Variational Approximations for Transfer in Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

The abstract paragraph should be indented ½ inch (3 picas) on both the left- and right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Introduction

Recent advancements have allowed reinforcement learning (RL) [29] to achieve impressive results in a wide variety of complex tasks, ranging from Atari [22] through the game of Go [28] to the control of sophisticated robotics systems [14, 20, 19]. 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 [31], 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 [18, 30], policies/options [10, 16], rewards [15], features [4], parameters [9, 13], and so on. We refer the reader to [31**?**] 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 [33], 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 [17], the authors assume that tasks are similar in their value functions and design a different hierarchical Bayesian model for transferring such information. More recently, [9], and its extension [13], 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 transfer algorithms that alleviate the need for strong assumptions and that easily adapt to different contexts motivates us to take a more general approach.

probabilistic? distributional?

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

Similarly to [17], 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 [23], 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, $\mathcal{R}_\tau : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ 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 $\pi : \mathcal{S} \to \mathcal{A}$ 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 π^*_τ 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., [24]) that Q^*_τ is the unique fixed-point of the optimal Bellman operator T_τ 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 $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ uniformly bounded by $\frac{R_{max}}{1-\gamma}$. In the following, to avoid cluttering the notation, we will drop the subscript τ when there is no ambiguity.

We consider a parametric family of Q-functions, $\mathcal{Q} = \left\{ Q_{\boldsymbol{w}} : \mathcal{S} \times \mathcal{A} \to \mathbb{R} \mid \boldsymbol{w} \in \mathbb{R}^d \right\}$, 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 $Q_{\boldsymbol{w}}$ is to the fixed-point of the Bellman operator. A possible measure is its Bellman error (or Bellman residual), defined by $B_{\boldsymbol{w}} \triangleq TQ_{\boldsymbol{w}} - Q_{\boldsymbol{w}}$. Notice that $Q_{\boldsymbol{w}}$ is optimal if and only if $B_{\boldsymbol{w}}(s,a) = 0$ for all s, a. If we assume the existence of a distribution ν over $\mathcal{S} \times \mathcal{A}$, a sound objective is to directly minimize the squared Bellman error of $Q_{\boldsymbol{w}}$ under ν, denoted by $\|B_{\boldsymbol{w}}\|^2_\nu$. 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., [21]). In practice, the Bellman error is typically replaced by the TD error $b(\boldsymbol{w})$, which approximates the former using a single transition sample $\langle s, a, s', r \rangle$, $b(\boldsymbol{w}) = r + \gamma \max_{a'} Q_{\boldsymbol{w}}(s', a') - Q_{\boldsymbol{w}}(s,a)$. 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 $\|B_{\boldsymbol{w}}\|^2_D = \frac{1}{N} \sum_{i=1}^N (r_i + \gamma \max_{a'} Q_{\boldsymbol{w}}(s'_i, a') - Q_{\boldsymbol{w}}(s_i, a_i))^2 = \frac{1}{N} \sum_{i=1}^N b_i(\boldsymbol{w})^2$. 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.

3.1 Algorithm

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

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

Assume, for the moment, that we know the distribution $p(\boldsymbol{w})$ and consider a dataset $D = \{(s_i, a_i, s'_i, r_i) \mid i = 1, 2, \dots N\}$ of samples from some task $\mathcal{M}_\tau \sim \mathcal{D}$ that we want to solve. Then, we can compute the posterior distribution over weights given such dataset by applying Bayes theorem as $p(\boldsymbol{w}|D) \propto p(D|\boldsymbol{w})p(\boldsymbol{w})$. Unfortunately, this cannot be directly used in practice since we do not have a model of the likelihood $p(D|\boldsymbol{w})$. In such case, it is very common to make strong assumptions on the MDPs or the Q-functions to get tractable posteriors. However, in our transfer settings, all distributions involved depend on the family of tasks under consideration and making such assumptions is likely to limit the applicability to specific problems. Thus, we take a different approach to derive a more general, but still well-grounded, solution. Recall that our final goal is to move the total probability mass over the weights minimizing some empirical loss measure, which in our case is the TD error $\|B_{\boldsymbol{w}}\|^2_D$. Then, given a prior $p(\boldsymbol{w})$, we know from PAC-Bayesian theory that the optimal Gibbs posterior q takes the form (e.g., [8]):

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

for some parameter $\Lambda > 0$. Since Λ is typically chosen to increase with the number of samples N, in the remaining, we set it to $\lambda^{-1}N$, for some constant $\lambda > 0$. Notice that, whenever the term $e^{-\Lambda \|B_{\boldsymbol{w}}\|^2_D}$ can be interpreted as the actual likelihood of D, q becomes a classic Bayesian posterior. Although we now have an appealing distribution, the integral at the denominator of (1) is intractable to compute even for simple Q-function models. Thus, we propose a variational approximation $q_{\boldsymbol{\xi}}$ by considering a simpler family of distributions parameterized by $\boldsymbol{\xi} \in \Xi$. Then, our problem reduces to finding the variational parameters $\boldsymbol{\xi}$ such that $q_{\boldsymbol{\xi}}$ minimizes the Kullback-Leibler (KL) divergence w.r.t. the Gibbs posterior q. From the theory of variational inference (e.g., [5]), this can be shown to be equivalent to minimizing the well-known (negative) *evidence lower bound* (ELBO):

