Abstract

While off-policy temporal difference (TD) methods have widely been used in reinforcement learning due to their efficiency and simple implementation, their Bayesian counterparts have not been utilized as frequently. One reason is that the non-linear max operation in the Bellman optimality equation makes it difficult to define conjugate distributions over the value functions. In this paper, we introduce a novel Bayesian approach to off-policy TD methods, called as ADFQ, which updates beliefs on state-action values, Q, through an online Bayesian inference method known as Assumed Density Filtering. In order to formulate a closed-form update, we approximately estimate analytic parameters of the posterior of the Q-beliefs. Uncertainty measures in the beliefs not only are used in exploration but also provide a natural regularization for learning. We show that ADFQ converges to Q-learning as the uncertainty measures of the Q-beliefs decrease. ADFQ improves common drawbacks of other Bayesian RL algorithms such as computational complexity. We also extend ADFQ with a neural network. Our empirical results demonstrate that the proposed ADFQ algorithm outperforms comparable algorithms on various domains including continuous state domains and games from the Arcade Learning Environment.

1 Introduction

Bayesian reinforcement learning (BRL) is a classic reinforcement learning (RL) technique that utilizes Bayesian inference to integrate new experiences with prior information about the problem in a probabilistic distribution. It explicitly quantifies the uncertainty of the learning parameters unlike standard RL approaches which do not properly account for uncertainty in the parameters. Explicit quantification of the uncertainty can help guide policies that consider the exploration-exploitation trade-off by exploring actions with higher uncertainty more often. Moreover, it can also regularize posterior updates by properly accounting for uncertainty.

Motivated by these potential advantages, a number of algorithms have been proposed in both model-based BRL [7, 28, 9, 15, 26] and model-free BRL [6, 10, 11, 12, 5, 13]. However, Bayesian approaches to off-policy temporal difference (TD) learning has been less well-studied compared to other methods due to difficulty in handling the max non-linearity in the Bellman optimality equation. Yet off-policy TD methods have been widely used in standard RL, including extensions integrating neural network function approximations such as Deep Q-Networks (DQN) [22, 23]. One recent influential algorithm for Bayesian off-policy TD learning is KTD-Q, an extension of Kalman Temporal Difference (KTD) [12]. KTD approximates the value function using the Kalman filtering scheme, and handles the non-linearity in the Bellman optimality equation by applying the Unscented Transform. Although the KTD framework is able to integrate some important features in RL, it requires numerous hyperparameters and is difficult to extend to function approximation methods with large numbers of parameters due to its high computational complexity.

Another field of probabilistic approaches to RL is Distributional RL which learns a value distribution or a return density function from the distributional Bellman equation. Recent work [2] proposed a
gradient-based categorical algorithm using a distributional perspective and showed the state-of-the-art performance in several games from the Arcade Learning Environment (ALE). The probabilistic approach to the value function is similar to our approach, however, the value distribution in their work represents the distribution of the random return that a learning agent receives, while the Q-belief defined in our approach is a belief distribution of a learning agent on a certain state-action pair. As we show in the experiments, we are able to utilize the uncertainty measures in the exploration, while only ε-greedy is used in their experiments.

In this paper, we introduce a novel approximate Bayesian off-policy TD learning algorithm, which we denote as ADFQ, that updates beliefs for Q (action-value function) and approximates their posteriors using an online Bayesian inference algorithm known as assumed density filtering (ADF). We handle the difficulty in finding a conjugate prior for the Bellman equation using ADF. In order to reduce the computational burden of estimating parameters of the approximated posterior, we propose a method to analytically estimate the parameters. Unlike Q-learning, the ADFQ update rule considers all possible actions for the next state and returns a soft-max behavior and regularization using the uncertainty measures of the Q-beliefs. This can alleviate the instability of the greedy update discussed by [16][29]. We prove the convergence of ADFQ to the optimal Q-values by showing that ADFQ becomes identical to Q-learning as all state and action pairs are visited infinitely often. In addition, ADFQ has better computational complexity compared with other BRL algorithms.

We implement ADFQ in a small discrete domain and then show how it can be extended to continuous or large discrete state environments using a neural network. There are previous works that implement Bayesian approaches to RL by using uncertainty in the neural network weights[1][24]. Our method differs in that it explicitly computes the variances of the Q-values and can use the them in exploration and the value update. In our experiments, ADFQ outperforms not only Q-learning and KTD-Q in tabular settings, but also DQN, and Double DQN ([17]) in large or continuous domains. Particularly, it showed dramatic improvements in a stochastic domain and domains with a large action set.

2 Background

2.1 Assumed Density Filtering

Assumed Density Filtering (ADF) is a general technique for approximating the true posterior with a tractable parametric distribution in Bayesian networks. It has been independently rediscovered for a number of applications and is also known as moment matching, online Bayesian learning, and weak marginalization [25][3][21]. Suppose that a hidden variable x follows a tractable parametric distribution p(x|θt) where θt is a set of parameters at time t. In the Bayesian framework, the distribution can be updated after observing some new data (Dt) using Bayes’ rule, p(x|θt, Dt) ∝ p(Dt|x, θt)p(x|θt). In online settings, a Bayesian update is typically performed after a new data point is observed, and the updated posterior is then used as a prior for the following iteration.

When the posterior computed by Bayes’ rule does not belong to the original parametric family, it can be approximated by a distribution belonging to the parametric family. In ADF, the posterior is projected onto the closest distribution in the family chosen by minimizing the reverse Kullback-Leibler divergence denoted as KL(ˆp||p) where ˆp is the original posterior distribution and p is a distribution in a parametric family of interest. Thus, for online Bayesian filtering, the parameters for the ADF estimate is given by θt+1 = argminθ KL(ˆp(θt, Dt)||p(θ)).

2.2 Q-learning

RL problems can be formulated in terms of a Markov Decision Process (MDP) described by the tuple, M = (𝒮, 𝒜, P, R, γ) where 𝒮 and 𝒜 are the state and action spaces, respectively, P : 𝒮 × 𝒜 × 𝒮 → [0, 1] is the state transition probability kernel, R : 𝒮 × 𝒜 → ℜ is a reward function, and γ ∈ [0, 1) is a discount factor. The value function is defined as Vπ(s) = Eπ[∞ ∑ t=0 γt r(s, a)] for all s ∈ 𝒮, the expected value of cumulative future rewards starting at a state s and following a policy π thereafter. The state-action value (Q) function is defined as the value for a state-action pair, Qπ(s, a) = Eπ[∞ ∑ t=0 γt r(s, a)] for all s ∈ 𝒮, a ∈ 𝒜. The objective of a learning agent in RL is to find an optimal policy π∗ = argmaxπ Vπ. Finding the optimal values, V∗(s) and Q∗(s, a), requires solving the Bellman optimality equation:

\[ Q^*(s, a) = \mathbb{E}_{s' \sim P(\cdot|s, a)}[R(s, a) + \gamma \max_{a' \in A} Q^*(s', a')] \quad (1) \]
We define where \( \phi \) is the standard Gaussian probability density function (PDF) and \( \Phi(\cdot) \) is the standard Gaussian cumulative distribution function (CDF) (derivation details are provided in Appendix A).

According to the Bellman optimality equation in Eq. 1, we can define a random variable for TD target, \( R(s, a) + \gamma \max_b Q(s', b) \), and the current \( Q(s, a) \) with a learning rate \( \alpha \in [0, 1] \) as shown below:

\[
Q(s, a) \leftarrow Q(s, a) + \alpha \left( R(s, a) + \gamma \max_b Q(s', b) - Q(s, a) \right)
\]

3 Bayesian Q-learning with Assumed Density Filtering

3.1 Belief Updates on \( Q \)

We define \( Q_{s,a} \), as a Gaussian random variable with mean \( \mu_{s,a} \) and variance \( \sigma_{s,a}^2 \) corresponding to the action value function \( Q(s, a) \) for \( s \in S \) and \( a \in A \). We assume that the random variables for different states and actions are independent and have different means and variances, \( Q_{s,a} \sim N(\mu_{s,a}, \sigma_{s,a}^2) \) where \( \mu_{s,a} \neq \mu_{s',a} \) if \( s \neq s' \) or \( a \neq a' \) \( \forall s \in S, \forall a \in A \).

According to the Bellman optimality equation in Eq. 1, \( V(s) = \max_a Q_{s,a} \), and \( V(s) \) is a set of mean and variance of \( Q_{s,a} \). In general, the probability density function for the maximum of Gaussian random variables, \( M = \max_{1 \leq k \leq N} X_k \) where \( X_k \sim N(\mu_k, \sigma_k^2) \), is no longer Gaussian:

\[
p(M = x) = \sum_{i=1}^{N} \frac{1}{\sigma_i} \phi \left( \frac{x - \mu_{i}}{\sigma_{i}} \right) \prod_{j \neq i}^{N} \Phi \left( \frac{x - \mu_{j}}{\sigma_{j}} \right)
\]

where \( \phi(\cdot) \) is the standard Gaussian probability density function (PDF) and \( \Phi(\cdot) \) is the standard Gaussian cumulative distribution function (CDF) (derivation details are provided in Appendix A).

For one-step Bayesian TD learning, the beliefs on \( Q = \{Q_{s,a}\}_{s \in S, a \in A} \) can be updated at time \( t \) after observing a reward \( r_t \) and the next state \( s_{t+1} \) using Bayes’ rule. In order to reduce notation, we drop the dependency on \( t \) denoting \( s_t = s, q_t = a, s_{t+1} = s', r_t = r \), yielding the causally related 4-tuple \( \tau = < s, a, r, s' > \). For deterministic MDPs (\( \mathcal{P} : S \times A \times S \to \{0, 1\} \)), we use the one-step TD target, \( r + \gamma V_{s'} \) to give the likelihood distribution, \( p(r + \gamma V_{s'} | q, \theta) = p_{V_{s'}}((q - r)/\gamma | s', q, \theta) \) where \( q \) is a value corresponding to \( Q_{s,a} \) and \( \theta \) is a set of mean and variance of \( Q \). For stochastic MDPs, we add small Gaussian white noise to the likelihood, \( r + \gamma V_{s'} + W \) where \( W \sim N(0, \sigma^2_w) \), and \( p(r + \gamma V_{s'} | q, \theta) = \int_{w} p(r + \gamma V_{s'} + w | q, \theta) p_W(w) \mathrm{d}w \). We will first derive the belief updates on Q-values in deterministic MDPs and then extend the result to the stochastic case. From the independence assumptions on \( Q \), the posterior update is reduced to an update for the belief on \( Q_{s,a} \):

\[
\hat{p}_{Q_{s,a}}(q | \theta, r, s') \propto p_{V_{s'}} \left( \frac{q - r}{\gamma} \bigg| q, s', \theta \right) p_{Q_{s,a}}(q | \theta)
\]

According to the Bellman optimality in Eq. 1 \( V_{s'} \) follows the distribution presented in Eq. 2. The resulting posterior distribution is given as follows (derivation details in Appendix B):

\[
\hat{p}_{Q_{s,a}}(q | \theta, r, s') = \frac{1}{Z} \sum_{b \in A} \frac{c_{\tau,b}}{\sigma_{\tau,b}} \phi \left( \frac{q - \bar{\mu}_{\tau,b}}{\sigma_{\tau,b}} \right) \prod_{b' \in A \setminus b} \Phi \left( \frac{q - (r + \gamma \mu_{s',b'})}{\gamma \sigma_{s',b'}} \right)
\]

where \( \frac{1}{Z} \sum_{b \in A} \frac{c_{\tau,b}}{\sigma_{\tau,b}} \phi \left( \frac{r + \gamma \mu_{s',b'}}{\gamma \sigma_{s',b'}} \right) \) is a normalization constant and

\[
c_{\tau,b} = \frac{1}{\sqrt{\sigma_{s,a}^2 + \gamma^2 \sigma_{s',b}^2}} \phi \left( \frac{(r + \gamma \mu_{s',b'}) - \mu_{s,a}}{\sqrt{\sigma_{s,a}^2 + \gamma^2 \sigma_{s',b}^2}} \right)
\]

\[
\bar{\mu}_{\tau,b} = \frac{\mu_{s,a}}{\sigma_{s,a}^2} \left( \frac{\mu_{s,a}}{\sigma_{s,a}^2} + \frac{r + \gamma \mu_{s',b'}}{\gamma^2 \sigma_{s',b}^2} \right) = \frac{1}{\sigma_{s,a}^2} + \frac{1}{\gamma^2 \sigma_{s',b}^2}
\]

Note that all next actions are considered in Eq. 3 unlike the conventional Q-learning update which only considers the subsequent action resulting in the maximum Q-value at the next step (\( \max_b Q(s', b) \)).
Figure 1: An example of the belief update in Eq.3 when $|A| = 3$, $\gamma = 0.9$ and prior (+ green) has $\mu_{s,a} = 0.0, \sigma_{s,a}^2 = 0.5$. Each column corresponds to a subsequent state and action pair, (a) $b = 1$: $\mu_{s',b} = 1.0, \sigma_{s',b}^2 = 2.0$, (b) $b = 2$: $\mu_{s',b} = 1.0, \sigma_{s',b}^2 = 0.1$, (c) $b = 3$: $\mu_{s',b} = 4.5, \sigma_{s',b}^2 = 0.1$.

