

---

# Transferring Value Functions by Variational Methods

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

We consider the problem of transferring value functions in reinforcement learning. We propose an approach that uses the given source tasks to learn a prior distribution over optimal value functions and provide an efficient variational approximation of the corresponding posterior in a new target task. We show our approach to be general, in the sense that it can be combined with complex parametric function approximators and distribution models, while providing two practical algorithms based on Gaussians and Gaussian mixtures. We theoretically analyze both by providing a finite-sample analysis and evaluate them empirically in four different domains.

Say something more about the advantages of our method

## 1 Introduction

Recent advancements have allowed reinforcement learning (RL) [34] to achieve impressive results in a wide variety of complex tasks, ranging from Atari [26] through the game of Go [33] to the control of sophisticated robotics systems [17, 23, 22]. The main limitation is that these RL algorithms still require an enormous amount of experience samples before successfully learning such complicated tasks. One of the most promising solutions to reduce the need of samples is transfer learning, which focuses on reusing past knowledge available to the agent in order to reduce the sample-complexity for learning new tasks. In the typical settings of transfer in RL [36], the agent is assumed to have already solved a set of *source tasks* generated from some unknown distribution. Then, given a *target task* (which is drawn from the same distribution, or a slightly different one), the agent can rely on knowledge from the source tasks to speed up the learning process. This reuse of knowledge constitutes a significant advantage over plain RL, where the agent learns each new task from scratch independently of any previous learning experience. Several algorithms have been proposed in the literature to transfer different elements involved in the learning process: experience samples [21, 35], policies/options [12, 19], rewards [18], features [5], parameters [11, 16], and so on. We refer the reader to [36] for a thorough survey on transfer in RL.

Under the assumption that tasks follow a specific distribution, an intuitive choice for designing a transfer algorithm is to attempt at characterizing the uncertainty over the target task. Then, an ideal algorithm would leverage prior knowledge from the source tasks to interact with the target task to reduce the uncertainty as quickly as possible. This simple intuition makes Bayesian methods appealing approaches for transfer in RL, and many previous works have been proposed in this direction. In [38], the authors assume tasks share similarities in their dynamics and rewards and propose a hierarchical Bayesian model for the distribution of these two elements. Similarly, in [20], the authors assume that tasks are similar in their value functions and design a different hierarchical Bayesian model for transferring such information. More recently, [11], and its extension [16], consider tasks whose dynamics are governed by some hidden parameters, and propose efficient Bayesian models for quickly learning such parameters in new tasks. However, most of these algorithms require specific, and sometimes restrictive, assumptions (e.g., on the distributions involved or the function approximators adopted), which might limit their practical applicability. The importance of having

probabilistic? distributional?

transfer algorithms that alleviate the need for strong assumptions and that easily adapt to different contexts motivates us to take a more general approach.

Similarly to [20], we assume tasks to share similarities in their value functions and use the given source tasks to learn a distribution over such functions. Then, we use this distribution as a prior for learning the target task and we propose a variational approximation of the corresponding posterior that is computationally efficient. Leveraging on recent ideas from randomized value functions [27], we design a Thompson Sampling-based algorithm which efficiently explores the target task by repeatedly sampling from the posterior and acting greedily w.r.t. (with respect to) the sampled value function. We show that our approach is very general, in the sense that it can work with any parametric function approximator and with any prior/posterior distribution models (in this paper we focus on the Gaussian and Gaussian mixture models). In addition to the algorithmic contribution, we give also a theoretical contribution by providing a finite-sample analysis of our approach and an experimental contribution showing its empirical performance on four domains with increasing level of difficulty.

## 2 Preliminaries

We consider a distribution  $\mathcal{D}$  over tasks, where each task  $\mathcal{M}_\tau$  is modeled as a discounted Markov Decision Process (MDP). We define an MDP as a tuple  $\mathcal{M}_\tau = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}_\tau, \mathcal{R}_\tau, p_0, \gamma \rangle$ , where  $\mathcal{S}$  is the state-space,  $\mathcal{A}$  is a finite set of actions,  $\mathcal{P}_\tau(\cdot|s, a)$  is the distribution of the next state  $s'$  given that action  $a$  is taken in state  $s$ ,  $\mathcal{R}_\tau : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  is the reward function,  $p_0$  is the initial-state distribution, and  $\gamma \in [0, 1)$  is the discount factor. We assume the reward function to be uniformly bounded by a constant  $R_{max} > 0$ . A deterministic policy  $\pi : \mathcal{S} \rightarrow \mathcal{A}$  is a mapping from states to actions. At the beginning of each episode of interaction, the initial state  $s_0$  is drawn from  $p_0$ . Then, the agent takes the action  $a_0 = \pi(s_0)$ , receives a reward  $\mathcal{R}_\tau(s_0, a_0)$ , transitions to the next state  $s_1 \sim \mathcal{P}(\cdot|s_0, a_0)$ , and the process is repeated. The goal is to find the policy maximizing the long-term return over a possibly infinite horizon:  $\max_\pi J(\pi) \triangleq \mathbb{E}_{\mathcal{M}_\tau, \pi} [\sum_{t=0}^{\infty} \gamma^t \mathcal{R}_\tau(s_t, a_t)]$ . To this end, we define the optimal value function of task  $\mathcal{M}_\tau$ ,  $Q_\tau^*(s, a)$ , as the expected return obtained by taking action  $a$  in state  $s$  and following an optimal policy thereafter. Then, an optimal policy  $\pi_\tau^*$  is a policy that is greedy with respect to the optimal value function, i.e.,  $\pi_\tau^*(s) = \operatorname{argmax}_a Q_\tau^*(s, a)$  for all states  $s$ . It can be shown (e.g., [28]) that  $Q_\tau^*$  is the unique fixed-point of the optimal Bellman operator  $T_\tau$  defined by  $T_\tau Q(s, a) = \mathcal{R}_\tau(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}_\tau} [\max_{a'} Q(s', a')]$  for any value function  $Q$ . From now on, we adopt the term  $Q$ -function to denote any plausible value function, i.e., any function  $Q : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  uniformly bounded by  $\frac{R_{max}}{1-\gamma}$ . In the following, to avoid cluttering the notation, we will drop the subscript  $\tau$  when there is no ambiguity.

We consider a parametric family of  $Q$ -functions,  $\mathcal{Q} = \{Q_w : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R} \mid w \in \mathbb{R}^d\}$ , and we assume each function in  $\mathcal{Q}$  to be uniformly bounded by  $\frac{R_{max}}{1-\gamma}$ . When learning the optimal value function, a quantity of interest is how close a given function  $Q_w$  is to the fixed-point of the Bellman operator. A possible measure is its Bellman error (or Bellman residual), defined by  $B_w \triangleq TQ_w - Q_w$ . Notice that  $Q_w$  is optimal if and only if  $B_w(s, a) = 0$  for all  $s, a$ . If we assume the existence of a distribution  $\nu$  over  $\mathcal{S} \times \mathcal{A}$ , a sound objective is to directly minimize the squared Bellman error of  $Q_w$  under  $\nu$ , denoted by  $\|B_w\|_\nu^2$ . Unfortunately, it is well-known that an unbiased estimator of this quantity requires two independent samples of the next state  $s'$  for each  $s, a$  (e.g., [25]). In practice, the Bellman error is typically replaced by the TD error  $b(w)$ , which approximates the former using a single transition sample  $\langle s, a, s', r \rangle$ ,  $b(w) = r + \gamma \max_{a'} Q_w(s', a') - Q_w(s, a)$ . Finally, given a dataset  $D = \langle s_i, a_i, r_i, s'_i \rangle_{i=1}^N$  of  $N$  samples, the squared TD error is computed as  $\|B_w\|_D^2 = \frac{1}{N} \sum_{i=1}^N (r_i + \gamma \max_{a'} Q_w(s'_i, a') - Q_w(s_i, a_i))^2 = \frac{1}{N} \sum_{i=1}^N b_i(w)^2$ . Whenever the distinction is clear from the context, with slight abuse of terminology, we refer to the squared Bellman error and squared TD error as Bellman error and TD error, respectively.

## 3 Variational Transfer Learning

In this section, we describe our variational approach to transfer in RL. In Section 3.1, we start by introducing our algorithm from a high-level perspective, in such a way that any choice of prior and posterior distributions is possible. Then, in Sections 3.2 and 3.3, we propose practical implementations based on Gaussians and mixtures of Gaussians, respectively. We conclude with some considerations on how to optimize the proposed objective in Section 3.4.

### 91 3.1 Algorithm

92 Let us observe that the distribution  $\mathcal{D}$  over tasks induces a distribution over optimal  $Q$ -functions.  
 93 Furthermore, for any MDP, learning its optimal  $Q$ -function is sufficient for solving the problem.  
 94 Thus, one can safely replace the distribution over tasks with the distribution over their optimal value  
 95 functions. In our parametric settings, we reduce the latter to a distribution  $p(\mathbf{w})$  over weights.

Should we just make the assumption that  $Q$ -functions share knowledge in their weights?

96 Assume, for the moment, that we know the distribution  $p(\mathbf{w})$  and consider a dataset  $D =$   
 97  $\{(s_i, a_i, s'_i, r_i) \mid i = 1, 2, \dots, N\}$  of samples from some task  $\mathcal{M}_\tau \sim \mathcal{D}$  that we want to solve. Then,  
 98 we can compute the posterior distribution over weights given such dataset by applying Bayes theorem  
 99 as  $p(\mathbf{w}|D) \propto p(D|\mathbf{w})p(\mathbf{w})$ . Unfortunately, this cannot be directly used in practice since we do not  
 100 have a model of the likelihood  $p(D|\mathbf{w})$ . In such case, it is very common to make strong assumptions  
 101 on the MDPs or the  $Q$ -functions to get tractable posteriors. However, in our transfer settings, all  
 102 distributions involved depend on the family of tasks under consideration and making such assump-  
 103 tions is likely to limit the applicability to specific problems. Thus, we take a different approach to  
 104 derive a more general, but still well-grounded, solution. Recall that our final goal is to move the total  
 105 probability mass over the weights minimizing some empirical loss measure, which in our case is the  
 106 TD error  $\|B_{\mathbf{w}}\|_D^2$ . Then, given a prior  $p(\mathbf{w})$ , we know from PAC-Bayesian theory that the optimal  
 107 Gibbs posterior  $q$  takes the form (e.g., [9]):

$$q(\mathbf{w}) = \frac{e^{-\Lambda \|B_{\mathbf{w}}\|_D^2} p(\mathbf{w})}{\int e^{-\Lambda \|B_{\mathbf{w}'}\|_D^2} p(d\mathbf{w}')}, \quad (1)$$

108 for some parameter  $\Lambda > 0$ . Since  $\Lambda$  is typically chosen to increase with the number of samples  
 109  $N$ , in the remaining, we set it to  $\lambda^{-1}N$ , for some constant  $\lambda > 0$ . Notice that, whenever the term  
 110  $e^{-\Lambda \|B_{\mathbf{w}}\|_D^2}$  can be interpreted as the actual likelihood of  $D$ ,  $q$  becomes a classic Bayesian posterior.  
 111 Although we now have an appealing distribution, the integral at the denominator of (1) is intractable  
 112 to compute even for simple  $Q$ -function models. Thus, we propose a variational approximation  $q_\xi$  by  
 113 considering a simpler family of distributions parameterized by  $\xi \in \Xi$ . Then, our problem reduces to  
 114 finding the variational parameters  $\xi$  such that  $q_\xi$  minimizes the Kullback-Leibler (KL) divergence  
 115 w.r.t. the Gibbs posterior  $q$ . From the theory of variational inference (e.g., [6]), this can be shown to  
 116 be equivalent to minimizing the well-known (negative) *evidence lower bound* (ELBO):

Maybe say how this is obtained

$$\min_{\xi \in \Xi} \mathcal{L}(\xi) = \mathbb{E}_{\mathbf{w} \sim q_\xi} [\|B_{\mathbf{w}}\|_D^2] - \frac{\lambda}{N} KL(q_\xi(\mathbf{w}) \parallel p(\mathbf{w})). \quad (2)$$

117 The approximate posterior trades-off between placing probability mass over those weights  $\mathbf{w}$  that  
 118 have low expected TD error (first term), and staying close to the prior distribution (second term).  
 119 Assuming that we can compute the gradients of (2) w.r.t. the variational parameters  $\xi$ , our objective  
 120 can be optimized using any stochastic optimization algorithm, as shown in the next subsections.

121 We now highlight our general transfer procedure in Algorithm 1, while deferring a description of  
 122 specific choices for the involved distributions to the next two subsections. Given a set of weights  $\mathcal{W}_s$   
 123 from the source tasks' optimal  $Q$ -functions, we start by estimating the prior distribution (line 1), and  
 124 we initialize the variational parameters by minimizing the KL divergence w.r.t. such distribution (line  
 125 2).<sup>1</sup> Then, at each time step of interaction, we re-sample the weights from the current approximate  
 126 posterior and act greedily w.r.t. the corresponding  $Q$ -function (lines 7,8). After collecting and storing  
 127 the new experience (lines 9-10), we draw a mini-batch of samples from the replay buffer (line 11), use  
 128 this to estimate the objective function gradient (line 12), and, finally, update the variational parameters  
 129 (line 13).

130 The key property of our approach is the weight resampling at line 7, which resembles the well-known  
 131 Thompson sampling approach adopted in multi-armed bandits [8] and closely relates to the recent  
 132 value function randomization [27]. At each time we guess what the task we are trying to solve based  
 133 on our current belief is and we act as if such guess were true. This mechanism allows an efficient  
 134 adaptive exploration of the target task. Intuitively, during the first steps of interaction, the agent  
 135 is very uncertain about the current task, and such uncertainty induces stochasticity in the chosen  
 136 actions, allowing rather informed exploration to take place. Consider, for instance, that actions that  
 137 are bad on average for all tasks are improbable to be sampled, while this cannot happen in uninformed

<sup>1</sup>If the prior and approximate posterior were in the same family of distributions we could simply set  $\xi$  to the prior parameters. However, we are not making this assumption at this point.

---

**Algorithm 1** Variational Transfer

---

**Require:** Target task  $\tau$ , source  $Q$ -function weights  $\mathcal{W}_s$ , batch size  $M$

---

```
1: Estimate prior  $p(\mathbf{w})$  from  $\mathcal{W}_s$ 
2: Initialize variational parameters:  $\xi \leftarrow \operatorname{argmin}_{\xi} KL(q_{\xi}||p)$ 
3: Initialize replay buffer:  $D = \emptyset$ 
4: repeat
5:   Sample initial state:  $s_0 \sim p_0^{(\tau)}$ 
6:   while  $s_h$  is not terminal do
7:     Sample weights:  $\mathbf{w} \sim q_{\xi}(\mathbf{w})$ 
8:     Take action  $a_h = \operatorname{argmax}_a Q_{\mathbf{w}}(s_h, a)$ 
9:     Observe transition  $s_{h+1} \sim \mathcal{P}^{(\tau)}(\cdot|s_h, a_h)$  and collect reward  $r_h = \mathcal{R}^{(\tau)}(s_h, a_h)$ 
10:    Add sample to the replay buffer:  $D \leftarrow D \cup \langle s_h, a_h, r_h, s_{h+1} \rangle$ 
11:    Sample mini-batch  $D' = \langle s_i, a_i, r_i, s'_{i+1} \rangle_{i=1}^M$  from  $D$ 
12:    Estimate the gradient  $\nabla_{\xi} \mathcal{L}(\xi)$  using  $D'$ 
13:    Update  $\xi$  in the direction of  $-\nabla_{\xi} \mathcal{L}(\xi)$  using any stochastic optimizer (e.g., ADAM)
14:   end while
15: until forever
```

---

138 exploration strategies, like  $\epsilon$ -greedy, before learning takes place. As the learning process goes on, the  
139 algorithm will quickly figure out which task is being faced, thus moving all the probability mass over  
140 the weights minimizing the TD error. From that point, sampling from the posterior is approximately  
141 equivalent to deterministically taking such weights, and no more exploration will be performed.

