# Bayesian Q-learning with Assumed Density Filtering

**Heejin Jeong** **Daniel D. Lee**
Department of Electrical and Systems Engineering
University of Pennslyvania
Philadelphia, PA 19104
{heejinj, ddlee}@seas.upenn.edu

## Abstract

While off-policy temporal difference methods have been broadly used in reinforcement learning due to their efficiency and simple implementation, their Bayesian counterparts have been relatively understudied. In this paper, we introduce a new Bayesian approach to off-policy TD methods using *Assumed Density Filtering*, called *ADFQ*, which updates beliefs on action-values (Q) through an online Bayesian inference method. Uncertainty measures in the beliefs not only are used in exploration but they provide a natural regularization in the belief updates. Our empirical results show the ADFQ outperforms comparing algorithms in several task domains. Moreover, our algorithms improve general drawbacks in BRL such as efficiency, usage of uncertainty, and nonlinearity.

## 1 Introduction

In reinforcement learning (RL), a learning subject seeks an optimal behavior by interacting with an environment which maximizes a *value* of a state - a sum of expected future outcomes starting from the state. Bayesian Reinforcement Learning (BRL) is one of the approaches in RL that deploys Bayesian inference in order to incorporate new information into prior. Various algorithms have been proposed in BRL [4, 16, 15, 5, 11, 3, 6, 7, 8, 2, 9]. However, to our knowledge, only few Bayesian approaches to *off-policy temporal difference (TD) learning* have been studied compared to other methods due to the non-linearity in the Bellman optimality equation. Yet off-policy TD methods have been widely used in the standard RL (e.g. Q-learning). One of the most recent influential algorithms in Bayesian off-policy TD learning would be KTD-Q extended from Kalman Temporal Difference (KTD) [8]. KTD approximates the value function (hidden states) using the Kalman filtering scheme. Although the KTD framework handles some important features in RL, it requires many hyperparameters and is computationally expensive. Another limitation is that it was proposed under a deterministic environment assumption and it was not extended for a stochastic case [8].

This paper presents a novel approximated Bayesian off-policy TD learning algorithm, termed as ADFQ, in finite state and action spaces which updates beliefs on Q-values and approximates their posteriors using an online Bayesian inference algorithm known as assumed density filtering (ADF) [14, 1, 13]. ADF, also known as *moment matching*, *online Bayesian learning*, and *weak marginalization*, is a general technique for approximating a true posterior ($\hat{p}$) to a tractable parametric distribution ($p$) in Bayesian networks by minimizing the reverse *Kullback-Leibler* divergence between them, $KL(\hat{p}||p)$. In the proposed ADFQ algorithms, ADF is used to solve the problem of the posterior inconsistency caused by the max operator in the Bellman optimality equation. We proposed two variants in ADFQ, *ADFQ-Numeric* and *ADFQ-Approx*, in terms of a way of computing approximation statistics. We experimented our algorithms on two discrete domains, and compared them with Q-learning and KTD-Q. It consistently outperformed the comparing algorithms on all the domains. We showed that ADFQ improved some of major drawbacks of BRL such as computational complexity as well as Q-learning could be a special case of ADFQ-Approx.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

## 2 Bayesian Q-learning with Assumed Density Filtering

### 2.1 Q-learning

RL problems can be formulated as a Markov Decision Process (MDP) described as a tuple, $M =< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$ where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to I\!R$ is the state transition probability kernel, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to I\!R$ is a reward function, and $\gamma \in [0, 1]$ is a discount factor. The *value* function under a policy $\pi$ is defined as $V^\pi(s) = \mathbf{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t)|s_0 = s]$ for all $s \in \mathcal{S}$. The *action-value* function is defined similarly, $Q^\pi(s, a) = \mathbf{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t)|s_0 = s, a_0 = a]$ for all $s \in \mathcal{S}, a \in \mathcal{A}$. The objective in RL is to find an optimal policy $\pi^* = \text{argmax}_\pi V^\pi$. This requires finding the optimal values, $V^*(\cdot)$ and $Q^*(\cdot, \cdot)$, solving the Bellman optimality equation:

$$Q^*(s, a) = \mathbf{E}_{s' \sim P(\cdot|s,a)}[R(s, a) + \gamma \max_{a' \in \mathcal{A}} Q(s', a')] \qquad V^*(s) = \max_a Q^*(s, a) \quad \forall s \in \mathcal{S} \tag{1}$$

where $s'$ is the subsequent state after executing the action $a$ at the state $s$.

*Q-learning* is the most popular off-policy TD learning technique due to its relatively easy implementation and guarantee of convergence to an optimal policy [19, 12]. Q-learning updates an action-value of the current state and action pair $Q(s, a)$ after observing a reward $R(s, a)$ and the next state $s'$ (one-step TD learning). The update is based on *TD error* - a difference between the *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 in Eq.2.

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left( R(s, a) + \gamma \max_b Q(s', b) - Q(s, a) \right) \tag{2}$$

### 2.2 Belief Updates on Q-values

We define $Q_{s,a} \sim \mathcal{N}(\mu_{s,a}, \sigma^2_{s,a})$ as a Gaussian random variable with mean $\mu_{s,a}$ and variance $\sigma^2_{s,a}$ corresponding to $Q(s, a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. We assume that the random variables for different states and actions are independent and have different mean and variance. Similarly, we can define a random variable for $V(s)$ as $V_s = \max_a Q_{s,a}$. We assume that $V_s$ is independent of $\{Q_{s,a}\}_{\forall a \in \mathcal{A}}$ given the related parameters $\{\mu_{s,a}, \sigma^2_{s,a}\}_{\forall a \in \mathcal{A}}$. In general, the maximum of Gaussian random variables is not Gaussian (see the Appendix.A).