This can lead to a more stable update rule as updating with only the maximum Q-value has inherent instability [16, 29]. The Bayesian update considers the scenario where the true maximum Q-value may not be the one with the highest estimated mean, and weights each subsequence Q-value accordingly. Each term for action $b$ inside the summation in Eq.3 has three important features. First of all, $\bar{\mu}_{\tau,b}$ is an inverse-variance weighted (IVW) average of the prior mean and the TD target mean. Therefore, the Gaussian PDF part becomes closer to the TD target distribution if it has a lower uncertainty than the prior, and vice versa as compared in the first row (a) and (b) of Fig.1. Next, the TD error, $\delta_{\tau,b} = (r + \gamma \mu_{s',b}) - \mu_{s,a}$, is naturally incorporated in the posterior distribution with the form of a Gaussian PDF in the weight $c_{\tau,b}$. Thus, a subsequent action which results in a smaller TD error contributes more to the update. The sensitivity of a weight value is determined by the prior and target uncertainties. An example case is described in the second row of Fig.1 where $\delta_{\tau,1} > \delta_{\tau,2} > \delta_{\tau,3}$ and $\sigma_{s',1} > \sigma_{s',2} = \sigma_{s',3}$. Finally, the product of Gaussian CDFs provides a soft-max operation. The red curve with dots in the third row of Fig.1 represents $\prod_{b' \neq b} \Phi(q | r + \gamma \mu_{\tau,b'} - \gamma \sigma_{\tau,b'})$ for each $b$. For a certain $q$ value (x-axis), the term returns a larger value for a larger $\mu_{s',b}$ as seen in the black circles.

### 3.2 Assumed Density Filtering on Q-Belief Updates

The posterior distribution in Eq.3, however, is no longer Gaussian. In order to continue the Bayesian update, we approximate the posterior with a Gaussian distribution using ADF. When the parametric family of interest is spherical Gaussian, it is shown that the ADF parameters are obtained by matching moments. Thus, the mean and variance of the approximate posterior are given by those of the true posterior, $\mathbb{E}_{q \sim \hat{p}_{Q,s,a}()}[q]$ and $\text{Var}_{q \sim \hat{p}_{Q,s,a}()}[q]$, respectively. It is fairly easy to analytically derive the posterior mean and variance when $|A| = 2$. The derivation is presented in Appendix C. However, to our knowledge, there is no closed-form expression for them when $|A| > 2$.

Similarly, we update the means and variances of Q using ADF for stochastic MDPs. However, the integral in the expected likelihood, $\int_{\mathcal{R}} p(r + \gamma V_{s'} + w | q, \theta) p_W(w) dw$, doesn’t have a closed-form expression in general except when $|A| = 2$ (see Appendix D). Thus, analytic solutions for the ADF parameters are not available. In the next sections, we prove the convergence of the means to the optimal Q-values for the case $|A| = 2$. Then, we show how to derive an analytic approximation for the ADF parameters which becomes exact in the small variance limit.

### 3.3 Convergence to Optimal Q-values

The convergence theorem of the Q-learning algorithm has previously been proven [31]. We, therefore, show that the online Bayesian update using ADF with the posterior in Eq.3 converges to Q-learning when $|A| = 2$. We apply an approximation from Lemma 1 in order to prove Theorem 1. Proofs for Lemma 1 and Theorem 1 are presented in Appendix E.
the smaller the variance of the next state (the higher the confidence), the more

When $|A| > 2$, the update can be solved by numerical approximation of the true posterior mean and variance using a number of samples. However, its computation becomes unwieldy due to the large number of samples needed for accurate estimates. This becomes especially problematic with small variances as the number of visits to corresponding state-action pairs grows. Therefore, in this section, we show how to accurately estimate the ADF parameters using an analytic approximation. This estimate becomes exact for small variances.

### 4 Analytic ADF Parameter Estimates

When $|A| > 2$, the update can be solved by numerical approximation of the true posterior mean and variance using a number of samples. However, its computation becomes unwieldy due to the large number of samples needed for accurate estimates. This becomes especially problematic with small variances as the number of visits to corresponding state-action pairs grows. Therefore, in this section, we show how to accurately estimate the ADF parameters using an analytic approximation. This estimate becomes exact for small variances.

#### 4.1 Analytic Approximation of Posterior

Using Lemma 1, the true posterior in Eq. 3 is approximated as the following distribution:

$$
\hat{p}_{\hat{q}_{s,a}}(q) = \frac{1}{Z} \sum_{b \in A} \frac{c_{r,b}}{\sqrt{2\pi}\hat{\sigma}_{r,b}} \exp \left\{ -\frac{(q - \hat{\mu}_{r,b})^2}{2\hat{\sigma}_{r,b}^2} - \sum_{b' \neq b} \frac{|r' + \gamma q_{s,a'} - q|^2_{b'}}{2\gamma^2\hat{\sigma}_{s,a'}^2} \right\}
$$

Each term for $b \in A$ inside the summation can then be approximated by a Gaussian PDF. Similar to Laplace’s method, we approximate each term as a Gaussian distribution by matching the maximum
We find \( \mu_m \) where

\[ H \]

The final approximated distribution is a Gaussian mixture model with \( \mu_m \)

The computational complexity of each update of the algorithm is \( O(2|S||A|) \) terms provides a softened maximum property over \( b \). As shown in Eq.11, it has the TD error penalizing term, \( c_{\tau,b} \), but also penalizes how far \( \mu^* \) is shifted towards larger TD target distributions. Moreover, the remaining terms provide a softened maximum property over \( b \). The final algorithm is summarized in Table 1.

4.2 Approximate Likelihood for Stochastic MDPs

In an asymptotic limit of \( \sigma_{\omega} / \sigma_{s',b} \to 0 \), \( \forall b \in A \) and \( |A| = 2 \), the likelihood distribution for stochastic MDPs is similar to that of the deterministic case but the variance of its Gaussian PDF term

Table 1: ADFQ algorithm

| Algorithm 1: ADFQ |
|-------------------|
| Initialize \( \mu_{s,a}, \sigma_{s,a} \) \( \forall s \in S \) and \( \forall a \in A \) |
| for each time step \( t \) do |
| \( a_t \sim \pi_{\text{action}}(s_t; \theta_t) \) |
| Perform the action and observe \( r_t \) and \( s_{t+1} \) |
| for each \( b \in A \) |
| Compute \( \mu_b^*, \sigma_b^*, k_b^* \) using Eq.9 [11] |
| Update \( \mu_{s_t,a_t} \) and \( \sigma_{s_t,a_t} \) using Eq.13 |
| |
We first examined the convergence to the optimal Q-values using randomly generated fixed trajectories. We test our algorithms in Maze (σ MDPs, α µ (or means) and the true optimal Q-values, and plotted the averaged results over 10 trials in Fig. 3. (denoted as ADFQ-Numeric). For both ADFQ and ADFQ-Numeric, we use two action policies: The ADFQ update on the mean

\[
\sum_{b \in A} \frac{\gamma}{\sqrt{\gamma^2 \sigma^2_{s',b} + \sigma_w^2}} \phi \left( \frac{q - (r + \gamma \mu_{s',b})}{\gamma \sigma_{s',b}} \right) \prod_{b' \neq b, b' \in A} \phi \left( \frac{q - (r + \gamma \mu_{s',b'})}{\gamma \sigma_{s',b'}} \right)
\]

Extending this result to the general case (|A| = n for n ∈ N), the posterior distribution, \( \hat{p}_{Q_{\tau,a}}(q) \), for stochastic MDPs is same with Eq.3 but \( \gamma^2 \sigma^2_{s',b} \) is replaced by \( \gamma^2 \sigma^2_{s',b} + \sigma_w^2 \) in \( \epsilon \), \( \mu \), and \( \sigma \). Therefore, \( \mu^*, \sigma^*, \) and \( k^*_b \) in the ADFQ algorithm (Table.1) are also changed accordingly.

4.3 Convergence of ADFQ

Theorem 1 extends to the ADFQ algorithm. The contraction behavior of the variances in the case of Theorem 1 is also empirically observed in ADFQ (Proof in Appendix E).

**Theorem 2.** The ADFQ update on the mean \( \mu_{s,a} \) ∀s ∈ S, ∀a ∈ A for |A| = 2 is equivalent to the Q-learning update if the variances approach 0 and if all state-action pairs are visited infinitely often. In other words, we have:

\[
\lim_{k \to \infty} \mu_{s,a;k+1} = (1 - \alpha_{\tau;k}) \mu_{s,a;k} + \alpha_{\tau;k} \left( r + \gamma \max_{b \in A} \mu_{s',b;k} \right)
\]

where \( \alpha_{\tau;k} = \frac{\sigma^2_{s,a;k}}{\left( \sigma^2_{s,a;k} + \gamma^2 \sigma^2_{s',b+k} + \sigma_w^2 \right)} \) and \( b^* = \arg\max_{b \in A} \mu_{s',b} \). For deterministic MDPs, \( \sigma_w = 0 \).

As we have observed the behavior of \( \alpha_{\tau} \) in Theorem 1 the learning rate \( \alpha_{\tau} \) again provides a natural learning rate with the ADFQ update. We can therefore think of Q-learning as a special case of ADFQ.

5 Experiments in a Discrete MDP

5.1 Algorithms and Domain

In addition to ADFQ, we evaluate a numerical approximation of the mean and variance of Eq.3 (denoted as ADFQ-Numeric). For both ADFQ and ADFQ-Numeric, we use two action policies: Bayesian Sampling (BS) selects \( a_t = \arg\max_a q_{s_t,a} \) where \( q_{s_t,a} \sim p_{Q_{\tau,a}}(\cdot|\theta_t) \), and \( \epsilon \)-greedy selects a random action with \( \epsilon \) probability and selects the action with the highest mean otherwise. In implementation, we fixed the initial variance to 100.0 and the variances are bounded by \( 10^{-10} \) since their values dramatically drop and eventually exceed the precision range of computers. For a stochastic case, we used a noise (\( \sigma_w^2 = 10^{-5} \)) and experience replay [20] with a batch size of 30.

For comparison, we test Q-learning with \( \epsilon \)-greedy and Boltzmann action policies. The learning rate decreases as the number of visits to a state-action pair increases \( \alpha_t = \alpha_0/(n_0 + t) \), \( \alpha_0 = 0.5 \) [19]. KTD-Q with \( \epsilon \)-greedy and its active learning scheme are also examined. The same hyperparameter values as the ones in the original paper are used if presented. All other hyperparameters are selected through cross-validation (presented in Appendix F).

We test our algorithms in Maze (\( \gamma = 0.95 \), Figure 3) from [6] with/without stochasticity in finite learning steps (\( T_H = 30000 \)). Since the KTD-Q algorithm was not able to handle a large discrete state space in reasonable time due to its high computational complexity, we reduced the state space to |\( S \)| = 112. The agent’s goal is to collect the flags "F" and escape the maze through the goal position "G" starting from "S". It receives a reward equivalent to the number of flags it has collected at "G". The agent remains at the current state if it performs an action toward a wall (black block). For a stochastic case, the agent slips with a probability 0.1 and moves to the right perpendicular direction.

5.2 Results

We first examined the convergence to the optimal Q-values using randomly generated fixed trajectories \(< s_0, a_0, r_0, s_1, \ldots >\) for all algorithms in order to evaluate only the update part of each algorithm. During learning, we computed the root mean square error (RMSE) between the estimated Q-values (or means) and the true optimal Q-values, and plotted the averaged results over 10 trials in Fig. 5.
Next, we evaluated the performance of each algorithm with different action policies during learning. At every 300 steps, the current policy was greedily evaluated where the maximum number of steps was bounded by 1.5 times of the optimal path length or it was terminated when the goal was reached. The entire experiment was repeated 10 times and the results were averaged.

As shown in Fig. 3, ADFQ converged to the optimal Q-values quicker than all other algorithms including Q-learning. In addition, ADFQ with $\epsilon$-greedy and ADFQ with BS showed similar results and converged to the optimal performance (3.0) faster than the comparing algorithms in both deterministic and stochastic cases. Q-learning with $\epsilon$-greedy learned an optimal policy almost as fast as ADFQ in the deterministic case, but the performance of ADFQ was improved dramatically in the stochastic case. KTD-Q diverged and performed poorly since its derivative-free approximation nature does not scale well with the number of parameters. In Appendix G, we show good performance of KTD-Q and its converging behavior in a small domain.