142 Finally, notice the generality of the proposed approach: as far as the objective  $\mathcal{L}$  is differentiable  
143 in the variational parameters  $\xi$ , and its gradients can be efficiently computed, any approximator for  
144 the  $Q$ -function and any prior/posterior distributions can be adopted. For the latter, we describe two  
145 practical choices in the next two sections.

### 146 3.2 Gaussian Variational Transfer

147 We now restrict to a specific choice of the prior and posterior families that makes our algorithm  
148 very efficient and easy to implement. We assume that optimal  $Q$ -functions (or better, their weights)  
149 follow a multivariate Gaussian distribution. That is, we model the prior as  $p(\mathbf{w}) = \mathcal{N}(\mu_p, \Sigma_p)$   
150 and we learn its parameters from the set of source weights using maximum likelihood estimation  
151 (with small regularization to make sure the covariance is positive definite). Then, our variational  
152 family is the set of all well-defined Gaussian distributions, i.e., the variational parameters are  
153  $\Xi = \{(\mu, \Sigma) \mid \mu \in \mathbb{R}^d, \Sigma \in \mathbb{R}^{d \times d}, \Sigma \succ 0\}$ . To prevent the covariance from becoming not positive  
154 definite, we consider its Cholesky decomposition  $\Sigma = \mathbf{L}\mathbf{L}^T$  and learn the lower-triangular Cholesky  
155 factor  $\mathbf{L}$  instead. In this case, deriving the gradient of the objective is very simple. Both the KL  
156 between two multivariate Gaussians and its gradients have a simple closed-form expression. The  
157 expected log-likelihood, on the other hand, can be easily differentiated by adopting the reparameteri-  
158 zation trick (e.g., [14, 29]). Although these results are well-known in the literature, we report them in  
159 App. ?? to have a more self-contained description.

### 160 3.3 Mixture of Gaussian Variational Transfer

161 Although the Gaussian assumption of the previous section is very appealing as it allows for a simple  
162 and efficient way of computing the variational objective and its gradients, in practice it rarely allows  
163 us to describe the prior distribution accurately. In fact, even for families of tasks in which the reward  
164 and transition models are Gaussian, the  $Q$ -values might be far from being normally distributed.  
165 Depending on the family of tasks under consideration and, since we are learning a distribution over  
166 weights, on the chosen function approximator, the prior might have arbitrarily complex shapes. When  
167 the information loss due to the Gaussian approximation becomes too severe, the algorithm is likely to  
168 fail at capturing any similarities between the tasks. We now propose a variant to successfully solve  
169 this problem, while keeping the algorithm efficient and simple enough to be applied in practice.

Given the source tasks' weights  $\mathcal{W}_s$ , we model our estimated prior as a mixture of Gaussians with one equally weighted isotropic Gaussian centered at each weight:  $p(\mathbf{w}) = \frac{1}{|\mathcal{W}_s|} \sum_{\mathbf{w}_s \in \mathcal{W}_s} \mathcal{N}(\mathbf{w}|\mathbf{w}_s, \sigma_p^2 \mathbf{I})$ . This model resembles a kernel density estimator [31] and, due to its nonparametric nature, it allows capturing arbitrarily complex distributions. Consistently with the prior, we model our approximate posterior as a mixture of Gaussians. However, we allow a different number of components (typically much less than the prior's), and we adopt full covariances instead of only diagonals so that our posterior has the potential to match complex distributions with fewer components. Using  $C$  components, our posterior is  $q_\xi(\mathbf{w}) = \frac{1}{C} \sum_{i=1}^C \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ , with variational parameters  $\boldsymbol{\xi} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_C, \boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_C)$ . Once again, we learn Cholesky factors instead of full covariances. Although this new model has the potential to capture much more complicated distributions, it poses a significant complication: the KL divergence between two mixtures of Gaussians has no closed-form expression. To solve this issue, we rely on an upper bound to such quantity, so that negative ELBO still upper bounds the KL between the approximate and the exact posterior. Among the many upper bounds available in the literature, we adopt the one proposed in [13], which we report here for the sake of completeness. We refer the reader to the original paper for the proof.

**Theorem 1** ([13]). *Let  $p = \sum_i c_i^{(p)} f_i^{(p)}$  and  $q = \sum_j c_j^{(q)} f_j^{(q)}$  be two mixture of Gaussian distributions, where  $f_i^{(p)} = \mathcal{N}(\boldsymbol{\mu}_i^{(p)}, \boldsymbol{\Sigma}_i^{(p)})$  denotes the  $i$ -th component of  $p$ ,  $c_i^{(p)}$  denotes its weight, and similarly for  $q$ . Introduce two vectors  $\chi^{(1)}$  and  $\chi^{(2)}$  such that  $c_i^{(p)} = \sum_j \chi_{j,i}^{(2)}$  and  $c_j^{(q)} = \sum_i \chi_{i,j}^{(1)}$ . Then:*

$$KL(p||q) \leq KL(\chi^{(2)}||\chi^{(1)}) + \sum_{i,j} \chi_{j,i}^{(2)} KL(f_i^{(p)}||f_j^{(q)}) \quad (3)$$

Our new algorithm replaces the KL with the above-mentioned upper bound. Each time we require its value, we have to recompute the parameters  $\chi^{(1)}$  and  $\chi^{(2)}$  that tighten the bound. As shown in [13], we can use a simple fixed-point procedure for this purpose. Finally, both terms in the objective are now linear combinations of functions of the variational parameters of different components, and their gradients easily derive from the ones of the Gaussian case. We report the derivation in App. ??.

### 3.4 Optimizing the TD error

From Sections 3.2 and 3.3, we know that differentiating the negative ELBO  $\mathcal{L}$  w.r.t.  $\boldsymbol{\xi}$  requires differentiating  $\|B_{\mathbf{w}}\|_D^2$  w.r.t.  $\mathbf{w}$ . Unfortunately, the TD error is well-known to be non-differentiable due to the presence of the max operator. This issue rarely is a problem since typical value-based algorithms are semi-gradient methods, i.e., they do not differentiate the targets (see, e.g., Chapter 11 of [34]). However, our transfer settings are quite different from common RL. In fact, our algorithm is likely to start from  $Q$ -functions that are very close to an optimum and aims only to adapt the weights in some direction of lower error (i.e., higher likelihood) so as to quickly converge to the solution of the target task. Unfortunately, this property does not hold for most semi-gradient algorithms. Even worse, many online RL algorithms combined with complex function approximators (e.g., DQNs) are well-known to be unstable, especially when approaching an optimum, and require many tricks and tuning to work well [30, 37]. This property is clearly undesirable in our case, as we only aim at adapting already good solutions. Thus, we consider using a residual gradient algorithm [4]. To differentiate the targets, we replace the optimal Bellman operator with the mellow Bellman operator introduced in [2], which adopts a softened version of max called *mellowmax*:

$$\text{mm}_a Q_{\mathbf{w}}(s, a) = \frac{1}{\kappa} \log \frac{1}{|\mathcal{A}|} \sum_a e^{\kappa Q_{\mathbf{w}}(s, a)} \quad (4)$$

where  $\kappa$  is a hyperparameter and  $|\mathcal{A}|$  is the number of actions. The mellow Bellman operator, which we denote as  $\tilde{T}$ , has several appealing properties that make it suitable for our settings: (i) it converges to the maximum as  $\kappa \rightarrow \infty$ , (ii) it has a unique fixed point, and (iii) it is *differentiable*. Denoting by  $\tilde{B}_{\mathbf{w}} = \tilde{T}Q_{\mathbf{w}} - Q_{\mathbf{w}}$  the Bellman residual w.r.t. the mellow Bellman operator  $\tilde{T}$ , we have that the corresponding TD error,  $\|\tilde{B}_{\mathbf{w}}\|_D^2$ , is now differentiable w.r.t.  $\mathbf{w}$ . Although residual algorithms have guaranteed convergence, they are typically much slower than their semi-gradient counterpart. [4] proposed to project the gradient in a direction that achieves higher learning speed, while preserving

questo dettaglio si potrebbe spostare nella parte sperimentale o togliere del tutto

216 convergence. This projection is obtained by including a parameter  $\psi \in [0, 1]$  in the TD error gradient:  
 217

$$\nabla_{\mathbf{w}} \left\| \tilde{B}_{\mathbf{w}} \right\|_D^2 = \frac{2}{N} \sum_{i=1}^N b_i(\mathbf{w}) \left( \gamma \psi \nabla_{\mathbf{w}} \text{mm}_{a'} Q_{\mathbf{w}}(s'_i, a') - \nabla_{\mathbf{w}} Q_{\mathbf{w}}(s_i, a_i) \right), \quad (5)$$

218 where  $b_i(\mathbf{w}) = r_i + \gamma \text{mm}_{a'} Q_{\mathbf{w}}(s'_i, a') - Q_{\mathbf{w}}(s_i, a_i)$ . Notice that  $\psi$  trades-off between the semi-  
 219 gradient ( $\psi = 0$ ) and the full residual gradient ( $\psi = 1$ ). A good criterion for choosing such  
 220 parameter is to start with values close to zero (to have faster learning) and move to higher values  
 221 when approaching the optimum (to guarantee convergence).

## 222 4 Theoretical Analysis

223 A first important question that we need to answer is whether replacing max with mellow-max in  
 224 the Bellman operator constitutes a strong approximation or not. It has been proved [2] that the  
 225 mellow Bellman operator is a non-expansion under the  $L_\infty$ -norm and, thus, has a unique fixed-point.  
 226 However, how such fixed-point differs from the one of the optimal Bellman operator remains an open  
 227 question. Since mellow-max monotonically converges to max as  $\kappa \rightarrow \infty$ , it would be desirable if  
 228 the fixed point of the corresponding operator also monotonically converged to the fixed point of the  
 229 optimal one. We confirm that this property actually holds in the following theorem.

230 **Theorem 2.** *Let  $Q^*$  be the fixed-point of the optimal Bellman operator  $T$ . Define the action-gap*  
 231 *function  $g(s)$  as the difference between the value of the best action and the second best action at*  
 232 *each state  $s$ . Let  $\tilde{Q}$  be the fixed-point of the mellow Bellman operator  $\tilde{T}$  with parameter  $\kappa > 0$  and*  
 233 *denote by  $\beta_\kappa > 0$  the inverse temperature of the induced Boltzmann distribution (as in [2]). Let  $\nu$  be*  
 234 *a probability measure over the state-action space. Then, for any  $p \geq 1$ :*

$$\left\| Q^* - \tilde{Q} \right\|_{\nu, p}^p \leq \frac{2\gamma R_{\max}}{(1-\gamma)^2} \left\| \frac{1}{1 + \frac{1}{|\mathcal{A}|} e^{\beta_\kappa g}} \right\|_{\nu, p}^p. \quad (6)$$

Correct mea-  
sure on the  
rhs

235 The proof is provided in App. ???. As expected,  $\tilde{Q}$  converges to  $Q^*$  exponentially fast when either  $\kappa$   
 236 (equivalently,  $\beta_\kappa$ ) increases, or the action gaps are enlarged. Notice that this result is of interest even  
 237 outside our specific settings.

238 The second question that we need to answer is whether we can provide any guarantee on our  
 239 algorithm's performance when given limited data. To address this point, we consider Algorithm 1  
 240 with a mixture of Gaussians and linear approximators. We assume only a finite dataset is available  
 241 and provide a finite-sample analysis bounding the distance between the fixed-point of the mellow  
 242 Bellman operator and  $Q$ -functions sampled from the variational distribution minimizing the objective  
 243 (2). Our main result is given in the following theorem.

244 **Theorem 3.** *Fix a target task  $\tau$  and let  $\tilde{Q}$  be the fixed-point of the corresponding mellow Bellman*  
 245 *operator. Assume linearly parameterized value functions  $Q_{\mathbf{w}}(s, a) = \mathbf{w}^T \phi(s, a)$  with uniformly*  
 246 *bounded weights  $|\mathbf{w}| \leq w_{\max}$  and uniformly bounded features  $|\phi(s, a)| \leq \phi_{\max}$ . Consider the*  
 247 *mixture version of Algorithm 1 using  $C$  components, source task weights  $\mathcal{W}_s$ , and bandwidth  $\sigma_p^2$*   
 248 *for the prior. Denote by  $\hat{\xi} = (\hat{\mu}_1, \dots, \hat{\mu}_C, \hat{\Sigma}_1, \dots, \hat{\Sigma}_C)$  the variational parameters minimizing the*  
 249 *objective of Equation (2) on a dataset  $D$  of  $N$  samples. Let  $\nu$  be a probability measure over  $\mathcal{S} \times \mathcal{A}$*   
 250 *and  $\mathbf{w}^* = \arg\inf_{\mathbf{w}} \left\| \tilde{B}_{\mathbf{w}} \right\|_{\nu}^2$ . Define  $v(\mathbf{w}^*) \triangleq \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, I)} [v(\mathbf{w})]$ , with  $v(\mathbf{w}) \triangleq \mathbb{E}_{\nu} [\text{Var}_{\mathcal{P}} [b(\mathbf{w})]]$ .*  
 251 *Then, there exist constants  $c_1, c_2, c_3, c_4$  such that, with probability at least  $1 - 2\delta$  over the choice of*  
 252 *weights  $\mathbf{w} \sim \frac{1}{C} \sum_i \mathcal{N}(\hat{\mu}_i, \hat{\Sigma}_i)$  and dataset  $D$ :*

$$\left\| Q_{\mathbf{w}} - \tilde{Q} \right\|_{\nu}^2 \leq \frac{1}{(1-\gamma)\delta} \left( 2 \left\| \tilde{B}_{\mathbf{w}^*} \right\|_{\nu}^2 + \frac{c_1}{N^2} + \frac{c_2 + \frac{c_4}{|\mathcal{W}_s|} \sum_j \|\mathbf{w}^* - \mathbf{w}_j\|}{N} + \frac{v(\mathbf{w}^*) + c_3 \sqrt{\log \frac{2}{\delta}}}{\sqrt{N}} \right). \quad (7)$$

253 We refer the reader to App. ??? for the proof and a specific definition of the constants. Four main  
 254 terms constitute our bound: the approximation error due to the limited hypothesis space (first term),  
 255 the variance (second term), the distance to the prior (third term), and a constant term decaying as  
 256  $\mathcal{O}(N^2)$ . Intuitively, the bound is tighter when the source tasks'  $Q$ -functions are, on average, close  
 257 to the optimal ones for the target task. In such case, the dominating error is due to the variance



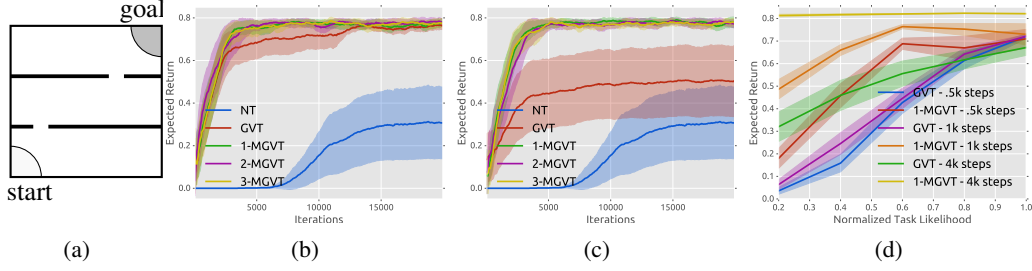


Figure 1: (a) the rooms environment, (b) transfer from 10 source tasks with both doors moving, (c) transfer from 10 source tasks with only one door moving, and (d) transfer performance as a function of how likely the target task is according to the prior.

of the estimates, and, thus, the algorithm is expected to achieve good performance rather quickly, as new data is collected. Furthermore, as  $N \rightarrow \infty$  the only error term remaining is, as usual, the irreducible approximation error due to the limited functional space. Notice that the variance term  $v(w^*)$  is due to the fact that we minimize a biased estimator of the Bellman error. If double sampling of the next state were possible (e.g., in simulation), such term could be removed. Finally, we want to point out that the proof of this theorem relies on a very general result (Lemma 3 in App. ??) that does not require any specific choice of distribution of approximator. Leveraging on such results, it is straightforward to provide a finite-sample analysis of the Gaussian version of our algorithm. For the sake of completeness, we report the derivation in App. ??.

