-exploitation strategy.

Consider the example in Figure 1(a) with our current estimates and uncertainties of the Q function over different actions. ε-greedy is not compelling since it assigns uniform probability to explore over 5 and 6, which are sub-optimal when the uncertainty estimates are available. In this setting, a possible remedy is Boltzmann exploration since it assigns lower probability to actions 5 and 6 but randomizes with almost the same probabilities over the remaining actions.

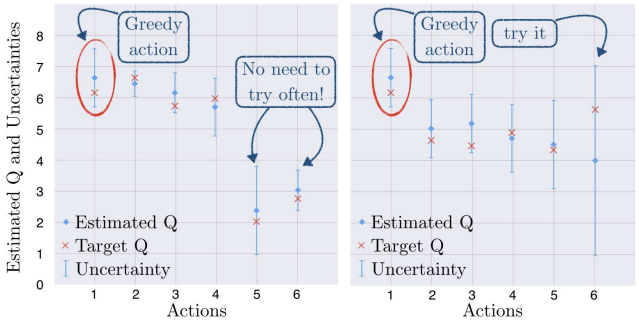

Figure 1. Thompson Sampling vs ε-greedy and Boltzmann exploration. (a) ε-greedy is wasteful since it assigns uniform probability to explore over 5 and 6, which are obviously sub-optimal when the uncertainty estimates are available. Boltzmann exploration randomizes over actions even if the optimal action is identifies. (b) Boltzmann exploration does not incorporate uncertainties over the estimated action-values and chooses actions 5 and 6 with similar probabilities while action 6 is significantly more uncertain. Thomson Sampling is a simple remedy to all these issues.

However, Boltzmann exploration is sub-optimal in settings where there is high uncertainty. For example if the current Q estimate is according to Figure 1(b), then Boltzmann exploration assigns almost equal probability to actions 5 and 6, even though action 6 has much higher uncertainty and needs to be explored more.

Thus, both ε-greedy and Boltzmann exploration strategies are sub-optimal since they do not maintain an uncertainty estimate over the Q estimation. In contrast, Thompson sampling uses both estimated Q function and its uncertainty estimates to carry out a more efficient exploration.

3. Bayesian Deep Q-Networks

Consider an MDP M as a tuple $\langle \mathcal{X}, \mathcal{A}, P, P_0, R, \gamma \rangle$, with state space $\mathcal{X}$, action space $\mathcal{A}$, transition kernel P, initial state distribution P_0, accompanied with reward function of R, and discount factor $0 \leq \gamma < 1$. Since it is mainly clear from the context, to ease to notation, we mainly use the same notation for random variables and their realizations. In value based model free RL, the core of most prominent approaches is to learn the Q-function through minimizing the Bellman residual (Schweitzer & Seidmann, 1985; Lagoudakis & Parr, 2003; Antos et al., 2008) and temporal difference (TD) update (Tesauro, 1995). Mnih et al. (2015) carries the same idea, and propose DQN where the Q-function is parameterized by a deep network. In order to reduce the bias of the estimator, DQN utilizes a target network Q^{target}, target value $y = r + \gamma Q^{target}(x', \hat{a})$, where the tuple (x, a, r, x') consists of a consecutive experiences, $\hat{a} = \arg\max_{a'} Q^{target}(x', a')$ and approaches the regression in the empirical estimates of the loss $\mathcal{L}(Q, Q^{target})$;

$$\mathcal{L}(Q, Q^{target}) = \mathbb{E}_{\pi}\left[(Q(x,a) - y)^2\right] \qquad (1)$$

A DQN agent, once in a while updates the Q^{target} network and sets it to the Q network, follows the regression in Eq.1 with the new target value and provides a biased estimator of the Q-value. To mitigate the bias in this estimator, Van Hasselt et al. (2016) proposes DDQN and instead use $\hat{a} = \arg\max_{a'} Q(x', a')$. We deploy this approach for the rest of this paper.

DQN architecture consists of a deep neural network where the Q-function is approximated as a linear function of the feature representation layer $\phi_\theta(x) \in \mathbb{R}^d$ parameterized by θ, i.e., for any pair of state-action (x, a), we have $Q(x, a) = \phi_\theta(x)^\top w_a$ with $w_a \in \mathcal{R}^d$, the parameter of the output layer. Consequently, the target model has the same architecture as the Q, and consists of $\phi_{\theta^{target}}(\cdot) \in \mathbb{R}^d$, the feature representation of the target

network, and ${w^{target}}_a$, $\forall a \in \mathcal{A}$ the target weight. Similar to DDQN, given a tuple of experience (x, a, r, x'), and $\hat{a} = \arg\max_{a'} {\phi_\theta}^\top w_{a'}$

$$Q(x,a) = \phi_\theta(x)^\top w_a \rightarrow y := r + \gamma {\phi_{\theta^{target}}(x')}^\top {w^{target}}_{\hat{a}}$$

The regression in Eq. 1 induces a linear regression in the learning of the output layer, i.e., w_a's. In this work, we utilize the DQN architecture and instead propose to use BLR (Rasmussen & Williams, 2006) in learning of the output layer. Through BLR, we efficiently approximate the distribution over the Q-values, capture the uncertainty over the Q-function estimation, and design a efficient exploration and exploitation strategy using Thompson Sampling.

By deploying BLR on the feature representation layer, we approximate the posterior distribution of each w_a, resulting in the posterior distribution of the Q-function. As in BLR methods, we maintain a Gaussian prior $\mathcal{N}(0, \sigma^2 I)$ with the target value $y \sim w_a^\top \phi_\theta(x) + \epsilon$ for each weight vector where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ is an i.i.d. Gaussian noise. Given a experience replay buffer $\mathcal{D} = \{x_\tau, a_\tau, y_\tau\}_{\tau=1}^D$, we construct $|\mathcal{A}|$ (number of actions) disjoint datasets $\mathcal{D}_a$ for each action with $a_\tau = a$. For each action a, we construct a matrix $\Phi_a^\theta \in \mathbb{R}^{d \times |\mathcal{D}_a|}$, the concatenation of feature vectors $\{\phi_\theta(x_i)\}_{i=1}^{|\mathcal{D}_a|}$, and $\mathbf{y}_a \in \mathbb{R}^{|\mathcal{D}_a|}$, the concatenation of target values in set $\mathcal{D}_a$. Finally, we approximate the posterior distribution of w_a as follows:

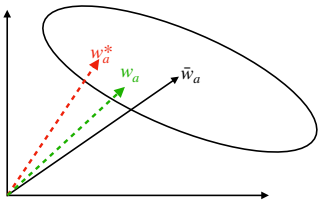

Figure 2. BDQN deploys Thompson Sampling to, sample w_a $\forall a \in A$ around the empirical mean $\overline{w}_a$ with w_a^* the underlying parameter of interest.

$$w_a \sim \mathcal{N}(\overline{w}_a, Cov_a), \quad \overline{w}_a := \frac{1}{\sigma_\epsilon^2} Cov_a \Phi_a^\theta \mathbf{y}_a,$$
$$Cov_a := \left(\frac{1}{\sigma_\epsilon^2}\Phi_a^\theta {\Phi_a^\theta}^\top + \frac{1}{\sigma^2} I\right)^{-1} \quad (2)$$

Fig. 2 expresses that the covariance matrix induces an ellipsoid around the estimated mean of the approximated posterior and samples drawn through Thompson Sampling are mainly close to this mean. A sample of $Q(x,a)$ is $w_a^\top \phi_\theta(x)$ where w_a is drawn from the posterior distribution Fig. 2. We require to sample a new Q function every $\mathcal{O}(episode\ length)$ time steps, theoretically proven in the next section, to make the algorithm explore efficiently. In BDQN, every T^{sample} times step, we draw a new w_a, $\forall a \in \mathcal{A}$ and follow the resulting policy, i.e., $a_{\text{TS}} := \max_a w_a^\top \phi_\theta(x)$. T^{sample} is chosen to be $\mathcal{O}(episode\ length)$ for all the Atari games, Appendix A.5. We simultaneously train the feature network under the loss $(y_\tau - w_{a_\tau}^\top \phi_\theta(x_\tau))^2$ with x_τ, a_τ, y_τ experiences from the replay buffer i.e.

$$\theta \leftarrow \theta - \alpha \cdot \nabla_\theta (y_\tau - w_{a_\tau}^\top \phi_\theta(x_\tau))^2 \quad (3)$$

We update the target network every T^{target} steps and set θ^{target} to θ. With the period of $T^{Bayes\ target}$, we update the posterior distribution using a minibatch of B randomly chosen experiences in the replay buffer, and set the $w_a^{target} = \overline{w}_a$, $\forall a \in \mathcal{A}$ which is the mean of the posterior distribution. We describe BDQN algorithm in Alg. 1 (more details in Section A.5).

noend 1 BDQN

Initialize θ, θ^{target}, w_a, w_a^{target}, Cov_a $\forall a$
Set the replay buffer $RB = \{\}$
for t = 1,2,3... **do**
  if t mod $T^{Bayes\ target} = 0$ **then**
    Update w_a^{target} and Cov_a, $\forall a$ using B samples
  if t mod $T^{sample} = 0$ **then**
    Draw $w_a \sim \mathcal{N}(w_a^{target}, Cov_a)$ $\forall a$
  Set $\theta^{target} \leftarrow \theta$ every T^{target}
  Execute $a_t = \arg\max_{a'} w_{a'}^\top \phi_\theta(x_t)$
  Store (x_t, a_t, r_t, x_{t+1}) in the RB
  Sample a minibatch $(x_\tau, a_\tau, r_\tau, x_{\tau+1})$ from the RB
  if $x_{\tau+1}$ is a terminal state **then**
    $y_\tau \leftarrow r_\tau$
  else
    $\hat{a} := \arg\max_{a'} w_{a'}^\top \phi_\theta(x_{\tau+1})$
    $y_\tau \leftarrow r_\tau + \gamma {w_{\hat{a}}^{target}}^\top \phi_{\theta^{target}}(x_{\tau+1})$
  Update $\theta \leftarrow \theta - \alpha \cdot \nabla_\theta (y_\tau - w_{a_\tau}^\top \phi_\theta(x_\tau))^2$

4. Regret Upper Bound

In this section we provide the analysis of Bayesian regret upper bound of PSRL Alg. 2 and frequentist regret upper bound of optimism Alg. 3 when the feature representation is given and fixed. Consider a finite horizon MDP $M := \langle \mathcal{X}, \mathcal{A}, P, P_0, R, \gamma, H \rangle$, with horizon length H and $0 \leq \gamma \leq 1$. In order to keep the notation simple, $\mathcal{X}$ and $\mathcal{A}$ also denote $\mathcal{X}^h$ and $\mathcal{A}^h$ for all h unless specified. In the following, $\|\cdot\|_2$ denotes the spectral norm and for any positive definite matrix χ, $\|\cdot\|_\chi$ denotes the χ matrix-weighted spectral norm. For any natural number H, $[H] = \{1, 2, \ldots, H\}$. We consider the class of MDPs where the optimal Q-function at each time step h is a linear transformation of $\phi(\cdot,\cdot) := \mathcal{X}^h \times \mathcal{A}^h \rightarrow \mathcal{R}^d$, i.e., $Q_{\pi^*}^{\omega^*}(x^h, a^h) := \phi(x^h, a^h)^\top {\omega^*}^h$, $\forall x^h, a^h \in \mathcal{X} \times \mathcal{A}$. Here ω^* and π^* denote a set of ${\omega^*}^h \in \mathbb{R}^d$ and ${\pi^*}^h : \mathcal{X} \rightarrow \mathcal{A}$

noend 2 PSRL

1: Input: the prior and likelihood
2: **for** episode: t= 1,2,... **do**
3: $\omega_t^h \sim$ posterior distribution, $\forall h \in [H]$
4: **for** $h = 0$ to the end of episode **do**
5: Follow π_t induced by ω_t^h
6: Update the posterior

as $\pi^*(x) := \arg\max_{a\in\mathcal{A}} Q_{\pi^{*h}}^{\omega^{*h}}(x^h, a^h)$. Let $V_{\pi^*}^{\omega^*}$ denote the corresponding value function. For MDPs, condition on x^h, a^h the distribution of

$$R^h + \gamma\phi(x^{h+1}, a^{h+1})^\top \omega^{*h+1}$$

can be written as

$$\phi(x^h, a^h)^\top \omega^{*h} + \nu^h$$

where ν^h is a mean zero random variable and R^h is the reward at time step h. Alg. 2 maintains a prior over the vectors $\omega^{*h}, \forall h$ and updates the posterior over time. At the beginning of an episode t, the agent draws $\omega_t^h, \forall h$, from the posterior, and follows their induced policy π_t^h, i.e., $a_t^h := \arg\max_{a\in\mathcal{A}} \phi^\top(x^h, a)\omega_t^h, \forall x^h \in \mathcal{X}$. Alg. 3, at the beginning of t'th episode, exploits the so-far collected samples and estimates ω^{*h} up to a high probability confidence intervals $\mathcal{C}_{t-1}^h$ i.e., $\omega^{*h} \in C_{t-1}^h$ with high probability $\forall h$. At each time step h, given a state x_t^h, the agent follows the optimistic policy; $\widetilde{\pi}_t^h(x_t^h) = \arg\max_{a\in\mathcal{A}} \max_{\omega\in\mathcal{C}_{t-1}^h} \phi^\top(X_t^h, a)\omega$. Through exploration and exploitation, we show that the confidence sets $\mathcal{C}_t^h, \forall h$, shrink with the rate of $\widetilde{\mathcal{O}}\left(1/\sqrt{t}\right)$ resulting in less and less per step regret (Lemma 1 in Appendix B). Define the following regression matrices

$$\chi_t^h := \sum_{i=1}^t \phi_i^h {\phi_i^h}^\top, \quad \overline{\chi}_t^h = \chi_t^h + \widetilde{\chi}^h$$

where $\widetilde{\chi}^h \in \mathbb{R}^{d\times d}$ is a ridge regularization matrix and usually is equal to λI. Similar to the linear bandit (Abbasi-Yadkori et al., 2011), consider the following generic assumptions,

- The noise model $\nu^h, \forall h$ induces a σ sub-Gaussian vector. (Assumption 1 in Appendix B)
- $\|\omega^{*h}\|_2 \leq L_\omega$, $\|\phi(x^h, a^h)\phi(x^h, a^h)^\top\|_2 \leq L$, $\forall x \in \mathcal{X}, a \in \mathcal{A}, \forall h$, a.s.
- Expected rewards and returns are in $[0, 1]$.