Maybe say how this is obtained

$$\min_{\boldsymbol{\xi} \in \Xi} \mathcal{L}(\boldsymbol{\xi}) = \mathbb{E}_{\boldsymbol{w} \sim q_{\boldsymbol{\xi}}} \left[\|B(\boldsymbol{w})\|^2_D \right] - \frac{\lambda}{N} KL\left(q_{\boldsymbol{\xi}}(\boldsymbol{w}) \,||\, p(\boldsymbol{w})\right). \tag{2}$$

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

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

The key property of our approach is the weight resampling at line 7, which resembles the well-known Thompson sampling approach adopted in multi-armed bandits [7] and closely relates to the recent value function randomization [23]. At each time we guess what the task we are trying to solve based on our current belief is and we act as if such guess were true. This mechanism allows an efficient adaptive exploration of the target task. Intuitively, during the first steps of interaction, the agent is very uncertain about the current task, and such uncertainty induces stochasticity in the chosen actions, allowing rather informed exploration to take place. Consider, for instance, that actions that are bad on average for all tasks are improbable to be sampled, while this cannot happen in uninformed

[1] If the prior and approximate posterior were in the same family of distributions we could simply set $\boldsymbol{\xi}$ to the prior parameters. However, we are not making this assumption at this point.

Algorithm 1 Variational Transfer

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

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

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

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

3.2 Gaussian Variational Transfer

We now restrict to a specific choice of the prior and posterior families that makes our algorithm very efficient and easy to implement. We assume that optimal Q-functions (or better, their weights) follow a multivariate Gaussian distribution. That is, we model the prior as $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ and we learn its parameters from the set of source weights using maximum likelihood estimation (with small regularization to make sure the covariance is positive definite). Then, our variational family is the set of all well-defined Gaussian distributions, i.e., the variational parameters are $\Xi = \left\{ (\boldsymbol{\mu}, \boldsymbol{\Sigma}) \mid \boldsymbol{\mu} \in \mathbb{R}^K, \boldsymbol{\Sigma} \in \mathbb{R}^{K \times K}, \boldsymbol{\Sigma} \succ 0 \right\}$. To prevent the covariance from becoming not positive definite, we consider its Cholesky decomposition $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^T$ and learn the lower-triangular Cholesky factor $\boldsymbol{L}$ instead. In this case, deriving the gradient of the objective is very simple. Both the KL between two multivariate Gaussians and its gradients have a simple closed-form expression. The expected log-likelihood, on the other hand, can be easily differentiated by adopting the reparameterization trick (e.g., [12, 25]). Although these results are well-known in the literature, we report them in App. **??** to have a more self-contained description.

3.3 Mixture of Gaussian Variational Transfer

Although the Gaussian assumption of the previous section is very appealing as it allows for a simple and efficient way of computing the variational objective and its gradients, we believe that such assumption almost never holds in practice. In fact, even for families of tasks in which the reward and transition models are Gaussian, the Q-values might be far from it. Depending on the family of tasks under consideration and, since we are learning a distribution over weights, on the chosen function approximator, the prior might have arbitrarily complex shapes. When the information loss due to the Gaussian approximation becomes too severe, the algorithm is likely too fail at capturing any similarities between the tasks. We now propose a variant to successfully solve this problem, while keeping the algorithm simple and efficient 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: $p(\boldsymbol{w}) = \frac{1}{|\mathcal{W}_s|} \sum_{\boldsymbol{w}_s \in \mathcal{W}_s} \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_s, \sigma_p^2 \boldsymbol{I})$. This resembles a kernel density estimator [27] and, due to its nonparametric nature, it allows capturing arbitrarily complex distribution. 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 less components. Using C components, our posterior is $q_{\boldsymbol{\xi}}(\boldsymbol{w}) = \frac{1}{C} \sum_{i=1}^{C} \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, 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 complex distributions, it poses a major complication: the KL divergence between two mixture 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 true posterior. Among the many upper bounds available in the literature, we adopt the one proposed in [11], which we report here for the sake of completeness. We refer the reader to the original paper for the proof.

Theorem 1 ([11]). *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:*

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

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 [11], this can be achieved by a simple fixed-point procedure. Finally, both terms in the objective are now linear combinations of functions of the variational parameters of different components, and their gradients are easily derived 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 $\|B(\boldsymbol{w})\|_D^2$ w.r.t. $\boldsymbol{w}$. Unfortunately, the TD error is well-known to be non-differentiable due to the presence of the max operator. This rarely represents a problem since typical value-based algorithms are actually semi-gradient methods, i.e., they do not differentiate the targets (see, e.g., Chapter 11 of [29]). However, our transfer settings are rather different than common RL. In fact, our algorithm is likely to start from Q-functions that are very close to an optimum, and the only thing that needs to be done is to adapt the weights in a direction of lower error (i.e., higher likelihood) so as to quickly converge to the solution of the task that is being solved. Unfortunately, this property cannot be guaranteed 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 [26, 32]. This is obviously an undesirable property in our case, as we only aim at adapting already good solutions. Thus, we consider using a residual gradient algorithm [3]. In order to differentiate the targets, we replace the optimal Bellman operator with the mellow Bellman operator introduced in [1], which adopts a softened version of max called *mellowmax*:

$$\operatorname*{mm}_a Q_{\boldsymbol{w}}(s,a) = \frac{1}{\kappa} \log \frac{1}{|\mathcal{A}|} \sum_a e^{\kappa Q_{\boldsymbol{w}}(s,a)} \tag{4}$$

where κ is a hyperparameter and $|\mathcal{A}|$ is the number of actions. The mellow Bellman operator, which we denote as $\widetilde{T}$, has several appealing properties that make it suitable for our settings: (i) it converges to the maximum as $\kappa \to \infty$, (ii) it has a unique fixed point, and (iii) it is *differentiable*. Denoting by $\widetilde{B}(\boldsymbol{w}) = \widetilde{T} Q_{\boldsymbol{w}} - Q_{\boldsymbol{w}}$ the Bellman residual w.r.t. the mellow Bellman operator $\widetilde{T}$, we have that the corresponding TD error, $\left\|\widetilde{B}(\boldsymbol{w})\right\|_D^2$, is now differentiable with respect to $\boldsymbol{w}$. Although residual algorithms have guaranteed convergence, they are typically much slower than their semi-gradient counterpart. [3] proposed to project the gradient in a direction that achieves higher learning speed,

while preserving convergence. This can be easily done by including a parameter $\psi \in [0, 1]$ in the TD error gradient such that:

$$\nabla_{\boldsymbol{w}} \left\| \widetilde{B}(\boldsymbol{w}) \right\|_D^2 = \frac{2}{N} \sum_{i=1}^{N} b_i(\boldsymbol{w}) \left(\gamma\psi \nabla_{\boldsymbol{w}} \operatorname*{mm}_{a'} Q_{\boldsymbol{w}}(s_i', a') - \nabla_{\boldsymbol{w}} Q_{\boldsymbol{w}}(s_i, a_i) \right) \right) \tag{5}$$

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

4 Theoretical Analysis

A first important question that we need to answer is whether replacing max with mellow-max in the Bellman operator constitutes a strong approximation or not. It has been proved [1] that the mellow Bellman operator is a non-expansion under the L_∞-norm and, thus, has a unique fixed-point. However, how such fixed-point differs from the one of the optimal Bellman operator remains an open question. Since mellow-max monotonically converges to max as $\kappa \to \infty$, it would be desirable if the corresponding operator also monotonically converged to the optimal one. We confirm that this property actually holds in the following theorem.

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

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

Correct measure on the rhs

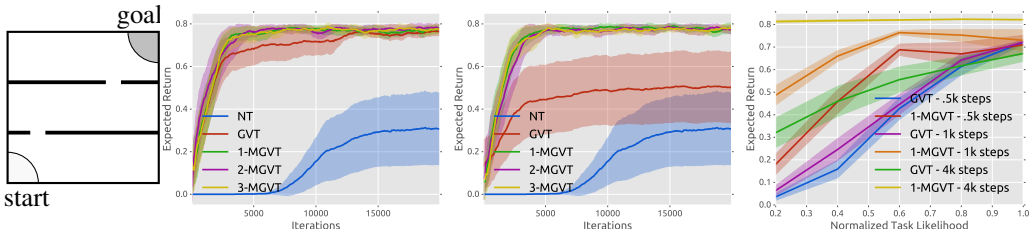
The proof is provided in App. **??**. As expected, $\widetilde{Q}$ converges to Q^* exponentially fast when either κ (equivalently, β_κ) increases, or the action gaps are enlarged. Notice that this result is of interest even outside our specific settings.

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

Theorem 3. *Fix a target task τ a let $\widetilde{Q}$ be the fixed-point of the corresponding mellow Bellman operator. Assume linearly parameterized value functions $Q_{\boldsymbol{w}}(s, a) = \boldsymbol{w}^T \boldsymbol{\phi}(s, a)$ with bounded weights $\boldsymbol{w} \leq w_{max}$ and uniformly bounded features $\boldsymbol{\phi}(s, a) \leq \phi_{\max}$. Consider the mixture version of Alg. 1 using C components, source task weights $\mathcal{W}_s$, and bandwidth σ_p^2 for the prior. Denote by $\widehat{\boldsymbol{\xi}} = (\widehat{\boldsymbol{\mu}}_1, \dots, \widehat{\boldsymbol{\mu}}_C, \widehat{\boldsymbol{\Sigma}}_1, \dots, \widehat{\boldsymbol{\Sigma}}_C)$ the variational parameters minimizing the objective of Eq. 2 on a dataset D of N samples. Let ν be a probability measure over $\mathcal{S} \times \mathcal{A}$ and $\boldsymbol{w}^* = \operatorname{arginf}_{\boldsymbol{w}} \left\| \widetilde{B}(\boldsymbol{w}) \right\|_\nu^2$. Define $v(\boldsymbol{w}^*) \triangleq \mathbb{E}_{\mathcal{N}(\boldsymbol{w}^*, \boldsymbol{I})} [v(\boldsymbol{w})]$, with $v(\boldsymbol{w}) \triangleq \mathbb{E}_\nu [Var_{\mathcal{P}} [b(\boldsymbol{w})]]$. 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 weights $\boldsymbol{w} \sim \frac{1}{C} \sum_i \mathcal{N}(\widehat{\boldsymbol{\mu}}_i, \widehat{\boldsymbol{\Sigma}}_i)$ and dataset D:*

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

We refer the reader to App. **??** for the proof and a specific definition of the constants. Four main terms constitute our bound: the approximation error due to the limited hypothesis space (first term), the variance (second term), the distance to the prior (third term), and a constant term decaying as



(a) (b) (c) (d)

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.