## 5 Experiments

In this section, we provide an experimental evaluation of our approach in four different domains with increasing level of difficulty. In all experiments, we compare our Gaussian variational transfer algorithm (GVT) and the version using a  $c$ -component mixture of Gaussians ( $c$ -MGVT) to plain no-transfer RL (NT) with  $\epsilon$ -greedy exploration. To the best of our knowledge, no existing transfer algorithm is directly comparable to our approach from an experimental perspective. Thus, we provide a discussion of related works in the next section.

### 5.1 The Rooms Problem

We consider an agent navigating in the environment depicted in Figure 1a. The agent starts in the bottom-left corner and must move from one room to another to reach the goal position in the top-right corner. The rooms are connected by small doors whose locations are unknown to the agent. The state-space is modeled as a  $10 \times 10$  continuous grid, while the action-space is the set of 4 movement directions (up, right, down, left). After each action, the agent moves by 1 in the chosen direction and the final position is corrupted by Gaussian noise  $\mathcal{N}(0, 0.2)$ . In case the agent hits a wall, its position remains unchanged. The reward is 1 when reaching the goal (after which the process terminates) and 0 otherwise, while the discount factor is  $\gamma = 0.99$ . In this experiment, we consider linearly parameterized  $Q$ -functions with 121 equally-spaced radial basis features.

Since the agent does not know the locations of the doors in advance and receives only very sparse feedback, it must efficiently explore the environment to figure out (i) their positions, and (ii) how to reach the goal. While this might be a complicated problem for plain RL, our transfer algorithm should be able to figure out the door positions quickly. In fact, notice that, although different, the optimal  $Q$ -functions for all tasks share some similarities. For example, once the agent has passed the last door before the goal, the  $Q$ -values are exactly the same in all tasks. The same does not hold for positions nearby the starting state. However, it is clear that there should be a preference over actions *up* and *right*, rather than *down* and *left* (which are worse in all tasks).

To prove that our guesses are correct, we generate a set of 50 source tasks for the three-room environment of Figure 1a by sampling both door locations uniformly in the allowed space, and solve all of them by directly minimizing the TD error as presented in Section 3.4. Then, we use our algorithms to transfer from 10 source tasks sampled from the previously generated set. The average return over the last 50 learning episodes as a function of the number of iterations is shown in

Should we discuss the main difference between the MGVT and GVT bounds here?

Any better motivation?

Good candidate to be removed

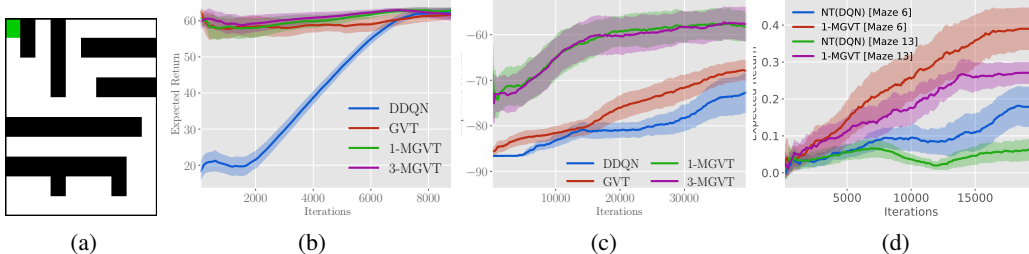


Figure 2: (a) Sample Maze (Maze 6), (b) transfer from 10 sources in Cartpole, (c) transfer from 10 source tasks in Mountain Car, and (d) transfer from 10 sources in Mazes 6 and 13—see App. ??.

Figure 1b. Each curve is the result of 20 independent runs, each one resampling the target and source tasks, with 95% confidence intervals. Further details on the parameters adopted in this experiment are given in App. ?. As expected, the no-transfer (NT) algorithm fails at learning the task in so few iterations due to the limited exploration provided by an  $\epsilon$ -greedy policy. On the other hand, all our algorithms achieve a significant speed-up and converge to the optimal performance in few iterations, with GVT being slightly slower. Interestingly, we notice that there is no advantage in adopting more than 1 component for the posterior in MGVT. This result is intuitive since, as soon as the algorithm figures out which is the target task, all the components move towards the same region.

To better understand the differences between GVT and MGVT, we now consider transferring from a slightly different distribution than the one from which target tasks are drawn. We generate 50 source tasks again but this time with the bottom door fixed at the center and the other one moving. Then, we repeat the previous experiment, allowing both doors to move when sampling target tasks. The results are shown in Figure 1c. Interestingly, MGVT seems almost unaffected by this change, proving that it has sufficient representation power to generalize to slightly different task distributions. The same does not hold for GVT, which now is not able to solve many of the sampled target tasks, as can be noticed from the higher variance. This result proves again that assuming Gaussian distributions can pose severe limitations in our transfer settings.

Finally, we analyze the transfer performance as a function of how likely the target task is according to the prior. We consider a two-room version of the environment of Figure 1a. Unlike before, we generate tasks by sampling the door position from a Gaussian with mean 5 and standard deviation 1.8, so that tasks with the door near the sides are very unlikely. Figure 1d shows the performance reached by GVT and 1-MGVT at fixed iterations as a function of how likely the target task is according to such distribution. As expected, GVT achieves poor performance on very unlikely tasks, even after many iterations. In fact, estimating a single Gaussian distribution definitely entails some information loss, especially about the unlikely tasks. On the other hand, MGVT keeps such information and, consequently, performs much better. Perhaps not surprisingly, MGVT reaches the optimal performance in  $4k$  iterations no matter what task is being solved.

## 5.2 Classic Control

We now consider two well-known classic control environments: Cartpole and Mountain Car [34]. For both, we generate 20 source tasks by uniformly sampling their physical parameters (cart mass, pole mass, pole length for Cartpole and car speed for Mountain Car). We parameterize  $Q$ -functions using neural networks with 1-layer of 32 hidden units for Cartpole and 64 for Mountain Car. A better description of these two environments and their parameters is given in App. ?. In this experiment, we use a Double Deep Q-Network [37] to provide a stronger no-transfer baseline for comparison. The results (same settings of Section 5.1) are shown in Figures 2b and 2c. For Cartpole (Figure 2b), all transfer algorithms are almost zero-shot. This result is expected since, although we vary the system parameters in a wide range, the optimal  $Q$ -values of states near the balanced position are almost the same for all tasks. On the contrary, in Mountain Car (Figure 2c) the optimal  $Q$ -functions become very different when changing the car speed. This phenomenon hinders the learning of GVT in the target task, while 1-MGVT achieves a good jump-start and converges in fewer iterations.



### 5.3 Maze Navigation

For this setting we propose a maze environment, such as those in Fig. ??, for a robotic agent to navigate. We created a set of 20 mazes—see App ??—of size  $10 \times 10$  with continuous space within which we ensured four groups: each group with the goal position in one of the corners of the maze. In order to simulate a robotic agent, we set an action space that allows to control the linear and angular motion by a fixed distance of 0.5 (*move forward*) and a fixed angle of  $\pm 22.5^\circ$  (*rotate counter- and clockwise*). Moreover, we enhanced the state space to include, in addition to the absolute position and orientation, a set of distance measurements to the nearest obstacles from the agent’s perspective with maximum range of 2 and a field of vision of 9 equally-spaced directions in  $[-90, 90^\circ]$  relative to its orientation. In this way, we provide the agent with information that a robot would normally collect from LIDARs, sonars and other sensors alike. Finally, in the same field of perception, we provide a boolean vector indicating whether the goal is within the range of observation. Regarding rewards, the agent gets 1 whenever it enters the goal area and 0 in any other situation and the MDP has a discount factor of  $\gamma = 0.99$ .

The main motivation for this set-up is that a robot is able to exploit locally-sensed information in addition to its estimation of the actual position—as opposed to the Rooms problem presented above—and it is such information that the robotic agent could learn to use and, also, exploit in different maze configurations leveraging its previous experience. Moreover, simple variations in the maze structure could, potentially, produce big differences in the value function with respect to the ones seen in previous tasks which makes the setting suitable to assess how robust our algorithm is to negative transfer.

The mazes were solved using Deep Q Networks (DQN) [1] with an MLP with hidden layers  $32 \times 32$  and, in order to perform the knowledge transfer, the target maze would be excluded from the source tasks given to the algorithm. Finally, the 1-MGVT algorithm was used to solve the target tasks—Mazes 6 and 13. As before, 20 independent runs with randomized sources to transfer are averaged and shown in Fig. ??, in this case 5 source tasks were used for each trial. While these mazes pose a challenge for our algorithm, we can see that 1-MGVT is capable to still capture useful information from the sources and adapt to the target mazes which is demonstrated by the increased slope in the learning curve.

cite DQN?

actually, decide how to improve these results

## 6 Related Works

Our approach is mostly related to [20]. Although we both assume the tasks to share similarities in their value functions, [20] consider only linear approximators and adopt a hierarchical Bayesian model of the corresponding weights’ distribution, which is assumed Gaussian. On the other hand, our variational approximation allows for more general distribution families and can be combined with non-linear approximators. Furthermore, [20] propose a Dirichlet process model for the case where weights cluster into different classes, which relates to our mixture formulation and proves again the importance of capturing more complicated task distributions. In [38], the authors propose a hierarchical Bayesian model for the distribution over tasks. Differently from our approach and [20], they consider a distribution over transition probabilities and rewards, rather than value functions. In the same spirit of our method, they consider a Thompson sampling-based procedure which, at each iteration, samples a new task from the posterior and solves it. However, [38] consider only finite MDPs, which poses a severe limitation on the algorithm’s applicability. On the contrary, our approach can handle high-dimensional tasks. In [11], the authors consider a family of tasks whose dynamics are governed by some hidden parameters and use Gaussian processes (GPs) to model such dynamics across tasks. Recently, [16] extended this approach by replacing GPs with Bayesian neural networks, so as to obtain a more scalable approach.

In the RL community, our approach is related to value function randomization[27], which extends the well-known LSTD [7] by adopting Bayesian linear regression to model the uncertainty over the predicted value function weights, and use that to perform a form of Thompson sampling. Such approach was recently extended by [3], where the approximator is replaced by a Bayesian neural network, leading to an algorithm capable of solving much more complicated problems. Both these algorithms rely on the Gaussian assumption and, since they work in plain RL settings, have no informative prior available. On the other hand, our variational approximation allows more complex

More comments about HiP-MDPs? Should we discuss more related works?

distributions (e.g., mixtures) to be adopted, while knowledge from the source tasks allows us to learn very informative priors.

## 7 Conclusion

In this work, we presented a variational method for transferring value functions in RL. We showed our approach to be general, in the sense that it can be combined with several distributions and function approximators, and we provided two practical algorithms based on Gaussians and mixtures of Gaussians, respectively. We analyzed both from a theoretical and empirical perspective, proving that the Gaussian one has severe limitations, while the mixture version is much more suitable for our transfer settings.

Since our algorithm effectively models the uncertainty over tasks, a relevant future work is to design an algorithm that explicitly explores the target task to reduce such uncertainty (e.g., [15]). Furthermore, our variation approach could be extended to model a distribution over optimal policies instead of value functions (e.g., [24]), which might allow better transferred behavior.

## References

- [1] Pierre Alquier, James Ridgway, and Nicolas Chopin. On the properties of variational approximations of gibbs posteriors. *Journal of Machine Learning Research*, 17(239):1–41, 2016.
- [2] Kavosh Asadi and Michael L Littman. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, pages 243–252, 2017.
- [3] Kamyar Azizzadenesheli, Emma Brunskill, and Animashree Anandkumar. Efficient exploration through bayesian deep q-networks. *arXiv preprint arXiv:1802.04412*, 2018.
- [4] Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.
- [5] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in neural information processing systems*, pages 4055–4065, 2017.
- [6] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.
- [7] Justin A Boyan. Least-squares temporal difference learning.
- [8] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.
- [9] Olivier Catoni. Pac-bayesian supervised classification: the thermodynamics of statistical learning. *arXiv preprint arXiv:0712.0248*, 2007.
- [10] Vincent Cottet and Pierre Alquier. 1-bit matrix completion: Pac-bayesian analysis of a variational approximation. *Machine Learning*, 107(3):579–603, 2018.
- [11] Finale Doshi-Velez and George Konidaris. Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. In *IJCAI: proceedings of the conference*, volume 2016, page 1432. NIH Public Access, 2016.
- [12] Fernando Fernández and Manuela Veloso. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 720–727. ACM, 2006.
- [13] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–317. IEEE, 2007.
- [14] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.
- [15] Rein Houthoofd, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pages 1109–1117, 2016.

- [16] Taylor W Killian, Samuel Daulton, George Konidaris, and Finale Doshi-Velez. Robust and efficient transfer learning with hidden parameter markov decision processes. In *Advances in Neural Information Processing Systems*, pages 6250–6261, 2017.
- [17] Jens Kober and Jan R Peters. Policy search for motor primitives in robotics. In *Advances in neural information processing systems*, pages 849–856, 2009.
- [18] George Konidaris and Andrew Barto. Autonomous shaping: Knowledge transfer in reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 489–496. ACM, 2006.
- [19] George Konidaris and Andrew G Barto. Building portable options: Skill transfer in reinforcement learning.
- [20] Alessandro Lazaric and Mohammad Ghavamzadeh. Bayesian multi-task reinforcement learning. In *ICML-27th International Conference on Machine Learning*, pages 599–606. Omnipress, 2010.
- [21] Alessandro Lazaric, Marcello Restelli, and Andrea Bonarini. Transfer of samples in batch reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 544–551. ACM, 2008.
- [22] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.
- [23] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [24] Yang Liu, Prajit Ramachandran, Qiang Liu, and Jian Peng. Stein variational policy gradient. *arXiv preprint arXiv:1704.02399*, 2017.
- [25] Odalric-Ambrym Maillard, Rémi Munos, Alessandro Lazaric, and Mohammad Ghavamzadeh. Finite-sample analysis of bellman residual minimization. In *Proceedings of 2nd Asian Conference on Machine Learning*, pages 299–314, 2010.
- [26] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.
- [27] Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. *arXiv preprint arXiv:1402.0635*, 2014.
- [28] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1994.
- [29] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.
- [30] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.
- [31] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.
- [32] Yevgeny Seldin, François Laviolette, Nicolo Cesa-Bianchi, John Shawe-Taylor, and Peter Auer. Pac-bayesian inequalities for martingales. *IEEE Transactions on Information Theory*, 58(12):7086–7093, 2012.
- [33] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.
- [34] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- [35] Matthew E Taylor, Nicholas K Jong, and Peter Stone. Transferring instances for model-based reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 488–505. Springer, 2008.
- [36] Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685, 2009.