ADFQ-Numeric initially resulted in a large jump in RMSE and learned very little. This can be explained by the fact that the mean of the maximum of Gaussian random variables is equal to or larger than the maximum of means of Gaussian random variables (i.e. $\mathbb{E}[M] = \max_{i=1 \ldots N} X_i \geq \max_{i=1 \ldots N} \mathbb{E}[X_i]$). While this can speed up learning in a certain type of domain, it impedes learning in Maze. When the agent performs an upward action, the agent receives no reward and remains at the current position. In this step, ADFQ-Numeric increases the Q-value while Q-learning and ADFQ decreases it. This results in the dramatic increase in the RMSE plot and requires more samples to find an optimal policy. ADFQ reduces this amount through the small-variance approximation.

6 ADFQ with Neural Networks

In this section, we extend our algorithm to a continuous or large state space environment with neural networks similar to Deep Q-Networks (DQN) proposed in [22, 23]. In the Deep ADFQ model with network parameters $\xi$, the output of the network is mean $\mu(s, a; \xi)$ and variance $\sigma^2(s, a; \xi)$ of each action for a given state $s$ as shown in Fig.4. In practice, we use $-\log(\sigma_{s,a})$ instead of $\sigma^2_{s,a}$ for the output in order to ensure positive values for the variance. As in DQN, we have a train network($\xi$) and a target network($\xi'$). Mean and variance values for $s$ and $s'$ from the target network are used as inputs into the ADFQ algorithm to compute the desired mean, $\mu^{ADFQ}$, and standard deviation,
ADFQ for the train network. We used experience replay (prioritized \([27]\) for Atari games) and a combined Huber loss functions of mean and variance.

In order to demonstrate the effectiveness of our algorithm, we tested on continuous state domains, CartPole and Acrobot, and on Atari games, Breakout(\(|A| = 4\), Pong(\(|A| = 6\)), Asterix(\(|A| = 9\), Enduro(\(|A| = 9\) from the OpenAI gym simulator \([4]\). For baselines, we used DQN and Double DQN (DDQN) with experience replay (prioritized for Atari games) implemented in OpenAI baselines \([8]\) with their default hyperparameters except for setting \(\gamma = 0.99\) for all tasks. We used \(\epsilon\)-greedy action policy with \(\epsilon\) annealed from 1.0 to 0.01 (to 0.02 for CartPole and Acrobot) for the baselines as well as ADFQ. Additionally, we examined Bayesian Sampling (BS) for ADFQ. Further details on the network architecture are provided in Appendix \([F]\). The algorithms were evaluated for 200K training steps in the continuous domains and for 5M frames (1.25M training steps) in the Atari games. Similar to the previous evaluation, each learning was greedily evaluated at every epoch for 5 times bounded by 10K steps in each trial, and their averaged results are presented in Fig.5. The entire experiment was repeated for 5 and 3 random seeds for the continuous domains and the Atari games, respectively. Rewards were normalized to \([-1, 0, 1]\) and different from raw scores of the games.

In all Atari games, ADFQ with BS showed dramatic increases in its performance at the beginning. The variance estimates may help in the initial learning stages as it can trust certain state action pairs heavily and update aggressively towards them. It also notably surpassed the other algorithms in the domains where accounting for uncertainty in exploration is more advantageous (the large number of actions). Particularly, in Enduro, ADFQ with Bayesian sampling achieved the near optimal performance within 1M frames and resulted a raw score of up to 7,181 (a video is attached)! This is
very impressive compared to the raw scores of the other state-of-the-art results after 200M training frames (Categorical DQN: 3,454, Prioritized Dueling Architecture: 2,306.4).

7 Discussion

We proposed an approach to Bayesian off-policy TD method called ADFQ. ADFQ surpassed the performance of Q-learning and KTD-Q in a small finite domain, and outperformed DQN and Double DQN in various continuous and large discrete domains. The presented ADFQ algorithm demonstrates several intriguing results.

Non-greedy Update. Unlike the conventional Q-learning algorithm, ADFQ incorporates the information of all possible actions for the subsequent state in the update with weights depending on TD errors and uncertainty measures (Eq.5 and Eq.13).

Regularization with uncertainty. ADFQ provides an intuitive update - a state-action pair with higher uncertainty in its $Q$ belief has a smaller weight contributing less to the update. Therefore, we make use of our uncertainty measures not only in exploration but also in the value update with natural regularization based on the current beliefs.

Convergence to Q-learning. We prove that ADFQ converges to Q-learning as the variances decrease and can be seen as a more general form of Q-learning.

Improved drawbacks of BRL. One of the major drawbacks of BRL approaches is their higher computational complexity than standard RL algorithms [14]. ADFQ is computationally more efficient than KTD-Q and requires only two hyperparameters to be chosen.

Scalability ADFQ is extended to Deep ADFQ with a neural network. It demonstrates that it makes use of the uncertainty information especially when the number of available actions is large and reasonable exploration is required.

We would like to highlight the fact that ADFQ is a Bayesian counterpart of Q-learning and is orthogonal to most other advancements made in Deep RL. ADFQ merely changes the loss function and we compare with basic architectures here to provide insight as to how it may improve the performance. ADFQ can be used in conjunction with other extensions and techniques such as Double DQN, multistep returns, and Dueling Architecture [30].

References

[1] K. Azizzadenesheli, E. Brunskill, and A. Anandkumar. Efficient exploration through bayesian deep q-networks. arXiv preprint arXiv:1802.04412, 2018.
[2] M. G. Bellemare, W. Dabney, and R. Munos. A distributional perspective on reinforcement learning.
[3] X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, Berkeley, CA, 1998.
[4] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym, 2016.
[5] G. Chowdhary, M. Liu, R. Grande, T. Walsh, J. How, and L. Carin. Off-policy reinforcement learning with gaussian process. IEEE/CAA Journal of Automatica Sinica, 1(3):227–238, 2014.
[6] R. Dearden, N. Friedman, and S. Russell. Bayesian q-learning. In AAAI/IAAI, pages 761–768, 1998.
[7] R. Dearden, N. Friedman, and D. Andre. Model based bayesian exploration. In Proceedings of the 15th conference on Uncertainty in artificial intelligence, pages 150–159. Morgan Kaufmann Publishers Inc., 1999.
[8] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu. Openai baselines. https://github.com/openai/baselines, 2017.
[9] M. Duff. Optimal learning: Computational procedures for bayes-adaptive markov decision processes. PhD thesis, University of Massachusetts, Amherst, 2002.
[10] Y. Engel, S. Mannor, and R. Meir. Bayes meets bellman: The gaussian process approach to temporal difference learning. In *Proceedings of the 20th International Conference on Machine Learning*, volume 20, 2003.

[11] Y. Engel, S. Mannor, and R. Meir. Reinforcement learning with gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 201–208, 2005.

[12] M. Geist and O. Pietquin. Kalman temporal differences. *Journal of artificial intelligence research*, 39:483–532, 2010.

[13] M. Ghavamzadeh and Y. Engel. Bayesian policy gradient algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, pages 457–464, 2006.

[14] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar. Bayesian reinforcement learning: A survey. *Foundation and Trends in Machine Learning*, 8(5-6):359–483, 2015.

[15] A. Guez, D. Silver, and P. Dayan. Efficient bayes-adaptive reinforcement learning using sample-based search. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1071–1079, 2012.

[16] A. Harutyunyan, M. G. Bellemare, T. Stepleton, and R. Munos. $Q(\rho)$ with off-policy corrections. In *International Conference on Algorithmic Learning Theory*, pages 305–320. Springer, 2016.

[17] H. V. Hasselt. Double q-learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[18] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

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

[20] L.-J. Lin. Reinforcement learning for robots using neural networks. Technical report, Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, 1993.

[21] P. S. Maybeck. Stochastic models, estimation and control. *Academic Press*, chapter 12.7, 1982.

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS) Deep Learning Workshop*, 2013.

[23] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7549):529–533, 2015.

[24] B. O’Donoghue, I. Osband, R. Munos, and V. Mnih. The uncertainty bellman equation and exploration. *arXiv preprint arXiv:1709.05380*, 2017.

[25] M. Opper. A bayesian approach to online learning. *On-Line Learning in Neural Networks*, 1999.

[26] P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete bayesian reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*, volume 20, pages 697–704, 2006.

[27] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[28] M. Strens. A bayesian framework for reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 943–950, 2000.

[29] J. N. Tsitsiklis. On the convergence of optimistic policy iteration. *Journal of Machine Learning Research*, 3(Jul):59–72, 2002.

[30] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, 2015.

[31] C. J. Watkins and P. Dayan. Q-learning. In *Machine Learning*, pages 279–292, 1992.
Appendix

A Maximum of Gaussian Random Variables

Let $X_i$ follows a Gaussian distribution, $\mathcal{N}(\mu_i, \sigma_i^2)$, and $\mu_i \neq \mu_j, \sigma_i \neq \sigma_j$ for any $i \neq j$. The distribution of the maximum of independent Gaussian random variables is derived as follows:

$$Pr\left(\max_{1 \leq i \leq N} X_i \leq x\right) = \prod_{i=1}^{N} Pr(X_i \leq x) = \prod_{i=1}^{N} \Phi\left(\frac{x - \mu_i}{\sigma_i}\right)$$

$$p\left(\max_{1 \leq i \leq N} X_i = x\right) = \frac{d}{dx} \left(Pr\left(\max_{1 \leq i \leq N} X_i \leq x\right)\right) = \sum_{i=1}^{N} \frac{1}{\sigma_i} \phi\left(\frac{x - \mu_i}{\sigma_i}\right) \prod_{i \neq j} \Phi\left(\frac{x - \mu_j}{\sigma_j}\right) \neq \text{Gaussian} \quad (15)$$

where $\phi(\cdot)$ is the standard Gaussian probability density function (PDF) and $\Phi(\cdot)$ is the standard Gaussian cumulative distribution function (CDF).

B Derivation of the Posterior Distribution of $Q$

In the “Belief Updates on Q-Values” section of the main paper, we have shown that

$$\hat{p}_{Q,s,a}(q|\theta, r, s') = \frac{1}{Z} \mathcal{D}_{\theta,r,s'} \left(\frac{q - r}{\gamma} \mid q, s', \theta\right) p_{Q,s,a}(q|\theta)$$

where $Z$ is a normalization constant. Applying the distributions over $V_{s'}$ and $Q_{s,a}$, we can derive the posterior:

$$\hat{p}_{Q,s,a}(q) = \frac{1}{Z} \sum_{b \in A} \frac{1}{\sigma_{s',b}} \phi\left(\frac{q - (\gamma \mu_{s',b})}{\gamma \sigma_{s',b}}\right) \prod_{b' \neq b, b' \in A} \Phi\left(\frac{q - (\gamma \mu_{s',b'})}{\gamma \sigma_{s',b'}}\right) \frac{1}{\sigma_{s,a}} \phi\left(\frac{q - \mu_{s,a}}{\sigma_{s,a}}\right)$$

$$= \frac{1}{Z \sqrt{2\pi \sigma_{s,a}}} \exp\left\{-\frac{1}{2} \frac{(\mu_{s,a} - (r + \gamma \mu_{s',b}))^2}{\sigma_{s,a}^2 + \gamma^2 \sigma_{s',b}^2}\right\} \phi\left(\frac{q - \mu_{\gamma,b}}{\sigma_{\gamma,b}}\right) \prod_{b' \neq b, b' \in A} \Phi\left(\frac{q - (r + \gamma \mu_{s',b'})}{\gamma \sigma_{s',b'}}\right)$$

where $Z$ is a normalization constant and

$$c_{\gamma,b} = \frac{1}{\sqrt{\sigma_{s,a}^2 + \gamma^2 \sigma_{s',b}^2}} \phi\left(\frac{(r + \gamma \mu_{s',b}) - \mu_{s,a}}{\sqrt{\sigma_{s,a}^2 + \gamma^2 \sigma_{s',b}^2}}\right)$$

$$\mu_{\gamma,b} = \frac{\sigma_{\gamma,b}^2}{\sigma_{s,a}^2} \phi\left(\frac{(r + \gamma \mu_{s',b}) - \mu_{s,a}}{\sqrt{\sigma_{s,a}^2 + \gamma^2 \sigma_{s',b}^2}}\right)$$

$$\sigma_{\gamma,b}^2 = \frac{1}{\sigma_{s,a}^2} + \frac{1}{\gamma^2 \sigma_{s',b}^2}$$

C Mean and Variance of the Posterior Distribution of $Q$

C.1 Moment Generating Function

The mean and variance of the posterior distribution (Eq.13) can be analytically found when $|A| = 2$. Consider a random variable $X_M = \max_{1 \leq k \leq N} X_k$ which density function (Eq.15) has a similar
form to the posterior distribution. The moment generating function of \( X_M \) is:

\[
M(t) = \int_{-\infty}^{\infty} e^{tx} \sum \frac{1}{\sigma_i} \phi\left( \frac{x - \mu_i}{\sigma_i} \right) \prod_{i \neq j} \Phi\left( \frac{x - \mu_j}{\sigma_j} \right) dx
\]

\[
= \sum \eta_i(t) \int_{-\infty}^{\infty} \frac{1}{\sigma_i} \phi\left( \frac{x - \mu_i}{\sigma_i} \right) \prod_{i \neq j} \Phi\left( \frac{x - \mu_j}{\sigma_j} \right) dx
\]

where

\[
\eta_i(t) = \exp\left\{ \mu_i t + \frac{t^2 \sigma_i^2}{2} \right\} \quad \text{and} \quad \mu_i' = \mu_i + t \sigma_i^2
\]