In the Bayesian perspective of the one-step TD learning, the beliefs on $\mathbf{Q} = \{Q_{s,a}\}_{\forall s \in \mathcal{S}, \forall a \in \mathcal{A}}$ can be updated at time $t$ after observing a reward $r_t$ and the next state $s_{t+1}$ using the Bayes rule. In order to reduce notation, we drop the dependency on $t$ denoting $s_t = s$, $a_t = a$, $s_{t+1} = s'$, $r_t = r$, and also define a causally related 4-tuple $\tau =< s, a, r, s' >$. In the one-step TD learning, the likelihood becomes $p(r + \gamma V_{s'}|\mathbf{q}, \theta) = p_{V_{s'}}((q - r)/\gamma|s', \mathbf{q}, \theta)$ where $\mathbf{q}$ corresponds to $\mathbf{Q}$ and $q$ is a value in $\mathbf{q}$ corresponding to $Q_{s,a}$. $\theta$ is a set of mean and variance of $\mathbf{Q}$. From the independence assumptions on $\mathbf{Q}$ and $\{V_s\}_{\forall s \in \mathcal{S}}$, the posterior update can be reduced to an update only for the belief on $Q_{s,a}$: $\hat{p}_{Q_{s,a}}(q|\theta, r, s') \propto p_{V_{s'}}(\,(q - r)/\gamma|\, q, s', \theta)\, p_{Q_{s,a}}(q|\theta)$. With the distributions over $V_{s'}$ and $Q_{s,a}$, the resulting posterior distribution is (derivation details in the Appendix.B):

$$\hat{p}_{Q_{s,a}}(q|\theta, r, s') = \frac{1}{Z} \sum_{b \in \mathcal{A}} \frac{c_{\tau,b}}{\bar{\sigma}_{\tau,b}} \phi\left( \frac{q - \bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}} \right) \prod_{b' \in \mathcal{A}, b' \neq b} \Phi\left( \frac{q - (r + \gamma \mu_{s',b'})}{\gamma \sigma_{s',b'}} \right) \tag{3}$$

where $\phi(\cdot)$ is the standard Gaussian probability density function (PDF) and $\Phi(\cdot)$ is the standard Gaussian cumulative distribution function (CDF). $Z$ is a normalization constant, $c_{\tau,b} = \phi\left( (r + \gamma \mu_{s',b} - \mu_{s,a}) \,/ \sqrt{\sigma^2_{s,a} + \gamma^2 \sigma^2_{s',b}} \right) / \sqrt{\sigma^2_{s,a} + \gamma^2 \sigma^2_{s',b}}$, and

$$\bar{\mu}_{\tau,b} = \bar{\sigma}^2_{\tau,b} \left( \frac{\mu_{s,a}}{\sigma^2_{s,a}} + \frac{r + \gamma \mu_{s',b}}{\gamma^2 \sigma^2_{s',b}} \right) \qquad \bar{\sigma}^2_{\tau,b} = \left( \frac{1}{\sigma^2_{s,a}} + \frac{1}{\gamma^2 \sigma^2_{s',b}} \right)^{-1} \tag{4}$$

Note that the TD errors, $(r + \gamma \mu_{s',b}) - \mu_{s,a}$, is naturally incorporated as a penalty in the weights of the summation of the posterior, $c_{\tau,b}$, with the form of Gaussian PDF. Unlike the Q-learning algorithm in Eq.2, all actions are considered for the subsequent state $s'$ in here and a subsequent action which results a smaller TD error contributes to the update more. In addition, since $\bar{\mu}_{\tau,b}$ is an inverse-variance weighted average, it is closer to the prior mean if uncertainty of the prior is smaller than that of a target, and vice versa.

However, the updated posterior does not belong to its original parametric family. We approximate the posterior to a Gaussian distribution using ADF. When the parametric family is a spherical Gaussian, it is easily shown that $\mu^* = \mathbf{E}_{\mathbf{q}\sim\hat{p}_{Q_{s,a}}(\cdot)}[\mathbf{q}]$ and $\sigma^{*2} = \mathrm{Var}_{\mathbf{q}\sim\hat{p}_{Q_{s,a}}(\cdot)}[\mathbf{q}]$. It is fairly easy to analytically derive the mean and the variance of the true posterior (Eq.3) when $|\mathcal{A}| = 2$ (derived in the Appendix.C). However, to our knowledge, when $|\mathcal{A}| > 2$, solutions become too complicated or are not known. In the next section, we present an approximated ADFQ algorithm which provides analytic solutions for the mean and the variance as well as reduces the algorithmic complexity.

## 3 Approximated ADFQ

When $\sigma$ in $X \sim \mathcal{N}(\mu, \sigma^2)$, 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 $s \in \mathcal{S}$ and $a \in \mathcal{A}$. The product of the Gaussian CDFs in the Eq.3 is approximated to 1 if $q \geq r + \gamma\mu_{s',b'}$ for all $b' \in \mathcal{A}, b' \neq b$, and 0 otherwise. However, when $q = \bar{\mu}_{\tau,b}$, we cannot simply apply the approximation since the PDF approaches infinity. Therefore, we define a function $f(\cdot)$ which is the approximation of the single term in the summation of Eq.3 and $f(q;\mu,\sigma) \equiv \mathcal{N}(q|\mu,\sigma)$ for $q \in [\mu-\epsilon, \mu+\epsilon], \epsilon \ll 1$ and 0 otherwise. Then, the posterior distribution is approximated to $p_{Q_{s,a}}(q|\theta', r, s') \approx \hat{p}_{Q_{s,a}}(q)$,

$$\hat{p}_{Q_{s,a}}(q) = \frac{1}{Z}\sum_{b\in\mathcal{A}} c_{\tau,b} f(q;\bar{\mu}_{\tau,b},\bar{\sigma}_{\tau,b}) \quad \text{for } q \in (-\infty,+\infty) \tag{5}$$

where the normalization factor $Z$ is $\sum_b c_{\tau,b}$. Further details on the approximation can be found in the Appendix.D. From integrals $\int x\phi(x)dx = -\phi(x) + C$ and $\int x^2\phi(x)dx = -x\phi(x) + \Phi(x) + C$, we obtain the mean and the variance of $\hat{p}_{s,a}(q)$,

$$\mathbf{E}_{q\sim\hat{p}_{Q_{s,a}}(\cdot)}[q] = \frac{\sum_b c_{\tau,b}\bar{\mu}_{\tau,b}}{\sum_b c_{\tau,b}} \qquad \mathrm{Var}_{q\sim\hat{p}_{Q_{s,a}}(\cdot)}[q] = \frac{\sum_b c_{\tau,b}\bar{\sigma}^2_{\tau,b}}{\sum_b c_{\tau,b}} \tag{6}$$