$\mathcal{O}(N^2)$. Intuitively, the bound is tighten when the source tasks' Q-functions are, on average, close to the optimal ones for the target task. In such case, the dominating error is due to the variance of the estimates, and, thus, the algorithm is expected to achieve good performance rather quickly, as new data is collected. Furthermore, as $N \to \infty$ the only error term remaining is, as usual, the irreducible approximation error due to the limited functional space. Notice that the variance term $v(\boldsymbol{w}^*)$ 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. **??**.

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

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 ϵ-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.

Any better motivation?

5.1 The Rooms Problem

We consider an agent navigating in the rooms environment depicted in Fig. **??**. 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 positions are unknown to the agent. The state-space is modeled as a 10×10 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 with 0.2 standard deviation. 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 where the doors are located 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 quickly figure out the door positions. 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. This does not hold for positions nearby the start 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).

Good candidate to be removed

In order to prove that our guesses are correct, we generate a set of 50 source tasks for the three-room environment of Fig. **??** by sampling both door positions uniformly in the allowed space, and solve all of them by directly minimizing the TD error as presented in Sec. 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 Fig. **??**. Each curve is the result of 20 independent runs, each 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 ϵ-greedy policy. On the other hand, all our algorithms achieve a significant speed-up and are able to 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 is intuitive since, when the algorithm quickly figures out which task is being solved, it will move all components in the same direction.

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 again 50 source tasks 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 Fig. **??**. Interestingly, MGVT seems almost unaffected by this change, proving that is 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 proves again that assuming Gaussian distributions can pose severe limitations in our transfer settings.

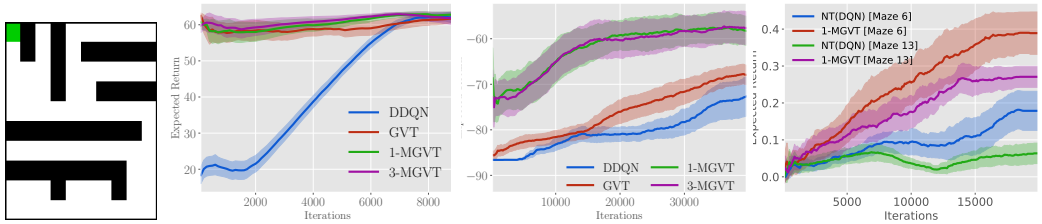
Finally, we analyze the transfer performance as a function of how likely the target task is according to the prior. We consider a two-room version of the environment of Fig. **??**. Differently from before, we generate tasks by sampling the door position from a Gaussian with mean 5 and standard deviation 1.8, so that tasks where the door is near the sides are very unlikely. Fig. **??** 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 [29]. 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 [32] to provide a stronger no-transfer baseline for comparison. The results (same settings of Sec. 5.1) are shown in Fig. **??** and **??**. For Cartpole (Fig. **??**), all transfer algorithms are almost zero-shot. This 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 (Fig. **??**) the optimal Q-functions become really different when changing the car speed. This causes GVT to struggle to solve the target task, while 1-MGVT achieves a good jump-start and convergences in less 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 10×10 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 $\pm 22.5^{\text{o}}$ (*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 $[-90, 90^{\text{o}}]$ 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



(a) (b) (c) (d)

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. **??**.

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) [] with an MLP with hidden layers 32×32 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 [17]. Although we both assume the tasks to share similarities in their value functions, [17] 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, [17] 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 [33], the authors propose a hierarchical Bayesian model for the distribution over tasks. Differently from our approach and [17], 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, [33] 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 [9], 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, [13] 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[23], which extends the well-known LSTD [6] 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 [2], 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

References

[1] Kavosh Asadi and Michael L Littman. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, pages 243–252, 2017.

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

[3] Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.

[4] 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.

[5] 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.

[6] Justin A Boyan. Least-squares temporal difference learning.

[7] 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.

[8] Olivier Catoni. Pac-bayesian supervised classification: the thermodynamics of statistical learning. *arXiv preprint arXiv:0712.0248*, 2007.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Jens Kober and Jan R Peters. Policy search for motor primitives in robotics. In *Advances in neural information processing systems*, pages 849–856, 2009.

[15] 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.

[16] George Konidaris and Andrew G Barto. Building portable options: Skill transfer in reinforcement learning.

[17] Alessandro Lazaric and Mohammad Ghavamzadeh. Bayesian multi-task reinforcement learning. In *ICML-27th International Conference on Machine Learning*, pages 599–606. Omnipress, 2010.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. *arXiv preprint arXiv:1402.0635*, 2014.

[24] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1994.

[25] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[26] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[27] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[28] 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.

[29] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[30] 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.

[31] Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685, 2009.

[32] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. 2016.

[33] Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. Multi-task reinforcement learning: a hierarchical bayesian approach. In *Proceedings of the 24th international conference on Machine learning*, pages 1015–1022. ACM, 2007.