- 487 [37] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning.  
488 2016.
- 489 [38] Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. Multi-task reinforcement learning: a  
490 hierarchical bayesian approach. In *Proceedings of the 24th international conference on Machine learning*,  
491 pages 1015–1022. ACM, 2007.

## 492 A Proofs

### 493 A.1 Proof of Theorem 2

494 **Theorem 2.** Let  $Q^*$  be the fixed-point of the optimal Bellman operator  $T$ . Define the action-gap  
 495 function  $g(s)$  as the difference between the value of the best action and the second best action at  
 496 each state  $s$ . Let  $\tilde{Q}$  be the fixed-point of the mellow Bellman operator  $\tilde{T}$  with parameter  $\kappa > 0$  and  
 497 denote by  $\beta_\kappa > 0$  the inverse temperature of the induced Boltzmann distribution (as in [2]). Let  $\nu$  be  
 498 a probability measure over the state-action space. Then, for any  $p \geq 1$ :

$$\|Q^* - \tilde{Q}\|_{\nu,p}^p \leq \frac{2\gamma R_{max}}{(1-\gamma)^2} \left\| \frac{1}{1 + \frac{1}{|\mathcal{A}|} e^{\beta_\kappa g}} \right\|_{\nu,p}^p. \quad (6)$$

Correct mea-  
sure on the  
rhs

499 *Proof.* We begin by noticing that:

$$\begin{aligned} \|Q^* - \tilde{Q}\|_{\nu,p}^p &= \|TQ^* - \tilde{T}\tilde{Q}\|_{\nu,p}^p \\ &= \|TQ^* - \tilde{T}Q^* + \tilde{T}Q^* - \tilde{T}\tilde{Q}\|_{\nu,p}^p \\ &\leq \|TQ^* - \tilde{T}Q^*\|_{\nu,p}^p + \|\tilde{T}Q^* - \tilde{T}\tilde{Q}\|_{\nu,p}^p \\ &\leq \|TQ^* - \tilde{T}Q^*\|_{\nu,p}^p + \gamma \|Q^* - \tilde{Q}\|_{\nu,p}^p \end{aligned}$$

500 where the first inequality follows from Minkowsky's inequality and the second one from the contrac-  
 501 tion property of the mellow Bellman operator. This implies that:

$$\|Q^* - \tilde{Q}\|_{\nu,p}^p \leq \frac{1}{1-\gamma} \|TQ^* - \tilde{T}Q^*\|_{\nu,p}^p \quad (8)$$

502 Let us bound the norm on the right-hand side separately. In order to do that, we will bound the  
 503 function  $|TQ^*(s, a) - \tilde{T}Q^*(s, a)|$  point-wisely for any state  $s, a$ . By applying the definition of the  
 504 optimal and mellow Bellman operators, we obtain:

$$\begin{aligned} |TQ^*(s, a) - \tilde{T}Q^*(s, a)| &= |R(s, a) + \gamma \mathbb{E} [\max_{a'} Q^*(s', a')] - R(s, a) - \gamma \mathbb{E} [\text{mm}_{a'} Q^*(s', a')]| \\ &= \gamma |\mathbb{E} [\max_{a'} Q^*(s', a')] - \mathbb{E} [\text{mm}_{a'} Q^*(s', a')]| \\ &\leq \gamma \mathbb{E} [|\max_{a'} Q^*(s', a') - \text{mm}_{a'} Q^*(s', a')|] \end{aligned} \quad (9)$$

505 Thus, bounding this quantity reduces to bounding  $|\max_a Q^*(s, a) - \text{mm}_a Q^*(s, a)|$  point-wisely for  
 506 any  $s$ . Recall that applying the mellow Bellman operator is equivalent to computing an expectation  
 507 under a Boltzmann distribution with inverse temperature  $\beta_\kappa$  induced by  $\kappa$  []. Thus, we can write:

Cite MM

$$\begin{aligned} |\max_a Q^*(s, a) - \text{mm}_a Q^*(s, a)| &= \left| \sum_a \pi^*(a|s) Q^*(s, a) - \sum_a \pi_{\beta_\kappa}(a|s) Q^*(s, a) \right| \\ &= \left| \sum_a Q^*(s, a) (\pi^*(a|s) - \pi_{\beta_\kappa}(a|s)) \right| \\ &\leq \sum_a |Q^*(s, a)| |\pi^*(a|s) - \pi_{\beta_\kappa}(a|s)| \\ &\leq \frac{R_{max}}{1-\gamma} \sum_a |\pi^*(a|s) - \pi_{\beta_\kappa}(a|s)| \end{aligned} \quad (10)$$

508 where  $\pi^*$  is the optimal (deterministic) policy w.r.t.  $Q^*$  and  $\pi_{\beta_\kappa}$  is the Boltzmann distribution induced  
 509 by  $Q^*$  with inverse temperature  $\beta_\kappa$ :

$$\pi_{\beta_\kappa}(a|s) = \frac{e^{\beta_\kappa Q^*(s, a)}}{\sum_{a'} e^{\beta_\kappa Q^*(s, a')}}.$$

510 Denote by  $a_1(s)$  the optimal action for state  $s$  under  $Q^*$ . We can then write:

$$\begin{aligned}
\sum_a |\pi^*(a|s) - \pi_{\beta_\kappa}(a|s)| &= |\pi^*(a_1(s)|s) - \pi_{\beta_\kappa}(a_1(s)|s)| + \sum_{a \neq a_1(s)} |\pi^*(a|s) - \pi_{\beta_\kappa}(a|s)| \\
&= |1 - \pi_{\beta_\kappa}(a_1(s)|s)| + \sum_{a \neq a_1(s)} |\pi_{\beta_\kappa}(a|s)| \\
&= 2|1 - \pi_{\beta_\kappa}(a_1(s)|s)|
\end{aligned} \tag{11}$$

511 Finally, let us bound this last term:

$$\begin{aligned}
|1 - \pi_{\beta_\kappa}(a_1(s)|s)| &= \left| 1 - \frac{e^{\beta_\kappa Q^*(s, a_1(s))}}{\sum_{a'} e^{\beta_\kappa Q^*(s, a')}} \right| \\
&= \left| 1 - \frac{e^{\beta_\kappa(Q^*(s, a_1(s)) - Q^*(s, a_2(s)))}}{\sum_{a'} e^{\beta_\kappa(Q^*(s, a') - Q^*(s, a_2(s)))}} \right| \\
&= \left| 1 - \frac{e^{\beta_\kappa g(s)}}{\sum_{a'} e^{\beta_\kappa(Q^*(s, a') - Q^*(s, a_2(s)))}} \right| \\
&= \left| 1 - \frac{e^{\beta_\kappa g(s)}}{e^{\beta_\kappa g(s)} + \sum_{a' \neq a_1(s)} e^{\beta_\kappa(Q^*(s, a') - Q^*(s, a_2(s)))}} \right| \\
&\leq \left| 1 - \frac{e^{\beta_\kappa g(s)}}{e^{\beta_\kappa g(s)} + |\mathcal{A}|} \right| \\
&= \left| \frac{1}{1 + \frac{1}{|\mathcal{A}|} e^{\beta_\kappa g(s)}} \right|
\end{aligned} \tag{12}$$

512 Combining Eq. (10), (11), and (12), we obtain:

$$\left| \max_a Q(s, a) - \min_a Q(s, a) \right| \leq \frac{2R_{max}}{1 - \gamma} \left| \frac{1}{1 + \frac{1}{|\mathcal{A}|} e^{\beta_\kappa g(s)}} \right|$$

513 Finally, using Eq. (9) we get:

$$\left| TQ^*(s, a) - \tilde{T}Q^*(s, a) \right| \leq \frac{2\gamma R_{max}}{1 - \gamma} \left| \frac{1}{1 + \frac{1}{|\mathcal{A}|} e^{\beta_\kappa g(s)}} \right|$$

514 Taking the norm and plugging this into Eq. (8) concludes the proof.  $\square$

## 515 A.2 Proof of Theorem 3

516 We begin by proving some important lemmas. Then, we use them to derive a finite-sample analysis  
517 of Alg. 1 with linearly parameterized value functions for both Gaussian distributions (Thm. 4) and  
518 Gaussian mixture models (Thm. 3).

519 **Lemma 1.** *Let  $p$  and  $\nu$  denote probability measures over weights and state-action pairs, respectively.*  
520 *Assume  $Q^*$  is the unique fixed-point of the optimal Bellman operator  $T$ . Then, for any  $\delta > 0$ , with*  
521 *probability at least  $1 - \delta$  over the choice of weights  $\mathbf{w} \sim p$ , the following holds:*

$$\|Q_{\mathbf{w}} - Q^*\|_\nu^2 \leq \frac{\mathbb{E}_p [\|B_{\mathbf{w}}\|_\nu^2]}{(1 - \gamma)\delta} \tag{13}$$

522 *Proof.* First notice that:

$$\begin{aligned}
\|Q_{\mathbf{w}} - Q^*\|_\nu^2 &= \|Q_{\mathbf{w}} + TQ_{\mathbf{w}} - TQ_{\mathbf{w}} - TQ^*\|_\nu^2 \\
&\leq \|Q_{\mathbf{w}} - TQ_{\mathbf{w}}\|_\nu^2 + \|TQ_{\mathbf{w}} - TQ^*\|_\nu^2 \\
&\leq \|Q_{\mathbf{w}} - TQ_{\mathbf{w}}\|_\nu^2 + \gamma \|Q_{\mathbf{w}} - Q^*\|_\nu^2 \\
&= \|B_{\mathbf{w}}\|_\nu^2 + \gamma \|Q_{\mathbf{w}} - Q^*\|_\nu^2
\end{aligned}$$



523 which implies that:

$$\|Q_{\mathbf{w}} - Q^*\|_{\nu}^2 \leq \frac{1}{1-\gamma} \|B_{\mathbf{w}}\|_{\nu}^2$$

524 Then we can write:

$$P\left(\|Q_{\mathbf{w}} - Q^*\|_{\nu}^2 > \epsilon\right) \leq P\left(\|B_{\mathbf{w}}\|_{\nu}^2 > \epsilon(1-\gamma)\right) \leq \frac{\mathbb{E}_p\left[\|B_{\mathbf{w}}\|_{\nu}^2\right]}{(1-\gamma)\epsilon}$$

525 Settings the right-hand side equal to  $\delta$  and solving for  $\epsilon$  concludes the proof.  $\square$

526 Lemma 1 can be straightforwardly extended to the case where the mellow Bellman operator is used  
527 instead of the optimal one. This is given in the following corollary.

528 **Corollary 1.** *Let  $p$  and  $\nu$  denote probability measures over weights and state-action pairs, respec-*  
529 *tively. Assume  $\tilde{Q}$  is the unique fixed-point of the mellow Bellman operator  $\tilde{T}$ . Then, for any  $\delta > 0$ ,*  
530 *with probability at least  $1 - \delta$  over the choice of weights  $\mathbf{w} \sim p$ , the following holds:*

$$\|Q_{\mathbf{w}} - \tilde{Q}\|_{\nu}^2 \leq \frac{\mathbb{E}_p\left[\|\tilde{B}_{\mathbf{w}}\|_{\nu}^2\right]}{(1-\gamma)\delta} \quad (14)$$

531 We now prove some important properties of the variational approximation introduced in Sec. 3.1.  
532 Our results generalize those of existing works that consider variational approximations of intractable  
533 Gibbs posteriors [1, 10]. From now on, we consider only  $Q$ -functions parameterized by weights  $\mathbf{w}$   
534 and assume them to be uniformly bounded by  $\frac{R_{max}}{1-\gamma}$ .

535 **Lemma 2.** *Let  $p$  and  $q$  be arbitrary distributions over weights  $\mathbf{w}$ , and  $\nu$  be a probability measure*  
536 *over  $\mathcal{S} \times \mathcal{A}$ . Consider a dataset  $D$  of  $N$  i.i.d. samples where state-action couples are distributed*  
537 *according to  $\nu$  and define  $v(\mathbf{w}) \triangleq \mathbb{E}_{\nu}[Var_{\mathcal{P}}[B_{\mathbf{w}}]]$ . Then, for any  $\delta > 0$ , with probability at least*  
538  *$1 - \delta$ , the following two inequalities hold simultaneously:*

$$\mathbb{E}_q\left[\|B_{\mathbf{w}}\|_{\nu}^2\right] \leq \mathbb{E}_q\left[\|B_{\mathbf{w}}\|_D^2\right] - \mathbb{E}_q[v(\mathbf{w})] + \frac{\lambda}{N} KL(q||p) + 4 \frac{R_{max}^2}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}} \quad (15)$$

539

$$\mathbb{E}_q\left[\|B_{\mathbf{w}}\|_D^2\right] \leq \mathbb{E}_q\left[\|B_{\mathbf{w}}\|_{\nu}^2\right] + \mathbb{E}_q[v(\mathbf{w})] + \frac{\lambda}{N} KL(q||p) + 4 \frac{R_{max}^2}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}} \quad (16)$$

540 *Proof.* From Hoeffding's inequality we have:

$$P\left(\left|\mathbb{E}_{\nu, \mathcal{P}}\left[\|B_{\mathbf{w}}\|_D^2\right] - \|B_{\mathbf{w}}\|_D^2\right| > \epsilon\right) \leq 2exp\left(-\frac{2N\epsilon^2}{\left(2\frac{R_{max}}{1-\gamma}\right)^4}\right)$$

541 which implies that, for any  $\delta > 0$ , with probability at least  $1 - \delta$ :

$$\left|\mathbb{E}_{\nu, \mathcal{P}}\left[\|B_{\mathbf{w}}\|_D^2\right] - \|B_{\mathbf{w}}\|_D^2\right| \leq 4 \frac{R_{max}^2}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}}$$

542 Under independence assumptions, the expected TD error can be re-written as:

$$\begin{aligned} \mathbb{E}_{\nu, \mathcal{P}}\left[\|B_{\mathbf{w}}\|_D^2\right] &= \mathbb{E}_{\nu, \mathcal{P}}\left[\frac{1}{N} \sum_{i=1}^N (r_i + \gamma \min_{a'} Q_{\mathbf{w}}(s'_i, a') - Q_{\mathbf{w}}(s_i, a_i))^2\right] \\ &= \mathbb{E}_{\nu, \mathcal{P}}\left[(R(s, a) + \gamma \min_{a'} Q_{\mathbf{w}}(s', a') - Q_{\mathbf{w}}(s, a))^2\right] \\ &= \mathbb{E}_{\nu}\left[\mathbb{E}_{\mathcal{P}}\left[B_{\mathbf{w}}^2\right]\right] \\ &= \mathbb{E}_{\nu}\left[Var_{\mathcal{P}}[B_{\mathbf{w}}] + \mathbb{E}_{\mathcal{P}}[B_{\mathbf{w}}]^2\right] \\ &= v(\mathbf{w}) + \|B_{\mathbf{w}}\|_{\nu}^2 \end{aligned}$$

543 where  $v(\mathbf{w}) \triangleq \mathbb{E}_\nu [\text{Var}_{\mathcal{P}} [B_{\mathbf{w}}]]$ . Thus:

$$\left| \|B_{\mathbf{w}}\|_\nu^2 + v(\mathbf{w}) - \|B_{\mathbf{w}}\|_D^2 \right| \leq 4 \frac{R_{max}^2}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}} \quad (17)$$

544 From the change of measure inequality [32], we have that, for any measurable function  $f(\mathbf{w})$  and  
545 any two probability measures  $p$  and  $q$ :

$$\log \mathbb{E}_p [e^{f(\mathbf{w})}] \geq \mathbb{E}_q [f(\mathbf{w})] - KL(q||p)$$

546 Thus, multiplying both sides of (17) by  $\lambda^{-1}N$  and applying the change of measure inequality with  
547  $f(\mathbf{w}) = \lambda^{-1}N \left| \|B_{\mathbf{w}}\|_\nu^2 + v(\mathbf{w}) - \|B_{\mathbf{w}}\|_D^2 \right|$ , we obtain:

$$\mathbb{E}_q [f(\mathbf{w})] - KL(q||p) \leq \log \mathbb{E}_p [e^{f(\mathbf{w})}] \leq 4 \frac{R_{max}^2 \lambda^{-1}N}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}}$$

548 where the second inequality holds since the right-hand side of (17) does not depend on  $\mathbf{w}$ . Finally,  
549 we can explicitly write:

$$\mathbb{E}_q \left[ \left| \|B_{\mathbf{w}}\|_\nu^2 + v(\mathbf{w}) - \|B_{\mathbf{w}}\|_D^2 \right| \right] \leq \frac{\lambda}{N} KL(q||p) + 4 \frac{R_{max}^2}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}}$$

550 from which the lemma follows straightforwardly.  $\square$

551 From Lemma 2 we can straightforwardly prove the following result which will be of fundamental  
552 importance in the remaining.