Note that the mean and variance of the approximated posterior are simply weighted sums of $\bar{\mu}_{\tau,b}$ and $\bar{\sigma}^2_{\tau,b}$ for all actions in $\mathcal{A}$, respectively, with weights dependent to TD error. We call the ADFQ algorithm which uses the approximated mean and variance as *ADFQ-Approx* and the ADFQ algorithm which numerically computes the mean and the variance directly from Eq.3 as *ADFQ-Numeric*.

**Algorithmic Complexity.** The space complexity for both ADFQ-Numeric and ADFQ-Approx is $O(|\mathcal{S}||\mathcal{A}|)$. The computational complexity of ADFQ-Numeric at one time step is $O(m|\mathcal{A}|)$ where $m$ is the number of samples for the numerical computation. The computational complexity of ADFQ-Approx is $O(|\mathcal{A}|)$ which is as efficient as the Q-learning algorithm. Both ADFQ-Numeric and ADFQ-Approx are more efficient than KTD-Q which computational complexity is $O(|\mathcal{S}|^2|\mathcal{A}|^3)$ and space complexity is $O(|\mathcal{S}|^2|\mathcal{A}|^2)$ in finite state and action spaces.

**Connection to Q-learning.** We can relate this result to the conventional Q-learning since the mean of the posterior is a linear combination of the prior mean (current $Q$) and the target means. Suppose that $c_{\tau,b} = 0$ for all $b \neq b^*$ in the summation of the approximated mean in Eq.6. Then we can define $\bar{\alpha}$ which corresponds to the learning rate in the conventional Q-learning as the below Eq.7. It provides an intuitive learning rate which converges to 1 when $\sigma_{s,a} \gg \sigma_{s',b^*}$ and to 0 when $\sigma_{s,a} \ll \sigma_{s',b^*}$.

$$\bar{\alpha} \equiv \frac{\bar{\sigma}^2_{b^*}}{\gamma^2\sigma^2_{s',b^*} + \bar{\sigma}^2_{b^*}} = \left(1 + \left(\frac{\gamma\sigma_{s',b^*}}{\sigma_{s,a}}\right)^2\right)^{-1} \tag{7}$$

## 4 Experiments

The ADFQ algorithms were tested with three different action policies: *Bayesian Sampling (BS)* which selects $a_t = \mathrm{argmax}_a\, q_{s_t,a}$ where $q_{s_t,a} \sim p_{Q_{s_t,a}}(\cdot|\theta_t)$, *semi-BS* which performs *BS* with a small probability and greedily selects the best action otherwise, and *$\epsilon$-greedy*. For compared algorithms, we experimented Q-learning with $\epsilon$-greedy and Boltzmann action policies and KTD-Q with $\epsilon$-greedy and its active learning scheme [8]. The initial covariance of the process and the observation noises in KTD-Q are set to be $0I$ and 1 respectively, following the original publication. For consistency, we initialized $Q$-values and mean parameters with $r_0/(1-\gamma)$ after the first nonzero reward $r_0$ was observed. For all algorithms, the discount factor was $\gamma = 0.9$. All other hyperparameters were selected through cross-validation and are reported in the Appendix.E.1.

Table 1: The mean sum of rewards and confidence interval over 10 trials for the best parameters

| | Q-learning | | ADFQ-Approx | | | KTD-Q | |
|---|---|---|---|---|---|---|---|
| | $\epsilon$-greedy | Boltzmann | $\epsilon$-greedy | semi-BS | BS | $\epsilon$-greedy | active |
| Loop | $302.3 \pm 12.17$ | $288.2 \pm 17.4$ | $\mathbf{338.0 \pm 0.0}$ | $329.2 \pm 13.8$ | $333.2 \pm 3.2$ | $281.6 \pm 5.2$ | $157.4 \pm 7.4$ |
| Maze | $239.7 \pm 81.4$ | $106.1 \pm 10.4$ | $\mathbf{274.8 \pm 80.3}$ | $264.0 \pm 67.3$ | $180.9 \pm 47.8$ | $20.5 \pm 16.4$ | $55.4 \pm 8.6$ |

We tested our algorithms in two domains. **Loop** ($T_H = 1000$) is a non-episodic domain with deterministic state transition used in [3]. It consists of 9 discrete states and 2 actions (a,b). There are +1 reward at $s = 4$ and +2 reward at $s = 8$. **Mini-Maze** ($T_H = 5000$) is designed inspired by Dearden's Maze [3] as shown in Fig.1 since the KTD-Q algorithm was not able to handle the Dearden's Maze in reasonable computational time. It is an episodic and stochastic domain with 112 states and 4 cardinal actions. The agent slips with a probability 0.1 and performs an action right-perpendicular to the original action. The agent starts at "S" and its goal is to collect the flags located in "F" and escape the maze through the goal state "G". It receives a reward at "G" equivalent to the number of flags it has collected. The Black blocks represent wall and the agent dose not move if it performs an action toward a wall.

The sum of rewards ($\sum_{t=0,\cdots,T_H} r_t$) obtained during learning is displayed in Table.1. For simplification, the results of ADFQ-Numeric are reported in the Appendix.E.2. It demonstrates that ADFQ-Approx with $\epsilon$-greedy and semi-BS action policies outperforms other algorithms. We also evaluated each algorithm semi-greedily (used $\epsilon$-greedy with $\epsilon = 0.1$) at every $T_H/100$ steps with the current policy during learning. The evaluation was repeated 10 times and each trial was terminated after $T_H/50$ steps and/or when a goal state was reached. ADFQ-Approx showed similar performances with Q-learning in the Loop but it slightly outperformed Q-learning in the Mini-Maze (Fig.1). In both domains, KTD-Q performed worse than others, especially in the larger domain.