A Proofs

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

Correct measure on the rhs

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

Proof. We begin by noticing that:

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

where the first inequality follows from Minkowsky's inequality and the second one from the contraction property of the mellow Bellman operator. This implies that:

$$\left\|Q^* - \widetilde{Q}\right\|_{\nu,p}^p \leq \frac{1}{1-\gamma} \left\|TQ^* - \widetilde{T}Q^*\right\|_{\nu,p}^p \tag{8}$$

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

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

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

Cite MM

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

where π^* is the optimal (deterministic) policy w.r.t. Q^* and π_{β_κ} is the Boltzmann distribution induced by Q^* with inverse temperature β_κ:

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

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

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

Finally, let us bound this last term:

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

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

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

Finally, using Eq. (9) we get:

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

Taking the norm and plugging this into Eq. (8) concludes the proof. □

Lemma 1. *Let p and ν denote probability measures over Q-functions and state-action pairs, respectively. Assume Q^* is the unique fixed-point of the optimal Bellman operator T. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a Q-function Q, the following holds:*

$$\|Q - Q^*\|_\nu^2 \leq \frac{\mathbb{E}_p\left[\|B(Q)\|_\nu^2\right]}{(1-\gamma)\delta} \tag{13}$$

Proof. First notice that:

$$\begin{aligned}\|Q - Q^*\| &= \|Q + TQ - TQ - TQ^*\| \\ &\leq \|Q - TQ\| + \|TQ - TQ^*\| \\ &\leq \|Q - TQ\| + \gamma \|Q - Q^*\| \\ &= \|B(Q)\| + \gamma \|Q - Q^*\| \end{aligned}$$

which implies that:

$$\|Q - Q^*\| \leq \frac{1}{1-\gamma} \|B(Q)\|$$

Then we can write:

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

Settings the right-hand side equal to δ and solving for ϵ concludes the proof. □

Corollary 1. *Let p and ν denote probability measures over Q-functions and state-action pairs, respectively. Assume $\widetilde{Q}$ is the unique fixed-point of the mellow Bellman operator $\widetilde{T}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a Q-function Q, the following holds:*

$$\left\| Q - \widetilde{Q} \right\|_\nu^2 \leq \frac{\mathbb{E}_p \left[\left\| \widetilde{B}(Q) \right\|_\nu^2 \right]}{(1-\gamma)\delta} \tag{14}$$

Lemma 2. *Assume Q-functions belong to a parametric space of functions bounded by $\frac{R_{max}}{1-\gamma}$. Let p and q be arbitrary distributions over the parameter space $\mathcal{W}$, and ν be a probability measure over $\mathcal{S} \times \mathcal{A}$. Consider a dataset D of N samples and define $v(\boldsymbol{w}) \triangleq \mathbb{E}_\nu \left[Var_{\mathcal{P}} \left[b(\boldsymbol{w}) \right] \right]$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following two inequalities hold simultaneously:*

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

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

Proof. From Hoeffding's inequality we have:

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

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

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

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

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

where $v(\boldsymbol{w}) \triangleq \mathbb{E}_\nu \left[Var_{\mathcal{P}} \left[b(\boldsymbol{w}) \right] \right]$. Thus:

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

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

Find a reference for this

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

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

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

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

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

from which the lemma follows straightforwardly. □

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

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

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

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

where the second inequality holds since $v(\boldsymbol{w}) > 0$, while the equality holds from the definition of $\widehat{\xi}$. We can now use Eq. (16) to bound $\mathbb{E}_{q_\xi}\left[\|B(\boldsymbol{w})\|_D^2\right]$, thus obtaining:

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

This concludes the proof. □

Theorem 4. *Fix a target task τ a let $\widetilde{Q}$ be the fixed-point of the corresponding mellow Bellman operator. Assume linearly parameterized value functions $Q_{\boldsymbol{w}}(s,a) = \boldsymbol{w}^T\boldsymbol{\phi}(s,a)$ with bounded weights $\boldsymbol{w} \leq w_{max}$ and uniformly bounded features $\boldsymbol{\phi}(s,a) \leq \phi_{\max}$. Consider the Gaussian version of Alg. 1 with prior $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ and denote by $(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}})$ the variational parameter minimizing the objective of Eq. 2 on a dataset D of N samples. Let ν be a probability measure over $\mathcal{S}\times\mathcal{A}$ and $\boldsymbol{w}^* = \operatorname{arginf}_{\boldsymbol{w}}\left\|\widetilde{B}(\boldsymbol{w})\right\|_\nu^2$. Define $v(\boldsymbol{w}^*) \triangleq \mathbb{E}_{\mathcal{N}(\boldsymbol{w}^*,\boldsymbol{I})}\left[v(\boldsymbol{w})\right]$, with $v(\boldsymbol{w}) \triangleq \mathbb{E}_\nu\left[Var_{\mathcal{P}}\left[b(\boldsymbol{w})\right]\right]$. Then, there exist constants c_1, c_2, c_3 such that, with probability at least $1-2\delta$ over the choice of weights $\boldsymbol{w} \sim \mathcal{N}(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}})$ and dataset D:*

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

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

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

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 Ξ. Here, we choose $\boldsymbol{\mu} = \boldsymbol{w}^*$ and $\boldsymbol{\Sigma} = c\boldsymbol{I}$, for some positive constant $c > 0$. Let us now bound these terms separately.