Furthermore, similar to ridge linear regression assumption for stochastic settings in Hsu et al. (2012), there exist finite

noend 3 OFU

1: Input: σ, λ and δ
2: **for** episode: t = 1,2,... **do**
3: **for** $h = 1$ to the end of episode **do**
4: choose optimistic $\widetilde{\omega}_t^h$ in $\mathcal{C}_{t-1}^h(\delta)$
5: Follow $\widetilde{\pi}_t^h$ induced by $\widetilde{\omega}_t^h$
6: Compute the confidence $\mathcal{C}_t^h(\delta)$, $\forall h \in [H]$

values of ρ_λ^h, $\forall h$ such that;

$$\sum_i^t \|\phi(x_t^h, \pi^*(x_t^h))\|^2_{\overline{\chi}_t^{h\,-1}} \leq \rho_\lambda^h, \ \forall h, t, \textit{with } \rho_\lambda^{H+1} = 0$$

Let $\overline{\rho}_\lambda^H(\gamma)$ denote the following combination of ρ_λ^h;

$$\overline{\rho}_\lambda^H(\gamma) := \sum_{i=1}^H (\gamma)^{H-i} \left(\frac{1}{H} + \frac{1}{H}\sum_{j=1}^i \prod_{k=1}^j (\gamma)^j \rho_\lambda^{H-(i-k)+1}\right)$$

For any prior and likelihood satisfying these assumptions, we have;

Theorem 1 (Bayesian Regret)**.** *For an episodic MDP with episode length H, discount factor γ, and feature map $\phi(x, a) \in \mathbb{R}^d$, after T episodes the posterior sampling on ω, Alg. 2, guarantees;*

$$\textbf{\textit{BayesReg}}_T := \mathbb{E}\left[\sum_t^T \left[V_{\pi^*}^{\omega^*} - V_{\widetilde{\pi}_t}^{\omega^*}\right]\right] = \mathcal{O}\left(d\sqrt{\overline{\rho}_\lambda^H(\gamma) HT}\log(T)\right)$$

Proof is given in the Appendix B.2.

Theorem 2 (Frequentist Regret)**.** *For an episodic MDP with episode length H, discount factor γ, feature map $\phi(x, a) \in \mathbb{R}^d$, the optimism on ω, Alg. 3, after T episodes, guarantees;*

$$\textbf{\textit{Reg}}_T := \mathbb{E}\left[\sum_t^T \left[V_{\pi^*}^{\omega^*} - V_{\widetilde{\pi}_t}^{\omega^*}\right] \middle| \omega^*\right] = \mathcal{O}\left(d\sqrt{\overline{\rho}_\lambda^H(\gamma) HT}\log(T)\right)$$

Proof is given in the Appendix B.1. These bounds are similar to those in linear bandits (Abbasi-Yadkori et al., 2011; Russo & Van Roy, 2014a) and linear quadratic control (Abbasi-Yadkori & Szepesvári, 2011), i.e. $\widetilde{\mathcal{O}}(d\sqrt{T})$. Since for $H = 1$, this problem reduces to linear bandit and for linear bandit the lower bound is $\Omega(d\sqrt{T})$ therefore, our bound is order-optimal in d and T for $H = 1$. To the best of our knowledge, there exists no lower bound known regarding $H \geq 1$ the optimality of these bounds is unknown. To derive these bounds, we insisted on deploying linear ridge regression to keep the analysis and the algorithm simple. This estimator results in a biased estimation of ω^{*h} for $h > 1$. In our analysis, we show that this bias vanishes with the desired rate, but results in the dependence in the $\overline{\rho}_\lambda^H(\gamma)$.

5. Experiments

We apply BDQN on a variety of Atari games in the Arcade Learning Environment (Bellemare et al., 2013) through OpenAI Gym[2] (Brockman et al., 2016). For the baseline, we evaluate BDQN on the measures of sample complexity and score against DDQN. All the implementations are programmed in MxNet framework (Chen et al., 2015) and are publicly available. The details on architecture, Appendix A.1, learning rate Appendix A.3, computation A.4 are also provided. In Appendix A.2 we describe how we spend less than two days on a single game and single machine for the hyper-parameter choices which is another evidence on the significance of BDQN.

Baselines: We implemented DDQN and BDQN exactly the same way as described in Van Hasselt et al. (2016). We also attempted to implement a few other deep RL methods that employ strategic exploration, e.g., (Osband et al., 2016; Bellemare et al., 2016). Unfortunately we encountered several implementation challenges where neither code nor the implementation details was publicly available. Despite the motivation of this work on the sample complexity, since we do not have access to the performance plots of these methods, the least is to report their final scores. To try to illustrate the performance of our approach we instead, extracted the best reported scores from a number of state-of-the-art deep RL methods and include them in Table 2, which is the only way to bring a comparison. We compare against DDQN, as well as DDQN$^+$ which is the reported scores of DDQN in Van Hasselt et al. (2016) at evaluation time where the $\varepsilon = 0.001$. Furthermore, we compared against scores of Bootstrap DQN (Osband et al., 2016), NoisyNet (Fortunato et al., 2017), CTS, Pixel, Reactor (Ostrovski et al., 2017) which are borrowed from the original papers. For NoisyNet, the scores of NoisyDQN are reported. We also provided the sample complexity, *SC*: the number of interactions BDQN requires to beat the human score (Mnih et al., 2015)(" $-$ " means BDQN could not beat human score) and *SC*$^+$: the number of interactions the BDQN requires to beat the score of DDQN$^+$. Note that these are not perfect comparisons, as there are additional details that are not included in the mentioned papers, i.e. it is hard to just compare the reported results (an issue that has been discussed extensively recently, e.g. (Henderson et al., 2017)).[3]. Moreover, when the regret analysis of an algorithm is considered, no evaluation phase required, and the reported results of BDQN are those while exploring. It is worth noting that, the scores during evaluation are much higher than those during the exploration and exploitation period, Appendix A.8. Furthermore, we also implemented DDQN drop-out as another proposed exploration-exploitation algorithm by Gal & Ghahramani (2016). Osband et al. (2016) investigates the sufficiency of the estimated uncertainty and hardness in driving suitable exploitation out of it. It has been observed that drop-out results in the ensemble of infinitely many models but all models almost the same (Dhillon et al., 2018; Osband et al., 2016). Consequently, it is not capable of capturing the statistical uncertainty in the Q function and falls short in outperforming the uniformly at random policy, Appendix A.6.

Results: The results are provided in Fig. 3. We observe that BDQN significantly improves the sample complexity of DDQN and reaches the highest recorded score of DDQN in a much fewer number of interactions than DDQN requires. We expected BDQN, due to its better exploration-exploitation strategy, to improve the regret and enhance the sample complexity, but we also observed a significant improvement in scores. It is worth noting that since BDQN is designed to minimize the regret, and also since the study in Fig. 3 are designed for sample complexity analysis, either of the reported BDQN and DDQN scores is while exploring. For example, in the game Pong, DDQN gives a score of 18.82 during the learning phase but setting ε to a quantity close to zero; it mostly gives the score of 21. In addition to the Table 2, we also provided the score ratio as well as the area under the performance plot ratio comparisons in Table 3.

For the game *Atlantis*, DDQN$^+$ gives score of $64.67k$ during the evaluation phase, while BDQN reaches score of $3.24M$ after $20M$ interactions. As it is been shown in Fig. 3, BDQN saturates for *Atlantis* after $20M$ interactions. We realized that BDQN reaches the internal *OpenAIGym* limit of $max\_episode$, where relaxing it improves score after $15M$ steps to $62M$, Appendix A.7. We observe that BDQN learns significantly better policies due to its efficient explore/exploit in a much shorter period of time. Since BDQN on game *Atlantis* promise a big jump around time step $20M$, we ran it five more times in order to make sure it was not just a coincidence Appendix A.7 Fig. 7. For the game Pong, we ran the experiment for a longer period but just plotted the beginning of it in order to observe the difference. Due to cost of deep RL methods, for some games, we run the experiment until a plateau is reached.

6. Related Work

The complexity of the exploration-exploitation trade-off has been deeply investigated in RL literature for both continuous and discrete MDPs (Kearns & Singh, 2002; Brafman & Tennenholtz, 2003; Asmuth et al., 2009;

[2]Environment details in the implementation code.

[3]We released our code with an extensive explanation. We also implemented bootstrapped DQN (Osband et al., 2016) and released the code but we were not able to reproduce their results beyond the performance of random policy

Table 2. Comparison of scores and sample complexities (scores in the first two columns are average of 100 consecutive episodes). The scores of DDQN$^+$ are the reported scores of DDQN in Van Hasselt et al. (2016) after running it for 200M interactions at evaluation time where the $\varepsilon = 0.001$. Bootstrap DQN (Osband et al., 2016), CTS, Pixel, Reactor (Ostrovski et al., 2017) are borrowed from the original papers. For NoisyNet (Fortunato et al., 2017), the scores of NoisyDQN are reported. Sample complexity, *SC*: the number of samples the BDQN requires to beat the human score (Mnih et al., 2015)(" $-$ " means BDQN could not beat human score). SC^+: the number of interactions the BDQN requires to beat the score of DDQN$^+$.

Game	BDQN	DDQN	DDQN$^+$	Bootstrap	NoisyNet	CTS	Pixel	Reactor	Human	SC	SC^+	Step
Amidar	**5.52k**	0.99k	0.7k	1.27k	1.5k	1.03k	0.62k	1.18k	1.7k	22.9M	4.4M	100M
Alien	3k	2.9k	2.9k	2.44k	2.9k	1.9k	1.7k	**3.5k**	6.9k	-	36.27M	100M
Assault	**8.84k**	2.23k	5.02k	8.05k	3.1k	2.88k	1.25k	3.5k	1.5k	1.6M	24.3M	100M
Asteroids	**14.1k**	0.56k	0.93k	1.03k	2.1k	3.95k	0.9k	1.75k	13.1k	58.2M	9.7M	100M
Asterix	**58.4k**	11k	15.15k	19.7k	11.0	9.55k	1.4k	6.2k	8.5k	3.6M	5.7M	100M
BeamRider	8.7k	4.2k	7.6k	**23.4k**	14.7k	7.0k	3k	3.8k	5.8k	4.0M	8.1M	70M
BattleZone	**65.2k**	23.2k	24.7k	36.7k	11.9k	7.97k	10k	45k	38k	25.1M	14.9M	50M
Atlantis	3.24M	39.7k	64.76k	99.4k	7.9k	1.8M	40k	**9.5M**	29k	3.3M	5.1M	40M
DemonAttack	11.1k	3.8k	9.7k	82.6k	26.7k	**39.3k**	1.3k	7k	3.4k	2.0M	19.9M	40M
Centipede	**7.3k**	6.4k	4.1k	4.55k	3.35k	5.4k	1.8k	3.5k	12k	-	4.2M	40M
BankHeist	0.72k	0.34k	0.72k	**1.21k**	0.64k	1.3k	0.42k	1.1k	0.72k	2.1M	10.1M	40M
CrazyClimber	124k	84k	102k	**138k**	121k	112.9k	75k	119k	35.4k	0.12M	2.1M	40M
ChopperCommand	**72.5k**	0.5k	4.6k	4.1k	5.3k	5.1k	2.5k	4.8k	9.9k	4.4M	2.2M	40M
Enduro	1.12k	0.38k	0.32k	1.59k	0.91k	0.69k	0.19k	**2.49k**	0.31k	0.82M	0.8M	30M
Pong	**21**	18.82	**21**	20.9	**21**	20.8	17	20	9.3	1.2M	2.4M	5M

Kakade et al., 2003; Ortner & Ryabko, 2012). Jaksch et al. (2010) investigates the regret analysis of MDPs with finite state and action where OFU principle is deployed to guarantee a regret upper bound, while Ortner & Ryabko (2012) relaxes it to a continuous state space and propose a sub-linear regret bound. Azizzadenesheli et al. (2016a) deploys OFU and propose a regret upper bound for Partially Observable MDPs (POMDPs) using spectral methods (Anandkumar et al., 2014). Furthermore, Bartók et al. (2014) tackles a general case of partial monitoring games and provides minimax regret guarantee. For linear quadratic models OFU is deployed to provide an optimal regret bound (Abbasi-Yadkori & Szepesvári, 2011).

In multi-arm bandit, there are compelling empirical pieces of evidence that Thompson Sampling sometimes provides better results than optimism-under-uncertainty approaches (Chapelle & Li, 2011), while also the performance guarantees are preserved (Agrawal & Goyal, 2012; Russo & Van Roy, 2014a). A natural adaptation of this algorithm to RL, posterior sampling RL (PSRL) Strens (2000) also shown to have good frequentist and Bayesian performance guarantees (Osband et al., 2013; Abbasi-Yadkori & Szepesvári, 2015).

Even though the theoretical RL addresses the exploration and exploitation trade-offs, these problems are still prominent in empirical reinforcement learning research (Mnih et al., 2015; Abel et al., 2016; Azizzadenesheli et al., 2016b). On the empirical side, the recent success in the video games has sparked a flurry of research interest. Following the success of Deep RL on Atari games (Mnih et al., 2015) and the board game Go (Silver et al., 2017), many researchers have begun exploring practical applications of deep reinforcement learning (DRL). Some investigated applications include, robotics (Levine et al., 2016), self-driving cars (Shalev-Shwartz et al., 2016), and safety (Lipton et al., 2016a). Inevitably for PSRL, the act of posterior sampling for policy or value is computationally intractable in large systems, so PSRL can not be easily leveraged to high dimensional problems (Ghavamzadeh et al., 2015; Engel et al., 2003; Dearden et al., 1998; Tziortziotis et al., 2013). To remedy these failings Osband et al. (2017) consider the use of randomized value functions. For finite state-action space MDP, (Osband et al., 2014) propose posterior sampling directly on the space of Q-functions and provide a Bayesian regret bound guarantee for finite state action MDPs. To approximate the posterior, they use BLR on one-hot encoding of state-action, also applicable to high dimension, and improve the computation complexity of PSRL. BDQN is strongly related and similar to this work, and is a generalization to continues state-action space MDPs.

To combat the computational and scalability shortcomings, Osband et al. (2016) suggests a bootstrapped-ensemble approach that trains several models in parallel to approximate the posterior distribution. Other works suggest using a variational approximation to the Q-networks (Lipton et al., 2016b) or a concurrent work on noisy network (Fortunato et al., 2017). However, most of these approaches significantly increase the computational cost of DQN and neither approach produced much beyond modest gains on Atari games. Interestingly, the Bayesian approach as a technique for learning a neural network has

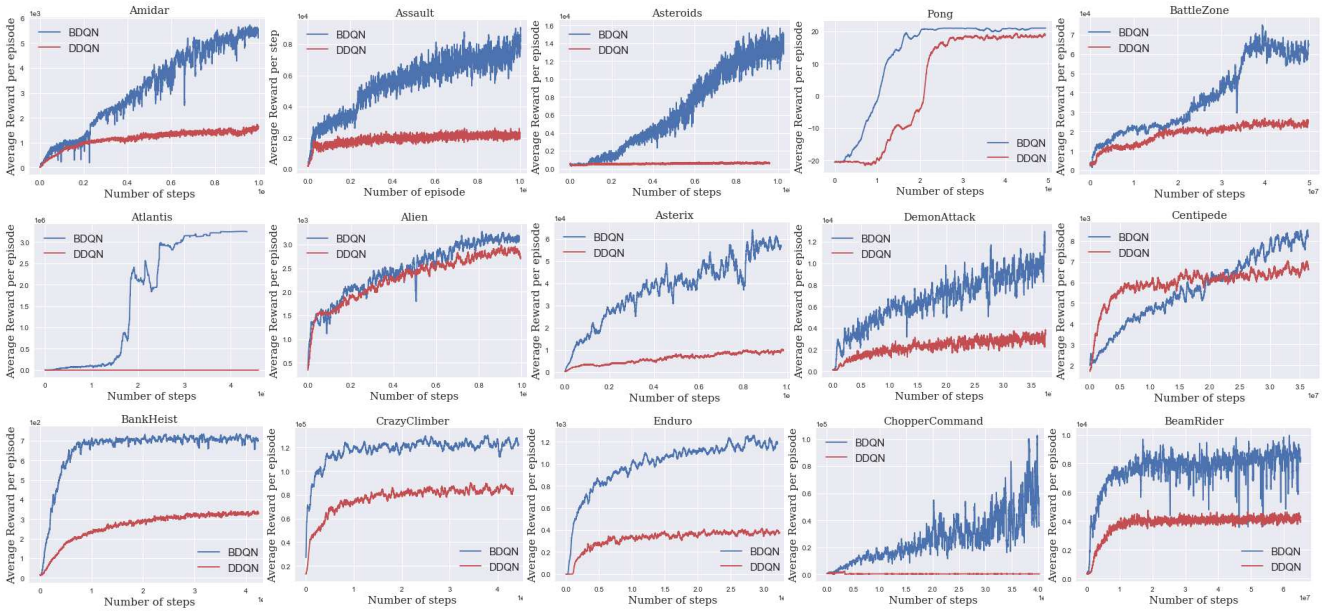

Figure 3. The comparison between DDQN and BDQN

been deployed for object recognition and image caption generation where its significant advantage has been verified Snoek et al. (2015).