## 5 Discussion

We proposed an approach to Bayesian off-policy TD method called ADFQ that surpassed the performance of Q-learning and KTD-Q in various task domains. The ADFQ demonstrates several intriguing results. First, unlike the Q-learning algorithm, the ADFQ incorporates the information of all possible actions for the next state in their update. Second, we made use of our uncertainty measures not only in exploration but also in the value update as natural regularization based on the current beliefs (Eq.4). Third, we were able to connect ADFQ-Approx algorithm to Q-learning and showed Q-learning could be a special case of our algorithm. Lastly, unlike other BRL approaches, ADFQ is computationally efficient and it requires only one additional hyperparameter, initial variance, than Q-learning.

There are several limitations in the proposed ADFQ. Convergence analysis is not provided in this paper and we considered only finite state and action spaces. We are currently extending our method to continuous domains in order to apply the method to a real world example.

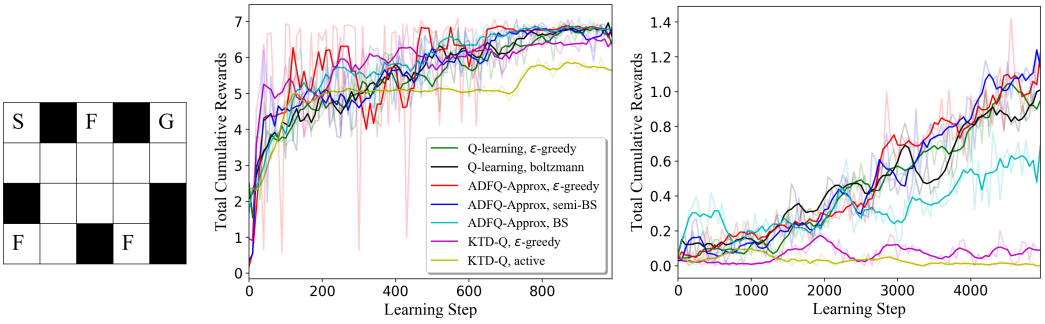


Figure 1: Left: Mini-Maze domain diagram. Middle: Semi-greedy evaluation on the Loop domain every $T_H/100$ steps during learning at each domain, averaged over 10 trials for each algorithm with an action selection rule. The curves were smoothed. Right: Evaluation on the Mini-Maze domain.

## References


[1] 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.

[2] 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.

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

[4] 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.

[5] M. Duff. Optimal learning: Computational procedures for bayes-adaptive markov decision processes. *PhD thesis, University of Massachusetts, Amherst*, 2002.

[6] 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.

[7] 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.

[8] M. Geist and P. Olivier. Kalman temporal differences. *Journal of artificial intelligence research*, 39: 483–532, 2010.

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

[10] 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.

[11] 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.

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

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

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

[15] 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.

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

[17] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1988.

[18] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

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

## A Maximum of Gaussian Random Variables

The distribution of the maximum of Gaussian random variables, $M = \max_{1\leq k\leq N} X_k$ where $X_k \sim \mathcal{N}(\mu_k, \sigma_k^2)$ for $1 \leq k \leq N$, is derived as follow:

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

$$p\left(\max_{1\leq k\leq N} X_k = x\right) = \sum_{i=1}^{N} \frac{1}{\sigma_i}\phi\left(\frac{x-\mu_i}{\sigma_i}\right)\prod_{i\neq j}^{N} \Phi\left(\frac{x-\mu_j}{\sigma_j}\right) \neq \text{Gaussian}$$

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 section 2 of the main paper, we have shown that

$$\hat{p}_{Q_{s,a}}(q|\theta, r, s') = \frac{1}{Z} p_{V_{s'}}\left(\frac{q-r}{\gamma}\middle| q, s', \theta\right) p_{Q_{s,a}}(q|\theta)$$

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

$$\begin{aligned}
\hat{p}_{Q_{s,a}}(q) &= \frac{1}{Z}\sum_{b\in\mathcal{A}} \frac{1}{\sigma_{s',b}}\phi\left(\frac{q-(r+\gamma\mu_{s',b})}{\gamma\sigma_{s',b}}\right)\prod_{b'\neq b, b'\in\mathcal{A}} \Phi\left(\frac{q-(r+\gamma\mu_{s',b'})}{\gamma\sigma_{s',b'}}\right)\frac{1}{\sigma_{s,a}}\phi\left(\frac{q-\mu_{s,a}}{\sigma_{s,a}}\right) \\
&= \frac{1}{Z\sqrt{2\pi}\sigma_{s,a}}\sum_{b\in\mathcal{A}} \frac{1}{\sigma_{s',b}} e^{-\frac{(\mu_{s,a}-(r+\gamma\mu_{s',b}))^2}{2(\sigma_{s,a}^2+\gamma^2\sigma_{s',b}^2)}}\phi\left(\frac{q-\bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\right)\prod_{b'\neq b, b'\in\mathcal{A}} \Phi\left(\frac{q-(r+\gamma\mu_{s',b'})}{\gamma\sigma_{s',b'}}\right) \\
&= \frac{1}{Z}\sum_{b\in\mathcal{A}} \frac{c_{\tau,b}}{\bar{\sigma}_{\tau,b}}\phi\left(\frac{q-\bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\right)\prod_{b'\neq b, b'\in\mathcal{A}} \Phi\left(\frac{q-(r+\gamma\mu_{s',b'})}{\gamma\sigma_{s',b'}}\right) \qquad (8)
\end{aligned}$$

The mean and variance of the posterior distribution when $|\mathcal{A}| = 2$ can be found using the results in the next section.