Bounding the expected TD error We have:

$$\begin{aligned}
\mathbb{E}_{\mathcal{N}(\boldsymbol{w}^*, c\boldsymbol{I})}\left[\left\|\widetilde{B}(\boldsymbol{w}^*)\right\|_\nu^2\right] &= \mathbb{E}_{\mathcal{N}(\boldsymbol{w}^*, c\boldsymbol{I})}\left[\mathbb{E}_\nu\left[(\widetilde{T}Q_{\boldsymbol{w}} - Q_{\boldsymbol{w}})^2\right]\right] \\
&= \mathbb{E}_\nu\left[\mathbb{E}_{\mathcal{N}(\boldsymbol{w}^*, c\boldsymbol{I})}\left[(\widetilde{T}Q_{\boldsymbol{w}} - Q_{\boldsymbol{w}})^2\right]\right] \\
&= \mathbb{E}_\nu\left[\mathbb{E}^2_{\mathcal{N}(\boldsymbol{w}^*, c\boldsymbol{I})}\left[\widetilde{T}Q_{\boldsymbol{w}} - Q_{\boldsymbol{w}}\right]\right] + \mathbb{E}_\nu\left[Var_{\mathcal{N}(\boldsymbol{w}^*, c\boldsymbol{I})}\left[\widetilde{T}Q_{\boldsymbol{w}} - Q_{\boldsymbol{w}}\right]\right]
\end{aligned} \tag{20}$$

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

$$\begin{aligned}
\mathbb{E}_{\mathcal{N}(\boldsymbol{w}^*, c\boldsymbol{I})}\left[\widetilde{T}Q_{\boldsymbol{w}} - Q_{\boldsymbol{w}}\right] &= \mathbb{E}_{\mathcal{N}(\boldsymbol{w}^*, c\boldsymbol{I})}\left[R(s,a) + \gamma\mathbb{E}_{s'}\min_{a'}\boldsymbol{w}^T\boldsymbol{\phi}(s',a') - \boldsymbol{w}^T\boldsymbol{\phi}(s,a)\right] \\
&= R(s,a) + \gamma\mathbb{E}_{\mathcal{N}(\boldsymbol{w}^*, c\boldsymbol{I})}\left[\mathbb{E}_{s'}\min_{a'}\boldsymbol{w}^T\boldsymbol{\phi}(s',a')\right] - \boldsymbol{w}^{*T}\boldsymbol{\phi}(s,a)
\end{aligned} \tag{21}$$

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

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

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

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

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

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

This implies:

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

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

$$\mathbb{E}_\nu\left[\mathbb{E}^2_{\mathcal{N}(\boldsymbol{w}^*, c\boldsymbol{I})}\left[\widetilde{T}Q_{\boldsymbol{w}} - Q_{\boldsymbol{w}}\right]\right] \leq 2\left\|\widetilde{B}(\boldsymbol{w}^*)\right\|_\nu^2 + \frac{1}{2}\gamma^2\kappa^2 c^2\boldsymbol{\phi}^2_{max}$$

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

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

From Cauchy-Schwarz inequality:

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

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

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

We can finally plug everything into 20, thus obtaining:

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{w}^*, c\boldsymbol{I})}\left[\left\|\widetilde{B}(\boldsymbol{w}^*)\right\|_\nu^2\right] \leq 2\left\|\widetilde{B}(\boldsymbol{w}^*)\right\|_\nu^2 + \frac{1}{2}\gamma^2\kappa^2 c^2 \boldsymbol{\phi}_{max}^2 + c\left(\boldsymbol{w}_{max}\boldsymbol{\phi}_{max}(1+\gamma)\right)^2$$

Bounding the KL divergence We have:

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

Now, putting all together into 19:

$$\begin{aligned}
\mathbb{E}_{q_{\hat{\boldsymbol{\xi}}}}\left[\left\|B(\boldsymbol{w})\right\|_\nu^2\right] &\leq 2\left\|\widetilde{B}(\boldsymbol{w}^*)\right\|_\nu^2 + \frac{1}{2}\gamma^2\kappa^2 c^2\boldsymbol{\phi}_{max}^2 + c\left(\boldsymbol{w}_{max}\boldsymbol{\phi}_{max}(1+\gamma)\right)^2 + \upsilon(\boldsymbol{w}^*)\sqrt{c} \\
&\quad + \frac{\lambda}{N}K\log\frac{\sigma_{max}}{c} + \frac{\lambda}{N}K\frac{c}{\sigma_{min}} + \frac{\lambda}{N}\left\|\boldsymbol{w}^* - \boldsymbol{\mu}_p\right\|_{\boldsymbol{\Sigma}_p^{-1}} + 2\frac{R_{max}^2}{(1-\gamma)^2}\sqrt{\frac{\log\frac{2}{\delta}}{N}}
\end{aligned}$$

where $\upsilon(\boldsymbol{w}^*) = \mathbb{E}_{\mathcal{N}(\boldsymbol{w}^*, \boldsymbol{I})}\left[v(\boldsymbol{w})\right]$. Since the bound holds for any $c > 0$, we can set it to $1/N$, thus obtaining:

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

Finally, defining the constants $c_1 = \frac{1}{2}\gamma^2\kappa^2\boldsymbol{\phi}_{max}^2 + \frac{\lambda K}{\sigma_{min}}$, $c_2 = \boldsymbol{w}_{max}^2\boldsymbol{\phi}_{max}^2(1+\gamma)^2 + \lambda K\log(\sigma_{max} + \log N)$, and $c_3 = 2\frac{R_{max}^2}{(1-\gamma)^2}$, we obtain:

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

c_2 actually contains N. Should we make this more explicit?