Concurrently, Levine et al. (2017) proposes least squares temporal difference which learns a linear model on the feature representation in order to estimate the Q-function while ε-greedy exploration is employed and improvement on 5 tested Atari games is provided. Out of these 5 games, one is common with our set of 15 games which BDQN outperform it by factor of 360% (w.r.t. the score reported in their paper). As motivated by theoretical understanding, our empirical study shows that performing Bayesian regression instead, and sampling from the result, can yield a substantial benefit, indicating that it is not just the higher data efficiency at the last layer, but that leveraging an explicit uncertainty representation over the value function is of substantial benefit. As stated before, in spite of the novelties proposed by the methods, mentioned in this section, neither of them, including TS based approaches, produced much beyond modest gains on Atari games while BDQN provides significant improvements in terms of both sample complexity and final performance.

7. Conclusion

In this work, we proposed BDQN, a practical Thompson sampling based RL algorithm which provides efficient exploration/exploitation in a computationally efficient manner. It involved making simple modifications to the DDQN architecture by replacing the linear regression learning of the last layer with Bayesian linear regression. Under a Gaussian prior, we obtained fast closed-form updates for the approximated posterior distribution. We demonstrated significantly faster training and much better performance in many games compared to the reported results in a vast number of state-of-the-art baselines. We also established theoretical guarantees for episodic MDPs with continuous state and action spaces in the case where the feature representation is fixed. We derived an order-optimal frequentist and Bayesian regret bound of $\widetilde{\mathcal{O}}(d\sqrt{N})$ after N time steps.

In the future, we plan to extend the current analysis and provide a frequent regret bound for Thompson sampling where instead of sampling from the posterior, we sample from a Gaussian approximation of the posterior (Abeille & Lazaric, 2017). We also aim to extend this analysis to the general class of functions and move beyond linear models. For the general class of functions, optimism in the face of uncertainty has been deployed to guarantee a tight probably approximately correct (PAC) bound (Jiang et al., 2016) in the finite action settings but the proposed algorithm requires solving NP-hard internal optimization problems. We intend to extend the current study of Thompson study algorithms to the general class of function and provide computationally feasible algorithms with efficient sample complexity guarantees. It is worth noting that in the current analysis of our work, we insisted on keeping the algorithm as simple as linear regression which resulted in the bias estimation of the model parameters. Although we showed that the bias terms vanish with the required rates, the analysis in (Antos et al., 2008) proposes a more sophisticated parameter estimation approach via a min-max alternative which results in an unbiased estimation of the model parameters. In the future works, we plan to deploy this approach and provide even more efficient

learning algorithms with better constants, and the horizon dependency in the sample complexity bounds.

Acknowledgments

The authors would like to thank Zachary C. Lipton, Marlos C. Machado, Ian Osband, Gergely Neu, Kristy Choi and, particularly Akshay Krishnamurthy during ICLR2019 openreview, for their feedbacks, suggestions, and helps. K. Azizzadenesheli is supported in part by NSF Career Award CCF-1254106 and AFOSR YIP FA9550-15-1-0221. This research has been conducted when the first author was a visiting researcher at Stanford University and Caltech. A. Anandkumar is supported in part by Microsoft Faculty Fellowship, Google faculty award, Adobe grant, NSF Career Award CCF-1254106, and AFOSR YIP FA9550-15-1-0221. All the experimental study have been done using Caltech AWS credits grant.

References

Abbasi-Yadkori, Y. and Szepesvári, C. Regret bounds for the adaptive control of linear quadratic systems. In *COLT 2011 - The 24th Annual Conference on Learning Theory, June 9-11, 2011, Budapest, Hungary*, 2011.

Abbasi-Yadkori, Y. and Szepesvári, C. Bayesian optimal control of smoothly parameterized systems. In *UAI*, pp. 1–11, 2015.

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24 - NIPS*, pp. 2312–2320, 2011.

Abeille, M. and Lazaric, A. Linear thompson sampling revisited. In *AISTATS 2017-20th International Conference on Artificial Intelligence and Statistics*, 2017.

Abel, D., Agarwal, A., Diaz, F., Krishnamurthy, A., and Schapire, R. E. Exploratory gradient boosting for reinforcement learning in complex domains. *arXiv*, 2016.

Agrawal, S. and Goyal, N. Analysis of thompson sampling for the multi-armed bandit problem. In *COLT*, 2012.

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

Antos, A., Szepesvári, C., and Munos, R. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 2008.

Asmuth, J., Li, L., Littman, M. L., Nouri, A., and Wingate, D. A bayesian sampling approach to exploration in reinforcement learning. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009.

Azizzadenesheli, K., Lazaric, A., and Anandkumar, A. Reinforcement learning of pomdps using spectral methods. In *Proceedings of the 29th Annual Conference on Learning Theory (COLT)*, 2016a.

Azizzadenesheli, K., Lazaric, A., and Anandkumar, A. Reinforcement learning in rich-observation mdps using spectral methods. *arXiv preprint arXiv:1611.03907*, 2016b.

Bartók, G., Foster, D. P., Pál, D., Rakhlin, A., and Szepesvári, C. Partial monitoringclassification, regret bounds, and algorithms. *Mathematics of Operations Research*, 2014.

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

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *J. Artif. Intell. Res.(JAIR)*, 2013.

Brafman, R. I. and Tennenholtz, M. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research*, 3:213–231, 2003.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016.

Chapelle, O. and Li, L. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pp. 2249–2257, 2011.

Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C., and Zhang, Z. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv*, 2015.

Chu, W., Li, L., Reyzin, L., and Schapire, R. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 208–214, 2011.

Dani, V., Hayes, T. P., and Kakade, S. M. Stochastic linear optimization under bandit feedback. 2008.

de la Pena, V. H., Klass, M. J., and Lai, T. L. Self-normalized processes: exponential inequalities, moment

bounds and iterated logarithm laws. *Annals of probability*, pp. 1902–1933, 2004.

Dearden, R., Friedman, N., and Russell, S. Bayesian q-learning. In *AAAI/IAAI*, pp. 761–768, 1998.

Dhillon, G. S., Azizzadenesheli, K., Lipton, Z. C., Bernstein, J., Kossaifi, J., Khanna, A., and Anandkumar, A. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

Engel, Y., Mannor, S., and Meir, R. Bayes meets bellman: The gaussian process approach to temporal difference learning. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, 2003.

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

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

Garcıa, J. and Fernández, F. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Ghavamzadeh, M., Mannor, S., Pineau, J., Tamar, A., et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 2015.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. Deep reinforcement learning that matters. *arXiv*, 2017.

Hsu, D., Kakade, S. M., and Zhang, T. Random design analysis of ridge regression. In *Conference on Learning Theory*, pp. 9–1, 2012.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 2010.

Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. Contextual decision processes with low bellman rank are pac-learnable. *arXiv*, 2016.

Kakade, S., Kearns, M. J., and Langford, J. Exploration in metric state spaces. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 306–312, 2003.

Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

Lagoudakis, M. G. and Parr, R. Least-squares policy iteration. *Journal of machine learning research*, 4(Dec): 1107–1149, 2003.

Levine, N., Zahavy, T., Mankowitz, D. J., Tamar, A., and Mannor, S. Shallow updates for deep reinforcement learning. *arXiv*, 2017.

Levine et al., S. End-to-end training of deep visuomotor policies. *JMLR*, 2016.

Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670. ACM, 2010.

Lipton, Z. C., Azizzadenesheli, K., Kumar, A., Li, L., Gao, J., and Deng, L. Combating reinforcement learning's sisyphean curse with intrinsic fear. *arXiv preprint arXiv:1611.01211*, 2016a.

Lipton, Z. C., Gao, J., Li, L., Li, X., Ahmed, F., and Deng, L. Efficient exploration for dialogue policy learning with bbq networks & replay buffer spiking. *arXiv preprint arXiv:1608.05081*, 2016b.

Machado, M. C., Bellemare, M. G., Talvitie, E., Veness, J., Hausknecht, M., and Bowling, M. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *arXiv preprint arXiv:1709.06009*, 2017.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

Ortner, R. and Ryabko, D. Online regret bounds for undiscounted continuous reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 1763–1771, 2012.

Osband, I. and Van Roy, B. Model-based reinforcement learning and the eluder dimension. In *Advances in Neural Information Processing Systems*, pp. 1466–1474, 2014a.

Osband, I. and Van Roy, B. Near-optimal reinforcement learning in factored mdps. In *Advances in Neural Information Processing Systems*, pp. 604–612, 2014b.

Osband, I., Russo, D., and Van Roy, B. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, 2013.

Osband, I., Van Roy, B., and Wen, Z. Generalization and exploration via randomized value functions. *arXiv*, 2014.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems*, 2016.

Osband, I., Russo, D., Wen, Z., and Van Roy, B. Deep exploration via randomized value functions. *arXiv*, 2017.

Ostrovski, G., Bellemare, M. G., Oord, A. v. d., and Munos, R. Count-based exploration with neural density models. *arXiv*, 2017.

Peña, V. H., Lai, T. L., and Shao, Q.-M. *Self-normalized processes: Limit theory and Statistical Applications*. Springer Science & Business Media, 2009.

Rasmussen, C. E. and Williams, C. K. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

Rusmevichientong, P. and Tsitsiklis, J. N. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

Russo, D. and Van Roy, B. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39 (4):1221–1243, 2014a.

Russo, D. and Van Roy, B. Learning to optimize via information-directed sampling. pp. 1583–1591, 2014b.

Schweitzer, P. J. and Seidmann, A. Generalized polynomial approximations in markovian decision processes. *Journal of mathematical analysis and applications*, 110(2): 568–582, 1985.

Shalev-Shwartz, S., Shammah, S., and Shashua, A. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv*, 2016.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *Nature*, 2017.

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M., and Adams, R. Scalable bayesian optimization using deep neural networks. In *ICML*, 2015.

Strens, M. A bayesian framework for reinforcement learning. In *ICML*, 2000.

Tesauro, G. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 1933.

Tziortziotis, N., Dimitrakakis, C., and Blekas, K. Linear bayesian reinforcement learning. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2013.

Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *AAAI*, 2016.

A. Appendix

In the main text, for simplicity, we use the terms "BLR" for i.i.d. samples and BLR for non i.i.d. samples exchangeable, even though, technically, the do not have equal meaning. In RL, the data is not i.i.d, and we extend the BLR to non i.i.d. setting by deploying additional Martingale type argument and handles the data with temporal dependency.

Table 3. $1st$ column: score ratio of BDQN to DDQN run for same number of time steps. $2nd$ column: score ratio of BDQN to DDQN$^+$. $3rd$ column: score ratio of BDQN to human scores reported at Mnih et al. (2015). $4th$ column: Area under the performance plot ration (AuPPr) of BDQN to DDQN. AuPPr is the integral of area under the performance plot ration. For Pong, since the scores start form -21, we shift it up by 21. $5th$ column: Sample complexity, *SC*: the number of samples the BDQN requires to beat the human score (Mnih et al., 2015)(" $-$ " means BDQN could not beat human score). $6th$ column: *SC*$^+$: the number of samples the BDQN requires to beat the score of DDQN$^+$. We run both BDQN and DDQN for the same number of times steps, stated in the last column.

Game	$\frac{\text{BDQN}}{\text{DDQN}}$	$\frac{\text{BDQN}}{\text{DDQN}^+}$	$\frac{\text{BDQN}}{\text{HUMAN}}$	AuPPr	SC	SC^+	Steps
Amidar	558%	788%	325%	280%	22.9M	4.4M	100M
Alien	103%	103%	43%	110%	-	36.27M	100M
Assault	396%	176%	589%	290%	1.6M	24.3M	100M
Asteroids	2517%	1516%	108%	680%	58.2M	9.7M	100M
Asterix	531%	385%	687%	590%	3.6M	5.7M	100M
BeamRider	207%	114%	150%	210%	4.0M	8.1M	70M
BattleZone	281%	253%	172%	180%	25.1M	14.9M	50M
Atlantis	80604%	49413%	11172%	380%	3.3M	5.1M	40M
DemonAttack	292%	114%	326%	310%	2.0M	19.9M	40M
Centipede	114%	178%	61%	105%	-	4.2M	40M
BankHeist	211%	100%	100%	250%	2.1M	10.1M	40M
CrazyClimber	148%	122%	350%	150%	0.12M	2.1M	40M
ChopperCommand	14500%	1576%	732%	270%	4.4M	2.2M	40M
Enduro	295%	350%	361%	300%	0.82M	0.8M	30M
Pong	112%	100%	226%	130%	1.2M	2.4M	5M

A.1. Network architecture:

The input to the network part of BDQN is $4 \times 84 \times 84$ tensor with a rescaled and averaged over channels of the last four observations. The first convolution layer has 32 filters of size 8 with a stride of 4. The second convolution layer has 64 filters of size 4 with stride 2. The last convolution layer has 64 filters of size 3 followed by a fully connected layer with size 512. We add a BLR layer on top of this.

A.2. Choice of hyper-parameters:

For BDQN, we set the values of W^{target} to the mean of the posterior distribution over the weights of BLR with covariances Cov and draw W from this posterior. For the fixed W and W^{target}, we randomly initialize the parameters of network part of BDQN, θ, and train it using RMSProp, with learning rate of 0.0025, and a momentum of 0.95, inspired by (Mnih et al., 2015) where the discount factor is $\gamma = 0.99$, the number of steps between target updates $T^{target} = 10k$ steps, and weights W are re-sampled from their posterior distribution every T^{sample} steps. We update the network part of BDQN every 4 steps by uniformly at random sampling a mini-batch of size 32 samples from the replay buffer. We update the posterior distribution of the weight set W every $T^{Bayes\ target}$ using mini-batch of size B (if the size of replay buffer is less than B at the current step, we choose the minimum of these two), with entries sampled uniformly form replay buffer. The experience replay contains the $1M$ most recent transitions. Further hyper-parameters are equivalent to ones in DQN setting.