## C Moment of $p(X = \max(X_1, X_2))$

The distribution of the maximum of $N$ Gaussian random variables $\{X_k|X_k \sim \mathcal{N}(\mu_k, \sigma_k^2), k = 1, \cdots, N\}$ is:

$$p(X_M = \max_k(X_k) = x) = \sum_i \frac{1}{\sigma_i}\phi\Big(\frac{x-\mu_i}{\sigma_i}\Big)\prod_{i\neq j}\Phi\Big(\frac{x-\mu_j}{\sigma_j}\Big)$$

and the moment of $X_M$ is:

$$\begin{aligned}
M(t) &= \int_{\infty}^{\infty} e^{tx}\sum_i \frac{1}{\sigma_i}\phi\Big(\frac{x-\mu_i}{\sigma_i}\Big)\prod_{i\neq j}\Phi\Big(\frac{x-\mu_j}{\sigma_j}\Big)dx \\
&= \sum_i \eta_i(t)\int_{\infty}^{\infty}\frac{1}{\sigma_i}\phi\Big(\frac{x-(\mu_i+t\sigma_i^2)}{\sigma_i}\Big)\prod_{i\neq j}\Phi\Big(\frac{x-\mu_j}{\sigma_j}\Big)dx
\end{aligned}$$

where

$$\eta_i(t) = \exp\left\{\mu_i t + \frac{t^2\sigma_i^2}{2}\right\}$$

When $N = 2$, the moment is:

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

Let $M(t) = M_1(t) + M_2(t)$ and differentiate with respect to one of the means. For the first term,

$$\begin{aligned}
\frac{\partial M_1(t)}{\partial\mu_2} &= -\frac{\eta_1(t)}{\sigma_1\sigma_2}\int_{-\infty}^{\infty}\phi\Big(\frac{x-\mu_1'}{\sigma_1}\Big)\phi\Big(\frac{x-\mu_2}{\sigma_2}\Big)dx \\
&= -\frac{\eta_1(t)\sigma_{12}}{\sqrt{2\pi}\sigma_1\sigma_2}\exp\left\{-\frac{1}{2}\frac{(\mu_2-\mu_1')^2}{\sigma_1^2+\sigma_2^2}\right\}
\end{aligned}$$

If we integrate the above equation with respect to $\mu_2$ again,

$$\begin{aligned}
M_1(t) &= \int \frac{\partial M_1(t)}{\partial \mu_2} d\mu_2 \\
&= -\frac{\eta_1(t)\sigma_{12}}{\sigma_1\sigma_2}\sqrt{\sigma_1^2+\sigma_2^2}\int \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}}\exp\left\{-\frac{(\mu_2-\mu_1')^2}{2(\sigma_1^2+\sigma_2^2)}d\mu_2\right\} \\
&= -\eta_1(t)\Phi\left(\frac{\mu_2-\mu_1'}{\sqrt{\sigma_1^2+\sigma_2^2}}\right) \\
&= \exp\left\{\mu_1 t + \frac{t^2\sigma_1^2}{2}\right\}\Phi\left(\frac{\mu_1+t\sigma_1^2-\mu_2}{\sqrt{\sigma_1^2+\sigma_2^2}}\right)
\end{aligned}$$

From this result, we can find its mean and variance by differentiating with respect to $t$. For example, for the mean,

$$\begin{aligned}
\mathbf{E}(X_M) &= \frac{d}{dt}\left(M_1(t)+M_2(t)\right)\Big|_{t=0} \\
&= \mu_1\Phi\left(\frac{\mu_1-\mu_2}{\sqrt{\sigma_1^2+\sigma_2^2}}\right)+\mu_2\Phi\left(\frac{\mu_2-\mu_1}{\sqrt{\sigma_1^2+\sigma_2^2}}\right)+\sqrt{\sigma_1^2+\sigma_2^2}\phi\left(\frac{\mu_1-\mu_2}{\sqrt{\sigma_1^2+\sigma_2^2}}\right)
\end{aligned}$$

# D Posterior Approximation for Small Variance

In the section 4 of the main paper, we considered a case in which $\sigma_{s,a} \ll 1 \forall s \in \mathcal{S}, a \in \mathcal{A}$ and defined the function $f(\cdot)$ for the product of a Gaussian PDF and Gaussian CDFs (see the main paper for details):

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

The posterior distribution in Eq.8 is approximated when the variance values are very small in three different range of $q$. Let $b^* \equiv \text{argmax}_{b\in\mathcal{A}}\,\mu_{s',b}$ and $b^{**} \equiv \text{argmax}_{b\in\mathcal{A}, b\neq b^*}\,\mu_{s',b}$. When $q \in (-\infty, r+\gamma\mu_{s',b^{**}})$, the product of the CDFs always approaches to 0, and thus the posterior becomes:

$$\hat{p}_{Q_{s,a}}(q) \approx \frac{1}{Z}\sum_i c_i f(q;\bar{\mu}_i)$$

When $q \in [r+\gamma\mu_{s',b^{**}}, r+\gamma\mu_{s',b^*})$, the product of CDFs approaches to 0 except when $b' = b^*$.

$$\hat{p}_{Q_{s,a}}(q) \approx \frac{1}{Z}\left\{c_{\tau,b^*}\frac{1}{\bar{\sigma}_{\tau,b^*}}\phi\left(\frac{q-\bar{\mu}_{\tau,b^*}}{\bar{\sigma}_{\tau,b^*}}\right)+\sum_{b\neq b^*}c_{\tau,b}f(q;\bar{\mu}_{\tau,b})\right\}$$

Finally, for $q \in [r+\gamma\mu_{s',b^*}, +\infty)$, the product of CDFs always approaches to 1.

$$\hat{p}_{Q_{s,a}}(q) \approx \frac{1}{Z}\sum_b c_{\tau,b}\frac{1}{\bar{\sigma}_{\tau,b}}\phi\left(\frac{q-\bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\right)$$