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

$$\left\|Q_{\boldsymbol{w}}-\widetilde{Q}\right\|_{\nu}^{2} \leq \frac{\mathbb{E}_{q_{\widehat{\boldsymbol{\xi}}}}\left[\|B(\boldsymbol{w})\|_{\nu}^{2}\right]}{(1-\gamma)\delta}$$

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

$$\left\|Q_{\boldsymbol{w}}-\widetilde{Q}\right\|_{\nu}^{2} \leq \frac{1}{(1-\gamma)\delta}\left(2\left\|\widetilde{B}(\boldsymbol{w}^{*})\right\|_{\nu}^{2}+\frac{c_{1}}{N^{2}}+\frac{c_{2}+\lambda\left\|\boldsymbol{w}^{*}-\boldsymbol{\mu}_{p}\right\|_{\boldsymbol{\Sigma}_{p}^{-1}}}{N}+\frac{\upsilon(\boldsymbol{w}^{*})+c_{3}\sqrt{\log\frac{2}{\delta}}}{\sqrt{N}}\right)$$

□

Theorem 3. *Fix a target task τ a let $\widetilde{Q}$ be the fixed-point of the corresponding mellow Bellman operator. Assume linearly parameterized value functions $Q_{\boldsymbol{w}}(s,a)=\boldsymbol{w}^{T}\boldsymbol{\phi}(s,a)$ with bounded weights $\boldsymbol{w}\leq w_{max}$ and uniformly bounded features $\boldsymbol{\phi}(s,a)\leq\phi_{\max}$. Consider the mixture version of Alg. 1 using C components, source task weights $\mathcal{W}_s$, and bandwidth σ_p^2 for the prior. Denote by $\widehat{\boldsymbol{\xi}}=(\widehat{\boldsymbol{\mu}}_1,\ldots,\widehat{\boldsymbol{\mu}}_C,\widehat{\boldsymbol{\Sigma}}_1,\ldots,\widehat{\boldsymbol{\Sigma}}_C)$ the variational parameters minimizing the objective of Eq. 2 on a dataset D of N samples. Let ν be a probability measure over $\mathcal{S}\times\mathcal{A}$ and $\boldsymbol{w}^{*}=\operatorname{arginf}_{\boldsymbol{w}}\left\|\widetilde{B}(\boldsymbol{w})\right\|_{\nu}^{2}$. Define $\upsilon(\boldsymbol{w}^{*})\triangleq\mathbb{E}_{\mathcal{N}(\boldsymbol{w}^{*},\boldsymbol{I})}\left[v(\boldsymbol{w})\right]$, with $v(\boldsymbol{w})\triangleq\mathbb{E}_{\nu}\left[Var_{\mathcal{P}}\left[b(\boldsymbol{w})\right]\right]$. 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 weights $\boldsymbol{w}\sim\frac{1}{C}\sum_i\mathcal{N}(\widehat{\boldsymbol{\mu}}_i,\widehat{\boldsymbol{\Sigma}}_i)$ and dataset D:*

$$\left\|Q_{\boldsymbol{w}}-\widetilde{Q}\right\|_{\nu}^{2} \leq \frac{1}{(1-\gamma)\delta}\left(2\left\|\widetilde{B}(\boldsymbol{w}^{*})\right\|_{\nu}^{2}+\frac{c_{1}}{N^{2}}+\frac{c_{2}+\frac{c_{4}}{|\mathcal{W}_s|}\sum_{j}\|\boldsymbol{w}^{*}-\boldsymbol{w}_{j}\|}{N}+\frac{\upsilon(\boldsymbol{w}^{*})+c_{3}\sqrt{\log\frac{2}{\delta}}}{\sqrt{N}}\right) \quad (7)$$

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

$$\begin{aligned}\mathbb{E}_{q_{\widehat{\boldsymbol{\xi}}}}\left[\|B(\boldsymbol{w})\|_{\nu}^{2}\right] &\leq \inf_{\boldsymbol{\xi}\in\Xi}\left\{\mathbb{E}_{q_{\boldsymbol{\xi}}}\left[\|B(\boldsymbol{w})\|_{\nu}^{2}\right]+\mathbb{E}_{q_{\boldsymbol{\xi}}}\left[v(\boldsymbol{w})\right]+2\frac{\lambda}{N}KL(q_{\boldsymbol{\xi}}||p)\right\}+2\frac{R_{max}^{2}}{(1-\gamma)^{2}}\sqrt{\frac{\log\frac{2}{\delta}}{N}}\\ &\leq \mathbb{E}_{\mathcal{N}(\boldsymbol{w}^{*},c\boldsymbol{I})}\left[\|B(\boldsymbol{w})\|_{\nu}^{2}\right]+\mathbb{E}_{\mathcal{N}(\boldsymbol{w}^{*},c\boldsymbol{I})}\left[v(\boldsymbol{w})\right]+2\frac{\lambda}{N}KL\left(\mathcal{N}(\boldsymbol{w}^{*},c\boldsymbol{I})\,||\,p\right)\\ &\quad+2\frac{R_{max}^{2}}{(1-\gamma)^{2}}\sqrt{\frac{\log\frac{2}{\delta}}{N}} \end{aligned} \quad (23)$$