For the BLR, we have noise variance σ_ϵ, variance of prior over weights σ, sample size B, posterior update period $T^{Bayes\ target}$, and the posterior sampling period T^{sample}. To optimize for this set of hyper-parameters we set up a very simple, fast, and cheap hyper-parameter tuning procedure which proves the robustness of BDQN. To find the first three, we set up a simple hyper-parameter search. We used a pretrained DQN model for the game of *Assault*, and removed the last fully connected layer in order to have access to its already trained feature representation. Then we tried combination of $B = \{T^{target}, 10 \cdot T^{target}\}$, $\sigma = \{1, 0.1, 0.001\}$, and $\sigma_\epsilon = \{1, 10\}$ and test for 1000 episode of the game. We set these parameters to their best $B = 10 \cdot T^{target}, \sigma = 0.001, \sigma = 1$.

The above hyper-parameter tuning is cheap and fast since it requires only a few times the B number of forwarding passes. For the remaining parameters, we ran BDQN (with weights randomly initialized) on the same game, *Assault*, for $5M$ time steps, with a set of $T^{Bayes\ target} = \{T^{target}, 10 \cdot T^{target}\}$ and $T^{sample} = \{\frac{T^{target}}{10}, \frac{T^{target}}{100}\}$, where BDQN performed better with choice of $T^{Bayes\ target} = 10 \cdot T^{target}$. For both choices of T^{sample}, it performs almost equal and we choose the higher one to reduce the computation cost. We started off with the learning rate of 0.0025 and did not tune for that. Thanks to the efficient Thompson sampling exploration and closed form BLR, BDQN can learn a better policy in an even shorter period of time. In contrast, it is well known for DQN based methods that changing the learning rate causes a major degradation in the performance (Fig. 4). The proposed hyper-parameter search is very simple and an exhaustive hyper-parameter search is likely to provide even better performance.

A.3. Learning rate:

It is well known that DQN and DDQN are sensitive to the learning rate and change of learning rate can degrade the performance to even worse than random policy. We tried the same learning rate as BDQN, 0.0025, for DDQN and observed that its performance drops. Fig. 4 shows that the DDQN with higher learning rates learns as good as BDQN at the very beginning but it can not maintain the rate of improvement and degrade even worse than the original DDQN with learning rate of 0.00025.

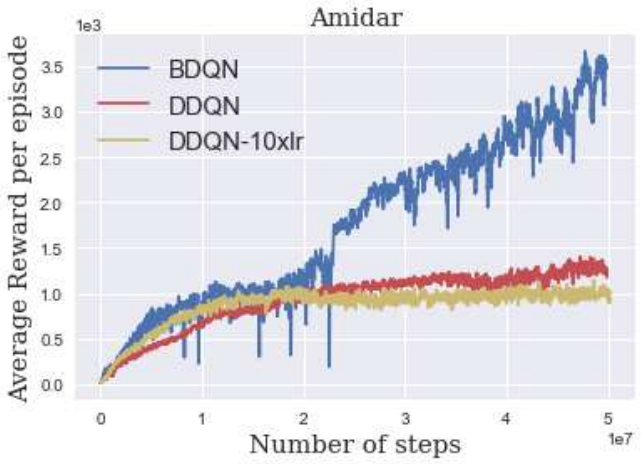

Figure 4. Effect of learning rate on DDQN

A.4. Computational and sample cost comparison:

For a given period of game time, the number of the backward pass in both BDQN and DQN are the same where for BDQN it is cheaper since it has one layer (the last layer) less than DQN. In the sense of fairness in sample usage, for example in duration of $10 \cdot T^{Bayes\ target} = 100k$, all the layers of both BDQN and DQN, except the last layer, sees the same number of samples, but the last layer of BDQN sees 16 times fewer samples compared to the last layer of DQN. The last layer of DQN for a duration of $100k$, observes $25k = 100k/4$ (4 is back prob period) mini batches of size 32, which is $16 \cdot 100k$, where the last layer of BDQN just observes samples size of $B = 100k$. As it is mentioned in Alg. 1, to update the posterior distribution, BDQN draws B samples from the replay buffer and needs to compute the feature vector of them. Therefore, during the $100k$ interactions for the learning procedure, DDQN does $32 * 25k$ of forward passes and $32 * 25k$ of backward passes, while BDQN does same number of backward passes (cheaper since there is no backward pass for the final layer) and $36 * 25k$ of forward passes. One can easily relax it by parallelizing this step along the main body of BDQN or deploying on-line posterior update methods.

A.5. Thompson sampling frequency:

The choice of Thompson sampling update frequency can be crucial from domain to domain. Theoretically, we show that for episodic learning, the choice of sampling at the beginning of each episode, or a bounded number of episodes is desired.

If one chooses T^{sample} too short, then computed gradient for backpropagation of the feature representation is not going to be useful since the gradient get noisier and the loss function is changing too frequently. On the other hand, the network tries to find a feature representation which is suitable for a wide range of different weights of the last layer, results in improper waste of model capacity. If the Thompson sampling update frequency is too low, then it is far from being Thompson sampling and losses the randomized exploration property. We are interested in a choice of T^{sample} which is in the order of upper bound on the average length of each episode of the Atari games. The current choice of T^{sample} is suitable for a variety of Atari games since the length of each episode is in range of $\mathcal{O}(T^{sample})$ and is infrequent enough to make the feature representation robust to big changes.

For the RL problems with shorter a horizon we suggest to introduce two more parameters, $\tilde{T}^{sample}$ and each $\tilde{w}_a$ where $\tilde{T}^{sample}$, the period that of each $\tilde{w}_a$ is sampled out of posterior, is much smaller than T^{sample} and $\tilde{w}_a, \forall a$ are used for Thompson sampling where w_a , $\forall a$ are used for backpropagation of feature representation. For game Assault, we tried using $\tilde{T}^{sample}$ and each $\tilde{w}_a$ but did not observe much a difference, and set them to T^{sample} and each w_a. But for RL setting with a shorter horizon, we suggest using them.

A.6. Dropout as a randomized exploration strategy

Dropout, as another randomized exploration method, is proposed by Gal & Ghahramani (2016), but Osband et al. (2016) argue about the deficiency of the estimated uncertainty and hardness in driving a suitable exploration and exploitation trade-off from it (Appendix A in (Osband et al., 2016)). They argue that Gal & Ghahramani (2016) does not address the fundamental issue that for large networks trained to convergence all dropout samples may converge to every single datapoint. As also observed by (Dhillon et al., 2018), dropout might results in a ensemble of many models, but all almost the same (converge to the very same model behavior). We also implemented the dropout version of DDQN, Dropout-DDQN, and ran it on four randomly chosen Atari games (among those we ran for less than $50M$ time steps). We observed that the randomization in Dropout-DDQN is deficient and results in performances worse than DDQN on these four Atari games, Fig. 5. In Table 4 we compare the performance of BDQN, DDQN, DDQN$^+$, and Dropout-DDQN, as well as the performance of the random policy, borrowed from Mnih et al. (2015). We observe that the Dropout-DDQN not only does not outperform the plain ε-greedy DDQN, it also sometimes underperforms the random policy. For the game Pong, we also ran Dropout-DDQN for $50M$ time steps but its average performance did not get any better than -17. For the experimental study we used the default dropout rate of 0.5 to mitigate its collapsing issue.

Table 4. The comparison of BDQN, DDQN, Dropout-DDQN and random policy. Dropout-DDQN as another randomization strategy provides a deficient estimation of uncertainty and results in poor exploration/exploitation trade-off.

Game	BDQN	DDQN	DDQN$^+$	Dropout-DDQN	Random Policy	Step
CrazyClimber	124k	84k	102k	19k	11k	40M
Atlantis	3.24M	39.7k	64.76k	7.7k	12.85k	40M
Enduro	1.12k	0.38k	0.32k	0.27k	0	30M
Pong	21	18.82	21	-18	-20.7	5M

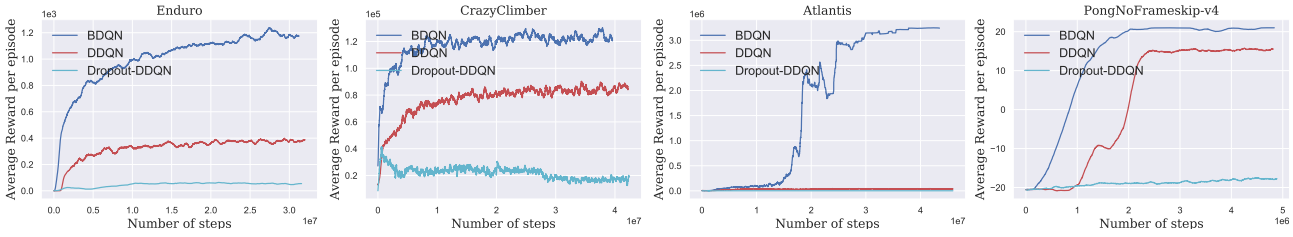

Figure 5. The comparison between DDQN, BDQN and Dropout-DDQN

A.7. Further investigation on Atlantis:

After removing the maximum episode length limit for the game Atlantis, BDQN gets the score of 62M. This episode is long enough to fill half of the replay buffer and make the model perfect for the later part of the game but losing the crafted skill for the beginning of the game. We observe in Fig. 6 that after losing the game in a long episode, the agent forgets

a bit of its skill and loses few games but wraps up immediately and gets to score of $30M$. To overcome this issue, one can expand the replay buffer size, stochastically store samples in the reply buffer where the later samples get stored with lowest chance, or train new models for the later parts of the episode. There are many possible cures for this interesting observation and while we are comparing against DDQN, we do not want to advance BDQN structure-wise.

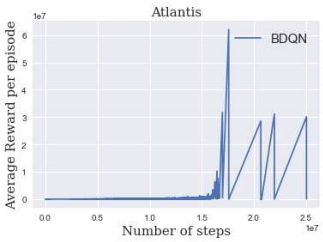

Figure 6. BDQN on Atlantis after removing the limit on max of episode length hits the score of $62M$ in $16M$ samples.

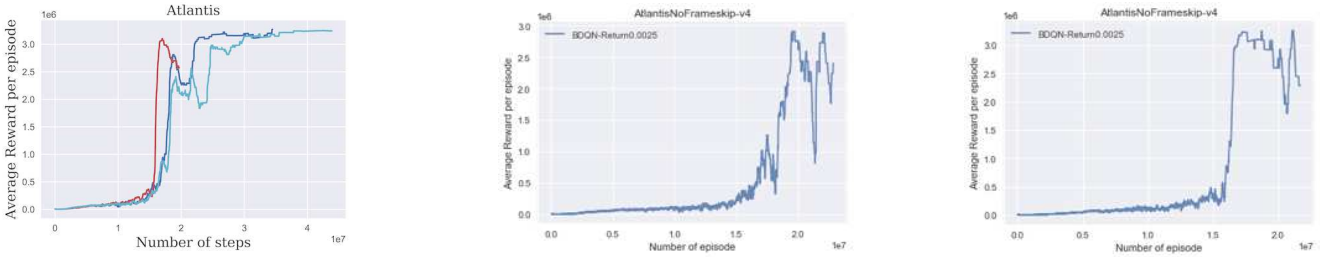

Figure 7. A couple of more runs of BDQN where the jump around $15M$ constantly happens

A.8. Further discussion on Reproducibility

In Table 2, we provide the scores of bootstrap DQN (Osband et al., 2016) and NoisyNet[4](Fortunato et al., 2017) along with BDQN. These score are directly copied from their original papers and we did not make any change to them. We also desired to report the scores of count-based method (Ostrovski et al., 2017), but unfortunately there is no table of score in that paper in order to provide them here.

In order to make it easier for the readers to compare against the results in Ostrovski et al. (2017), we visually approximated their plotted curves for CTS, $Pixel$, and $Reactor$, and added them to the Table 2. We added these numbers just for the convenience of the readers. Surely we do not argue any scientific meaning for them and leave it to the readers to interpret them.

Table 2 shows a significant improvement of BDQN over these baselines. Despite the simplicity and negligible computation overhead of BDQN over DDQN, we can not scientifically claim that BDQN outperforms these baselines by just looking at the scores in Table2 because we are not aware of their detailed implementation as well as environment details. For example, in this work, we directly implemented DDQN by following the implementation details mentioned in the original DDQN paper and the scores of our DDQN implementation during the evaluation time almost matches the scores of DDQN reported in the original paper. But the reported scores of implemented DDQN in Osband et al. (2016) are much different from the reported score in the original DDQN paper.

A.9. A short discussion on safety

In BDQN, as mentioned in Eq. 2, the prior and likelihood are conjugate of each others. Therefore, we have a closed form posterior distribution of the discounted return, $\sum_{t=0}^{N} \gamma^t r_t | x_0 = x, a_0 = a, \mathcal{D}_a$, approximated as

[4]This work does not have scores of Noisy-net with DDQN objective function but it has Noisy-net with DQN objective which are the scores reported in Table 2

$$\mathcal{N}\left(\frac{1}{\sigma_\epsilon^2}\phi^\theta(x)^\top \Xi_a \Phi_a^\theta \mathbf{y}_a, \phi^\theta(x)^\top \Xi_a \phi^\theta(x)\right)$$

One can use this distribution and come up with a safe RL criterion for the agent (Garcıa & Fernández, 2015). Consider the following example; for two actions with the same mean, if the estimated variance over the return increases, then the action becomes more unsafe Fig. 8. By just looking at the low and high probability events of returns under different actions we can approximate whether an action is safe to take.

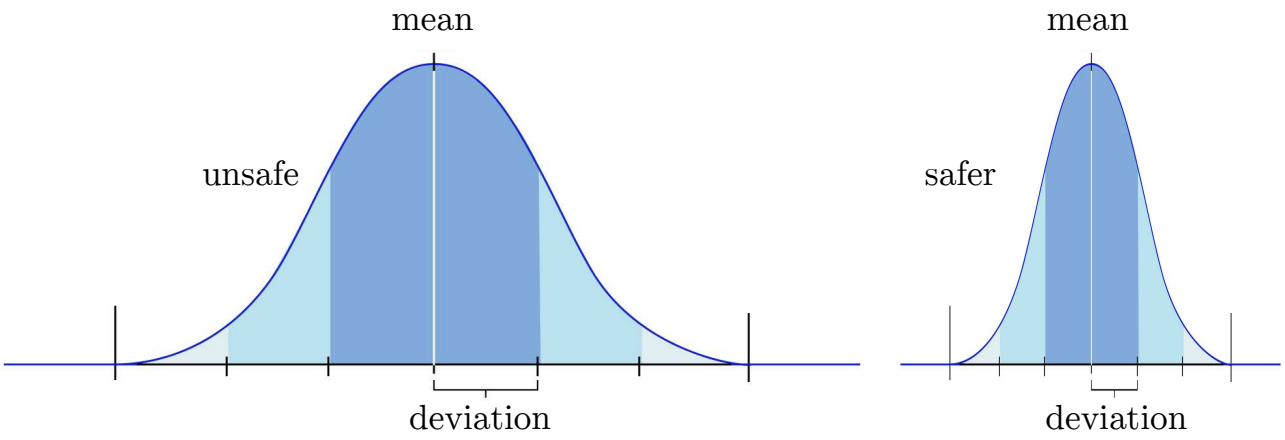

Figure 8. Two actions, with the same mean, but the one with higher variance on the return might be less safe than the one with narrower variance on the return.