However, since $\bar{\sigma}_{\tau,b} \ll 1$ when $\sigma_{s,a} \ll 1$ and $\sigma_{s',b} \ll 1$ for all $b \in \mathcal{A}$, the Gaussian PDF is approximately equal to $f(\cdot)$:

$$\frac{1}{\bar{\sigma}_{\tau,b}}\phi\Big(\frac{q-\bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\Big) \approx f(q;\bar{\mu}_{\tau,b})$$

Therefore, the posterior distribution is approximated to

$$\hat{p}_{Q_{s,a}}(q) \approx \frac{1}{Z}\sum_b c_{\tau,b}f(q;\bar{\mu}_{\tau,b}) \quad \text{for } q \in (-\infty,+\infty) \tag{9}$$

The normalization factor is:

$$\begin{aligned}
Z &= \sum_{b\in\mathcal{A}} c_{\tau,b}\int_{-\infty}^{\infty} f(q;\bar{\mu}_{\tau,b})dq \\
&= \sum_{b\in\mathcal{A}} c_{\tau,b}\int_{\bar{\mu}_{\tau,b}-\epsilon}^{\bar{\mu}_{\tau,b}+\epsilon}\frac{1}{\bar{\sigma}_{\tau,b}}\phi\left(\frac{q-\bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\right)dq \\
&= \sum_{b\in\mathcal{A}} c_{\tau,b}\Big(\Phi\Big(\frac{\epsilon}{\bar{\sigma}_{\tau,b}}\Big)-\Phi\Big(-\frac{\epsilon}{\bar{\sigma}_{\tau,b}}\Big)\Big) \\
\lim_{\sigma_{s,a},\sigma_{s',b}\to 0} Z &= \sum_{b\in\mathcal{A}} c_{\tau,b}
\end{aligned} \tag{10}$$

| Algorithm | Parameter | Loop | Mini-Maze |
|---|---|---|---|
| Q-learning, $\epsilon$-greedy | $\epsilon$ | 0.05 | 0.05 |
| | $\alpha$ | 0.5 | 0.5 |
| Q-learning, Boltzmann | $\tau$ | 0.1 | 0.3 |
| | $\alpha$ | 0.5 | 0.5 |
| ADFQ-Numeric, $\epsilon$-greedy | $\epsilon$ | 0.0 | 0.2 |
| | $\sigma_0^2$ | 0.1 | 0.1 |
| ADFQ-Approx, $\epsilon$-greedy | $\epsilon$ | 0.0 | 0.0 |
| | $\sigma_0^2$ | 100.0 | 10.0 |
| ADFQ-Numeric, semi-BS | $\epsilon$ | 0.05 | 0.15 |
| | $\sigma_0^2$ | 0.1 | 0.1 |
| ADFQ-Approx, semi-BS | $\epsilon$ | 0.05 | 0.05 |
| | $\sigma_0^2$ | 10.0 | 100.0 |
| ADFQ-Numeric, BS | $\sigma_0^2$ | 0.1 | 0.1 |
| ADFQ-Approx, BS | $\sigma_0^2$ | 100.0 | 0.1 |
| KTD-Q, $\epsilon$-greedy | $\epsilon$ | 0.15 | 0.15 |
| | $\sigma_0^2$ | 10.0 | 10.0 |
| KTD-Q, active | $\sigma_0^2$ | 1.0 | 1.0 |

Table 2: The best performing parameters of all algorithms in each domain

# E Experiment Details and Additional Results

## E.1 Best Parameters

We chose the best parameter values via cross-validation. The results are summarized in Table.2. $\alpha$ is the learning rate of Q-learning, $\sigma_0^2$ represents initial variance of ADFQ and KTD-Q algorithms, $\tau$ is the temperature constant in Boltzmann distribution and $\kappa$ is a scaling factor used for Unscented Transform in KTD-Q. The test ranges are: $\alpha = [0.1, 0.3, 0.5]$, $\epsilon = [0.0, 0.05, 0.1, 0.15, 0.2]$, $\tau = [0.1, 0.3, 0.5]$, $\sigma_0^2 = [0.1, 1.0, 10.0]$.

## E.2 Additional Results of ADFQ

The semi-greedy evaluation results of ADFQ-Numeric are shown in Fig.2 and compared with the results ADFQ-Approx. In addition, both algorithms were evaluated with $\gamma = 0.5$. The total cumulative rewards in Table.3 as well as the semi-greedy evaluation plots in Fig.3 show that ADFQ-Numeric performs similarly with ADFQ-Approx unlike the case where $\gamma = 0.9$.

# F Comparison of Approximated Distributions with True Distribution

The true posterior in Eq.8 (TRUE, red), the approximated posteriors in ADFQ-Numeric (green) and in ADFQ-Approx (blue) are compared in Fig.4-8 under five various conditions. We consider a case where $|\mathcal{A}| = 4$. Using the notations in Eq.3, we set $\mu_{s,a} = \mu_{s',b}$ and $\sigma_{s,a} = \sigma_{s',b}$ for all $b \in \mathcal{A}$ except in Fig.8. In Fig.4, the variance values vary across the subfigures. As variance is assumed to be small in ADFQ-Approx, the smaller the variance,

| | Loop ($\gamma = 0.5$) | Loop (0.9) | Mini-Maze (0.5) | Mini-Maze (0.9) |
|---|---|---|---|---|
| ADFQ-Numeric, $\epsilon$-greedy | $381.0 \pm 0.0$ | $325.5 \pm 13.5$ | $309.3 \pm 26.1$ | $187.4 \pm 92.8$ |
| ADFQ-Numeric, semi-BS | $381.3 \pm 1.4$ | $329.5 \pm 0.8$ | $\mathbf{328.2 \pm 35.0}$ | $220.2 \pm 41.1$ |
| ADFQ-Numeric, BS | $380.9 \pm 1.7$ | $328.4 \pm 0.8$ | $283.5 \pm 56.4$ | $204.7 \pm 72.8$ |
| ADFQ-Approx, $\epsilon$-greedy | $\mathbf{384.0 \pm 0.0}$ | $\mathbf{338.0 \pm 0.0}$ | $318.2 \pm 29.2$ | $\mathbf{274.8 \pm 80.3}$ |
| ADFQ-Approx, semi-BS | $\mathbf{384.0 \pm 0.0}$ | $329.2 \pm 13.8$ | $321.3 \pm 17.7$ | $264.0 \pm 67.3$ |
| ADFQ-Approx, BS | $381.9 \pm 1.7$ | $333.2 \pm 3.2$ | $260.7 \pm 85.4$ | $180.9 \pm 47.8$ |