553 **Lemma 3.** *Let  $p$  be a prior distribution over the parameter space  $\mathcal{W}$ , and  $\nu$  be a probability measure  
554 over  $\mathcal{S} \times \mathcal{A}$ . Assume  $\hat{\xi}$  is the minimizer of  $ELBO(\xi) = \mathbb{E}_{q_\xi} [\|B_{\mathbf{w}}\|_D^2] + \frac{\lambda}{N} KL(q_\xi||p)$  for a dataset  
555  $D$  of  $N$  samples. Define  $v(\mathbf{w}) \triangleq \mathbb{E}_\nu [\text{Var}_{\mathcal{P}} [B_{\mathbf{w}}]]$ . Then, for any  $\delta > 0$ , with probability at least  
556  $1 - \delta$ :*

$$\mathbb{E}_{q_{\hat{\xi}}} [\|B_{\mathbf{w}}\|_\nu^2] \leq \inf_{\xi \in \Xi} \left\{ \mathbb{E}_{q_\xi} [\|B_{\mathbf{w}}\|_\nu^2] + \mathbb{E}_{q_\xi} [v(\mathbf{w})] + 2 \frac{\lambda}{N} KL(q_\xi||p) \right\} + 8 \frac{R_{max}^2}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}}$$

557 *Proof.* Let us use Lemma 2 for the specific choice  $q = q_{\hat{\xi}}$ . From Eq. (15), we have:

$$\begin{aligned} \mathbb{E}_{q_{\hat{\xi}}} [\|B_{\mathbf{w}}\|_\nu^2] &\leq \mathbb{E}_{q_{\hat{\xi}}} [\|B_{\mathbf{w}}\|_D^2] - \mathbb{E}_{q_{\hat{\xi}}} [v(\mathbf{w})] + \frac{\lambda}{N} KL(q_{\hat{\xi}}||p) + 4 \frac{R_{max}^2}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}} \\ &\leq \mathbb{E}_{q_{\hat{\xi}}} [\|B_{\mathbf{w}}\|_D^2] + \frac{\lambda}{N} KL(q_{\hat{\xi}}||p) + 4 \frac{R_{max}^2}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}} \\ &= \inf_{\xi \in \Xi} \left\{ \mathbb{E}_{q_\xi} [\|B_{\mathbf{w}}\|_D^2] + \frac{\lambda}{N} KL(q_\xi||p) \right\} + 4 \frac{R_{max}^2}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}} \end{aligned}$$

558 where the second inequality holds since  $v(\mathbf{w}) > 0$ , while the equality holds from the definition of  $\hat{\xi}$ .

559 We can now use Eq. (16) to bound  $\mathbb{E}_{q_{\hat{\xi}}} [\|B_{\mathbf{w}}\|_D^2]$ , thus obtaining:

$$\mathbb{E}_{q_{\hat{\xi}}} [\|B_{\mathbf{w}}\|_\nu^2] \leq \inf_{\xi \in \Xi} \left\{ \mathbb{E}_{q_\xi} [\|B_{\mathbf{w}}\|_\nu^2] + \mathbb{E}_{q_\xi} [v(\mathbf{w})] + 2 \frac{\lambda}{N} KL(q_\xi||p) \right\} + 8 \frac{R_{max}^2}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}}$$

560 This concludes the proof.  $\square$

561 It is worth noting the generality of Lemma 3: in bounding the expected Bellman error we did not need  
562 to assume any particular distribution, nor did we have to assume any particular function approximator.

563 We are now ready to state our main result. We start from the Gaussian case and then straightforwardly  
564 extend the proof to the mixture one.

Make sure  
that the con-  
stants multi-  
plying sqrt  
are correct  
now

**Theorem 4.** Fix a target task  $\tau$  a let  $\tilde{Q}$  be the fixed-point of the corresponding mellow Bellman operator. Assume linearly parameterized value functions  $Q_{\mathbf{w}}(s, a) = \mathbf{w}^T \phi(s, a)$  with bounded weights  $\|\mathbf{w}\| \leq w_{\max}$  and uniformly bounded features  $\|\phi(s, a)\| \leq \phi_{\max}$ . Consider the Gaussian version of Alg. 1 with prior  $p(\mathbf{w}) = \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$  and denote by  $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$  the variational parameter minimizing the objective of Eq. 2 on a dataset  $D$  of  $N$  i.i.d. samples distributed according to  $\tau$  and  $\nu$ . Let  $\mathbf{w}^* = \arg\inf_{\mathbf{w}} \left\| \tilde{B}_{\mathbf{w}} \right\|_{\nu}^2$  and define  $v(\mathbf{w}^*) \triangleq \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, \frac{1}{N} \mathbf{I})} [v(\mathbf{w})]$ , with  $v(\mathbf{w}) \triangleq \mathbb{E}_{\nu} [\text{Var}_{\mathcal{P}} [B_{\mathbf{w}}]]$ . Then, there exist constants  $c_1, c_2, c_3$  such that, with probability at least  $1 - 2\delta$  over the choice of weights  $\mathbf{w} \sim \mathcal{N}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$  and dataset  $D$ :

$$\left\| Q_{\mathbf{w}} - \tilde{Q} \right\|_{\nu}^2 \leq \frac{1}{(1-\gamma)\delta} \left( 2 \left\| \tilde{B}_{\mathbf{w}^*} \right\|_{\nu}^2 + v(\mathbf{w}^*) + c_1 \sqrt{\frac{\log \frac{2}{\delta}}{N}} + \frac{c_2 + \lambda \|\mathbf{w}^* - \boldsymbol{\mu}_p\|_{\boldsymbol{\Sigma}_p^{-1}}}{N} + \frac{c_3}{N^2} \right) \quad (18)$$

*Proof.* Using Lemma 3 with variational parameters  $\hat{\boldsymbol{\xi}} = (\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$ , we have:

$$\begin{aligned} \mathbb{E}_{q_{\hat{\boldsymbol{\xi}}}} [\|B_{\mathbf{w}}\|_{\nu}^2] &\leq \inf_{\boldsymbol{\xi} \in \Xi} \left\{ \mathbb{E}_{q_{\boldsymbol{\xi}}} [\|B_{\mathbf{w}}\|_{\nu}^2] + \mathbb{E}_{q_{\boldsymbol{\xi}}} [v(\mathbf{w})] + 2 \frac{\lambda}{N} KL(q_{\boldsymbol{\xi}} \| p) \right\} + 8 \frac{R_{\max}^2}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}} \\ &\leq \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} [\|B_{\mathbf{w}}\|_{\nu}^2] + \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} [v(\mathbf{w})] + 2 \frac{\lambda}{N} KL(\mathcal{N}(\mathbf{w}^*, c\mathbf{I}) \| p) \\ &\quad + 8 \frac{R_{\max}^2}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}} \end{aligned} \quad (19)$$

where the second inequality is due to the fact that, since Lemma 3 contains an infimum over the variational parameters, we can upper bound its right-hand side by choosing any specific  $\boldsymbol{\xi}$  from  $\Xi$ . Here, we choose  $\boldsymbol{\mu} = \mathbf{w}^*$  and  $\boldsymbol{\Sigma} = c\mathbf{I}$ , for some positive constant  $c > 0$ . Let us now bound these terms separately.

**Bounding the expected TD error** We have:

$$\begin{aligned} \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} [\left\| \tilde{B}_{\mathbf{w}} \right\|_{\nu}^2] &= \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} [\mathbb{E}_{\nu} [(\tilde{T}Q_{\mathbf{w}} - Q_{\mathbf{w}})^2]] \\ &= \mathbb{E}_{\nu} [\mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} [(\tilde{T}Q_{\mathbf{w}} - Q_{\mathbf{w}})^2]] \\ &= \mathbb{E}_{\nu} [\mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})}^2 [\tilde{T}Q_{\mathbf{w}} - Q_{\mathbf{w}}]] + \mathbb{E}_{\nu} [\text{Var}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} [\tilde{T}Q_{\mathbf{w}} - Q_{\mathbf{w}}]] \end{aligned} \quad (20)$$

Let us bound these two terms point-wisely for each  $s, a$ . For the first expectation, we have:

$$\begin{aligned} \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} [\tilde{T}Q_{\mathbf{w}} - Q_{\mathbf{w}}] &= \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} [R(s, a) + \gamma \mathbb{E}_{s'} \text{mm}_{a'} \mathbf{w}^T \phi(s', a') - \mathbf{w}^T \phi(s, a)] \\ &= R(s, a) + \gamma \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} [\mathbb{E}_{s'} \text{mm}_{a'} \mathbf{w}^T \phi(s', a')] - \mathbf{w}^{*T} \phi(s, a) \end{aligned} \quad (21)$$

To bound the second term, we adopt Jensen's inequality:

$$\begin{aligned} \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} [\mathbb{E}_{s'} \text{mm}_{a'} \mathbf{w}^T \phi(s', a')] &= \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} \left[ \mathbb{E}_{s'} \frac{1}{\kappa} \log \frac{1}{|\mathcal{A}|} \sum_{a'} e^{\kappa \mathbf{w}^T \phi(s', a')} \right] \\ &\leq \mathbb{E}_{s'} \frac{1}{\kappa} \log \frac{1}{|\mathcal{A}|} \sum_{a'} \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} [e^{\kappa \mathbf{w}^T \phi(s', a')}] \end{aligned} \quad (22)$$

Now, since we know that  $\mathbf{w}^T \phi(s', a') \sim \mathcal{N}(\mathbf{w}^{*T} \phi(s', a'), c \phi(s', a')^T \phi(s', a'))$ ,  $e^{\kappa \mathbf{w}^T \phi(s', a')}$  follows a log-normal distribution with mean  $e^{\kappa \mathbf{w}^{*T} \phi(s', a') + \frac{1}{2} \kappa^2 c \phi(s', a')^T \phi(s', a')}$ . Thus:

$$\begin{aligned} \mathbb{E}_{s'} \frac{1}{\kappa} \log \frac{1}{|\mathcal{A}|} \sum_{a'} \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, cI)} \left[ e^{\kappa \mathbf{w}^T \phi(s', a')} \right] &= \mathbb{E}_{s'} \frac{1}{\kappa} \log \frac{1}{|\mathcal{A}|} \sum_{a'} e^{\kappa \mathbf{w}^{*T} \phi(s', a') + \frac{1}{2} \kappa^2 c \phi(s', a')^T \phi(s', a')} \\ &\leq \mathbb{E}_{s'} \frac{1}{\kappa} \log \frac{1}{|\mathcal{A}|} \sum_{a'} e^{\kappa \mathbf{w}^{*T} \phi(s', a')} e^{\frac{1}{2} \kappa^2 c \phi_{max}^2} \\ &= \mathbb{E}_{s'} \frac{1}{\kappa} \log \frac{1}{|\mathcal{A}|} \sum_{a'} e^{\kappa \mathbf{w}^{*T} \phi(s', a')} + \frac{1}{2} \kappa c \phi_{max}^2 \\ &= \mathbb{E}_{s'} \min_{a'} \mathbf{w}^{*T} \phi(s', a') + \frac{1}{2} \kappa c \phi_{max}^2 \end{aligned}$$

Plugging this into 22 and then into 21, we obtain:

$$\begin{aligned} \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, cI)} \left[ \tilde{T} Q_{\mathbf{w}} - Q_{\mathbf{w}} \right] &\leq R(s, a) + \gamma \mathbb{E}_{s'} \min_{a'} \mathbf{w}^{*T} \phi(s', a') + \frac{1}{2} \gamma \kappa c \phi_{max}^2 - \mathbf{w}^{*T} \phi(s, a) \\ &= \tilde{B}_{\mathbf{w}^*} + \frac{1}{2} \gamma \kappa c \phi_{max}^2 \end{aligned}$$

This implies:

$$\begin{aligned} \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, cI)}^2 \left[ \tilde{T} Q_{\mathbf{w}} - Q_{\mathbf{w}} \right] &\leq \left( \tilde{B}(\mathbf{w}^{*T}) + \frac{1}{2} \gamma \kappa c \phi_{max}^2 \right)^2 \\ &\leq 2 \tilde{B}^2(\mathbf{w}^*) + \frac{1}{2} \gamma^2 \kappa^2 c^2 \phi_{max}^4 \end{aligned}$$

where the second inequality follows from Cauchy-Schwarz inequality. Going back to 20, the first term can now be upper bounded by:

$$\mathbb{E}_{\nu} \left[ \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, cI)}^2 \left[ \tilde{T} Q_{\mathbf{w}} - Q_{\mathbf{w}} \right] \right] \leq 2 \left\| \tilde{B}_{\mathbf{w}^*} \right\|_{\nu}^2 + \frac{1}{2} \gamma^2 \kappa^2 c^2 \phi_{max}^4$$

Let us now consider the variance term of 20 and derive a bound that holds point-wisely for any  $s, a$ . We have:

$$\begin{aligned} Var_{\mathcal{N}(\mathbf{w}^*, cI)} \left[ \tilde{T} Q_{\mathbf{w}} - Q_{\mathbf{w}} \right] &= Var_{\mathcal{N}(\mathbf{w}^*, cI)} \left[ R(s, a) + \gamma \mathbb{E}_{s'} \min_{a'} \mathbf{w}^T \phi(s', a') - \mathbf{w}^T \phi(s, a) \right] \\ &= Var_{\mathcal{N}(\mathbf{w}^*, cI)} \left[ \gamma \mathbb{E}_{s'} \min_{a'} \mathbf{w}^T \phi(s', a') - \mathbf{w}^T \phi(s, a) \right] \\ &= Var_{\mathcal{N}(\mathbf{w}^*, cI)} \left[ \gamma \mathbb{E}_{s'} \min_{a'} \mathbf{w}^T \left( \phi(s', a') - \frac{1}{\gamma} \phi(s, a) \right) \right] \\ &= \gamma^2 Var_{\mathcal{N}(\mathbf{w}^*, I)} \left[ \mathbb{E}_{s'} \min_{a'} \sqrt{c} \mathbf{w}^T \left( \phi(s', a') - \frac{1}{\gamma} \phi(s, a) \right) \right] \end{aligned}$$