B. Bayesian and frequentist regrets, Proof of Theorems 1 and 2

Modeling: We consider an episodic MDP with episode length of H, accompanied with discount factor $0 \leq \gamma \leq 1$. The optimal Q function $Q^*(\cdot,\cdot) \to \mathbb{R}$ is the state action conditional expected return under the optimal policy. For any time step h, and any state and action pairs x^h, a^h we have;

$$Q^*(x^h, a^h) = \mathbb{E}\left[\sum_{h'=}^{H} \gamma^{h'-h} R_{h'} \middle| X^h = x^h, A^h = a^h, \pi^*\right]$$

where X^h and A^h denote the state and action random variables. Since we assume the environment is an MDP, following the Bellman optimality we have for $h < H$

$$Q^*(x^h, a^h) = \mathbb{E}\left[R_h + \gamma Q^*(X^{h+1}, \pi^*(X^{h+1})) \middle| X^h = x^h, A^h = a^h, \pi^*\right],$$

and for $h = H$

$$Q^*(x^H, a^H) = \mathbb{E}\left[R_H \middle| X^H = x^H, A^H = a^H, \pi^*\right]$$

where the optimal policy $\pi^*(\cdot)$ denotes a deterministic mapping from states to actions. Following the Bellman optimality, conditioned on $X^h = x^h, A^h = a^h$, one can rewrite the reward at time step h, R^h, as follows;

$$R^h = Q^*(x^h, a^h) - \gamma \mathbb{E}\left[Q^*(X^{h+1}, \pi^*(X^{h+1})) \middle| X^h = x^h, A^h = a^h, \pi^*\right] + R_\nu^h$$

Where R_ν^h is the noise in the reward, and it is a mean zero random variable due to bellman optimality. In other word, condition on $X^h = x^h, A^h = a^h$, the distribution of one step reward can be described as;

$$R^h + \gamma \mathbb{E}\left[Q^*(X^{h+1}, \pi^*(X^{h+1})) \middle| X^h = x^h, A^h = a^h, \pi^*\right] = Q^*(x^h, a^h) + R_\nu^h$$

where the equality is in distribution. We also can extend the randomness in the noise of the reward one step beyond and include the randomness in the transition. It means, condition on $X^h = x^h, A^h = a^h$ and following π^* at a time step h, for the distribution of one step return we have

$$R^h + \gamma Q^*(X^{h+1}, \pi^*(X^{h+1})) = Q^*(x^h, a^h) + \nu^h \tag{4}$$

where here the ν^h encodes the randomness in the reward as well as the transition kernel, and it is a mean zero random variable. The equality is in distribution. If instead of following π^* after time step h, we follow policies other than π^*, e.g., π then condition on $X^h = x^h, A^h = a^h$ and following π at a time step h, for the distribution of one step return we have

$$R^h + \gamma Q^*(X^{h+1}, \pi(X^{h+1})) = Q^*(x^h, a^h) + \nu^h \tag{5}$$

where then noise process ν^h is not mean zero anymore (it is biased), except for final time step H. We can deduce the bias in $R^h + \gamma Q^*(X^{h+1}, \pi(X^{h+1}))$ and condition on $X^h = x^h, A^h = a^h$ and following π^* at a time step h, for the distribution of one step return we have;

$$\begin{aligned} &R^h + \gamma Q^*(X^{h+1}, \pi(X^{h+1})) \\ &\quad = Q^*(x^h, a^h) + \nu^h + \gamma Q^*(X^{h+1}, \pi(X^{h+1})) - \gamma Q^*(X^{h+1}, \pi^*(X^{h+1})) \end{aligned} \tag{6}$$

where ν^h is an unbiased zero mean random variable due to equality of two mentioned reward distributions in Eq. 4 and Eq. 5.

In the following, we consider the case when the optimal Q function is representable as a linear transformation of given feature representation $\phi(x^h, a^h) \in \mathcal{R}^d$ for any pair of state and actions x^h, a^h at any time step h. In other words, at any time step h we have

$$Q^*(x^h, a^h) = \phi(x^h, a^h)^\top \omega^{*h},\ x^h \in \mathcal{X}^h,\ a^h \in \mathcal{A}^h$$

To keep the notation simple, since all policies, Q functions, and feature presentations are function of h, we encode the h into the state x. We consider the feature represent and the weight vectors of $\omega^{*1}, \omega^{*2}, \ldots, \omega^{*H}$ satisfy the following conditions;

$$\|\phi(x^h, a^h)\phi(x^h, a^h)^\top\|_2^2 \leq L\ ,\ \textit{and}\ \|\omega^{*1}\|_2, \ldots, \|\omega^{*H}\|_2 \leq L_\omega,\ \forall h \in [H],\ x^h \in \mathcal{X}^h,\ a^h \in \mathcal{A}^h$$

Moreover, for the optimal actions and time step h, ρ_λ^h denotes the spectral bound on;

$$\sum_i^t \|\phi(x_t^h, \pi^*(x_t^h))\|^2_{\overline{\chi}_t^{h-1}} \leq \rho_\lambda^h,\ \forall h, t$$

Let $\overline{\rho}_\lambda^H(\gamma)$ denote the following combination of ρ_λ^h;

$$\begin{aligned} \overline{\rho}_\lambda^H(\gamma) := \frac{1}{H}\Big[&\gamma^{H-1} \\ &+ \gamma^{H-2}\left(1 + \gamma\rho^H\right) \\ &+ \gamma^{H-3}\left(1 + \gamma\rho^{H-1} + \gamma^2\rho^{H-2}\rho^{H-1}\right) \\ &+ \ldots \\ &+ \gamma^0\left(1 + \gamma\rho^2 + \ldots + \gamma^{H-1}\rho^2 \ldots \rho^{H-2}\rho^{H-1}\right)\Big] \\ &= \sum_{i=1}^H (\gamma)^{H-i}\left(\frac{1}{H} + \frac{1}{H}\sum_{j=1}^i \prod_{k=1}^j (\gamma)^j \rho_\lambda^{H-(i-k)+1}\right) \end{aligned}$$

with default value of $\rho_\lambda^{H+1} = 0$. It is worth noting that a lose upper bound on $\overline{\rho}_\lambda^H$ when $\lambda = 1$ is $\mathcal{O}\left(\left(\max_h\{\rho_\lambda^h\}\right)^H\right)$.

In the following, we denote the agent policy at h'th time step of t'th episode as π_t^h . We show how to estimate ω^{*h} , $\forall h \in [H]$ using data collected under π_t^h. We denote $\hat{\omega}_t^h$ as the estimation of ω^{*h} for any h at an episode t. We show over time for any h our estimations concentrate around the true paramters ω^{*h}. We further show how to deploy this concentration an construct two algorithms, one based on PSRL, and another based on Optimism OFU to guaranteed Bayesian and frequentist regret upper bounds respectively. The main body of the following analyses on concentration of measure are based on the contextual linear bandit analyses (Abbasi-Yadkori et al., 2011; Chu et al., 2011; Li et al., 2010; Rusmevichientong & Tsitsiklis, 2010; Dani et al., 2008; Russo & Van Roy, 2014a) and self normalized processes (de la Pena et al., 2004; Peña et al., 2009).

For the following we consider the case where $\gamma = 1$ and then show how to extend the results to the discounted case.

Abstract Notation: We use subscript, e.g., t to represent an event in t^{th} episode and superscript, e.g., h to represent an event at h^{th} time step. For example In the following, $\phi(X_t^h, A_t^h)$ represents the feature vector observed at h^{th} time step of t^{th} episode and in short ϕ_t^h represents the same thing. Moreover, we denote π_t as the agent policy during episode t and the concatenation of ϕ_t^h, $\forall h \in [H]$. Similarly, for the optimal policy, we have π^* as the concatenation of ϕ^{*h}, $\forall h \in [H]$. In addition, we have ω_t and ω^* as the concatenation of model parameters in episode t and the optimal parameters.

The target values, condition on $X_t^h = x_t^h, A_t^h = a_t^h$ is as follows;

$$\widetilde{\nu}_t^h = r_t^h + \phi(x_t^{h+1}, \pi^*(x_t^{h+1}))^\top \omega^{*h}$$

Assumption 1 (Sub-Gaussian random variable)**.** *The modified target value $\widetilde{\nu}_t^h$ in Eq. 6, at time step h of t^{th} episode, conditioned on the event up to time step h of the episode t is sub-Gaussian random variable, i.e. there exists a parameter $\sigma \geq 0$ such that $\forall \alpha \in \mathcal{R}^d$*

$$\mathbb{E}\left[\exp\left(\alpha^\top \phi_t^h \widetilde{\nu}_t^h / \sigma - (\alpha^\top \phi_t^h)^2/2\right) \middle| \mathcal{F}_{t-1}^{h-1}\right] \leq 1$$

where $\widetilde{\nu}_t^h$ is $\mathcal{F}_{t-1}^h$-measurable, and $\phi(X_t^h, A_t^h)$ is $\mathcal{F}_{t-1}^{h-1}$-measurable.

A similar assumption on the noise model is considered in the prior analyses of linear bandit (Abbasi-Yadkori et al., 2011).

Expected Reward and Return Assumption: The expected reward and expected return are in $[0, 1]$.

Regret Definition Let V_π^ω denote the value of policy π under a model parameter ω. The regret definition for the frequentist regret is as follows;

$$\textbf{Reg}_T := \mathbb{E}\left[\sum_t^T \left[V_{\pi^*}^{\omega^*} - V_{\pi_t}^{\omega^*}\right] | \omega^*\right]$$

Where π_t is the agent policy during episode t.

When there is a prior over the ω^* the expected Bayesian regret might be the target of the study. The Bayesian regret is as follows;

$$\textbf{BayesReg}_T := \mathbb{E}\left[\sum_t^T \left[V_{\pi^*}^{\omega^*} - V_{\pi_t}^{\omega^*}\right]\right]$$

B.1. Optimism: Regret bound of Alg. 3

In optimism we approximate the desired model parameters $\omega^{*1}, \omega^{*2}, \ldots, \omega^{*H}$ up to their high probability confidence set $\mathcal{C}_t^1(\delta), \mathcal{C}_t^2(\delta), \ldots, \mathcal{C}_t^H(\delta)$ where $\omega^{*h} \in \mathcal{C}_t^h(\delta)$, $\forall h \in [H]$ with probability at least $1 - \delta$. In optimism we choose the most optimistic models from these plausible sets. Let $\widetilde{\omega}_t^1, \widetilde{\omega}_t^2, \ldots, \widetilde{\omega}_t^H$ denote the chosen most optimistic parameters during an episode t while r_t^h and $\phi(x_t^h, a_t^h)$ $\forall h \in [H]$ denote the features and reward observed during the episode t. The most optimistic models are;

$$\widetilde{\omega}_t^h = \arg \max_{\omega \in \mathcal{C}_{t-1}^h(\delta)} \max_{a \in \mathcal{A}^h} \phi(x_t^h, a_t^h)\omega$$

and also the most pessimistic models $\check{\omega}_t^1, \check{\omega}_t^2, \ldots, \check{\omega}_t^H$;

$$\check{\omega}_t^h = \arg \min_{\omega \in \mathcal{C}_{t-1}^h(\delta)} \max_{a \in \mathcal{A}^h} \phi(x_t^h, a_t^h)\omega$$

Now we define the target values as follows;

$$\overline{\nu}_t^h = r_t^h + \phi(x_t^{h+1}, \check{\pi}_t(x_t^{h+1}))^\top \check{\omega}_{t-1}^h$$

where $\check{\pi}_t$ is the policy corresponding to $\check{\omega}_t^h,\ \forall h$. For $h = H$ we have $\overline{\nu}_t^H = r_t^H$. For $h = H$ we have $\widetilde{\nu}_t^H = r_t^H$. The modified target values and the target value have the following relationship

$$\begin{aligned}
\overline{\nu}_t^h &= \overline{\nu}_t^h - \widetilde{\nu}_t^h + \widetilde{\nu}_t^h \\
&= r_t^h + \phi(x_t^{h+1}, \check{\pi}_{t-1}(x_t^{h+1}))^\top \check{\omega}_{t-1}^h - r_t^h - \phi(x_t^{h+1}, \pi^*(x_t^{h+1}))^\top \omega^{*h} + \widetilde{\nu}_t^h \\
&= \phi(x_t^{h+1}, \check{\pi}_t(x_t^{h+1}))^\top \check{\omega}_{t-1}^h - \phi(x_t^{h+1}, \pi^*(x_t^{h+1}))^\top \omega^{*h} + \widetilde{\nu}_t^h
\end{aligned}$$

and when $h = H$ then then $\overline{\nu}_t^h = \widetilde{\nu}_t^h$.

Let $\mathbf{\Phi}_t^h \in \mathbb{R}^{t \times d}$ denote the row-wised concatenation of $\{\phi_i^h\}_{i=1}^t$, $\overline{\boldsymbol{\nu}}_t^h \in \mathbb{R}^t$ a column of target values $\{\overline{\nu}_i^h\}_{i=1}^t$, $\widetilde{\boldsymbol{\nu}}_t^h \in \mathbb{R}^t$ a column of modified target values $\{\widetilde{\nu}_i^h\}_{i=1}^t$, and $\boldsymbol{R}_t^h \in \mathbb{R}^t$ a column of $\{r_i^h\}_{i=1}^t$. Let us restate the following quantities for the self-normalized processes;

$$S_t^h := \sum_i^t \widetilde{\nu}_i^h \phi_i^h = {\mathbf{\Phi}_t^h}^\top \widetilde{\boldsymbol{\nu}}_t^h, \quad \chi_t^h := \sum_{i=1}^t \phi_i^h {\phi_i^h}^\top = {\mathbf{\Phi}_t^h}^\top \mathbf{\Phi}_t^h, \quad \overline{\chi}_t^h = \chi_t^h + \widetilde{\chi}^h$$

where $\widetilde{\chi}^h$ is a ridge regularization matrix and usually is equal to λI.

Lemma 1 (Confidence intervals)**.** *Let $\widehat{\omega}_t^h$ denote the estimation of ω^{*h} given $\mathbf{\Phi}_t^1, \ldots, \mathbf{\Phi}_t^H$ and $\overline{\boldsymbol{\nu}}_t^1, \ldots, \overline{\boldsymbol{\nu}}_t^H$;*

$$\widehat{\omega}_t^h := \left({\mathbf{\Phi}_t^h}^\top \mathbf{\Phi}_t^h + \lambda I\right)^{-1} {\mathbf{\Phi}_t^h}^\top \overline{\boldsymbol{\nu}}_t^h$$

with probability at least $1 - \delta/H$

$$\|\widehat{\omega}_t^h - \omega^{*h}\|_{\overline{\chi}_t^h} \leq \theta_t^h(\delta) : \sigma\sqrt{2\log(H/\delta) + d\log(1 + tL^2/\lambda)} + \lambda^{1/2} L_\omega + \theta_t^{h+1}(\delta)\sqrt{\rho^{h+1}}$$

and

$$\mathcal{C}_t^h(\delta) := \{\omega \in \mathbb{R}^d : \|\widehat{\omega}_t^h - \omega^{*h}\|_{\overline{\chi}_t^h} \leq \theta_t^h(\delta)\}$$

where for $h = H$

$$\|\widehat{\omega}_t^H - \omega^{*H}\|_{\overline{\chi}_t^H} \leq \theta_t^H(\delta) : \sigma\sqrt{2\log(H/\delta) + d\log(1 + tL^2/\lambda)} + \lambda^{1/2} L_\omega$$

for all $t \leq T$. Furthermore, $\theta_t^{H+1} = 0, \forall t$.