Table 3: The mean and confidence interval of total cumulative rewards over 10 trials with $\gamma = 0.5$ and 0.9. The number of learning steps are: **Loop** - 1000 steps, **Mini-Maze** - 5000 steps

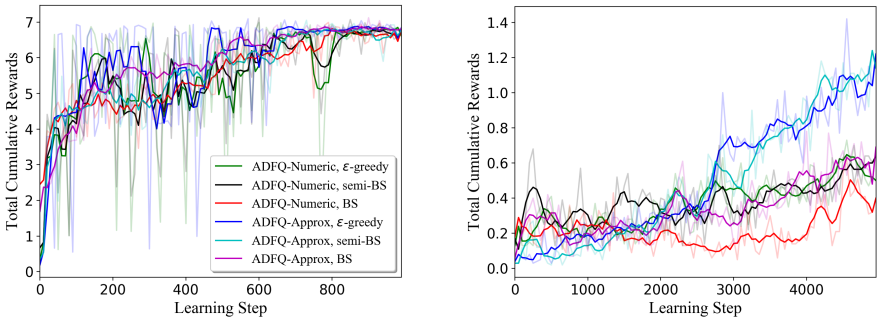


Figure 2: Cumulative rewards in semi-greedy evaluation during learning at each domain, averaged over 10 trials for ADFQ-Numeric and ADFQ-Approx with $\gamma = 0.9$. The curves were smoothed by a simple moving average with window size 4. Left: Loop, Right: Mini-Maze

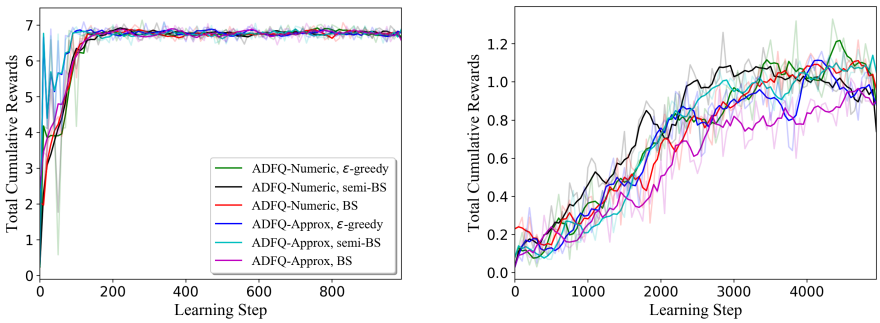


Figure 3: Cumulative rewards in semi-greedy evaluation during learning at each domain, averaged over 10 trials for ADFQ-Numeric and ADFQ-Approx with $\gamma = 0.5$. The curves were smoothed by a simple moving average with window size 4. Left: Loop, Right: Mini-Maze

the better ADFQ-Approx approximates TRUE. In Fig.5, all conditions are the same as those in Fig.4, but the discount factor is $\gamma = 0.5$. An approximated distribution with a smaller discount factor converges to the true distribution quicker. In Fig.6, different mean values were tested with a fixed discount factor and variance. As shown, the larger the mean, the better ADFQ-Approx approximates TRUE. Fig.7 tested on non-zero reward. The result shows that non-zero reward has a negative effect on the approximation and it affects more with a smaller discount factor. Lastly, different values are used for each mean and variance in Fig.8. The left subfigure corresponds to the case when the parameters of $(s, a)$ have been updated more than those of the next state, $(s', b)$ for all $b \in \mathcal{A}$. The variance of ADFQ-Approx is the same as TRUE but the distribution is slightly shifted to the left. In the opposite case (the middle subfigure), the variance values are still same, but ADFQ-Approx distribution is shifted more towards the left. When all the mean and the variance are different (the right subfigure), ADFQ-Numeric fails to approximate TRUE well, but ADFQ-Approx shows a better approximation. In the all cases except the right figure in Fig.8, ADFQ-Numeric shows almost identical distribution to the true posterior. Therefore, we can infer that ADF methods reasonably approximate the true posterior distribution.

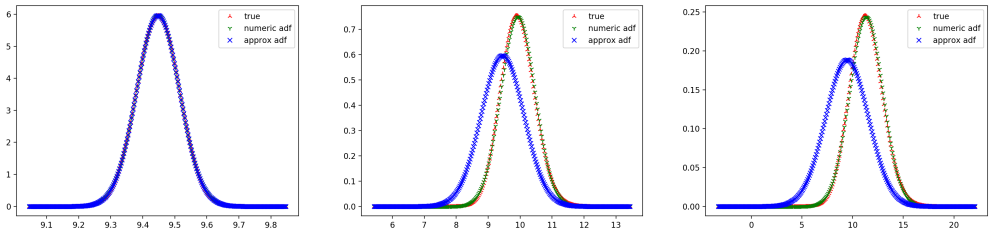


Figure 4: Posterior distributions drawn by TRUE (red), ADFQ-Numeric (green), and ADFQ-Approx (blue) with $r = 0$, $\gamma = 0.9$, $\mu_{s,a} = \mu_{s',b\in\mathcal{A}} = 10.0$. **Left:** $\sigma^2_{s,a} = \sigma^2_{s',b\in\mathcal{A}} = 0.01$, **Middle:** $\sigma^2_{s,a} = \sigma^2_{s',b\in\mathcal{A}} = 1.0$, **Right:** $\sigma^2_{s,a} = \sigma^2_{s',b\in\mathcal{A}} = 10.0$