When \( N = 2 \),

\[
M(t) = \int_{-\infty}^{\infty} e^{tx} \left( \frac{1}{\sigma_1} \phi\left( \frac{x - \mu_1}{\sigma_1} \right) \phi\left( \frac{x - \mu_2}{\sigma_2} \right) + \frac{1}{\sigma_2} \phi\left( \frac{x - \mu_2}{\sigma_2} \right) \Phi\left( \frac{x - \mu_1}{\sigma_1} \right) \right) dx
\]

(16)

Since the two terms are symmetric, let \( M(t) = M_1(t) + M_2(t) \) and differentiate each term with respect to \( \mu_2 \) and \( \mu_1 \), respectively. For the first term,

\[
\frac{\partial M_1(t)}{\partial \mu_2} = -\frac{\eta_1(t)}{\sigma_2} \int_{-\infty}^{\infty} \phi\left( \frac{x - \mu_1}{\sigma_1} \right) \phi\left( \frac{x - \mu_2}{\sigma_2} \right) dx
\]

\[
= -\frac{\eta_1(t) \sigma_{12}}{\sqrt{2\pi} \sigma_1 \sigma_2} \exp\left\{ -\frac{1}{2} \left( \frac{\mu_1' - \mu_2}{\sigma_1} \right)^2 \right\} \int_{-\infty}^{\infty} \phi\left( \frac{x - \mu_{12}}{\sigma_{12}} \right) dx
\]

\[
= -\frac{\eta_1(t) \sigma_{12}}{\sqrt{2\pi} \sigma_1 \sigma_2} \exp\left\{ -\frac{1}{2} \left( \frac{\mu_1' - \mu_2}{\sigma_1^2 + \sigma_2^2} \right)^2 \right\}
\]

(17)

C.2 Moments of the Posterior Distribution

We apply this result to the posterior distribution by replacing the variables in \( M_1(t) \) as:

\[
\mu_1 \rightarrow \bar{\mu}_{r,1} \quad \sigma_1 \rightarrow \bar{\sigma}_{r,1} \quad \mu_2 \rightarrow r + \gamma \mu_2 \quad \sigma_2 \rightarrow \gamma \sigma_2
\]

and replacing the variables in \( M_2(t) \) similarly. Then, we obtain the normalizing factor:

\[
Z = c_{r,1} \Phi_{r,1} + c_{r,2} \Phi_{r,2}
\]

(19)

where we define the following notations for simplicity:

\[
\Phi_{r,1} = \Phi\left( \frac{\bar{\mu}_{r,1} - (r + \gamma \mu_{s,2})}{\bar{\sigma}_{r,1}^2 + \gamma^2 \sigma_{s,2}^2} \right), \quad \phi_{r,1} = \frac{1}{\sqrt{\bar{\sigma}_{r,1}^2 + \gamma^2 \sigma_{s,2}^2}} \phi\left( \frac{\bar{\mu}_{r,1} - (r + \gamma \mu_{s,2})}{\sqrt{\bar{\sigma}_{r,1}^2 + \gamma^2 \sigma_{s,2}^2}} \right)
\]

and \( \Phi_{r,2} \) and \( \phi_{r,2} \) are also similarly defined. The exact mean of the posterior distribution is derived by solving the first derivative of the moment generating function with respect to \( t \) at \( t = 0 \):

\[
M_1'(t) = c_{r,1} \eta_1(t) \phi\left( \frac{\bar{\mu}_{r,1} - (r + \gamma \mu_{s,2})}{\sqrt{\bar{\sigma}_{r,1}^2 + \gamma^2 \sigma_{s,2}^2}} \right) + c_{r,1} \eta_1(t) \phi\left( \frac{\bar{\sigma}_{r,1}^2}{\sqrt{\bar{\sigma}_{r,1}^2 + \gamma^2 \sigma_{s,2}^2}} \phi\left( \frac{\bar{\mu}_{r,1} - (r + \gamma \mu_{s,2})}{\sqrt{\bar{\sigma}_{r,1}^2 + \gamma^2 \sigma_{s,2}^2}} \right)
\]

(18)
Thus, the second moment is:

\[ M''_t(t) = c_{r,1}\eta''_1(t)\Phi\left(\frac{\mu'_t - (r + \gamma \mu,')}{\sqrt{\sigma^2_t + \gamma^2 \sigma^2_{s',2}}}\right) + 2c_{r,1}\eta'_1(t)\frac{\sigma^2_{r,1}}{\sqrt{\sigma^2_t + \gamma^2 \sigma^2_{s',2}}} \phi\left(\frac{\mu'_t - (r + \gamma \mu,')}{\sqrt{\sigma^2_t + \gamma^2 \sigma^2_{s',2}}}\right) + c_{r,1}\eta_1(t)\frac{\sigma^2_{r,1}}{(\sigma^2_t + \gamma^2 \sigma^2_{s',2})^{2/3}} \phi\left(\frac{\mu'_t - (r + \gamma \mu,')}{\sqrt{\sigma^2_t + \gamma^2 \sigma^2_{s',2}}}\right) \]

The variance of the posterior is also derived by solving the second derivative of the moment generating function:

\[ \mathbb{E}_{q \sim p_{Q,a}()}[q^2] = \frac{c_{r,1}}{Z} \left( (\mu^2_{r,1} + \mu^2_{r,2})\Phi_{r,1} + 2\mu_{r,1}\mu_{r,2}\Phi_{r,2} - \frac{\sigma^4_{r,2}}{\sigma^2_{s',2} + \gamma^2 \sigma^2_{s',1}} (\mu_{r,2} - (r + \gamma \mu,'))\Phi_{r,2} \right) \]

and the variance is \( \mathbb{E}_{q \sim p_{Q,a}()}[q^2] - (\mathbb{E}_{q \sim p_{Q,a}()}[q])^2 \).

**D Q-beliefs for Stochastic MDPs**

For stochastic MDPs, we add small Gaussian white noise to the likelihood, \( r + \gamma V_s + W \) where \( W \sim \mathcal{N}(0, \sigma_w^2) \), and the likelihood distribution is obtained by solving the following integral:

\[ p(r + \gamma V_s | q, \theta) = \int_{-\infty}^{\infty} \sum_{b \in A} \frac{1}{\sigma_{s',b}} \phi\left( \frac{w - (q - (r + \gamma \mu,')}{\gamma \sigma_{s',b}} \right) \times \prod_{b' \neq b} \Phi\left( \frac{w - (q - (r + \gamma \mu,')}{\gamma \sigma_{s',b'}} \right) \frac{1}{\sigma_w} \phi\left( \frac{w}{\sigma_w} \right) dw \]

\[ = \int_{-\infty}^{\infty} \sum_{b \in A} \frac{1}{\sigma_{s',b}} \phi\left( \frac{w - \bar{w}_b}{\bar{v}_b} \right) \prod_{b' \neq b} \left( 1 - \Phi\left( \frac{w - (q - (r + \gamma \mu,')}{\gamma \sigma_{s',b'}} \right) \right) dw \]

where

\[ l_b = \frac{1}{\sqrt{\sigma_w^2 + \gamma^2 \sigma_{s',b}^2}} \phi\left( \frac{q - (r + \gamma \mu,')}{\sqrt{\sigma_w^2 + \gamma^2 \sigma_{s',b}^2}} \right) \]

\[ \bar{w}_b = \frac{q - (r + \gamma \mu,')}{\gamma^2 \sigma_{s',b}^2} \]

\[ \bar{v}_b = \frac{1}{\gamma^2 \sigma_{s',b}^2} \]

**D.1 Expected Likelihood for \(|A| = 2\)**

The distribution inside the integral has a similar form with the posterior distribution Eq.[3] but more complicated, and we have mentioned that a closed form solution for its integral is not available when \(|A| > 2\). Therefore, we derive an analytic solution of the expected likelihood when \(|A| = 2\) and approximate to a simpler form so that it can be generalized to an arbitrary number of actions.

Using Eq.[18] for finding the zeroth moment, we obtain:

\[ p(r + \gamma V_s | q, \theta) = l_1 \Phi\left( \frac{-\bar{w}_1 - (q - (r + \gamma \mu,'))}{\sqrt{\bar{v}_1^2 + \gamma^2 \sigma_{s',2}^2}} \right) + l_2 \Phi\left( \frac{-\bar{w}_2 - (q - (r + \gamma \mu,'))}{\sqrt{\bar{v}_2^2 + \gamma^2 \sigma_{s',1}^2}} \right) \]
Inside the CDF term is a function of $q$:

$$\frac{-\bar{w}_1 - (q - (r + \gamma \mu_{s',2}))}{\sqrt{\bar{v}_1^2 + \gamma^2 \sigma_{s',2}^2}} = \frac{1}{\sqrt{\bar{v}_1^2 + \gamma^2 \sigma_{s',2}^2}} \left( 1 - \frac{\bar{v}_1^2}{\gamma^2 \sigma_{s',1}^2} \right) q - \left( r + \gamma \mu_{s',2} - \frac{\bar{v}_1^2}{\gamma^2 \sigma_{s',1}^2} (r + \gamma \mu_{s',1}) \right)$$

We define

$$\mu_{2}^{w} = \left( 1 - \frac{\bar{v}_1^2}{\gamma^2 \sigma_{s',1}^2} \right)^{-1} \left( r + \gamma \mu_{s',2} - \frac{\bar{v}_1^2}{\gamma^2 \sigma_{s',1}^2} (r + \gamma \mu_{s',1}) \right)$$

and express the likelihood distribution Eq.24 as:

$$p(r + \gamma V_s | q, \theta) = l_1 \Phi \left( \frac{q - \mu_{1}^{w}}{\sigma_{2}^{w}} \right) + l_2 \Phi \left( \frac{q - \mu_{2}^{w}}{\sigma_{2}^{w}} \right)$$

Then, we can find the solutions of the posterior mean and variance for the stochastic case when $|A| = 2$ by replacing $r + \gamma \mu_{s',2}$ and $\gamma \sigma_{s',2}$ with $\mu_{2}^{w}$ and $\sigma_{2}^{w}$, respectively in Eq.20 and Eq.21.

### D.2 Asymptotic Limits

In one asymptotic limit of $\sigma_{w}/\sigma_{s',b} \rightarrow 0$,

$$\lim_{\sigma_{w}/\sigma_{s',b} \rightarrow 0} \frac{\bar{v}_1^2}{\bar{v}_1^2 + \gamma^2 \sigma_{s',s'}^2} = \lim_{\sigma_{w}/\sigma_{s',b} \rightarrow 0} \frac{\gamma^2 \sigma_{s',b}^2 \sigma_{w}^2}{\gamma^2 \sigma_{s',b}^2 + \sigma_{w}^2} = 0$$

and therefore,

$$\lim_{\sigma_{w}/\sigma_{s',b} \rightarrow 0} \Phi \left( \frac{-\bar{w}_b - (q - (r + \gamma \mu_{s',b}))}{\sqrt{\bar{v}_1^2 + \gamma^2 \sigma_{s',s'}^2}} \right) = \Phi \left( \frac{q - (r + \gamma \mu_{s',b})}{\gamma \sigma_{s',b}} \right)$$

and the likelihood distribution becomes

$$\sum_{b \in \{1, 2\}} \frac{1}{\sigma_{2}^{w} + \gamma^2 \sigma_{s',b}^2} \Phi \left( \frac{q - (r + \gamma \mu_{s',b})}{\sqrt{\sigma_{2}^{w} + \gamma^2 \sigma_{s',b}^2}} \right) = \Phi \left( \frac{q - (r + \gamma \mu_{s',b})}{\gamma \sigma_{s',b}} \right)$$

The posterior distribution derived from this likelihood has the same form with Eq.3 but it uses $\gamma^2 \sigma_{s',s'}^2 + \sigma_{w}^2$ instead of $\gamma^2 \sigma_{s',s'}^2$ in $c_{\tau,b}$, $\bar{c}_{\tau,b}$, and $\bar{\sigma}_{\tau,b}$:

$$c_{\tau,b} = \frac{1}{\sqrt{\sigma_{s,a}^2 + \gamma^2 \sigma_{s',s'}^2 + \sigma_{w}^2}} \phi \left( \frac{(r + \gamma \mu_{s',b}) - \mu_{s,a}}{\sqrt{\sigma_{s,a}^2 + \gamma^2 \sigma_{s',s'}^2 + \sigma_{w}^2}} \right)$$

$$\bar{c}_{\tau,b} = \bar{\sigma}_{\tau,b} \left( \frac{\mu_{s,a}}{\sigma_{s,a}^2} + \frac{r + \gamma \mu_{s',b}}{\gamma^2 \sigma_{s',s'}^2 + \sigma_{w}^2} \right)$$

$$\bar{\sigma}_{\tau,b} = \left( \frac{1}{\sigma_{s,a}^2} + \frac{1}{\sigma_{s,s'}^2 + \sigma_{w}^2} \right)^{-1}$$

It is identical to the posterior of the deterministic case when $\sigma_{w} = 0$.