Let Θ_T denote the event that the confidence bounds in Lemma 1 holds at least until T^{th} episode.

Lemma 2 (Determinant Lemma(Lemma 11 in (Abbasi-Yadkori et al., 2011)))**.** *For a sequence ϕ_t^h we have*

$$\sum_t^T \log\left(1 + \|\phi_t^h\|^2_{\overline{\chi}_{t-1}^{-1}}\right) \leq d\log(\lambda + TL^2/d) \tag{7}$$

Lemma 1 states that under event Θ_T, $\|\widehat{\omega}_t^h - {\omega^*}^h\|_{\overline{\chi}_t^h} \leq \theta_t^h(\delta)$. Furthermore, we define state and policy dependent optimistic parameter $\widetilde{\omega}_t(\pi)$ as follows;

$$\widetilde{\omega}_t^h(\pi) := \arg \max_{\omega \in C_{t-1}^h(\delta)} \phi(X_t^h, \pi(X_t^h))^\top \omega$$

Following OFU, Alg. 3, we set $\pi_t^h = \widetilde{\pi}_t^h,\ \forall h$, denotes the optimistic policy. By the definition we have

$$V_{\widetilde{\pi}_t^h}^{\widetilde{\omega}_t^h(\widetilde{\pi}_t^h)}(X_t^h) := \phi(X_t^h, \widetilde{\pi}_t^h(X_t^h))^\top \widetilde{\omega}_t^h(\widetilde{\pi}_t^h) \geq V_{\pi^{*h}}^{\widetilde{\omega}_t(\pi^{*h})}(X_t^h)$$

We use this inequality to derive an upper bound for the regret;

$$\begin{aligned}
\mathbf{Reg}_T :&= \mathbb{E}\left[\sum_t^T \left[\underbrace{V_{\pi^*}^{\omega^*}(X_t^1) - V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1)}_{\Delta_t^{h=1}}\right] |\omega^*\right] \\
&\leq \mathbb{E}\left[\sum_t^T \left[V_{\widetilde{\pi}_t^1}^{\widetilde{\omega}_t^1(\widetilde{\pi}_t^1)}(X_t^1) - V_{\pi^{*1}}^{\widetilde{\omega}_t^1(\pi^{*1})}(X_t^1) + V_{\pi^*}^{\omega^*}(X_t^1) - V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1)\right] |\omega^*\right] \\
&= \mathbb{E}\left[\sum_t^T \left[V_{\widetilde{\pi}_t^1}^{\widetilde{\omega}_t^1(\widetilde{\pi}_t^1)}(X_t^1) - V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1) + \underbrace{V_{\pi^*}^{\omega^*}(X_t^1) - V_{\pi^{*1}}^{\widetilde{\omega}_t^h(\pi^{*1})}(X_t^1)}_{\leq 0}\right] |\omega^*\right]
\end{aligned}$$

Resulting in

$$\mathbf{Reg}_T \leq \mathbb{E}\left[\sum_t^T \left[V_{\widetilde{\pi}_t^1}^{\widetilde{\omega}_t^1(\widetilde{\pi}_t^1)}(X_t^1) - V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1)\right] |\omega^*\right]$$

Let us defined $V_\pi^{\omega^*}(X_t^{h'}; h)$ as the value function at $X_t^{h'}$, following policies π for the first h time steps then switching to the optimal policy.

$$\begin{aligned}
\mathbf{Reg}_T &\leq \mathbb{E}\left[\sum_t^T \left[V_{\widetilde{\pi}_t^1}^{\widetilde{\omega}_t^1(\widetilde{\pi}_t^1)}(X_t^1) - V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1)\right] |\omega^*\right] \\
&= \mathbb{E}\left[\sum_t^T \left[V_{\widetilde{\pi}_t^1}^{\widetilde{\omega}_t^1(\widetilde{\pi}_t^1)}(X_t^1) - V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1; 1) + V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1; 1) - V_{\widetilde{\pi}_t^1}^{{\omega^*}^1}(X_t^1)\right] |\omega^*\right]
\end{aligned}$$

Given the linear model of the Q function we have;

$$\begin{aligned}
\mathbf{Reg}_T &\leq \mathbb{E}\left[\sum_t^T \left[V_{\widetilde{\pi}_t^1}^{\widetilde{\omega}_t^1(\widetilde{\pi}_t^1)}(X_t^1) - V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1; 1) + V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1; 1) - V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1)\right] |\omega^*\right] \\
&= \mathbb{E}\left[\sum_t^T \left[\phi(X_t^1, \widetilde{\pi}_t^1(X_t^1))^\top \widetilde{\omega}_t^1(\widetilde{\pi}_t^1) - \phi(X_t^1, \widetilde{\pi}_t^1(X_t^1))^\top {\omega^*}^1 + V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1; 1) - V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1)\right] |\omega^*\right] \\
&= \mathbb{E}\left[\sum_t^T \left[\phi(X_t^1, \widetilde{\pi}_t^1(X_t^1))^\top \left(\widetilde{\omega}_t^1(\widetilde{\pi}_t) - {\omega^*}^1\right) + \underbrace{V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1; 1) - V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1)}_{(\Delta_t^{h=2})}\right] |\omega^*\right]
\end{aligned}$$

For $\Delta_t^{h=2}$ we deploy the similar decomposition and upper bound as $\Delta_t^{h=1}$

$$\Delta_t^2 := V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1; 1) - V_{\widetilde{\pi}_t}^{\omega^*}(X_t^1)$$

Since for both of $V^{\omega^*}_{\widetilde{\pi}_t}(X^1_t;1)$ and $V^{\omega^*}_{\widetilde{\pi}_t}(X^1_t)$ we follow the same policy on the same model for 1 time step, the reward at the first time step has the same distribution, therefore we have;

$$\Delta^{h=2}_t = E\left[V^{\omega^*}_{\widetilde{\pi}_t}(X^2_t;1) - V^{\omega^*}_{\widetilde{\pi}_t}(X^2_t)|X^1_t, A^1_t = \widetilde{\pi}_t(X^1_t), \omega^*\right]$$

resulting in

$$\begin{aligned}\Delta^{h=2}_t &= E\left[V^{\omega^*}_{\widetilde{\pi}_t}(X^2_t;1) - V^{\omega^*}_{\widetilde{\pi}_t}(X^2_t)|X^1_t, A^1_t = \widetilde{\pi}_t(X^1_t), \omega^*\right] \\ &\leq \mathbb{E}\left[\phi(X^2_t, \widetilde{\pi}^2_t(X^2_t))^\top \left(\widetilde{\omega}^2_t(\widetilde{\pi}^2_t) - \omega^{*2}\right) + \underbrace{V^{\omega^*}_{\widetilde{\pi}_t}(X^2_t;2) - V^{\omega^*}_{\widetilde{\pi}_t}(X^2_t)}_{(\Delta^{h=3}_t)}|X^1_t, A^1_t = \widetilde{\pi}_t(X^1_t), \omega^*\right]\end{aligned}$$

Similarly we can defined $\Delta^{h=3}_t, \dots \Delta^{h=H}_t$. Therefore;

$$\begin{aligned}\mathbf{Reg}_T &\leq \mathbb{E}\left[\sum_t^T\sum_h^H \phi(X^h_t, \widetilde{\pi}_t(X^h_t))^\top \left(\widetilde{\omega}^h_t(\widetilde{\pi}^h_t) - \omega^{*h}\right)|\omega^*\right] \\ &= \mathbb{E}\left[\sum_t^T\sum_h^H \phi(X^h_t, \widetilde{\pi}_t(X^h_t))^\top {\overline{\chi}^h_{t-1}}^{-1/2} {\overline{\chi}^h_{t-1}}^{1/2}\left(\widetilde{\omega}^h_t(\widetilde{\pi}^h_t) - \omega^{*h}\right)|\omega^*\right] \\ &\leq \mathbb{E}\left[\sum_t^T\sum_h^H \|\phi(X^h_t, \widetilde{\pi}_t(X^h_t))\|_{{\overline{\chi}^h_{t-1}}^{-1}} \|\widetilde{\omega}^h_t(\widetilde{\pi}^h_t) - \omega^{*h}\|_{\overline{\chi}^h_{t-1}}|\omega^*\right] \\ &\leq \mathbb{E}\left[\sum_t^T\sum_h^H \|\phi(X^h_t, \widetilde{\pi}_t(X^h_t))\|_{{\overline{\chi}^h_{t-1}}^{-1}} 2\theta^h_{t-1}(\delta)|\omega^*\right]\end{aligned} \tag{8}$$

Since the maximum expected cumulative reward, condition on states of a episode is at most 1, we have;

$$\begin{aligned}\mathbf{Reg}_T &\leq \mathbb{E}\left[\sum_t^T\sum_h^H \min\{\|\phi(X^h_t, \widetilde{\pi}_t(X^h_t))\|_{{\overline{\chi}^h_{t-1}}^{-1}} 2\theta^h_{t-1}(\delta), 1\}|\omega^*\right] \\ &\leq \mathbb{E}\left[\sum_t^T\sum_h^H 2\theta^h_{t-1}(\delta)\min\{\|\phi(X^h_t, \widetilde{\pi}_t(X^h_t))\|_{{\overline{\chi}^h_{t-1}}^{-1}}, 1\}|\omega^*\right]\end{aligned}$$

Moreover, at time T, we can use Jensen's inequality, exploit the fact that $\theta^h_t(\delta)$ is an increasing function of t and have

$$\mathbf{Reg}_T \leq 2\mathbb{E}\left[\sqrt{T\sum_h^H 2\theta^h_T(\delta)^2 \sum_t^T \min\{\|\phi(X^h_t, \widetilde{\pi}_t(X^h_t))\|^2_{{\overline{\chi}^h_{t-1}}^{-1}}, 1\}}|\omega^*\right] \tag{9}$$

Now, using the fact that for any scalar α such that $0 \leq \alpha \leq 1$, then $\alpha \leq 2\log(1+\alpha)$ we can rewrite the latter part of Eq. 9

$$\sum_t^T \min\{\|\phi(X^h_t, \widetilde{\pi}_t(X^h_t))\|^2_{{\overline{\chi}^h_{t-1}}^{-1}}, 1\} \leq 2\sum_t^T \log\left(1 + \|\phi(X^h_t, \widetilde{\pi}_t(X^h_t))\|^2_{{\overline{\chi}^h_{t-1}}^{-1}}\right)$$

By applying the Lemma 2 and substituting the RHS of Eq. 7 into Eq. 9, we get

$$\begin{aligned}\mathbf{Reg}_T &\leq 2\mathbb{E}\left[\sqrt{T\sum_{h}^{H} 2\theta_T^h(\delta)^2 d\log(\lambda + TL^2/d)|\omega^*}\right] \\ &\leq 2\mathbb{E}\left[\sqrt{Td\log(\lambda+TL^2/d)\sum_{h}^{H} 4\left(\sigma\sqrt{2\log\left(H/\delta\right)+d\log\left(1+TL^2/\lambda\right)}+\lambda^{1/2}L_\omega\right)^2 + 4\theta_T^{h+1}(\delta)^2\rho^{h+1}|\omega^*}\right]\end{aligned} \tag{10}$$

and deriving the upper bounds

$$\mathbf{Reg}_T \leq 2\left(\sigma\sqrt{2\log\left(1/\delta\right)+d\log\left(1+TL^2/\lambda\right)}+\lambda^{1/2}L_\omega\right)\sqrt{4\overline{\rho}_\lambda^H(1)THd\log(\lambda+TL^2/d)} \tag{11}$$

with probability at least $1-\delta$. If we set $\delta = 1/T$ then the probability that the event Θ_T holds is $1-1/T$ and we get regret of at most the RHS of Eq. 11, otherwise with probability at most $1/T$ we get maximum regret of T, therefore

$$\mathbf{Reg}_T \leq 1 + 2\left(\sigma\sqrt{2\log\left(1/\delta\right)+d\log\left(1+TL^2/\lambda\right)}+\lambda^{1/2}L_\omega\right)\sqrt{4\overline{\rho}_\lambda^H(1)THd\log(\lambda+TL^2/d)}$$

For the case of discounted reward, substituting $\overline{\rho}_\lambda^H(\gamma)$ instead of $\overline{\rho}_\lambda^H(1)$ results in the theorem statement.

B.2. Bayesian Regret of Alg. 2

The analysis developed in the previous section, up to some minor modification, e.g., change of strategy to PSRL, directly applies to Bayesian regret bound, with a farther expectation over models.

When there is a prior over the ${\omega^*}^h,\ \forall h$ the expected Bayesian regret might be the target of the study.

$$\begin{aligned}\mathbf{BayesReg}_T :&= \mathbb{E}\left[\sum_{t}^{T}\left[V_{\pi^*}^{\omega^*} - V_{\pi_t}^{\omega^*}\right]\right] \\ &\mathbb{E}\left[\sum_{t}^{T}\left[V_{\pi^*}^{\omega^*} - V_{\pi_t}^{\omega^*}\middle|\mathcal{H}_t\right]\right]\end{aligned}$$

Here $\mathcal{H}_t$ is a multivariate random sequence which indicates history at the beginning of episode t and $\pi_t^h, \forall h$ are the policies following PSRL. For the remaining π_t^h denotes the PSRL policy at each time step h. As it mentioned in the Alg. 2, at the beginning of an episode, we draw $\omega_t^h\ \forall h$ from the posterior and the corresponding policies are;

$$\pi_t^h(X_t^h) := \arg\max_{a\in\mathcal{A}} \phi(X_t^h, a)^\top \omega_t^h$$

Condition on the history $\mathcal{H}_t$, i.e., the experiences by following the agent policies $\pi_{t'}^h$ for each episode $t' \leq t$, we estimate the $\widehat{\omega}_t^h$ as follows;

$$\widehat{\omega}_t^h := \left({\mathbf{\Phi}_t^h}^\top \mathbf{\Phi}_t^h + \lambda I\right)^{-1} {\mathbf{\Phi}_t^h}^\top \overline{\boldsymbol{\nu}}_t^h$$

Lemma 1 states that under event Θ_T, $\|\widehat{\omega}_t^h - {\omega^*}^h\|_{\overline{\chi}_t} \leq \theta_t^h(\delta),\ \forall h$. Conditioned on $\mathcal{H}_t$, the ω_t^h and ${\omega^*}^h$ are equally distributed, then we have

$$\mathbb{E}\left[V_{{\pi^*}^h}^{\widetilde{\omega}_t^h(\pi^{*h})} = \phi(X_t^h, {\pi^*}^h(X_t^h))^\top \widetilde{\omega}_t^h({\pi^*}^h)\middle|\mathcal{H}_t\right] = \mathbb{E}\left[V_{\pi_t^h}^{\widetilde{\omega}_t^h(\pi_t^h)} = \phi(X_t^h, \pi_t^h(X_t^h))^\top \widetilde{\omega}_t^h(\pi_t^h)\middle|\mathcal{H}_t\right]$$