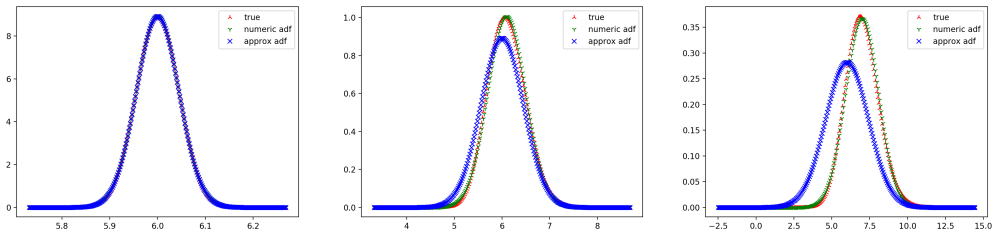


Figure 5: Posterior distributions drawn by TRUE (red), ADFQ-Numeric (green), and ADFQ-Approx (blue) with $r = 0$, $\gamma = 0.5$, $\mu_{s,a} = \mu_{s',b\in\mathcal{A}} = 10.0$. **Left:** $\sigma^2_{s,a} = \sigma^2_{s',b\in\mathcal{A}} = 0.01$, **Middle:** $\sigma^2_{s,a} = \sigma^2_{s',b\in\mathcal{A}} = 1.0$, **Right:** $\sigma^2_{s,a} = \sigma^2_{s',b\in\mathcal{A}} = 10.0$

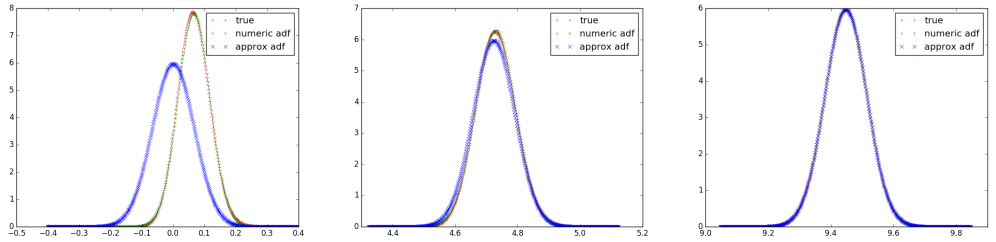


Figure 6: Posterior distributions drawn by TRUE (red), ADFQ-Numeric (green), and ADFQ-Approx (blue) with $r = 0$, $\gamma = 0.9$, $\sigma^2_{s,a} = \sigma^2_{s',b\in\mathcal{A}} = 0.01$. **Left:** $\mu_{s,a} = \mu_{s',b\in\mathcal{A}} = 0.0$, **Middle:** $\mu_{s,a} = \mu_{s',b\in\mathcal{A}} = 5.0$, **Right:** $\mu_{s,a} = \mu_{s',b\in\mathcal{A}} = 10.0$

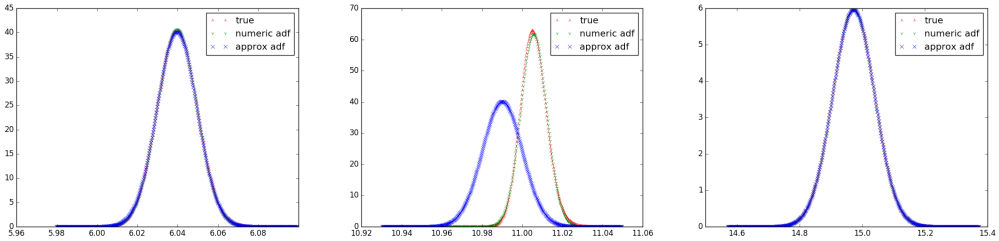


Figure 7: Posterior distributions drawn by TRUE (red), ADFQ-Numeric (green), and ADFQ-Approx (blue) with $\mu_{s,a} = \mu_{s',b\in\mathcal{A}} = 10.0$, $\sigma^2_{s,a} = \sigma^2_{s',b\in\mathcal{A}} = 0.01$. **Left:** $\gamma = 0.1$, $r = 5.0$, **Middle:** $\gamma = 0.1$, $r = 10.0$, **Right:** $\gamma = 0.9$, $r = 10.0$,

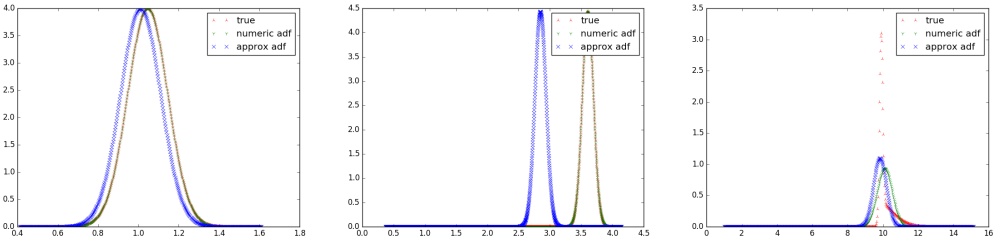


Figure 8: Posterior distributions drawn by TRUE (red), ADFQ-Numeric (green), and ADFQ-Approx (blue) with $r = 0$, $\gamma = 0.9$. **Left:** Relatively small $\mu_{s,a} = 1.0$ and $\sigma^2_{s,a} = 0.01$ compared to $\mu_{s',b\in A} = [10.0, 12.0, 11.0, 10.0]$ and $\sigma^2_{s',b\in A} = 10.0$, **Middle:** Relatively large $\mu_{s,a} = 10.0$ and $\sigma^2_{s,a} = 10.0$ compared to $\mu_{s',b\in A} = [1.0, 2.0, 1.0, 4.0]$ and $\sigma^2_{s',b\in A} = 0.01$, **Right:** Arbitrary, $\mu_{s,a} = 9.0$, $\sigma^2_{s,a} = 1.0$, $\mu_{s',b\in A} = [11.0, 2.0, 15.0, 4.0]$, $\sigma^2_{s',b\in A} = [0.01, 1.0, 10.0, 0.1]$