In the other asymptotic limit,

$$\lim_{\sigma_{s',b}/\sigma_{w} \rightarrow 0} \bar{v}_1^2 + \gamma^2 \sigma_{s',s'}^2 = \gamma^2 \sigma_{s',s'}^2 + \gamma^2 \sigma_{s',s'}^2$$

and

$$\lim_{\sigma_{s',b}/\sigma_{w} \rightarrow 0} \bar{v}_1^2 = 1$$
and since we set $\sigma_w$ as a small number, $\sigma_{s',b}/\sigma_w \to 0$ infers $\sigma_{s',b} \to 0$, and therefore, the likelihood distribution becomes Gaussian:

$$\lim_{\sigma_{s',b}/\sigma_w \to 0} p(r + \gamma V_{s'}|q, \theta) = \frac{1}{\sqrt{2\pi \sigma^2_w}} \phi \left( \frac{q - (r + \gamma \mu_{s',b})}{\sqrt{\gamma^2 \sigma_{s',b}^2 + \sigma_w^2}} \right)$$

where $b^+ = \arg\max_{b \in \{1, 2\}} \mu_{s',b}$. Therefore, the posterior distribution becomes Gaussian with mean at $\tilde{\mu}_{\tau,b^+}$ and variance at $\tilde{\sigma}_{\tau,b^+}^2$ in Eq. 29.

### D.3 Approximate Likelihood

In order to have closed-form expressions for the ADFQ update, we extend the asymptotic result for $|A| = 2$ presented in the previous section to the general case ($|A| = n$ for $n \in \mathbb{N}$) with an assumption of $\sigma_w \ll \sigma_b \forall b \in A$. Therefore, the approximate likelihood for stochastic MDPs is:

$$p(r + \gamma V_{s'}|q, \theta) = \sum_{b \in A} \frac{\gamma}{\sqrt{\gamma^2 \sigma_{s',b}^2 + \sigma_w^2}} \phi \left( \frac{q - (r + \gamma \mu_{s',b})}{\sqrt{\gamma^2 \sigma_{s',b}^2 + \sigma_w^2}} \right) \prod_{b' \neq b, b' \in A} \Phi \left( \frac{q - (r + \gamma \mu_{s',b'})}{\gamma \sigma_{s',b'}} \right)$$

Then, the posterior distribution for stochastic MDPs is derived as:

$$\hat{p}_{Q,s,a}(q) = \frac{1}{Z} \sum_{b \in A} \frac{\gamma}{\sqrt{\gamma^2 \sigma_{s',b}^2 + \sigma_w^2}} \phi \left( \frac{q - (r + \gamma \mu_{s',b})}{\sqrt{\gamma^2 \sigma_{s',b}^2 + \sigma_w^2}} \right) \prod_{b' \neq b, b' \in A} \Phi \left( \frac{q - (r + \gamma \mu_{s',b'})}{\gamma \sigma_{s',b'}} \right)$$

where $Z$ is a normalization constant and

$$c_{r,b} = \frac{1}{\sqrt{\gamma^2 \sigma_{s,a}^2 + \gamma^2 \sigma_{s',b}^2 + \sigma_w^2}} \phi \left( \frac{(r + \gamma \mu_{s',b}) - \mu_{s,a}}{\sqrt{\gamma^2 \sigma_{s,a}^2 + \gamma^2 \sigma_{s',b}^2 + \sigma_w^2}} \right)$$

$$\mu_{r,b} = \frac{\sigma_w^2}{\sigma_{s,a}^2 + \gamma^2 \sigma_{s',b}^2 + \sigma_w^2} \left( \frac{\mu_{s,a} + \gamma \mu_{s',b}}{\gamma^2} \right)$$

$$\sigma_{r,b}^2 = \left( \frac{1}{\sigma_{s,a}^2} + \frac{1}{\gamma^2 \sigma_{s',b}^2 + \sigma_w^2} \right)^{-1}$$

### E Proofs

#### E.1 Lemma 1

**Lemma 1.** Let $X$ be a random variable following a normal distribution, $N(\mu, \sigma^2)$. Then we have:

$$\lim_{\sigma \to 0} \Phi \left( \frac{x - \mu}{\sigma} \right) - \exp \left\{ -\frac{1}{2} \left[ \frac{x - \mu}{\sigma} \right]^2 \right\} = 0$$

where $[x]_+ = \max(0, x)$ is the ReLU nonlinearity.
Proof.

\[
\lim_{\sigma \to 0} \frac{x - \mu}{\sigma} = -\infty
\]

Let’s define \( y \equiv (x - \mu)/\sigma \),

\[
\Phi(y < 0) = \int_{-\infty}^{y} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt
\]

\[
= \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} t' \right\} dt'
\]

\[
= \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} y^2 \right\} \int_{-\infty}^{0} \exp \left\{ (y + \frac{1}{2} t') t' \right\} dt'
\]

\[
= \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} y^2 \right\} \int_{-\infty}^{0} \exp \left\{ -yt' \right\} dt'
\]

\[
= -\frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} y^2 \right\}
\]

\[
= \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} y^2 \right\}
\]

\[
\lim_{y \to -\infty} \Phi(y) = \lim_{y \to -\infty} \int_{-\infty}^{y} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt = 1 = e^0
\]

Therefore,

\[
\lim_{\sigma \to 0} \Phi \left( \frac{x - \mu}{\sigma} \right) - \exp \left\{ -\frac{1}{2} \left[ \frac{x - \mu}{\sigma} \right]^2 \right\} = 0
\]

\[\square\]

E.2 Theorem 1

**Theorem 1.** Suppose that the mean and variance of \( Q_{s,a} \) \( \forall s \in S, \forall a \in A \) are iteratively updated by the mean and variance of \( \hat{p}_{Q_{s,a}} \) after observing \( r \) and \( s' \) at every step. When \(|A| = 2\), the update rule of the means is equivalent to the Q-learning update if all state-action pairs are visited infinitely often and the variances approach 0. In other words, at the \( k \)th update on \( \mu_{s,a} \):

\[
\lim_{k \to \infty, \{\sigma\} \to 0} \mu_{s,a;k+1} = (1 - \alpha_{r;k}) \mu_{s,a;k} + \alpha_{r;k} \left( r + \gamma \max_{b \in A} \mu_{s',b;k} \right)
\]

where \( \alpha_{r;k} = \sigma_{s,a;k}^2 / \left( \sigma_{s,a;k}^2 + \gamma^2 \sigma_{s',b;k}^2 + \sigma_w^2 \right) \) and \( b^+ = \arg\max_{b \in A} \mu_{s',b} \). For deterministic MDPs, \( \sigma_w = 0 \).

**Proof.** We first show the convergence of the algorithm in a deterministic MDP, and then extend the result to a stochastic case.

For simplicity, we define new notations as:

\[
y_s \equiv r + \gamma \mu_{s',b} - \mu_{s,a}, \quad \nu_0 \equiv \sigma_{s,a}^2, \quad \nu_b \equiv \sigma_{s,a}^2 + \gamma^2 \sigma_{s',b}^2
\]

In the section C we obtained the exact solutions for the posterior mean and variance when \(|A| = 2\) (Eq 20 and Eq 21). When \( \sigma_{s,a}, \sigma_{s',a_1}, \sigma_{s',a_2} \to 0 \), the posterior mean is approximated as:

\[
\frac{\bar{\mu}_{r,1} \Phi_{r,1} + \bar{\mu}_{r,2} \Phi_{r,2}}{c_{r,1} \Phi_{r,1} + c_{r,2} \Phi_{r,2}}
\]

(32)
Then, using the Lemma 1, \( c_{\tau,1} \Phi_{\tau,1} \) is approximated as:

\[
\frac{1}{\sqrt{2\pi(\sigma_{s,a}^2 + \gamma^2 \sigma_{s',1}^2)}} \exp \left\{ -\frac{(r + \gamma \mu_{s',1} - \mu_{s,a})^2}{2(\sigma_{s,a}^2 + \gamma^2 \sigma_{s',1}^2)} - \frac{[r + \gamma \mu_{s',2} - \hat{\mu}_{\tau,1}]^2}{2(\sigma_{s,1}^2 + \gamma^2 \sigma_{s',2}^2)} \right\} \\
= \frac{1}{\sqrt{2\pi v_1}} \exp \left\{ -\frac{y_1^2}{2v_1} - \frac{[y_2 - \alpha_1 y_1]^2}{2v_1^{-1}(v_1 v_2 - v_0^2)} \right\} \quad \text{where } \alpha_b = \frac{\sigma_{s,a}^2}{\sigma_{s,a}^2 + \gamma^2 \sigma_{s',b}^2} = v_0 \quad v_0 (33)
\]

Since the RHS of the equation is a sum of exponential function with the denominator of the inside term is proportional to a negative inverse variance, \( \mathbb{E}_{\hat{q} \sim \hat{p}_{q*,a}^1}(\hat{q}) \) is approximated to \( \hat{\mu}_{\tau,2} = (1 - \alpha_2)\mu_{s,a} + \alpha_2(r + \gamma \mu_{s',a}) \) if \( c_{\tau,1} \Phi_{\tau,1} \ll c_{\tau,2} \Phi_{\tau,2} \) which is identical with the Q-learning update. Therefore, proving Theorem 1 is equivalent to proving the following statement. If \( \mu_{s',2} > \mu_{s',1} \) and \( \sigma_{s,a}, \sigma_{s',1} \), and \( \sigma_{s',2} \) approach to 0, then \( c_{\tau,1} \Phi_{\tau,1}/c_{\tau,2} \Phi_{\tau,2} \) approaches to 0. From the Eq 32 and Eq 33,

\[
\log \left( \frac{\sqrt{v_1} c_1 \Phi_1}{v_2 c_2 \Phi_2} \right) = -\frac{y_1^2}{2v_1} \left( 1 + \frac{v_1}{v_1 v_2 - v_0^2} - \frac{v_1 v_2}{v_1 v_2 - v_0^2} \right) + \frac{y_2^2}{2v_2} \left( 1 + \frac{v_0}{v_1 v_2 - v_0^2} - \frac{v_1 v_2}{v_1 v_2 - v_0^2} \right)
\]

Therefore,

\[
c_1 \Phi_1 = \sqrt{v_2} \quad \text{and} \quad \mu_{(new)} = \frac{\mu_{\tau,1} \sqrt{v_2} + \hat{\mu}_{\tau,2} \sqrt{v_1}}{\sqrt{v_1} + \sqrt{v_2}}
\]

Since \( \hat{\mu}_{\tau,1} \geq \mu_{s,a} \) and \( \hat{\mu}_{\tau,2} \geq \mu_{s,a} \), the newly updated mean is located somewhere between \( \mu_{\tau,1} \) and \( \hat{\mu}_{\tau,2} \) and always \( \mu_{(new)} \geq \mu_{s,a} \). Therefore, if \( \mu_{s,a} < r + \gamma \mu_{s',1} \) and \( \hat{\mu}_{\tau,2} \leq r + \gamma \mu_{s',1} \), then \( \mu_{(new)} \geq \mu_{s,a} \) until \( \hat{\mu}_{\tau,2} \) becomes larger than \( r + \gamma \mu_{s',1} \).

(i) For \( \mu_{s,a} < r + \gamma \mu_{s',1} \) and \( \hat{\mu}_{\tau,2} < r + \gamma \mu_{s',1} \),

\[
(RHS) = -\frac{y_1^2}{2v_1} \left( 1 + \frac{v_1}{v_1 v_2 - v_0^2} - \frac{v_1 v_2}{v_1 v_2 - v_0^2} \right) + \frac{y_2^2}{2v_2} \left( 1 + \frac{v_0}{v_1 v_2 - v_0^2} - \frac{v_1 v_2}{v_1 v_2 - v_0^2} \right)
\]

Therefore,

\[
c_1 \Phi_1 = \frac{\sqrt{v_2}}{v_1} \quad \text{and} \quad \mu_{(new)} = \frac{\mu_{\tau,1} \sqrt{v_2} + \hat{\mu}_{\tau,2} \sqrt{v_1}}{\sqrt{v_1} + \sqrt{v_2}}
\]

Since \( \hat{\mu}_{\tau,1} \geq \mu_{s,a} \) and \( \hat{\mu}_{\tau,2} \geq \mu_{s,a} \), the newly updated mean is located somewhere between \( \mu_{\tau,1} \) and \( \hat{\mu}_{\tau,2} \) and always \( \mu_{(new)} \geq \mu_{s,a} \). Therefore, if \( \mu_{s,a} < r + \gamma \mu_{s',1} \) and \( \hat{\mu}_{\tau,2} \leq r + \gamma \mu_{s',1} \), then \( \mu_{(new)} \geq \mu_{s,a} \) until \( \hat{\mu}_{\tau,2} \) becomes larger than \( r + \gamma \mu_{s',1} \).