Therefore, for the regret we have

$$\begin{aligned}\textbf{BayesReg}_T :&= \sum_t^T \mathbb{E}\left[\underbrace{V_{\pi^*}^{\omega^*}(X_t^1) - V_{\pi_t}^{\omega^*}(X_t^1)}_{\Delta_t^{h=1}}|\mathcal{H}_t\right] \\ &= \sum_t^T \mathbb{E}\left[V_{\pi_t^1}^{\widetilde{\omega}_t^1(\pi_t^1)}(X_t^1) - V_{\pi^{*1}}^{\widetilde{\omega}_t^1(\pi^{*1})}(X_t^1) + V_{\pi^*}^{\omega^*}(X_t^1) - V_{\pi_t}^{\omega^*}(X_t^1)|\mathcal{H}_t\right] \\ &= \sum_t^T \mathbb{E}\left[V_{\pi_t^1}^{\widetilde{\omega}_t^1(\pi_t^1)}(X_t^1) - V_{\pi_t}^{\omega^*}(X_t^1) + \underbrace{V_{\pi^*}^{\omega^*}(X_t^1) - V_{\pi^{*1}}^{\widetilde{\omega}_t^1(\pi^{*1})}(X_t^1)}_{\leq 0}|\mathcal{H}_t\right]\end{aligned}$$

Resulting in

$$\textbf{BayesReg}_T \leq \sum_t^T \mathbb{E}\left[V_{\pi_t^1}^{\widetilde{\omega}_t^1(\pi_t^1)}(X_t^1) - V_{\pi_t}^{\omega^*}(X_t^1)|\mathcal{H}_t\right]$$

Similar to optimism and defining $V_\pi^\omega(X_t^{h'};h)$ we have;

$$\begin{aligned}\textbf{BayesReg}_T &\leq \mathbb{E}\left[\sum_t^T \left[V_{\pi_t^1}^{\widetilde{\omega}_t^1(\pi_t^1)}(X_t^1) - V_{\pi_t}^{\omega^*}(X_t^1)|\mathcal{H}_t\right]\right] \\ &= \mathbb{E}\left[\sum_t^T \left[V_{\pi_t^1}^{\widetilde{\omega}_t^1(\pi_t^1)}(X_t^1) - V_{\pi_t}^{\omega^*}(X_t^1;1) + V_{\pi_t}^{\omega^*}(X_t^1;1) - V_{\pi_t}^{\omega^*}(X_t^1)|\mathcal{H}_t\right]\right] \\ &\leq \mathbb{E}\left[\sum_t^T \left[\phi(X_t^1,\pi_t^1(X_t^1))^\top \widetilde{\omega}_t^1(\pi_t^1) - \phi(X_t^1,\pi_t^1(X_t^1))^\top \omega^{*1} + V_{\pi_t}^{\omega^*}(X_t^1;1) - V_{\pi_t}^{\omega^*}(X_t^1)|\mathcal{H}_t\right]\right] \\ &= \mathbb{E}\left[\sum_t^T \left[\phi(X_t^1,\pi_t^1(X_t^1))^\top \left(\widetilde{\omega}_t^1(\pi_t^1) - \omega^{*1}\right) + \underbrace{V_{\pi_t}^{\omega^*}(X_t^1;1) - V_{\pi_t}^{\omega^*}(X_t^1)}_{(\Delta_t^{h=2})}|\mathcal{H}_t\right]\right]\end{aligned}$$

For $\Delta_t^{h=2}$ we deploy the similar decomposition and upper bound as $\Delta_t^{h=1}$

$$\Delta_t^2 := V_{\pi_t}^{\omega^*}(X_t^1;1) - V_{\pi_t}^{\omega^*}(X_t^1)$$

Since for both of $V_{\pi_t}^{\omega^*}(X_t^1;1)$ and $V_{\pi_t}^{\omega^*}(X_t^1)$ we follow the same distribution over policies and models for 1 time step, the reward at the first time step has the same distribution, therefore we have;

$$\Delta_t^{h=2} = E\left[V_{\pi_t}^{\omega^*}(X_t^2;1) - V_{\pi_t}^{\omega^*}(X_t^2)|X_t^1, A_t^1 = \pi_t(X_t^1), \mathcal{H}_t\right]$$

resulting in

$$\begin{aligned}\Delta_t^{h=2} &= E\left[V_{\pi_t}^{\omega^*}(X_t^2;1) - V_{\pi_t}^{\omega^*}(X_t^2)|X_t^1, A_t^1 = \pi_t(X_t^1), \mathcal{H}_t\right] \\ &\leq \mathbb{E}\left[\phi(X_t^2,\pi_t^2(X_t^2))^\top \left(\widetilde{\omega}_t^2(\pi_t^2) - \omega^{*2}\right) + \underbrace{V_{\pi_t}^{\omega^{*2}}(X_t^2;2) - V_{\pi_t}^{\omega^*}(X_t^2)}_{(\Delta_t^{h=3})}|X_t^1, A_t^1 = \pi_t(X_t^1), \mathcal{H}_t\right]\end{aligned}$$

Similarly we can defined $\Delta_t^{h=3}, \dots \Delta_t^{h=H}$. The condition on $\mathcal{H}_t$ was required to come up with the mentioned decomposition through Δ_t^h and it is not needed anymore, therefore;

$$
\begin{aligned}
\textbf{BayesReg}_T &\leq \mathbb{E}\left[\sum_t^T \sum_h^H \phi(X_t^h, \widetilde{\pi}_t(X_t^h))^\top \left(\widetilde{\omega}_t^h(\widetilde{\pi}_t^h) - \omega^{*h}\right)\right] \\
&= \mathbb{E}\left[\sum_t^T \sum_h^H \phi(X_t^h, \widetilde{\pi}_t(X_t^h))^\top {\overline{\chi}_{t-1}^h}^{-1/2} {\overline{\chi}_{t-1}^h}^{1/2} \left(\widetilde{\omega}_t^h(\widetilde{\pi}_t^h) - \omega^{*h}\right)\right] \\
&\leq \mathbb{E}\left[\sum_t^T \sum_h^H \|\phi(X_t^h, \widetilde{\pi}_t(X_t^h))\|_{{\overline{\chi}_{t-1}^h}^{-1}} \|\widetilde{\omega}_t^h(\widetilde{\pi}_t^h) - \omega^{*h}\|_{\overline{\chi}_{t-1}^h}\right] \\
&\leq \mathbb{E}\left[\sum_t^T \sum_h^H \|\phi(X_t^h, \widetilde{\pi}_t(X_t^h))\|_{{\overline{\chi}_{t-1}^h}^{-1}} 2\theta_{t-1}^h(\delta)\right]
\end{aligned} \tag{12}
$$

Again similar to optimism we have the maximum expected cumulative reward condition on states of a episode is at most 1, we have;

$$
\begin{aligned}
\textbf{BayesReg}_T &\leq \mathbb{E}\left[\sum_t^T \sum_h^H \min\{\|\phi(X_t^h, \widetilde{\pi}_t(X_t^h))\|_{{\overline{\chi}_{t-1}^h}^{-1}} 2\theta_{t-1}^h(\delta), 1\}\right] \\
&\leq \mathbb{E}\left[\sum_t^T \sum_h^H 2\theta_{t-1}^h(\delta) \min\{\|\phi(X_t^h, \widetilde{\pi}_t(X_t^h))\|_{{\overline{\chi}_{t-1}^h}^{-1}}, 1\}\right]
\end{aligned}
$$

Moreover, at time T, we can use Jensen's inequality, exploit the fact that $\theta_t(\delta)$ is an increasing function of t and have

$$
\begin{aligned}
\textbf{BayesReg}_T &\leq 2\mathbb{E}\left[\sqrt{T\sum_h^H 2\theta_T^h(\delta)^2 \sum_t^T \min\{\|\phi(X_t^h, \widetilde{\pi}_t(X_t^h))\|^2_{{\overline{\chi}_{t-1}^h}^{-1}}, 1\}}\right] \\
&\leq 2\mathbb{E}\left[\sqrt{T\sum_h^H 2\theta_T^h(\delta)^2 d\log(\lambda + TL^2/d)}\right] \\
&\leq 2\mathbb{E}\left[\sqrt{Td\log(\lambda + TL^2/d)\sum_h^H 4\left(\sigma\sqrt{2\log(H/\delta) + d\log(1 + TL^2/\lambda)} + \lambda^{1/2}L_\omega\right)^2 + 4\theta_T^{h+1}(\delta)^2\rho^{h+1}}\right] \\
&\leq 2\left(\sigma\sqrt{2\log(1/\delta) + d\log(1 + TL^2/\lambda)} + \lambda^{1/2}L_\omega\right)\sqrt{4\overline{\rho}_\lambda^H(1)THd\log(\lambda + TL^2/d)}
\end{aligned} \tag{13}
$$

Under event Θ_T which holds with probability at least $1 - \delta$. If we set $\delta = 1/T$ then the probability that the event Θ holds is $1 - 1/T$ and we get regret of at most the RHS of Eq. 13, otherwise with probability at most $1/T$ we get maximum regret of T, therefore

$$
\textbf{BayesReg}_T \leq 1 + 2\left(\sigma\sqrt{2\log(1/\delta) + d\log(1 + TL^2/\lambda)} + \lambda^{1/2}L_\omega\right)\sqrt{4\overline{\rho}_\lambda^H(1)THd\log(\lambda + TL^2/d)}
$$

For the case of discounted reward, substituting $\overline{\rho}_\lambda^H(\gamma)$ instead of $\overline{\rho}_\lambda^H(1)$ results in the theorem statement.

B.3. Proof of Lemmas

Lemma 3. *Let $\alpha \in \mathcal{R}^d$ be any vector and for any $t \geq 0$ define*

$$
M_t^h(\alpha) := \exp\left(\sum_i^t \left[\frac{\alpha^\top \phi_i^h \overline{\nu}_i^h}{\sigma} - \frac{1}{2}\|{\phi_i^h}^\top \alpha\|_2^2\right]\right), \ \forall h
$$

Then, for a stopping time under the filtration $\{\mathcal{F}_t^h\}_{t=0}^{\infty}$, $M_\tau^h(\alpha) \leq 1$.

Proof. Lemma 3 We first show that $\{M_t^h(\alpha)\}_{t=0}^{\infty}$ is a supermartingale sequence. Let

$$D_i^h(\alpha) = \exp\left(\frac{\alpha^\top \phi_i^h \overline{\nu}_i^h}{\sigma} - \frac{1}{2}\|{\phi_i^h}^\top \alpha\|_2^2\right)$$

Therefore, we can rewrite $\mathbb{E}\left[M_t^h(\alpha)\right]$ as follows:

$$\mathbb{E}\left[M_t^h(\alpha)|\mathcal{F}_{t-1}^h\right] = \mathbb{E}\left[D_1^h(\alpha)\dots D_{t-1}^h(\alpha)D_t^h(\alpha)|\mathcal{F}_{t-1}^h\right] = D_1^h(\alpha)\dots D_{t-1}^h(\alpha)\mathbb{E}\left[D_t^h(\alpha)|\mathcal{F}_{t-1}^h\right] \leq M_{t-1}^h(\alpha)$$

The last inequality follows since $\mathbb{E}\left[D_t^h(\alpha)|\mathcal{F}_{t-1}^h\right] \leq 1$ due to Assumption 1, therefore since for the first time step $\mathbb{E}\left[M_1^h(\alpha)\right] \leq 1$, then $\mathbb{E}\left[M_t^h(\alpha)\right] \leq 1$. For a stopping time τ, Define a variable $\overline{M}_t^\alpha = M_{\min\{t,\tau\}}^h(\alpha)$ and since $\mathbb{E}\left[M_\tau^h(\alpha)\right] = \mathbb{E}\left[\liminf_{t\to\infty} \overline{M}_t^h(\alpha)\right] \leq \liminf_{t\to\infty} \mathbb{E}\left[\overline{M}_t^h(\alpha)\right] \leq 1$, therefore the Lemma 3 holds. □

Lemma 4. *[Extension to Self-normalized bound in (Abbasi-Yadkori et al., 2011)] For a stopping time* τ *and filtration* $\{\mathcal{F}_t^h\}_{t=0,h=1}^{\infty,h}$*, with probability at least* $1 - H\delta$

$$\|S_\tau^h\|^2_{\overline{\chi}_\tau^{h}{}^{-1}} \leq 2\sigma^2 \log\left(\frac{\det\left(\overline{\chi}_\tau^h\right)^{1/2}\det(\widetilde{\chi}^h)^{-1/2}}{\delta}\right)$$

Proof. of Lemma 4. Given the definition of the parameters of self-normalized process, we can rewrite $M_t^h(\alpha)$ as follows;

$$M_t^h(\alpha) = \exp\left(\frac{\alpha^\top S_t^h}{\sigma} - \frac{1}{2}\|\alpha\|^2_{\chi_t^h}\right)$$

Consider Ω^h as a Gaussian random vector and $f(\Omega^h = \alpha)$ denotes the density with covariance matrix of $\overline{\chi}^{h}{}^{-1}$. Define $M_t^h := \mathbb{E}\left[M_t^h(\Omega^h)|\mathcal{F}_\infty\right]$. Therefore we have $\mathbb{E}\left[M_t^h\right] = \mathbb{E}\left[\mathbb{E}\left[M_t^h(\Omega^h)|\Omega^h\right]\right] \leq 1$. Therefore

$$\begin{aligned}
M_t^h &= \int_{\mathbb{R}^d} \exp\left(\frac{\alpha^\top S_t^h}{\sigma} - \frac{1}{2}\|\alpha\|^2_{\chi_t^h}\right) f(\alpha)d\alpha \\
&= \int_{\mathbb{R}^d} \exp\left(\frac{1}{2}\|\alpha - \chi_t^{h}{}^{-1} S_t/\sigma\|^2_{\chi_t^h} + \frac{1}{2}\|S_t^h/\sigma\|^2_{\chi_t^{h}{}^{-1}}\right) f(\alpha)d\alpha \\
&= \sqrt{\frac{\det\left(\widetilde{\chi}^h\right)}{(2\pi)^d}} \exp\left(\frac{1}{2}\|S_t^h/\sigma\|_{\chi_t^{h}{}^{-1}}\right) \int_{\mathbb{R}^d} \exp\left(\frac{1}{2}\|\alpha - \chi_t^{h}{}^{-1} S_t^h/\sigma\|^2_{\chi_t^h} + \frac{1}{2}\|\alpha\|^2_{\widetilde{\chi}^h}\right) d\alpha
\end{aligned}$$

Since χ_t^h and $\widetilde{\chi}^h$ are positive semi definite and positive definite respectively, we have

$$\begin{aligned}
\|\alpha - \chi_t^{h}{}^{-1} S_t^h/\sigma\|^2_{\chi_t^h} + \|\alpha\|^2_{\widetilde{\chi}^h} &= \|\alpha - \left(\widetilde{\chi}^h + \chi_t^{h}{}^{-1}\right) S_t^h/\sigma\|^2_{\widetilde{\chi}^h + \chi_t^h} + \|\chi_t^{h}{}^{-1} S_t^h/\sigma\|^2_{\chi_t^h} - \|S_t^h/\sigma\|^2_{\left(\widetilde{\chi}^h + \chi_t^h\right)^{-1}} \\
&= \|\alpha - \left(\widetilde{\chi}^h + \chi_t^{h}{}^{-1}\right) S_t^h/\sigma\|^2_{\widetilde{\chi}^h + \chi_t^h} + \|\mathcal{S}_t^h/\sigma\|^2_{\chi_t^{h}{}^{-1}} - \|S_t^h/\sigma\|^2_{\left(\widetilde{\chi}^h + \chi_t^h\right)^{-1}}
\end{aligned}$$