From Cauchy-Schwarz inequality:

$$\begin{aligned} \sqrt{c} \left| \mathbf{w}^T \left( \phi(s', a') - \frac{1}{\gamma} \phi(s, a) \right) \right| &\leq \sqrt{c} \|\mathbf{w}\| \left\| \phi(s', a') - \frac{1}{\gamma} \phi(s, a) \right\| \\ &\leq \sqrt{c} \mathbf{w}_{max} \phi_{max} \frac{1 + \gamma}{\gamma} \end{aligned}$$

Then, the random variable over which the variance is computed is limited in  $[-\sqrt{c} \mathbf{w}_{max} \phi_{max} \frac{1 + \gamma}{\gamma}, \sqrt{c} \mathbf{w}_{max} \phi_{max} \frac{1 + \gamma}{\gamma}]$  and the variance can be straightforwardly bounded using Popoviciu's inequality:

$$Var_{\mathcal{N}(\mathbf{w}^*, cI)} \left[ \tilde{T} Q_{\mathbf{w}} - Q_{\mathbf{w}} \right] \leq \gamma^2 \frac{1}{4} \left( 2 \sqrt{c} \mathbf{w}_{max} \phi_{max} \frac{1 + \gamma}{\gamma} \right)^2 = c (\mathbf{w}_{max} \phi_{max} (1 + \gamma))^2$$

We can finally plug everything into 20, thus obtaining:

$$\mathbb{E}_{\mathcal{N}(\mathbf{w}^*, cI)} \left[ \left\| \tilde{B}_{\mathbf{w}^*} \right\|_{\nu}^2 \right] \leq 2 \left\| \tilde{B}_{\mathbf{w}^*} \right\|_{\nu}^2 + \frac{1}{2} \gamma^2 \kappa^2 c^2 \phi_{max}^4 + c (\mathbf{w}_{max} \phi_{max} (1 + \gamma))^2$$

594 **Bounding the KL divergence** We have:

$$\begin{aligned}
KL(\mathcal{N}(\mathbf{w}^*, c\mathbf{I}) \parallel p) &= KL(\mathcal{N}(\mathbf{w}^*, c\mathbf{I}) \parallel \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)) \\
&= \frac{1}{2} \left( \log \frac{|\boldsymbol{\Sigma}_p|}{c^d} + c \text{Tr}(\boldsymbol{\Sigma}_p^{-1}) + \|\mathbf{w}^* - \boldsymbol{\mu}_p\|_{\boldsymbol{\Sigma}_p^{-1}}^2 - d \right) \\
&\leq \frac{1}{2} d \log \frac{\sigma_{max}}{c} + \frac{1}{2} d \frac{c}{\sigma_{min}} + \frac{1}{2} \|\mathbf{w}^* - \boldsymbol{\mu}_p\|_{\boldsymbol{\Sigma}_p^{-1}}^2
\end{aligned}$$

595 Now, putting all together into 19:

$$\begin{aligned}
\mathbb{E}_{q_{\hat{\xi}}} [\|B_{\mathbf{w}}\|_{\nu}^2] &\leq 2 \|\tilde{B}_{\mathbf{w}^*}\|_{\nu}^2 + \frac{1}{2} \gamma^2 \kappa^2 c^2 \phi_{max}^4 + c(\mathbf{w}_{max} \phi_{max} (1 + \gamma))^2 + \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} [v(\mathbf{w})] \\
&\quad + \frac{\lambda}{N} d \log \frac{\sigma_{max}}{c} + \frac{\lambda}{N} d \frac{c}{\sigma_{min}} + \frac{\lambda}{N} \|\mathbf{w}^* - \boldsymbol{\mu}_p\|_{\boldsymbol{\Sigma}_p^{-1}}^2 + 8 \frac{R_{max}^2}{(1 - \gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}}
\end{aligned}$$

596 Since the bound holds for any  $c > 0$ , we can set it to  $1/N$ , thus obtaining:

$$\begin{aligned}
\mathbb{E}_{q_{\hat{\xi}}} [\|B_{\mathbf{w}}\|_{\nu}^2] &\leq 2 \|\tilde{B}_{\mathbf{w}^*}\|_{\nu}^2 + v(\mathbf{w}^*) + \frac{1}{N^2} \left( \frac{1}{2} \gamma^2 \kappa^2 \phi_{max}^4 + \frac{\lambda d}{\sigma_{min}} \right) \\
&\quad + \frac{1}{N} \left( \mathbf{w}_{max}^2 \phi_{max}^2 (1 + \gamma)^2 + \lambda d (\log \sigma_{max} + \log N) + \lambda \|\mathbf{w}^* - \boldsymbol{\mu}_p\|_{\boldsymbol{\Sigma}_p^{-1}}^2 \right) \\
&\quad + 8 \frac{R_{max}^2}{(1 - \gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}}
\end{aligned}$$

597 Finally, defining the constants  $c_1 = \frac{8R_{max}^2}{\sqrt{2}(1-\gamma)^2}$ ,  $c_2 = \mathbf{w}_{max}^2 \phi_{max}^2 (1 + \gamma)^2 + \lambda d (\log \sigma_{max} + \log N)$ ,

598 and  $c_3 = \frac{1}{2} \gamma^2 \kappa^2 \phi_{max}^4 + \frac{\lambda d}{\sigma_{min}}$ , we obtain:

$$\mathbb{E}_{q_{\hat{\xi}}} [\|B_{\mathbf{w}}\|_{\nu}^2] \leq 2 \|\tilde{B}_{\mathbf{w}^*}\|_{\nu}^2 + v(\mathbf{w}^*) + c_1 \sqrt{\frac{\log \frac{2}{\delta}}{N}} + \frac{c_2 + \lambda \|\mathbf{w}^* - \boldsymbol{\mu}_p\|_{\boldsymbol{\Sigma}_p^{-1}}^2}{N} + \frac{c_3}{N^2}$$

599 Let us now apply Corollary 1. We have that, with probability at least  $1 - \delta$ :

$$\|Q_{\mathbf{w}} - \tilde{Q}\|_{\nu}^2 \leq \frac{\mathbb{E}_{q_{\hat{\xi}}} [\|B_{\mathbf{w}}\|_{\nu}^2]}{(1 - \gamma)\delta}$$

600 Thus, we probability at least  $1 - 2\delta$ :

$$\|Q_{\mathbf{w}} - \tilde{Q}\|_{\nu}^2 \leq \frac{1}{(1 - \gamma)\delta} \left( 2 \|\tilde{B}_{\mathbf{w}^*}\|_{\nu}^2 + v(\mathbf{w}^*) + c_1 \sqrt{\frac{\log \frac{2}{\delta}}{N}} + \frac{c_2 + \lambda \|\mathbf{w}^* - \boldsymbol{\mu}_p\|_{\boldsymbol{\Sigma}_p^{-1}}^2}{N} + \frac{c_3}{N^2} \right)$$

601 □

602 **Theorem 5.** Fix a target task  $\tau$  a let  $\tilde{Q}$  be the fixed-point of the corresponding mellow Bellman  
603 operator. Assume linearly parameterized value functions  $Q_{\mathbf{w}}(s, a) = \mathbf{w}^T \phi(s, a)$  with bounded  
604 weights  $\|\mathbf{w}\| \leq w_{max}$  and uniformly bounded features  $|\phi(s, a)| \leq \phi_{max}$ . Consider the mixture  
605 version of Alg. 1 using  $C$  components, source task weights  $\mathcal{W}_s$ , and bandwidth  $\sigma_p^2$  for the prior.

606 Denote by  $\hat{\xi} = (\hat{\boldsymbol{\mu}}_1, \dots, \hat{\boldsymbol{\mu}}_C, \hat{\boldsymbol{\Sigma}}_1, \dots, \hat{\boldsymbol{\Sigma}}_C)$  the variational parameters minimizing the objective of  
607 Eq. 2 on a dataset  $D$  of  $N$  i.i.d. samples distributed according to  $\tau$  and  $\nu$ . Let  $\mathbf{w}^* = \arg\inf_{\mathbf{w}} \|\tilde{B}_{\mathbf{w}}\|_{\nu}^2$   
608 and define  $v(\mathbf{w}^*) \triangleq \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, \frac{1}{N}\mathbf{I})} [v(\mathbf{w})]$ , with  $v(\mathbf{w}) \triangleq \mathbb{E}_{\nu} [\text{Var}_{\mathcal{P}} [B_{\mathbf{w}}]]$ . Then, there exist constants  
609  $c_1, c_2, c_3$  such that, with probability at least  $1 - 2\delta$  over the choice of weights  $\mathbf{w} \sim \frac{1}{C} \sum_i \mathcal{N}(\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i)$   
610 and dataset  $D$ :

$$\|Q_{\mathbf{w}} - \tilde{Q}\|_{\nu}^2 \leq \frac{1}{(1 - \gamma)\delta} \left( 2 \|\tilde{B}_{\mathbf{w}^*}\|_{\nu}^2 + v(\mathbf{w}^*) + c_1 \sqrt{\frac{\log \frac{2}{\delta}}{N}} + \frac{c_2 + 2\lambda \text{softmin}_j \left\{ \frac{1}{2\sigma_p^2} \|\mathbf{w}^* - \mathbf{w}_j\| \right\}}{N} + \frac{c_3}{N^2} \right) \quad (23)$$

611 where  $\text{softmin}_i x \triangleq \sum_j \frac{e^{-x_j}}{\sum_j e^{-x_j}} x_i$ .

612 *Proof.* Similarly to the previous proof, we can apply Lemma 3 with variational parameters  $\hat{\xi} =$   
 613  $(\hat{\mu}_1, \dots, \hat{\mu}_C, \hat{\Sigma}_1, \dots, \hat{\Sigma}_C)$ , while choosing the same specific parameters for the right-hand side:  
 614  $\mu_i = \mathbf{w}^*$  and  $\Sigma_i = c\mathbf{I}$  for all  $i = 1, \dots, C$ . Then, we obtain:

$$\begin{aligned} \mathbb{E}_{q_{\xi}} [\|B\mathbf{w}\|_{\nu}^2] &\leq \inf_{\xi \in \Xi} \left\{ \mathbb{E}_{q_{\xi}} [\|B\mathbf{w}\|_{\nu}^2] + \mathbb{E}_{q_{\xi}} [v(\mathbf{w})] + 2\frac{\lambda}{N} KL(q_{\xi} || p) \right\} + 8\frac{R_{max}^2}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}} \\ &\leq \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} [\|B\mathbf{w}\|_{\nu}^2] + \mathbb{E}_{\mathcal{N}(\mathbf{w}^*, c\mathbf{I})} [v(\mathbf{w})] + 2\frac{\lambda}{N} KL(\mathcal{N}(\mathbf{w}^*, c\mathbf{I}) || p) \\ &\quad + 8\frac{R_{max}^2}{(1-\gamma)^2} \sqrt{\frac{\log \frac{2}{\delta}}{2N}} \end{aligned} \quad (24)$$

615 The only difference w.r.t. Eq. (19) of Thm. 4 is the KL divergence term, which now contains a  
 616 mixture distribution. From Thm. 1 we have:

$$KL(\mathcal{N}(\mathbf{w}^*, c\mathbf{I}) || p) \leq KL(\chi^{(2)} || \chi^{(1)}) + \sum_j \chi_j^{(2)} KL(\mathcal{N}(\mathbf{w}^*, c\mathbf{I}) || \mathcal{N}(\mathbf{w}_j, \sigma_p^2 \mathbf{I})) \quad (25)$$

617 where the vectors  $\chi^{(1)}$  and  $\chi^{(2)}$  are the ones defined in Thm. 1. Notice that, since we reduced the  
 618 posterior to one component, we can get rid of the index  $i$ . Using the definitions of these two vectors  
 619 from Sec. 8 of [13], we have:

$$\chi_j^{(1)} = \frac{1}{|\mathcal{W}_s|} \forall j = 1, \dots, |\mathcal{W}_s|$$

620

$$\chi_j^{(2)} = \frac{e^{-KL(\mathcal{N}(\mathbf{w}^*, c\mathbf{I}) || \mathcal{N}(\mathbf{w}_j, \sigma_p^2 \mathbf{I}))}}{\sum_{j'} e^{-KL(\mathcal{N}(\mathbf{w}^*, c\mathbf{I}) || \mathcal{N}(\mathbf{w}_{j'}, \sigma_p^2 \mathbf{I}))}} \forall j = 1, \dots, |\mathcal{W}_s| \quad (26)$$

621 Since the KL divergence is:

$$KL(\mathcal{N}(\mathbf{w}^*, c\mathbf{I}) || \mathcal{N}(\mathbf{w}_j, \sigma_p^2 \mathbf{I})) = \frac{1}{2} \left( d \log \frac{\sigma_p^2}{c} + d \frac{c}{\sigma_p^2} + \frac{1}{\sigma_p^2} \|\mathbf{w}^* - \mathbf{w}_j\|^2 - d \right)$$

622 Eq. 26 can be rewritten as:

$$\chi_j^{(2)} = \frac{e^{-\frac{1}{2\sigma_p^2} \|\mathbf{w}^* - \mathbf{w}_j\|^2}}{\sum_{j'} e^{-\frac{1}{2\sigma_p^2} \|\mathbf{w}^* - \mathbf{w}_{j'}\|^2}} \forall j = 1, \dots, |\mathcal{W}_s|$$

623 Let us bound the two terms of (25) separately. For the first one, we have:

$$\begin{aligned} KL(\chi^{(2)} || \chi^{(1)}) &= \sum_j \chi_j^{(2)} \log \frac{\chi_j^{(2)}}{\chi_j^{(1)}} \\ &= \sum_j \chi_j^{(2)} \log \chi_j^{(2)} - \sum_j \chi_j^{(2)} \log \frac{1}{|\mathcal{W}_s|} \\ &\leq \log |\mathcal{W}_s| \end{aligned}$$

624 where the inequality holds since the first term is negative. For the second one:

$$\begin{aligned} \sum_j \chi_j^{(2)} KL(\mathcal{N}(\mathbf{w}^*, c\mathbf{I}) || \mathcal{N}(\mathbf{w}_j, \sigma_p^2 \mathbf{I})) &= \frac{1}{2} \sum_j \chi_j^{(2)} \left( d \log \frac{\sigma_p^2}{c} + d \frac{c}{\sigma_p^2} + \frac{1}{\sigma_p^2} \|\mathbf{w}^* - \mathbf{w}_j\|^2 - d \right) \\ &\leq \frac{1}{2} d \log \frac{\sigma_p^2}{c} + \frac{1}{2} d \frac{c}{\sigma_p^2} + \sum_j \chi_j^{(2)} \frac{1}{2\sigma_p^2} \|\mathbf{w}^* - \mathbf{w}_j\|^2 \\ &= \frac{1}{2} d \log \frac{\sigma_p^2}{c} + \frac{1}{2} d \frac{c}{\sigma_p^2} + \text{softmax}_j \left\{ \frac{1}{2\sigma_p^2} \|\mathbf{w}^* - \mathbf{w}_j\|^2 \right\} \end{aligned}$$