(ii) For \( r + \gamma \mu_{s',1} \leq \hat{\mu}_{\tau,1} < r + \gamma \mu_{s',2} \) (\( \hat{\mu}_{\tau,2} > r + \gamma \mu_{s',1} \) from this condition),

\[
(RHS) = -\frac{y_1^2}{2v_1} - \frac{(y_2 - \alpha_1 y_1)^2}{2v_1^{-1}(v_1 v_2 - v_0^2)} + \frac{y_2^2}{2v_2}
\]

Therefore, \( R(HS) < 0 \) and

\[
\lim_{\sigma_{s,a}, \sigma_{s',1}, \sigma_{s',2} \to 0} c_1 \Phi_1 = \lim_{v_0, v_1, v_2 \to 0} \left[ \sqrt{v_2} \exp \left\{ -\frac{(y_1 - \alpha_2 y_2)^2}{2v_2^{-1}(v_1 v_2 - v_0^2)} \right\} \right] = 0
\]

(iii) For \( \mu_{s,a} > r + \mu_{s',2} \) and \( \hat{\mu}_{\tau,1} \geq r + \gamma \mu_{s',2} \) (\( \hat{\mu}_{\tau,2} > r + \gamma \mu_{s',1} \) from this condition),

\[
(RHS) = -\frac{y_1^2}{2v_1} + \frac{y_2^2}{2v_2}
\]

\[
= -\frac{y_1^2}{2v_1} \left( 1 - \frac{v_1 y_2^2}{v_2 y_1^2} \right)
\]
If \( \frac{y_2^2}{v_2} < \frac{y_1^2}{v_1} \), then \( RHS < 0 \) with \( \sigma_{s,a}, \sigma_{s',1}, \sigma_{s',2} \to 0 \), and thus \( c_{\tau,1} \Phi_{\tau,1}/c_{\tau,2} \Phi_{\tau,2} \) approaches to 0 as the previous case. If \( \frac{y_2^2}{v_2} \geq \frac{y_1^2}{v_1} \),
\[
\frac{c_1 \Phi_1}{c_2 \Phi_2} = C \sqrt{\frac{v_2}{v_1}} \quad \text{for some constant } C
\]

Therefore,
\[
\mu_{s,a}^{(\text{new})} = \frac{\bar{\mu}_{\tau,1} C + \bar{\mu}_{\tau,2}}{C + 1}
\]

Similar to the first case, \( \mu_{s,a}^{(\text{new})} \) will be located somewhere between \( \bar{\mu}_{\tau,1} \) and \( \bar{\mu}_{\tau,2} \) and always \( \mu_{s,a}^{(\text{new})} < \mu_{s,a} \) until \( \bar{\mu}_{\tau,1} \) becomes smaller than or equal to \( r + \gamma \mu_{s',2} \).

In conclusion, when the variables satisfy either (i) or (iii), the mean value is contracted to the range corresponding to (ii) which is identical to the Q-learning update.

For stochastic MDPs, \( r + \gamma \mu_{s',b} \) and \( \gamma \sigma_{s',b} \) in the CDF terms are replaced by \( \mu_{s',b}^w \) and \( \gamma \sigma_{s',b}^w \), respectively as \( \sigma_{s',b}^w \) approaches 0 as \( \sigma_{s,a}, \sigma_{s',1}, \sigma_{s',2} \to 0 \) and therefore, the above proofs are applied. However, the proofs are invalid when \( \sigma_{s',b}/\sigma_w = 0 \) since the CDF terms in the likelihood distribution are no longer functions of \( q \). As we have shown in the section D.2, the posterior mean and variance are
\[
\mu_{s,a}^{(\text{new})} = \bar{\mu}_{\tau,b^+} = \bar{\sigma}_{\tau,b^+} + \frac{\mu_{s,a}^2}{\sigma_{s,a}^2} \left( r + \gamma \mu_{s',b^+} \right)
\]

where \( b^+ = \arg\max_{b \in \mathcal{A}} \mu_{s',b} \). Therefore, the update rule using the posterior mean for stochastic MDPs is identical to the Q-learning update rule with the corresponding learning rate, \( \alpha_{\tau} \):
\[
\alpha_{\tau} = \frac{\sigma_{s,a}^2}{\sigma_{s,a}^2 + \gamma^2 \sigma_{s',b^+}^2 + \sigma_w^2}
\]

**E.3 Theorem 2: Convergence of ADFQ**

**Theorem 2.** The ADFQ update on the mean \( \mu_{s,a} \) \( \forall s \in \mathcal{S}, \forall a \in \mathcal{A} \) for \( |\mathcal{A}| = 2 \) is equivalent to the Q-learning update if the variances approach 0 and if all state-action pairs are visited infinitely often. In other words, we have:
\[
\lim_{k \to \infty} \mu_{s,a;k+1} = (1 - \alpha_{\tau,k}) \mu_{s,a;k} + \alpha_{\tau,k} \left( r + \gamma \max_{b \in \mathcal{A}} \mu_{s',b;k} \right)
\]

where \( \alpha_{\tau,k} = \frac{\sigma_{s,a,k}^2}{\sigma_{s,a,k}^2 + \gamma^2 \sigma_{s',b;k}^2 + \sigma_w^2} \) and \( b^+ = \arg\max_{b \in \mathcal{A}} \mu_{s',b} \). For deterministic MDPs, \( \sigma_w = 0 \).

**Proof.** Similar to the proof for the exact update case, we will show that the ratios of the coefficients, \( k_1^*/k_{b_{\text{max}}}^* \) becomes 0 \( \forall b \in \mathcal{A}, b \neq b_{\text{max}} \) where \( b_{\text{max}} = \arg\max_{b \in \mathcal{A}} \mu_{s',b} \), and \( \mu_b^* \to \bar{\mu}_b \) as \( \sigma_{s,a}, \sigma_{s',b} \) \( \forall b \in \mathcal{A} \) goes to 0. When \( |\mathcal{A}| = 2 \) and \( \mu_{s',2} > \mu_{s',1} \),
\[
\frac{k_1^*}{k_2^*} = \frac{\sigma_1^* \sigma_{s',2}^*}{\sigma_2^* \sigma_{s',1}^*} \exp \left\{ \frac{y_1^2}{2v_1} + \frac{y_2^2}{2v_2} - \frac{(\mu_1^* - \bar{\mu}_{\tau,1})^2}{2\sigma_{\tau,1}^2} - \frac{(\mu_2^* - \bar{\mu}_{\tau,2})^2}{2\sigma_{\tau,2}^2} \right\}
\]
\[
\left\{ -\frac{r + \gamma \mu_{s',2} - \mu_1^*}{\gamma^2 \sigma_{s',2}^*} + \frac{r + \gamma \mu_{s',1} - \mu_2^*}{\gamma^2 \sigma_{s',1}^*} \right\}
\]

According to the definition of \( \mu_b^* \),
\[
\mu_1^* - \bar{\mu}_{\tau,1} = \frac{\sigma_{\tau,1}}{\gamma^2 \sigma_{s',2}^*} \left[ r + \gamma \mu_{s',2} - \mu_1^* \right] + \frac{\sigma_{\tau,2}}{\gamma^2 \sigma_{s',1}^*} \left[ r + \gamma \mu_{s',1} - \mu_2^* \right]
\]
and $\mu_0^* \geq \mu_{\tau,b}$. Therefore,

$$
\log \left( \frac{k_1^* \sigma_{s',1}^2 \sigma_{s',2}^2}{k_2^* \sigma_{s',2} \sigma_{s',1}^2} \right) = - \frac{y_1^2}{2v_1} + \frac{y_2^2}{2v_2} - \frac{[r + \gamma \mu_{s',2} - \mu_1^*]^2}{2\gamma^2 \sigma_{s',2}^2} \left( 1 + \frac{\sigma_{s',2}^2}{\gamma^2 \sigma_{s',1}^2} \right) + \frac{[r + \gamma \mu_{s',1} - \mu_2^*]^2}{2\gamma^2 \sigma_{s',1}^2} \left( 1 + \frac{\sigma_{s',1}^2}{\gamma^2 \sigma_{s',1}^2} \right)
$$

When $\mu_0^* < r + \gamma \mu_{s',b}$

$$
\mu_0^* = \left( \frac{1}{\sigma_{s',b}^*} + \frac{1}{\gamma^2 \sigma_{s',b}^*} \right)^{-1} \left( \frac{\mu_{s',b}^* + r + \gamma \mu_{s',b}^*}{\sigma_{s',b}^* + \gamma^2 \sigma_{s',b}^*} \right)
$$

When $\mu_0^* \geq r + \gamma \mu_{s',b}$, $\mu_0^* = \mu_{s',b}$.

For $\mu_0^* < r + \gamma \mu_{s',2}$ and $\mu_0^* < r + \gamma \mu_{s',1}$, it is also, $\mu_{s,a} \leq \mu_{s,1} \leq \mu_{s,2} < r + \gamma \mu_{s',1} < r + \gamma \mu_{s',2}$. Then, using Eq.35, we have

$$
\log \left( \frac{k_1^* \sigma_{s',1}^2 \sigma_{s',2}^2}{k_2^* \sigma_{s',2} \sigma_{s',1}^2} \right) = - \frac{y_1^2}{2v_1} + \frac{y_2^2}{2v_2} \frac{(y_2 - \alpha_1 y_1)^2}{2v_1 (v_1 v_2 - v_0^2)} + \frac{y_2^2}{2v_2} \frac{(y_1 - \alpha_2 y_2)^2}{2v_2 (v_1 v_2 - v_0^2)}
$$

which is same with (i) of the proof of Theorem 1. The new mean will be weighted sum of $\mu_0^*, \mu_0^*$. Since $\mu_{s,a}$ is smaller than both $\mu_{s,1}$ and $\mu_{s,2}$, $\mu_{s,a}^{(new)} > \mu_{s,a}$ until $r + \gamma \mu_{s',1} > \mu_{s,2}$. For the other cases, the same directions in the proof of Theorem 1 are applied.

We can apply the same proof procedures to the stochastic case using $\gamma^2 \sigma_{s',b}^2 + \sigma_w^2$ instead of $\gamma^2 \sigma_{s',b}^2$ in $\bar{\mu}_{s,b}, \sigma_{s,b}$, and $c_{s,b}$. Therefore, the mean update rule converges to the Q-learning update and the corresponding learning rate is:

$$
\alpha_\tau = \frac{\sigma_{s,a}^2}{\sigma_{s,a}^2 + \gamma^2 \sigma_{s',b}^2 + \sigma_w^2}
$$

\[\square\]

F  Experimental Details

F.1 Maze Experiment

We chose values for the hyperparameters that resulted the best performance and summarized in Table 2. $\alpha$ is a learning rate of Q-learning, $\epsilon$ is a probability factor in $\epsilon$-greedy method, $\tau$ is the temperature constant in Boltzmann distribution and $\kappa$ is a scaling factor used for Unscented Transform in KTD-Q. The test ranges are: $\epsilon = [0.0, 0.1, 0.2, 0.5]$, $\tau = [0.1, 0.3, 0.5]$, $\kappa = [1, n/2, n]$ where $n = 448$ is the number of parameters of KTD-Q ($|S| \times |A|$ in discrete case). We initialized the mean and Q values with 3.0, the maximum possible reward of the domain.

| Algorithm   | Parameter | Maze - Deterministic | Maze - Stochastic |
|-------------|-----------|----------------------|-------------------|
| Q-learning, $\epsilon$-greedy | $\epsilon$ | 0.0 | 0.0 |
| Q-learning, Boltzmann | $\tau$ | 0.1 | 0.1 |
| ADFQ-Numeric, $\epsilon$-greedy | $\epsilon$ | 0.2 | 0.2 |
| ADFQ $\epsilon$-greedy | $\epsilon$ | 0.2 | 0.2 |
| KTD-Q, $\epsilon$-greedy | $\epsilon$ | 0.1 | 0.1 |
| KTD-Q, active | $\kappa$ | 224 | 224 |
| KTD-Q, active | $\kappa$ | 448 | 224 |
F.2 Neural Network Architecture and Details

In the all domains, we used the default settings of the OpenAI baselines \[8\] for DQN and Double DQN, and made minimal changes for ADFQ. We used ReLU nonlinearities and the Adam optimizer with minibatches size of 32. We used experience replay for the Cartpole and Acrobot environments, and prioritized experience replay for the Atari games.