Therefore,

$$\begin{aligned}
M_t^h &= \sqrt{\frac{\det\left(\widetilde{\chi}^h\right)}{(2\pi)^d}} \exp\left(\frac{1}{2}\|S_t^h/\sigma\|_{\left(\widetilde{\chi}^h + \chi_t^h\right)^{-1}}\right) \int_{\mathbb{R}^d} \exp\left(\frac{1}{2}\|\alpha - \left(\widetilde{\chi}^h + \chi_t^{h}{}^{-1}\right) S_t^h/\sigma\|^2_{\widetilde{\chi}^h + \chi_t^h}\right) d\alpha \\
&= \left(\frac{\det\left(\widetilde{\chi}^h\right)}{\det\left(\widetilde{\chi}^h + \chi_t^h\right)}\right)^{1/2} \exp\left(\frac{1}{2}\|S_t/\sigma\|^2_{\left(\widetilde{\chi}^h + \chi_t^h\right)^{-1}}\right)
\end{aligned}$$

Since $\mathbb{E}\left[M_\tau^h\right] \leq 1$ we have

$$\mathbb{P}\left(\|S_\tau^h\|^2_{\left(\overline{\chi}_\tau^h\right)^{-1}} \leq 2\sigma \log\left(\frac{\det\left(\overline{\chi}_\tau^h\right)^{1/2}}{\delta \det\left(\widetilde{\chi}^h\right)^{1/2}}\right)\right) \leq \frac{\mathbb{E}\left[\exp\left(\frac{1}{2}\|S_\tau^h/\sigma\|^2_{\left(\overline{\chi}_\tau^h\right)^{-1}}\right)\right]}{\frac{\left(\det\left(\overline{\chi}_\tau^h\right)\right)^{1/2}}{\delta\left(\det\left(\widetilde{\chi}^h\right)\right)^{1/2}}} \leq \delta$$

Where the Markov inequality is deployed for the final step. The stopping is considered to be the time step as the first time in the sequence when the concentration in the Lemma 4 does not hold. □

Proof of Lemma 1. Consider the following estimator $\widehat{\omega}_t^h$:

$$\begin{aligned}
\widehat{\omega}_t^h &= \left(\boldsymbol{\Phi}_t^{h\top}\boldsymbol{\Phi}_t^h + \lambda I\right)^{-1}\left(\boldsymbol{\Phi}_t^{h\top}\overline{\boldsymbol{\nu}}_t^h\right) \\
&= \left(\boldsymbol{\Phi}_t^{h\top}\boldsymbol{\Phi}_t^h + \lambda I\right)^{-1}\left(\boldsymbol{\Phi}_t^{h\top}\left(\overline{\boldsymbol{\nu}}_t^h - \widetilde{\boldsymbol{\nu}}_t^h + \widetilde{\boldsymbol{\nu}}_t^h\right)\right) \\
&= \left(\boldsymbol{\Phi}_t^{h\top}\boldsymbol{\Phi}_t^h + \lambda I\right)^{-1}\left(\boldsymbol{\Phi}_t^{h\top}\widetilde{\boldsymbol{\nu}}_t^h\right) \\
&+ \left(\boldsymbol{\Phi}_t^{h\top}\boldsymbol{\Phi}_t^h + \lambda I\right)^{-1}\left(\boldsymbol{\Phi}_t^{h\top}(\overline{\boldsymbol{\nu}}_t^h - \widetilde{\boldsymbol{\nu}}_t^h)\right) \\
&+ \left(\boldsymbol{\Phi}_t^{h\top}\boldsymbol{\Phi}_t^h + \lambda I\right)^{-1}\left(\boldsymbol{\Phi}_t^{h\top}\boldsymbol{\Phi}_t^h + \lambda I\right)\omega^{*h} \\
&- \lambda\left(\boldsymbol{\Phi}_t^{h\top}\boldsymbol{\Phi}_t^h + \lambda I\right)^{-1}\omega^{*h}
\end{aligned}$$

therefore, for any vector $\zeta^h \in \mathbb{R}^d$

$$\begin{aligned}
\zeta^{h\top}\widehat{\omega}_t^h - \zeta^{h\top}\omega^{*h} &= \zeta^{h\top}\left(\boldsymbol{\Phi}_t^{h\top}\boldsymbol{\Phi}_t^h + \lambda I\right)^{-1}\left(\boldsymbol{\Phi}_t^{h\top}\widetilde{\boldsymbol{\nu}}_t^h\right) \\
&+ \zeta^{h\top}\left(\boldsymbol{\Phi}_t^{h\top}\boldsymbol{\Phi}_t^h + \lambda I\right)^{-1}\left(\boldsymbol{\Phi}_t^{h\top}(\overline{\boldsymbol{\nu}}_t^h - \widetilde{\boldsymbol{\nu}}_t^h)\right) \\
&- \zeta^{h\top}\lambda\left(\boldsymbol{\Phi}_t^{h\top}\boldsymbol{\Phi}_t^h + \lambda I\right)^{-1}\omega^{*h}
\end{aligned}$$

As a results, applying Cauchy-Schwarz inequality and inequalities $\|\omega^{*h}\|_{\overline{\chi}_t^{h-1}} \leq \frac{1}{\lambda(\overline{\chi}_t^h)}\|\omega^{*h}\|_2 \leq \frac{1}{\lambda}\|\omega^{*h}\|_2$ we get

$$\begin{aligned}
|\zeta^{h\top}\widehat{\omega}_t^h - \zeta^{h\top}\omega^{*h}| &\leq \|\zeta^h\|_{\overline{\chi}_t^{h-1}}\|\boldsymbol{\Phi}_t^{h\top}\widetilde{\boldsymbol{\nu}}_t^h\|_{\overline{\chi}_t^{h-1}} + \|\zeta^h\|_{\overline{\chi}_t^{h-1}}\|\boldsymbol{\Phi}_t^{h\top}(\overline{\boldsymbol{\nu}}_t^h - \widetilde{\boldsymbol{\nu}}_t^h)\|_{\overline{\chi}_t^{h-1}} + \lambda\|\zeta^h\|_{\overline{\chi}_t^{h-1}}\|\omega^{*h}\|_{\overline{\chi}_t^{h-1}} \\
&\leq \|\zeta^h\|_{\overline{\chi}_t^{h-1}}\left(\|\boldsymbol{\Phi}_t^{h\top}\widetilde{\boldsymbol{\nu}}_t^h\|_{\overline{\chi}_t^{h-1}} + \|\boldsymbol{\Phi}_t^{h\top}(\overline{\boldsymbol{\nu}}_t^h - \widetilde{\boldsymbol{\nu}}_t^h)\|_{\overline{\chi}_t^{h-1}} + \lambda^{1/2}\|\omega^{*h}\|_2\right)
\end{aligned}$$

where applying self normalized Lemma 4, for all h with probability at least $1 - H\delta$

$$|\zeta^{h\top}\widehat{\omega}_t^h - \zeta^{h\top}\omega^{*h}| \leq \|\zeta^h\|_{\overline{\chi}_t^{h-1}}\left(2\sigma\log\left(\frac{\det\left(\overline{\chi}_t^h\right)^{1/2}}{\det\left(\widetilde{\chi}\right)^{1/2}}\right) + \lambda^{1/2}L_\omega + \|\boldsymbol{\Phi}_t^{h\top}(\overline{\boldsymbol{\nu}}_t^h - \widetilde{\boldsymbol{\nu}}_t^h)\|_{\overline{\chi}_t^{h-1}}\right)$$

hold for any ζ^h. By plugging in $\zeta^h = \overline{\chi}_t^h\left(\widehat{\omega}_t^h - \omega^{*h}\right)$ we get the following;

$$\|\widehat{\omega}_t^h - \omega^{*h}\|_{\overline{\chi}_t^h} \leq \theta_t^h(\delta) = \sigma\sqrt{2\log\left(\frac{\det(\overline{\chi}_t^h)^{1/2}\det(\lambda I)^{-1/2}}{\delta}\right)} + \lambda^{1/2}L_\omega + \|\boldsymbol{\Phi}_t^{h\top}(\overline{\boldsymbol{\nu}}_t^h - \widetilde{\boldsymbol{\nu}}_t^h)\|_{\overline{\chi}_t^{h-1}}$$

For $h = H$ the last term in the above equation $\|\boldsymbol{\Phi}_t^{H\top}(\overline{\boldsymbol{\nu}}_t^H - \widetilde{\boldsymbol{\nu}}_t^H)\|_{\overline{\chi}_t^{H-1}}$ is zero therefore we have;

$$\|\widehat{\omega}_t^H - \omega^{*H}\|_{\overline{\chi}_t^H} \leq \theta_t^H(\delta) = \sigma\sqrt{2\log\left(\frac{\det(\overline{\chi}_t^H)^{1/2}\det(\lambda I)^{-1/2}}{\delta}\right)} + \lambda^{1/2}L_\omega$$

For $h < H$ we need to account for the bias introduced by $\|\boldsymbol{\Phi}_t^{h\top}(\overline{\boldsymbol{\nu}}_t^h - \widetilde{\boldsymbol{\nu}}_t^h)\|_{\overline{\chi}_t^{h-1}}$. Due to the pesimism we have

$$\phi(x_t^{h+1}, \pi^*(x_t^{h+1}))^\top\omega^{*h+1} \geq \phi(x_t^{h+1}, \check{\pi}_t(x_t^{h+1}))^\top\check{\omega}_{t-1}^{h+1}$$

also due to the optimal action of the pesimistic model we have

$$\phi(x_t^{h+1}, \pi^*(x_t^{h+1}))^\top \omega^{*h+1} - \phi(x_t^{h+1}, \check{\pi}_t(x_t^{h+1}))^\top \check{\omega}_{t-1}^{h+1} \leq \phi(x_t^{h+1}, \pi^*(x_t^{h+1}))^\top \omega^{*h+1} - \phi(x_t^{h+1}, \pi^*(x_t^{h+1}))^\top \check{\omega}_{t-1}^{h+1}$$

With high probability we have;

$$\begin{aligned}
&\|\overline{\chi}_t^{H-1^{-1/2}} \boldsymbol{\Phi}_t^{H-1^\top} (\overline{\boldsymbol{\nu}}_i^{H-1} - \widetilde{\boldsymbol{\nu}}_i^{H-1})\|_2 \\
&\leq \|\overline{\chi}_t^{H-1^{-1/2}} \boldsymbol{\Phi}_t^{H-1^\top}\|_2 \|(\overline{\boldsymbol{\nu}}_i^{H-1} - \widetilde{\boldsymbol{\nu}}_i^{H-1})\|_2 \\
&\leq \|(\overline{\boldsymbol{\nu}}_i^{H-1} - \widetilde{\boldsymbol{\nu}}_i^{H-1})\|_2 \\
&= \left(\sum_i^t \left(\phi(x_t^{h+1}, \pi^*(x_t^{h+1}))^\top \omega^{*h+1} - \phi(x_t^{h+1}, \check{\pi}_t(x_t^{h+1}))^\top \check{\omega}_{t-1}^{h+1} \right)^2 \right)^{1/2} \\
&\leq \left(\sum_i^t \left(\phi(x_t^{h+1}, \pi^*(x_t^{h+1}))^\top \omega^{*h+1} - \phi(x_t^{h+1}, \pi^*(x_t^{h+1}))^\top \check{\omega}_{t-1}^{h+1} \right)^2 \right)^{1/2} \\
&= \left(\sum_i^t \left(\phi(x_t^{h+1}, \pi^*(x_t^{h+1}))^\top \left(\omega^{*h+1} - \check{\omega}_{t-1}^{h+1} \right) \right)^2 \right)^{1/2} \\
&= \left(\sum_i^t \left(\phi(x_t^{h+1}, \pi^*(x_t^{h+1}))^\top \overline{\chi}_t^{h+1^{-1/2}} \overline{\chi}_t^{h+1^{1/2}} \left(\omega^{*h+1} - \check{\omega}_{t-1}^{h+1} \right) \right)^2 \right)^{1/2} \\
&\leq \left(\sum_i^t \left(\|\phi(x_t^{h+1}, \pi^*(x_t^{h+1}))\|_{\overline{\chi}_t^{h+1^{-1}}} \theta_t^{h+1}(\delta) \right)^2 \right)^{1/2} \\
&= \theta_t^{h+1}(\delta) \left(\sum_i^t \|\phi(x_t^{h+1}, \pi^*(x_t^{h+1}))\|^2_{\overline{\chi}_t^{h+1^{-1}}} \right)^{1/2} \\
&\leq \theta_t^{h+1}(\delta) \sqrt{\rho_\lambda^{h+1}}
\end{aligned}$$

Therefore

$$\|\widehat{\omega}_t^h - \omega^{*h}\|_{\overline{\chi}_t^h} \leq \theta_t^h(\delta) = \sigma \sqrt{2 \log \left(\frac{\det(\overline{\chi}_t^h)^{1/2} \det(\lambda I)^{-1/2}}{\delta} \right)} + \lambda^{1/2} L_\omega + \theta_t^{h+1}(\delta) \sqrt{\rho_\lambda^{h+1}}$$

The $\det(\overline{\chi}_t^h)$ can be written as $\det(\overline{\chi}_t^h) = \prod_j^d \alpha_j$ therefore, $trace(\overline{\chi}_t) = \sum_j^d \alpha_j$. We also know that;

$$(\prod_j^d \alpha_j)^{1/d} \leq \frac{\sum_j^d \alpha_j}{d}$$

A matrix extension to Lemma 10 and 11 in (Abbasi-Yadkori et al., 2011) results in $\det(\overline{\chi}_t^h) \leq \left(\frac{trace(\overline{\chi}_t^h)}{d} \right)^d$ while we have $trace(\overline{\chi}_t^h) = trace(\lambda I) + \sum_i^t trace\left({\phi^h}_i^\top \phi^h{}_i \right) \leq d\lambda + tL$,

$$\det(\overline{\chi}_t^h) \leq \left(\frac{d\lambda + tL}{d} \right)^d \tag{14}$$

therefore the main statement of Lemma 1 goes through. □

Proof. Lemma 2 We have the following for the determinant of $\det\left(\overline{\chi}_T^h\right)$ through first matrix determinant lemma and second Sylvester's determinant identity;

$$\begin{aligned}\det\left(\overline{\chi}_T^h\right) &= \det\left(\overline{\chi}_{T-1} + \phi_T^h {\phi_T^h}^\top\right) \\ &= \det\left(1 + {\phi_T^h}^\top {\overline{\chi}_{T-1}^h}^{-1} \phi_T^h\right) \det\left(\overline{\chi}_{T-1}^h\right) \\ &= \prod_t^T \det\left(1 + \|\phi_t^h\|^2_{\overline{\chi}_t^{h\,-1}}\right) \det\left(\widetilde{\chi}\right)\end{aligned}$$

Using the fact that $\log(1+\vartheta) \leq \vartheta$ we have

$$\sum_t^T \log\left(1 + \|\phi_t^h\|^2_{\overline{\chi}_t^{h\,-1}}\right) \leq \left(\log\left(\det\left(\overline{\chi}_T^h\right)\right) - \log\left(\det\left(\widetilde{\chi}^h\right)\right)\right) \leq d\log(\lambda + TL^2/d) \tag{15}$$

and the statement follows.

□