625 Putting the two terms together:

$$KL(\mathcal{N}(\mathbf{w}^*, c\mathbf{I}) \parallel p) \leq \log |\mathcal{W}_s| + \frac{1}{2}d \log \frac{\sigma_p^2}{c} + \frac{1}{2}d \frac{c}{\sigma_p^2} + \text{softmin}_j \left\{ \frac{1}{2\sigma_p^2} \|\mathbf{w}^* - \mathbf{w}_j\| \right\}$$

626 Notice that, from now on, one can simply apply the proof of Thm. 4 with  $\sigma_{max} = \sigma_{min} = \sigma_p^2$   
 627 and  $\frac{1}{2} \|\mathbf{w}^* - \boldsymbol{\mu}_p\|_{\Sigma_p^{-1}}$  replaced by  $\text{softmin}_j \left\{ \frac{1}{2\sigma_p^2} \|\mathbf{w}^* - \mathbf{w}_j\| \right\}$ . Thus, by redefining the three  
 628 constants to  $c_1 = \frac{8R_{max}^2}{\sqrt{2}(1-\gamma)^2}$ ,  $c_2 = \mathbf{w}_{max}^2 \phi_{max}^2 (1 + \gamma)^2 + \lambda d (\log \sigma_p^2 + \log N) + 2\lambda \log |\mathcal{W}_s|$ ,  
 629 and  $c_3 = \frac{1}{2} \gamma^2 \kappa^2 \phi_{max}^4 + \frac{\lambda d}{\sigma_p^2}$ , we can write that, with probability at least  $1 - 2\delta$ :

$$\|Q_{\mathbf{w}} - \tilde{Q}\|_{\nu}^2 \leq \frac{1}{(1-\gamma)\delta} \left( 2 \|\tilde{B}_{\mathbf{w}^*}\|_{\nu}^2 + v(\mathbf{w}^*) + c_1 \sqrt{\frac{\log \frac{2}{\delta}}{N}} + \frac{c_2 + 2\lambda \text{softmin}_j \left\{ \frac{1}{2\sigma_p^2} \|\mathbf{w}^* - \mathbf{w}_j\| \right\}}{N} + \frac{c_3}{N^2} \right)$$

630

□

## 631 B Additional Details on the Algorithms

### 632 B.1 Gaussian Variational Transfer

633 Under Gaussian distributions, all quantities of interest for using Alg. 1 can be computed very easily.  
 634 The KL divergence between the prior and approximate posterior can be computed in closed-form as:

$$KL(q_{\xi}(\mathbf{w}) \parallel p(\mathbf{w})) = \frac{1}{2} \left( \log \frac{|\Sigma_p|}{|\Sigma|} + \text{Tr}(\Sigma_p^{-1} \Sigma) + (\boldsymbol{\mu} - \boldsymbol{\mu}_p)^T \Sigma_p^{-1} (\boldsymbol{\mu} - \boldsymbol{\mu}_p) - d \right) \quad (27)$$

635 for  $\xi = (\boldsymbol{\mu}, \mathbf{L})$  and  $\Sigma = \mathbf{L}\mathbf{L}^T$ . Its gradients with respect to the variational parameters are:

$$\nabla_{\boldsymbol{\mu}} KL(q_{\xi}(\mathbf{w}) \parallel p(\mathbf{w})) = \Sigma_p^{-1} (\boldsymbol{\mu} - \boldsymbol{\mu}_p) \quad (28)$$

636

$$\nabla_{\mathbf{L}} KL(q_{\xi}(\mathbf{w}) \parallel p(\mathbf{w})) = \Sigma_p^{-1} \mathbf{L} - (\mathbf{L}^{-1})^T \quad (29)$$

637 Finally, the gradients w.r.t. the expected likelihood term of the variational objective (2) can be  
 638 computed using the reparameterization trick (e.g., [14, 29]):

$$\nabla_{\boldsymbol{\mu}} \mathbb{E}_{\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{L}\mathbf{L}^T)} [\|B_{\mathbf{w}}\|_D^2] = \mathbb{E}_{\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [\nabla_{\mathbf{w}} \|B_{\mathbf{w}}\|_D^2] \quad \text{for } \mathbf{w} = \mathbf{L}\mathbf{v} + \boldsymbol{\mu} \quad (30)$$

639

$$\nabla_{\mathbf{L}} \mathbb{E}_{\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{L}\mathbf{L}^T)} [\|B_{\mathbf{w}}\|_D^2] = \mathbb{E}_{\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [\nabla_{\mathbf{w}} \|B_{\mathbf{w}}\|_D^2 \cdot \mathbf{v}^T] \quad \text{for } \mathbf{w} = \mathbf{L}\mathbf{v} + \boldsymbol{\mu} \quad (31)$$

### 640 B.2 Mixture of Gaussian Variational Transfer

641 For the implementation of the Mixture of Gaussian Variational Transfer, we use the upper bound  
 642 on the KL divergence between two Mixtures of Gaussians, as in Theorem1, to obtain an upper  
 643 bound on the negative ELBO in Equation2. Consider we have  $C$  components for the posterior  
 644 family  $q_{\xi}(\mathbf{w}) = \frac{1}{C} \sum_{i=1}^C \mathcal{N}(\mathbf{w} | \boldsymbol{\mu}_i, \Sigma_i)$  and a prior distribution, constructed from the set of weights  
 645  $\mathcal{W}_s = \{\mathbf{w}_1, \dots, \mathbf{w}_{|\mathcal{W}_s|}\}$  of the sources' optimal  $Q$ -functions,  $p(\mathbf{w}) = \frac{1}{|\mathcal{W}_s|} \sum_{j=1}^{|\mathcal{W}_s|} \mathcal{N}(\mathbf{w} | \mathbf{w}_j, \sigma_p^2 \mathbf{I})$ .

$$KL(q_{\xi}(\mathbf{w}) \parallel p(\mathbf{w})) \leq KL(\chi^{(2)} \parallel \chi^{(1)}) + \sum_{i=1}^C \sum_{j=1}^{|\mathcal{W}_s|} \chi_{j,i}^{(2)} KL(\mathcal{N}(\mathbf{w} | \boldsymbol{\mu}_i, \Sigma_i) \parallel \mathcal{N}(\mathbf{w} | \mathbf{w}_j, \sigma_p^2 \mathbf{I})) \quad (32)$$

646 And substituting (32) in the negative ELBO in 2 we get the following upper bound.

$$\begin{aligned} \mathcal{L}(\xi) &\leq \tilde{\mathcal{L}}(\xi) = \mathbb{E}_{\mathbf{w} \sim q_{\xi}} [\|B_{\mathbf{w}}\|_D^2] \\ &\quad + \frac{\lambda}{N} KL(\chi^{(2)} \parallel \chi^{(1)}) + \frac{\lambda}{N} \sum_{i=1}^C \sum_{j=1}^{|\mathcal{W}_s|} \chi_{j,i}^{(2)} KL(\mathcal{N}(\mathbf{w} | \boldsymbol{\mu}_i, \Sigma_i) \parallel \mathcal{N}(\mathbf{w} | \mathbf{w}_j, \sigma_p^2 \mathbf{I})) \end{aligned} \quad (33)$$

Finally, using this upper bound as objective of our optimization problem, we can then exploit the linearity of the expectation operator to obtain

$$\begin{aligned}\tilde{\mathcal{L}}(\xi) = & \frac{1}{C} \sum_{i=1}^C \mathbb{E}_{\mathbf{w} \sim \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)} \left[ \|B_{\mathbf{w}}\|_D^2 \right] \\ & + \frac{\lambda}{N} KL(\chi^{(2)} || \chi^{(1)}) + \frac{\lambda}{N} \sum_{i=1}^C \sum_{j=1}^{|\mathcal{W}_s|} \chi_{j,i}^{(2)} KL(\mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) || \mathcal{N}(\mathbf{w}|\mathbf{w}_j, \sigma_p^2 \mathbf{I}))\end{aligned}\quad (34)$$

that is easily differentiable with respect to  $\xi = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_C, \boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_C)$  using the Eq. 28, 29, 30, 31 derived for the Gaussian case.

## C Additional Details on the Experiments

In the present section we present the values of the hyper-parameters that we use for the experiments in this paper.

It is important to notice that for the implementation of the algorithms we use a Replay Buffer with a fixed size and batches of a given size sampled randomly from it to perform the gradient steps and use ADAM as the optimizer with the default parameters.

### C.1 The Rooms Problem

For these experiments, in order to train the source tasks for the Rooms Problem, we directly minimize the expected TD error based on the *mellow* Bellman operator by stochastic gradient descent. We set to use a *Batch Size* of 50, a *Buffer Size* of 50000,  $\psi = 0.5$  and a learning rate  $\alpha = 0.001$ . Additionally, for exploration we use an *Exploration Fraction* of 0.7.

For the transfer algorithm GVT, we set a *Batch Size* of 50 and a *Buffer Size* of 10000. We use  $\psi = 0.5$ ,  $\lambda = 10^{-4}$  and 10 *weights* to estimate the expected TD error. For the learning rates we use  $\alpha_{\mu} = 0.001$  for the mean of the posterior Gaussian and  $\alpha_L = 0.0001$  to learn its Cholesky factor L. Furthermore, we restrict the minimum value reachable for the eigenvalues of these factor to be  $\sigma_{min}^2 = 0.0001$ . In the case of MGVT we use, instead,  $\lambda = 10^{-6}$ ,  $\alpha_{\mu} = 0.001$  and  $\alpha_L = 0.1$ . Finally, for the prior's covariances we set  $\sigma_p^2 = 10^{-5}$ .

Besides the results that we show in Sec. 5.1, we present in this section further empirical evaluation.

Firstly, we show the results of the evaluation performance. We compute this as the average reward gotten when the agent acts using the greedy policy using the parameters at the iteration of evaluation. In the case of GVT, we take the mean of the posterior Gaussian and, in MGVT, we compute the mean of the posterior by averaging the means of its components and use it for evaluation.

In Fig. ?? we show the results when evaluating the Rooms Problem performance when the sources used to transfer have sample tasks resulting of both doors being sampled uniformly. It is easily noticeable that both GVT and MGVT perform much better in comparison with the no transfer performance and shows that the mean behavior of our posterior distribution, indeed, converges to the actual optimal solution.

In Fig.3b we show the evaluation for the generalization experiment when the sampled source tasks have a door fixed and the target task is generated by sampling both doors' positions. From this we can clearly appreciate that MGVT is able to quickly converge to the optimal solution in this more complicated setting, whereas GVT fails to adapt as consistently as MGVT, thus the higher variance and distance to the optimal value. In this scenario, using a Gaussian to model the prior over-constrains the algorithm to stay close to part of the function space that cannot solve optimally the target tasks sampled from the modified distribution.

Furthermore, we investigate the exploratory behavior induced by our transfer algorithms and how they differ between each other and in comparison with  $\epsilon$ -greedy exploration. In Fig. 4, we show the results of running the no transfer (NT) algorithm, GVT and 1-MGVT for  $2k$  iterations and we represent in the plot the positions visited by the agent.

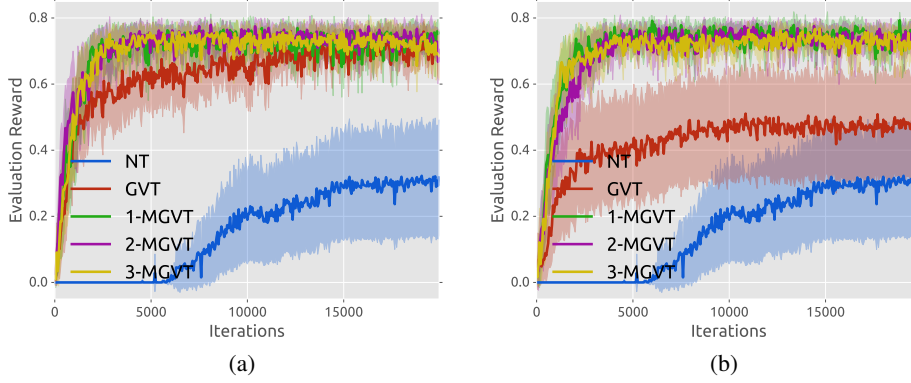


Figure 3: (a) Rooms Problem: Evaluation Reward, (b) Rooms Problem: Evaluation Reward in the generalization experiment

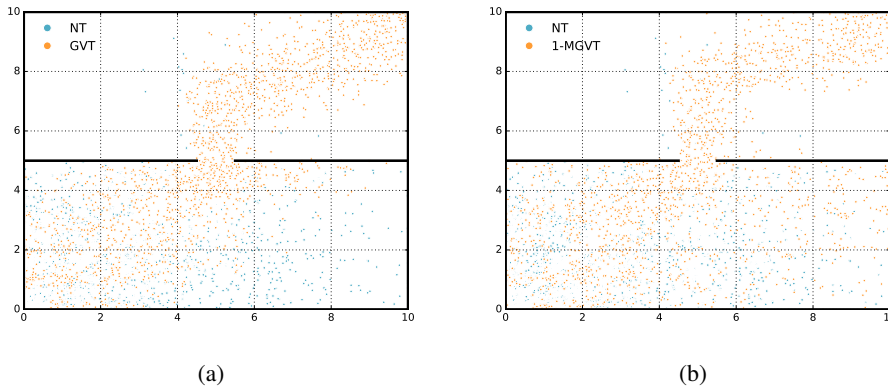


Figure 4: (a) Two-Rooms Problem:  $\epsilon$ -greedy vs. GVT, (b) Two-Rooms Problem:  $\epsilon$ -greedy vs. 1-MGVT experiment

689 Observing Fig. 4a, it is possible to understand the difference between the  $\epsilon$ -greedy exploration and  
 690 the resulting behavior from GVT. It is noticeable that NT is not capable to lead the agent to the goal  
 691 within the given iterations as most of the states visited are sparse within the first room, whereas GVT  
 692 is able to concentrate more of its effort in looking for the door around the middle of the wall. After  
 693 finding it, already in the second room, the positions concentrate in the path leading to the goal given  
 694 that the need for the exploration is less. This is not surprising as the value function should be equal  
 695 for all tasks after crossing the door.

696 In the other case, we have Fig. 4b that shows the same situation as with GVT, but it is quite interesting  
 697 to notice how sparser the exploration of 1-MGVT is with respect to GVT. Indeed, 1-MGVT is able to  
 698 actually explore the right part of the first room within these iterations, which might be seen as the  
 699 result of the prior model being able to capture more information than the Gaussian; hence, the higher  
 700 speed-up in convergence and robustness to changes in the distribution from which target tasks are  
 701 drawn. Indeed, as 1-MGVT is able to allow for more flexible exploration, it is capable to discover  
 702 how to best solve the task much faster than GVT.

703 Finally, in Fig. 5 we present the expected return as a function of the number of source tasks used  
 704 for GVT and MGVT. In particular, we show the resulting curves after 1000 iterations in Fig. 5a and  
 705 after 1950 iterations in Fig. 5b. It is interesting to notice the difference on performance between  
 706 MGVT and GVT whenever there is a small number of source tasks. MGVT clearly provides faster  
 707 adaptation in the presence of low prior knowledge as it can be discern from the gap seen in nearly  
 708 a 1000 iterations between the plots. It is as expected because approximating the prior Gaussian  
 709 distribution using maximum likelihood estimations with a low number of samples does not provide  
 710 enough precision. As the number of source tasks increases, as seen more clearly from Fig. 5b, the  
 711 performances become closer for this environment.