**Initialization.** In ADFQ, Xavier initialization was used for all weight variables, and all bias variables were initialized to zero except for the final hidden layer. The weights of the final hidden layer were initialized with \(0\) and its bias variables were initialized with two constant values which correspond to \(\mu_0\) and \(-\log(\sigma_0)\) where \(\mu_0\) is an initial mean and \(\sigma_0^2\) is an initial variance (e.g. an initial bias vector of the final layer is \(\vec{b} = [\mu_0, \cdots, \mu_0, -\log(\sigma_0), \cdots, -\log(\sigma_0)]^T\)). We set \(\sigma_0 = 30.0\) for Cartpole and Acrobot tasks, and \(\sigma_0 = 50.0\) for the Atari games.

**Network Architecture.** For the Cartpole and Acrobot environments, we used a simple single hidden layer with 64 neurons. For the Atari games, Breakout and Enduro, we used a network with three convolution layers followed by a 256 neuron linear layer. The first convolution layer contains 32 filters of size 8 with stride 4. The second convolution layer contains 64 filters of size 4 with stride 2. The final convolution layer contains 64 filters of size 3 with stride 1. Both network architectures are the default setting in OpenAI baselines.

G Additional Experiments

In addition to the Maze domain presented in the main paper, we tested ADFQ and the comparing algorithms in a smaller domain, Loop \[6\], shown in Fig. 6 for 10000 finite steps. It consists of 9 states and 2 actions (a,b). There are +1 reward at state 4 and +2 reward at state 8. For a stochastic case, a learning agent performs the other action with a probability 0.1.

As same with the experiment in the Maze domain, we evaluated convergence of the algorithms as well as performance of their current policies during learning, and plotted in Fig. 7. In the both cases, ADFQ converged to the optimal Q values faster than Q-learning. In the deterministic case, all algorithms converged to the optimal policy. However, unlike the result in Maze, KTD-Q with both \(\epsilon\)-greedy and active learning scheme outperformed Q-learning and ADFQ, and its Q-value estimation did not diverge from the optimal Q-values. In the stochastic case, KTD-Q failed to converge to the optimal Q-values as it is proposed under a deterministic environment assumption. The author proposed XKTD-V and XKTD-SARSA which are extended versions of KTD-V and KTD-SARSA, respectively, for a stochastic environment. Yet, KTD-Q was not able to be extended to XKTD-Q (see the section 4.3.2 in \[12\] for details). Despite the convergence issue, the KTD-Q with active learning scheme worked better than Q-learning and converged to an optimal policy. In addition to the experiment results in the main paper, these results imply that KTD-Q does not scale with the number of parameters even though it works well in smaller domains, and its convergence to the optimal Q-values is not guaranteed in stochastic domains.

![Figure 6: Loop domain diagram](image)

21
Additional Approach to Analytic ADF Parameter Estimates

In this section, we introduced another approximation approach to the true posterior, Eq.3 using a stronger approximation. The main approximation in the ADFQ algorithm eventually becomes same with the following algorithm as variance decreases.

When the variance of a Gaussian random variable, $X \sim \mathcal{N}(\mu, \sigma^2)$, approaches 0, its CDF and PDF are approximated to a Heaviside step function, $H(\cdot)$, and a dirac delta function, $\delta(\cdot)$, respectively. Suppose that $\sigma_{s,a} \ll 1$ for all $s \in S$ and $a \in A$. The product of the Gaussian CDFs in the Eq.3 is approximated to 1 if $q \geq r + \gamma \mu_{s',b'}$ for all $b' \in A, b' \neq b$, and 0 otherwise. However, when $q = \bar{\mu}_{\tau,b}$, we cannot simply apply the approximation since the PDF approaches infinity:

$$\lim_{\sigma_{\tau,b}',\sigma_{\tau,b}' \to 0} \frac{1}{\sigma_{\tau,b}} \phi \left( \frac{q - \bar{\mu}_{\tau,b}}{\sigma_{\tau,b}} \right) \cdot \prod_{b' \neq b} \Phi \left( \frac{q - (r + \gamma \mu_{s',b'})}{\gamma \sigma_{s',b'}} \right) = \infty \cdot 0 \neq 0$$

We define a function $f(\cdot)$ which is the approximation of the above term inside of the limit when the term of the product of the Gaussian CDFs approaches to 0 (e.g. $q < r + \gamma \mu_{s',b'}$ for all $b' \neq b$).

$$f(q; \mu, \sigma) = \begin{cases} \phi \left( \frac{q - \mu}{\sigma} \right) & \text{for } q \in [\mu - \epsilon, \mu + \epsilon], \epsilon \ll 1 \\ 0 & \text{otherwise} \end{cases} \quad (36)$$

Let $b^* = \arg\max_{b \in A} \mu_{s',b}$ and $b^{2*} = \arg\max_{b \in A, b \neq b^*} \mu_{s',b}$. We also define $t_{\tau,b} = r + \gamma \mu_{s',b}$ for a TD target where $\tau = <s,a,r,s'>$ is a causality tuple. Then, the true posterior distribution in Eq.3 is approximated to $\tilde{p}_{Q_{s,a}}$ in three different ranges of $q$.

1. $q \in (-\infty, t_{\tau,b^*})$

$$\tilde{p}_{Q_{s,a}}(q) = \frac{1}{Z} \sum_{b} \epsilon_{\tau,b} f(q; \bar{\mu}_{\tau,b}, \sigma_{\tau,b}) \quad (37)$$
2. \( q \in [t_{\tau,b^*}, t_{\tau,b^*}) \)

\[
\tilde{p}_{Q_{\tau,a}}(q) = \frac{1}{Z} \sum_{b \neq b^*} c_{\tau,b} f(q; \bar{\mu}_{\tau,b}, \bar{\sigma}_{\tau,b}) + \frac{1}{Z} \frac{c_{\tau,b^*}}{\bar{\sigma}_{\tau,b^*}} \phi\left(\frac{q - \bar{\mu}_{\tau,b^*}}{\bar{\sigma}_{\tau,b^*}}\right) \tag{38}
\]

3. \( q \in [t_{\tau,b^*}, +\infty) \)

\[
\tilde{p}_{Q_{\tau,a}} = \frac{1}{Z} \sum_{b} c_{\tau,b} f(q; \bar{\mu}_{\tau,b}, \bar{\sigma}_{\tau,b}) + \frac{1}{Z} \frac{c_{\tau,b^*}}{\bar{\sigma}_{\tau,b^*}} \phi\left(\frac{q - \bar{\mu}_{\tau,b^*}}{\bar{\sigma}_{\tau,b^*}}\right) \tag{39}
\]

Applying ADF to the approximated distribution, \( \tilde{p} \), we need to find only mean and variance of the distribution. The first and the second moments of \( \tilde{p}_{Q_{\tau,a}} \) are:

\[
\mathbb{E}_q[\tilde{p}_{Q_{\tau,a}}(\cdot)|q] \approx \frac{\sum_b c_{\tau,b} \nu_{\tau,b} \mu_b}{\sum_b c_{\tau,b} \nu_{\tau,b}} \quad \text{and} \quad \mathbb{E}_q[\tilde{p}_{Q_{\tau,a}}(\cdot)|q]^2 \approx \frac{\sum_b c_{\tau,b} \nu_{\tau,b} (\mu_b^2 + \sigma_b^2)}{\sum_b c_{\tau,b} \nu_{\tau,b}} \tag{40}
\]

where \( \nu_{\tau,b} = \begin{cases} 
1 - (1 - \epsilon)H(r + \gamma \mu_{s,b^*} - \bar{\mu}_{\tau,b}) & \text{for } b \neq b^* \\
1 - (1 - \epsilon)H(r + \gamma \mu_{s,b^*} - \bar{\mu}_{\tau,b}) & \text{for } b = b^*
\end{cases} \)

The variance is computed by \( \mathbb{E}_q[\tilde{p}_{Q_{\tau,a}}(\cdot)|q]^2 - (\mathbb{E}_q[\tilde{p}_{Q_{\tau,a}}(\cdot)|q])^2 \). Note that they are linear combinations of the IVW average values, \( \bar{\mu}_{\tau,b} \) and \( \bar{\sigma}_b^2 \) and its computational complexity for an each update is \( O(|A|) \). Detailed derivations are presented in the following subsections.

**H.1 Normalization**

\[
Z = \sum_{b \neq b^*} c_{\tau,b} \left\{ \int_{-\infty}^{t_{\tau,b^*}} f(q; \bar{\mu}_{\tau,b}, \bar{\sigma}_{\tau,b}) dq + \int_{t_{\tau,b^*}}^{\infty} \frac{1}{\bar{\sigma}_{\tau,b}} \phi\left(\frac{q - \bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\right) dq \right\} \\
+ c_{\tau,b^*} \left\{ \int_{-\infty}^{t_{\tau,b^*}} f(q; \bar{\mu}_{\tau,b^*}, \bar{\sigma}_{\tau,b^*}) dq + \int_{t_{\tau,b^*}}^{\infty} \frac{1}{\bar{\sigma}_{\tau,b^*}} \phi\left(\frac{q - \bar{\mu}_{\tau,b^*}}{\bar{\sigma}_{\tau,b^*}}\right) dq \right\}
\]

\[
= \sum_{b \neq b^*} c_{\tau,b} \left\{ H(t_{\tau,b^*} - (\bar{\mu}_{\tau,b} + \epsilon)) \int_{\bar{\mu}_{\tau,b} + \epsilon}^{\bar{\mu}_{\tau,b}} \frac{\epsilon}{\bar{\sigma}_{\tau,b}} \phi\left(\frac{q - \bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\right) dq + \int_{\bar{\mu}_{\tau,b} + \epsilon}^{\infty} \frac{1}{\bar{\sigma}_{\tau,b}} \phi\left(\frac{q - \bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\right) dq \right\} \\
+ c_{\tau,b^*} \left\{ H(t_{\tau,b^*} - (\bar{\mu}_{\tau,b^*} + \epsilon)) \int_{\bar{\mu}_{\tau,b^*} + \epsilon}^{\bar{\mu}_{\tau,b^*}} \frac{\epsilon}{\bar{\sigma}_{\tau,b^*}} \phi\left(\frac{q - \bar{\mu}_{\tau,b^*}}{\bar{\sigma}_{\tau,b^*}}\right) dq + \int_{\bar{\mu}_{\tau,b^*} + \epsilon}^{\infty} \frac{1}{\bar{\sigma}_{\tau,b^*}} \phi\left(\frac{q - \bar{\mu}_{\tau,b^*}}{\bar{\sigma}_{\tau,b^*}}\right) dq \right\}
\]

\[
= \sum_{b \neq b^*} c_{\tau,b} \left\{ \epsilon H(t_{\tau,b^*} - (\bar{\mu}_{\tau,b} + \epsilon)) \int_{\bar{\mu}_{\tau,b} - \epsilon}^{\bar{\mu}_{\tau,b}} \phi\left(\frac{\epsilon}{\bar{\sigma}_{\tau,b}}\right) dq + \int_{\bar{\mu}_{\tau,b} - \epsilon}^{\infty} \frac{1}{\bar{\sigma}_{\tau,b}} \phi\left(\frac{\epsilon}{\bar{\sigma}_{\tau,b}}\right) dq \right\} \\
+ c_{\tau,b^*} \left\{ \epsilon H(t_{\tau,b^*} - (\bar{\mu}_{\tau,b^*} + \epsilon)) \int_{\bar{\mu}_{\tau,b}^* - \epsilon}^{\bar{\mu}_{\tau,b}^*} \phi\left(\frac{\epsilon}{\bar{\sigma}_{\tau,b^*}}\right) dq + \int_{\bar{\mu}_{\tau,b}^* - \epsilon}^{\infty} \frac{1}{\bar{\sigma}_{\tau,b^*}} \phi\left(\frac{\epsilon}{\bar{\sigma}_{\tau,b^*}}\right) dq \right\}
\]

where \( H(\cdot) \) is a Heaviside step function which \( H(x) = 1 \) if \( x \geq 0 \) and \( H(x) = 0 \) otherwise. Since \( \bar{\sigma}_{\tau,b} \ll 1 \forall b \in A \) and \( \epsilon > 0 \)

\[
\Phi\left(\frac{t_{\tau,b^*} - \bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\right) \approx H(t_{\tau,b^*} - \bar{\mu}_{\tau,b}) \quad \Phi\left(\frac{\epsilon}{\bar{\sigma}_{\tau,b}}\right) \approx 1
\]

\[
Z \approx \sum_{b \neq b^*} c_{\tau,b} (\epsilon H(t_{\tau,b^*} - (\bar{\mu}_{\tau,b} + \epsilon)) - H(t_{\tau,b^*} - \bar{\mu}_{\tau,b})) + c_{\tau,b^*} (\epsilon H(t_{\tau,b^*} - (\bar{\mu}_{b^*} + \epsilon)) - H(t_{\tau,b^*} - \bar{\mu}_{b^*}))
\]

\[
\approx \sum_{b} c_{\tau,b} (1 - (1 - \epsilon)H(t_{\tau,b^*} - \bar{\mu}_{\tau,b})) + c_{\tau,b^*} (1 - (1 - \epsilon)H(t_{b^*} - \bar{\mu}_{b^*}))
\]