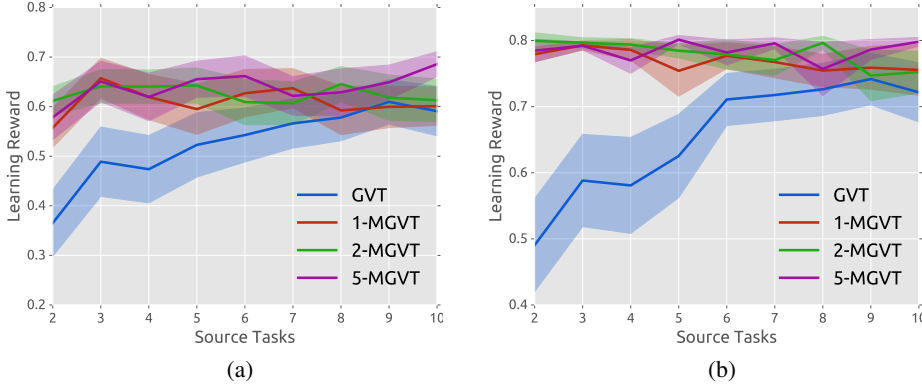


Figure 5: Expected return w.r.t. to the number of source tasks (a) 1000 iterations, (b) 1950 iterations

Finally, among the MGVT curves in Fig. 5, we see that, even if more components lead to slightly better performances, the advantage is not quite significant in this environment. As seen before, the actual performances will converge to optimality in all the MGVT cases. Clearly, as we expect all components to converge to the optimal parameters, the initial performance difference will, indeed, reduce fast.

## C.2 Classic Control

### C.2.1 Cartpole

For this environment we generate tasks by uniformly sampling the cart mass in the range  $[0.5, 1.5]$ , the pole mass in  $[0.1, 0.3]$  and the pole length in  $[0.2, 1.0]$ .

During the training of the source tasks, we use a *Batch Size* of 150 and *Buffer Size* of 50000. Specifically, for DDQN we use a *Target Update Frequency* of 500 and *Exploration Fraction* of 0.35 and a learning rate  $\alpha = 0.001$ . We use a Multilayer Perceptron (MLP) with ReLU as activation function and a hidden layer of 32 neurons, an input layer of 4 and the value of the 2 actions as the output layer.

For the transfer experiments, we set the *Batch Size* to 500, the number of *weights* sampled to approximate the expected TD error to 5,  $\lambda = 0.001$  and  $\psi = 0.5$ . We use  $\alpha_\mu = 0.001$  as the learning rate for the mean of the Gaussian posterior. For its the Cholesky factor  $L$  we use  $\alpha_L = 0.0001$  and set the limit that the minimum eigenvalue may reach to  $\sigma_{min}^2 = 0.0001$ . Additionally, for MGVT we set the variance of the prior components  $\sigma_p^2 = 10^{-5}$  and leave the learning rates of the posterior components' means and Cholesky factor the same as GVT.

In Fig. 6a, we show the evaluation reward for DDQN, GVT, 1-MGVT and 3-MGVT, which consist of the average performance of the greedy policy using the parameters in the iteration of evaluation. In the case of GVT, we use the mean of the Gaussian posterior for the greedy policy and for MGVT corresponds to the mean of the Mixture of Gaussians for the posterior, that reduces to the average of the components' means. From the figure we can see how the transfer methods provide a significant jump-start w.r.t. the DDQN evaluation. Although, a slight performance drop is seen as the algorithm adapts to the optimal Q-function, the algorithm quickly reaches optimality and converges to the actual optimal performance while DDQN, within these iterations, remains on average slightly suboptimal.

improve this

### C.2.2 Mountain Car

We generate tasks sampling uniformly the base speed of the actions in the range  $[0.001, 0.0015]$ .

For the sources, we train the tasks using DDQN with a *Target Update Frequency* of 500, a *Batch Size* of 32, a *Buffer Size* of 50000 and learning rate  $\alpha = 0.001$ . Moreover, we set the *Exploration Fraction* to 0.15. We use an MLP with single hidden layer of 64 neurons with ReLU activation function, an input layer of 4 neurons and as output the value of each of the 3 actions.

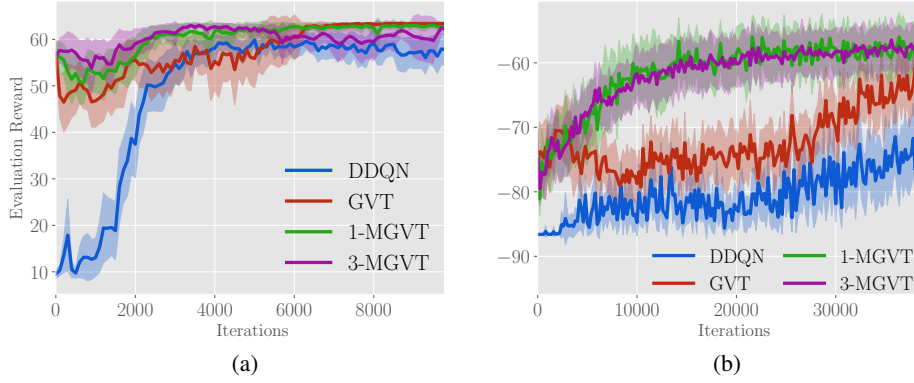


Figure 6: (a) Cartpole: Evaluation Reward, (b) Mountain Car: Evaluation Reward

For the transfer experiments, we set the *Batch Size* to 500, and use 10 *weights* to approximate the expected TD error,  $\lambda = 10^{-5}$  and  $\psi = 0.5$ . For the learning rates, we use  $\alpha_\mu = 0.001$  for learning the means of the Gaussians. In the case of the Cholesky factors  $L$ , we use  $\alpha_L = 0.0001$  and allow the eigenvalues to reach a minimum value of  $\sigma_{min}^2 = 0.0001$ . In the case of MGVT, additionally, we set the prior covariance to be  $\sigma_p^2 = 10^{-5}$ .

Finally, in Fig. 6b, we show the evaluation reward obtained during the executions, which, as before, corresponds to the average reward gotten acting with a greedy policy with the parameters of the network in the case of DDQN and the mean value of the posterior in the cases of GVT and MGVT in the iteration of evaluation. In this plot, we can observe directly how fast is MGVT able to converge its mean to the actual optimal performance and that even if GVT struggles to find learn in comparison with MGVT, still provides a clear advantage w.r.t. DDQN. Also, it is worth noting that the variances seen are result that different speeds of the car result in different time to reach the goal and, thus, different optimal performances.

Clearly, this variation in the values among tasks allows MGVT to excel w.r.t. to the other methods. The richer prior of MGVT allows to exploit efficiently the previous knowledge on the possible values attainable from the tasks and, therefore, quickly resolve how to best optimize for the target. This fact is a clear contrast with GVT as the search is more likely constrained to the mean values and moving away from that is, in fact, slower.

### C.3 Maze Navigation

For the maze navigation task in Sec. 5.3, here we enumerate the mazes that were designed to realize the experiments. In Fig. 7, there are 20 mazes with varying degree of difficulty and that were designed to hold some similarities that would be useful for transferring. Moreover, we ensure 4 groups of mazes that are characterized by their goal position.

For the experiments we use as an approximator an MLP with two hidden layers of 32 neurons with ReLU as activation functions, an input layer receiving the 22-dimensional state of the MDP and an output layer that produces the value of each of the 3 action. For training the sources we use DDQN with a *Batch size* of 70, a *Buffer Size* of 10000 and a *Target Update Frequency* of 100, setting the *Exploration Fraction* to 0.1 and learning rate  $\alpha = 0.001$ .

In transfer experiments we use  $\psi = 0.5$ , a *Batch Size* of 50, a *Buffer Size* of 50000 and use 10 sampled *weights* from the posterior to approximate the TD error. Moreover, we use  $\lambda = 10^{-6}$ . For GVT, in particular, we use  $\alpha_\mu = 0.001$  as the rate to learn the mean parameters of our Gaussian posterior. For the Cholesky factor  $L$ , instead, we use  $\alpha_L = 10^{-7}$  and set the minimum value reachable by its eigenvalues to be  $\sigma_{min} = 0.0001$ . In the case of MGVT experiments we set  $\alpha_\mu = 0.001$  to learn the mean of the different Gaussian components of our posterior and set  $\alpha_L = 10^{-6}$  for the Cholesky factors of its covariances. Finally, we use  $\sigma_p^2 = 10^{-5}$  as the prior variance.

In Fig. ??, we present the results of transferring from 5 source tasks to the mazes shown in Fig. 7a, Fig. 7g, Fig. 7i and Fig. 7n using GVT and MGVT. We show both the evaluation reward, i.e. the average performance obtained evaluating the mean parameters of the posterior in a given iteration of

also GVT?

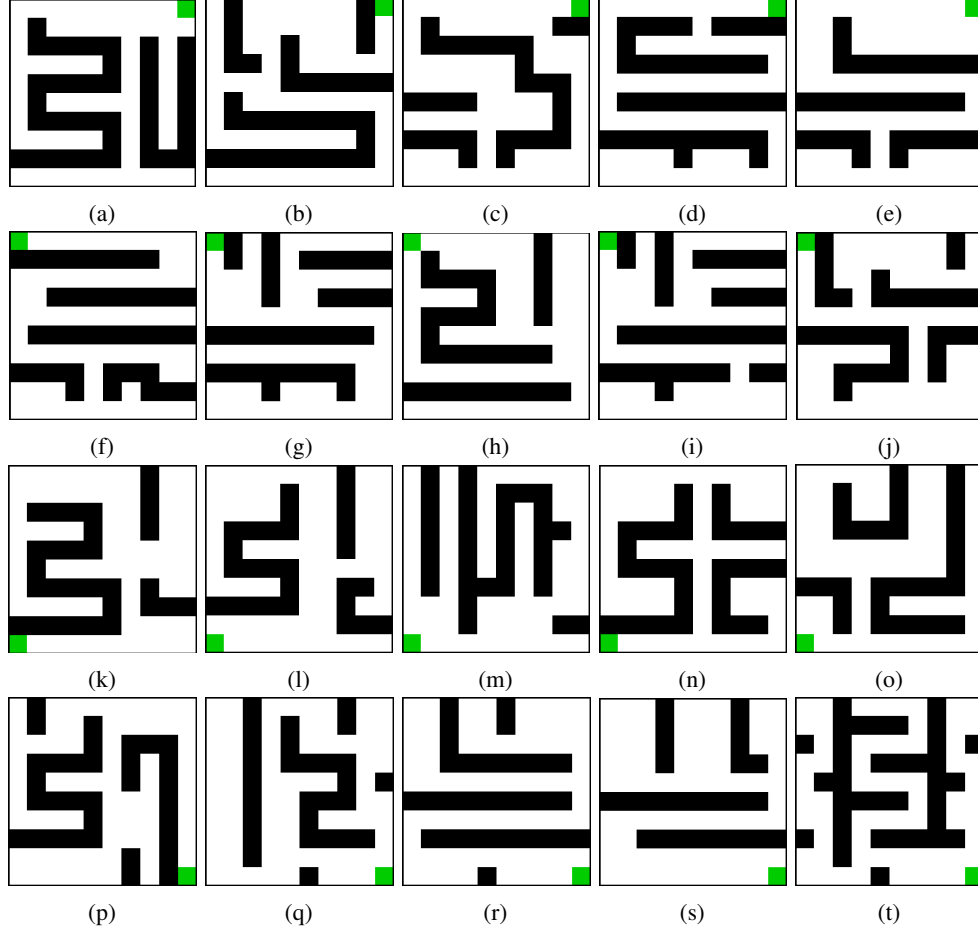


Figure 7: Set of mazes for the Maze Navigation task

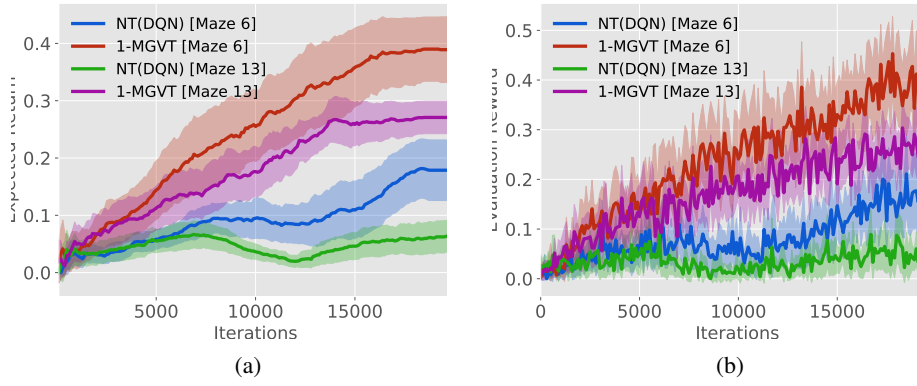


Figure 8: Maze 7a (a) Expected Return (b) Evaluation Reward

784 the algorithm by following a greedy policy w.r.t. those parameters, and the expected return. All the  
785 curves result from averaging 20 independent runs and randomly sampled sources.



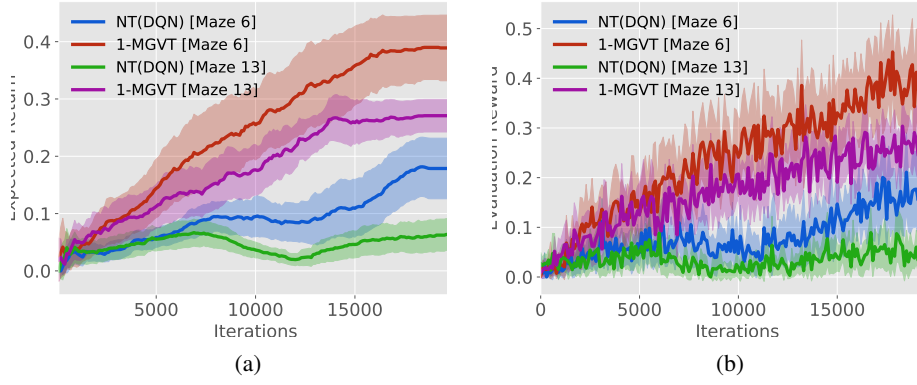


Figure 9: Maze 7g (a) Expected Return (b) Evaluation Reward

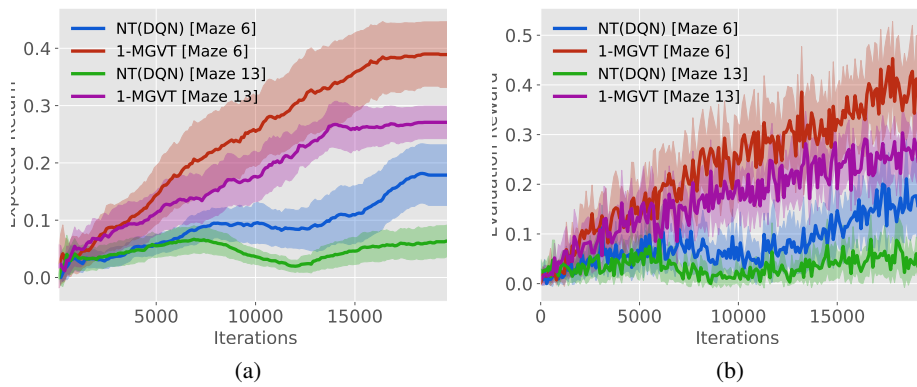


Figure 10: Maze 7n (a) Expected Return (b) Evaluation Reward