23
where
\[ \nu_{\tau,b} = \begin{cases} 1 - (1 - \epsilon)H(t_{\tau,b}^* - \bar{\mu}_{\tau,b}) & \text{for } b \neq b^* \\ 1 - (1 - \epsilon)H(t_{\tau,b} - \bar{\mu}_{\tau,b}) & \text{for } b = b^* \end{cases} \]

H.2 Mean

\[ \mathbb{E}_{\hat{P}_{Q,a}}[q] = \frac{1}{Z} \sum_{b \neq b^*} c_{\tau,b} \left\{ \epsilon H(t_{\tau,b}^* - (\bar{\mu}_{\tau,b} + \epsilon)) \int_{\bar{\mu}_{\tau,b} - \epsilon}^{\bar{\mu}_{\tau,b} + \epsilon} \frac{q}{\sigma_{\tau,b}} \phi \left( \frac{q - \bar{\mu}_{\tau,b}}{\sigma_{\tau,b}} \right) dq + \int_{\bar{\mu}_{\tau,b}^* - \epsilon}^{\bar{\mu}_{\tau,b}^* + \epsilon} \frac{q}{\sigma_{\tau,b^*}} \phi \left( \frac{q - \bar{\mu}_{\tau,b^*}}{\sigma_{\tau,b^*}} \right) dq \right\} + \frac{1}{Z} c_{\tau,b^*} \left\{ \epsilon H(t_{\tau,b^*} - (\bar{\mu}_{\tau,b^*} + \epsilon)) \int_{\bar{\mu}_{\tau,b^*} - \epsilon}^{\bar{\mu}_{\tau,b^*} + \epsilon} \frac{q}{\sigma_{\tau,b^*}} \phi \left( \frac{q - \bar{\mu}_{\tau,b^*}}{\sigma_{\tau,b^*}} \right) dq + \int_{\bar{\mu}_{\tau,b^*}^* - \epsilon}^{\bar{\mu}_{\tau,b^*}^* + \epsilon} \frac{q}{\sigma_{\tau,b^*}} \phi \left( \frac{q - \bar{\mu}_{\tau,b^*}}{\sigma_{\tau,b^*}} \right) dq \right\} \]

The mean of the two sided truncated normal distribution with the original normal distribution \( N(\mu, \sigma^2) \) is:

\[ \mathbb{E}[X | a < X < b] = \mu + \sigma \frac{\phi\left(\frac{a - \mu}{\sigma}\right) - \phi\left(\frac{b - \mu}{\sigma}\right)}{\Phi\left(\frac{b - \mu}{\sigma}\right) - \Phi\left(\frac{a - \mu}{\sigma}\right)} \]

With a normalization constant, \( Z_E \),

\[ \mathbb{E}[X | a < X < b] \cdot Z_E = \mu + \sigma \frac{\phi\left(\frac{a - \mu}{\sigma}\right) - \phi\left(\frac{b - \mu}{\sigma}\right)}{\Phi\left(\frac{b - \mu}{\sigma}\right) - \Phi\left(\frac{a - \mu}{\sigma}\right)} \]

Therefore,

\[ \int_{\bar{\mu}_{\tau,b} - \epsilon}^{\bar{\mu}_{\tau,b} + \epsilon} \frac{q}{\sigma_{\tau,b}} \phi \left( \frac{q - \bar{\mu}_{\tau,b}}{\sigma_{\tau,b}} \right) dq = \bar{\mu}_{\tau,b} \left( 2\Phi \left( \frac{\epsilon}{\sigma_{\tau,b}} \right) - 1 \right) + \sigma_{\tau,b} \cdot 0 \approx \bar{\mu}_{\tau,b} \]

and

\[ \mathbb{E}_{\hat{P}_{Q,a}}[q] = \frac{1}{Z} \sum_{b \neq b^*} c_{\tau,b} \left\{ \epsilon H(t_{\tau,b} - (\bar{\mu}_{\tau,b} + \epsilon)) \bar{\mu}_{\tau,b} + \bar{\mu}_{\tau,b} \left( 1 - \Phi \left( \frac{t_{\tau,b} - \bar{\mu}_{\tau,b}}{\sigma_{\tau,b}} \right) \right) + \sigma_{\tau,b} \phi \left( \frac{t_{\tau,b} - \bar{\mu}_{\tau,b}}{\sigma_{\tau,b}} \right) \right\} + \frac{1}{Z} c_{\tau,b^*} \left\{ \epsilon H(t_{\tau,b^*} - (\bar{\mu}_{\tau,b^*} + \epsilon)) \bar{\mu}_{\tau,b^*} + \bar{\mu}_{\tau,b^*} \left( 1 - \Phi \left( \frac{t_{\tau,b^*} - \bar{\mu}_{\tau,b^*}}{\sigma_{\tau,b^*}} \right) \right) + \sigma_{\tau,b^*} \phi \left( \frac{t_{\tau,b^*} - \bar{\mu}_{\tau,b^*}}{\sigma_{\tau,b^*}} \right) \right\} \approx \frac{1}{Z} \sum_{b} c_{\tau,b} \nu_{\tau,b} \bar{\mu}_{\tau,b} \]

H.3 Variance

First of all, the second moment is:

\[ \mathbb{E}_{\hat{P}_{Q,a}}[q^2] = \frac{1}{Z} \sum_{b \neq b^*} c_{\tau,b} \left\{ \epsilon H(t_{\tau,b} - (\bar{\mu}_{\tau,b} + \epsilon)) q^2 \frac{2}{\sigma_{\tau,b}} \phi \left( \frac{q - \bar{\mu}_{\tau,b}}{\sigma_{\tau,b}} \right) dq + \int_{\bar{\mu}_{\tau,b}}^{\infty} q^2 \frac{2}{\sigma_{\tau,b}} \phi \left( \frac{q - \bar{\mu}_{\tau,b}}{\sigma_{\tau,b}} \right) dq \right\} + \frac{1}{Z} c_{\tau,b^*} \left\{ \epsilon H(t_{\tau,b^*} - (\bar{\mu}_{\tau,b^*} + \epsilon)) q^2 \frac{2}{\sigma_{\tau,b^*}} \phi \left( \frac{q - \bar{\mu}_{\tau,b^*}}{\sigma_{\tau,b^*}} \right) dq + \int_{\bar{\mu}_{\tau,b^*}}^{\infty} q^2 \frac{2}{\sigma_{\tau,b^*}} \phi \left( \frac{q - \bar{\mu}_{\tau,b^*}}{\sigma_{\tau,b^*}} \right) dq \right\} \]

The variance of the truncated normal distribution is:

\[ \text{Var}[X | a < X < b] = \sigma^2 \left[ 1 + \frac{\phi\left(\frac{a - \mu}{\sigma}\right) - \phi\left(\frac{b - \mu}{\sigma}\right)}{\Phi\left(\frac{b - \mu}{\sigma}\right) - \Phi\left(\frac{a - \mu}{\sigma}\right)} - \left( \frac{\phi\left(\frac{a - \mu}{\sigma}\right) - \phi\left(\frac{b - \mu}{\sigma}\right)}{\Phi\left(\frac{b - \mu}{\sigma}\right) - \Phi\left(\frac{a - \mu}{\sigma}\right)} \right)^2 \right] \]

With a normalization constant, \( Z_V \),

\[ \mathbb{E}[X^2 | a < X < b] \cdot Z_V = \left( \mu^2 + \sigma^2 \right) \left( \Phi \left( \frac{b - \mu}{\sigma} \right) - \Phi \left( \frac{a - \mu}{\sigma} \right) \right) + \sigma^2 \left( \frac{\mu - \mu}{\sigma} \phi \left( \frac{a - \mu}{\sigma} \right) - \frac{b - \mu}{\sigma} \phi \left( \frac{b - \mu}{\sigma} \right) \right) + 2\mu \sigma \left( \phi \left( \frac{a - \mu}{\sigma} \right) - \phi \left( \frac{b - \mu}{\sigma} \right) \right) \]
Thus, when $t_{\tau,b^*} \geq \bar{\mu}_{\tau,b} + \epsilon$,

$$\int_{\bar{\mu}_{\tau,b} - \epsilon}^{\bar{\mu}_{\tau,b} + \epsilon} \frac{q^2}{\bar{\sigma}_{\tau,b}} \phi\left(\frac{q - \bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\right) dq = (\bar{\mu}_{\tau,b}^2 + \bar{\sigma}_{\tau,b}^2) \left(2 \Phi\left(\frac{\epsilon}{\bar{\sigma}_{\tau,b}}\right) - 1\right) - 2\epsilon \bar{\sigma}_{\tau,b} \phi\left(\frac{\epsilon}{\bar{\sigma}_{\tau,b}}\right)$$

$$\approx \bar{\mu}_{\tau,b}^2 + \bar{\sigma}_{\tau,b}^2$$

For the second term,

$$\int_{t_{\tau,b^*}}^{\infty} \frac{q^2}{\bar{\sigma}_{\tau,b}} \phi\left(\frac{q - \bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\right) dq = (\bar{\mu}_{\tau,b}^2 + \bar{\sigma}_{\tau,b}^2) \left(1 - \Phi\left(\frac{t_{\tau,b^*} - \bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\right)\right) + \bar{\sigma}_{\tau,b}(t_{\tau,b^*} - \bar{\mu}_{\tau,b}) \phi\left(\frac{t_{\tau,b^*} - \bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\right) + 2\bar{\mu}_{\tau,b} \bar{\sigma}_{\tau,b} \phi\left(\frac{t_{\tau,b^*} - \bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\right)$$

$$\approx (\bar{\mu}_{\tau,b}^2 + \bar{\sigma}_{\tau,b}^2) (1 - H(t_{\tau,b^*} - \bar{\mu}_{\tau,b}))$$

Therefore,

$$\mathbb{E}_{\tilde{p}_{Q,a}}[q^2] \approx \frac{1}{Z} \sum_{b \neq b^*} c_{\tau,b} (\bar{\mu}_{\tau,b}^2 + \bar{\sigma}_{\tau,b}^2) (\epsilon H(t_{\tau,b^*} - (\bar{\mu}_{\tau,b} + \epsilon)) + (1 - H(t_{\tau,b^*} - \bar{\mu}_{\tau,b})))$$

$$+ \frac{1}{Z} c_{\tau,b^*} (\bar{\mu}_{\tau,b^*}^2 + \bar{\sigma}_{\tau,b^*}^2) (\epsilon H(t_{\tau,b^*} - (\bar{\mu}_{\tau,b^*} + \epsilon)) + (1 - H(t_{\tau,b^*} - \bar{\mu}_{\tau,b^*})))$$

$$\approx \frac{1}{Z} \sum_{b \neq b^*} c_{\tau,b} (\bar{\mu}_{\tau,b}^2 + \bar{\sigma}_{\tau,b}^2) (1 - (1 - \epsilon) H(t_{\tau,b^*} - \bar{\mu}_{\tau,b}))$$

$$+ \frac{1}{Z} c_{\tau,b^*} (\bar{\mu}_{\tau,b^*}^2 + \bar{\sigma}_{\tau,b^*}^2) (1 - (1 - \epsilon) H(t_{\tau,b^*} - \bar{\mu}_{\tau,b^*}))$$

$$\approx \frac{1}{Z} \sum_{b} c_{\tau,b} \nu_{\tau,b} (\bar{\mu}_{\tau,b}^2 + \bar{\sigma}_{\tau,b}^2)$$

$$\text{Var}_{\tilde{p}_{Q,a}}[q] = \mathbb{E}_{\tilde{p}_{Q,a}}[q^2] - (\mathbb{E}_{\tilde{p}_{Q,a}}[q])^2$$

H.4 Experiments

We evaluated the convergence of the algorithm, denoted as ADFQ-V2, to the optimal Q-values as well as its performance during learning in the Maze domain as we did in the main paper (Fig.8). ADFQ-V2 converged to the optimal Q-values in both cases. In the bottom row of the figure, we compared the performance of ADFQ-V2 to that of ADFQ. ADFQ-V2 with $\epsilon$-greedy showed a similar performance, but ADFQ-V2 with BS showed a slower convergence to the optimal performance (3.0). This may be due to the stronger assumption on the variance in the algorithm update. However, ADFQ-V2 is computationally cheaper than ADFQ, $O(|A|)$, and it sometimes takes less absolute time to reach an optimal performance than ADFQ in some domains.
Figure 8: Top: Root Mean Square Error (RMSE) of $\mu$ from the optimal Q-values in deterministic (left) and in stochastic (right) Maze. Bottom: Greedy evaluation plots during learning. The curves were smoothed by a moving average with window 4 in deterministic (left), stochastic (right) Maze.