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

$$KL(\mathcal{N}(\boldsymbol{w}^{*},c\boldsymbol{I})\,||\,p)\leq KL(\chi^{(2)}||\chi^{(1)})+\sum_{j}\chi_{j}^{(2)}KL(\mathcal{N}(\boldsymbol{w}^{*},c\boldsymbol{I})\,||\,\mathcal{N}(\boldsymbol{w}_{j},\sigma_{p}^{2}\boldsymbol{I})) \quad (24)$$

where the vectors $\chi^{(1)}$ and $\chi^{(2)}$ are the ones defined in Thm. 1. Notice that, since we reduced the posterior to one component, the can get rid of the index i. Using the definitions of these two vectors from Sec. 8 of [11], we have:

$$\chi_{j}^{(1)}=\frac{1}{|\mathcal{W}_s|}\;\forall j=1,\ldots,|\mathcal{W}_s|$$

$$\chi_{j}^{(2)}=\frac{e^{-KL(\mathcal{N}(\boldsymbol{w}^{*},c\boldsymbol{I})\,||\,\mathcal{N}(\boldsymbol{w}_{j},\sigma_{p}^{2}\boldsymbol{I}))}}{\sum_{j'}e^{-KL(\mathcal{N}(\boldsymbol{w}^{*},c\boldsymbol{I})\,||\,\mathcal{N}(\boldsymbol{w}_{j'},\sigma_{p}^{2}\boldsymbol{I}))}}\;\forall j=1,\ldots,|\mathcal{W}_s| \quad (25)$$

Since the KL divergence is:

$$KL(\mathcal{N}(\boldsymbol{w}^{*},c\boldsymbol{I})\,||\,\mathcal{N}(\boldsymbol{w}_{j},\sigma_{p}^{2}\boldsymbol{I}))=\frac{1}{2}\left(d\log\frac{\sigma_{p}^{2}}{c}+d\frac{c}{\sigma_{p}^{2}}+\frac{1}{\sigma_{p}^{2}}\left\|\boldsymbol{w}^{*}-\boldsymbol{w}_{j}\right\|-d\right)$$

Eq. 25 can be rewritten as:

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

Let us bound the two terms of (24) separately. For the first one, we have:

$$\begin{aligned}
KL(\chi^{(2)}||\chi^{(1)}) &= \sum_j \chi_j^{(2)} \log \frac{\chi_j^{(2)}}{\chi_j^{(1)}} \\
&= \sum_j \chi_j^{(2)} \log \chi_j^{(2)} - \sum_j \chi_j^{(2)} \log \frac{1}{|\mathcal{W}_s|} \\
&= \sum_j \chi_j^{(2)} \log \frac{e^{-\frac{1}{\sigma_p^2}\|\boldsymbol{w}^* - \boldsymbol{w}_j\|}}{\sum_{j'} e^{-\frac{1}{\sigma_p^2}\|\boldsymbol{w}^* - \boldsymbol{w}_{j'}\|}} - \log \frac{1}{|\mathcal{W}_s|} \\
&= -\sum_j \chi_j^{(2)} \frac{1}{\sigma_p^2} \|\boldsymbol{w}^* - \boldsymbol{w}_j\| - \sum_j \chi_j^{(2)} \log \sum_{j'} e^{-\frac{1}{\sigma_p^2}\|\boldsymbol{w}^* - \boldsymbol{w}_{j'}\|} - \log \frac{1}{|\mathcal{W}_s|} \\
&= -\sum_j \chi_j^{(2)} \frac{1}{\sigma_p^2} \|\boldsymbol{w}^* - \boldsymbol{w}_j\| - \log \frac{1}{|\mathcal{W}_s|} \sum_{j'} e^{-\frac{1}{\sigma_p^2}\|\boldsymbol{w}^* - \boldsymbol{w}_{j'}\|} \\
&\leq -\sum_j \chi_j^{(2)} \frac{1}{\sigma_p^2} \|\boldsymbol{w}^* - \boldsymbol{w}_j\| + \frac{1}{|\mathcal{W}_s|} \sum_{j'} \frac{1}{\sigma_p^2} \|\boldsymbol{w}^* - \boldsymbol{w}_{j'}\|
\end{aligned}$$

where we applied Jensen's inequality in the last step ($-\log$ is a convex function). The second term of (24) is:

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

Putting it all together:

$$\begin{aligned}
KL(\mathcal{N}(\boldsymbol{w}^*, c\boldsymbol{I}) \,||\, p) &\leq \frac{1}{|\mathcal{W}_s|} \sum_j \frac{1}{\sigma_p^2} \|\boldsymbol{w}^* - \boldsymbol{w}_j\| - \frac{1}{2} \sum_j \chi_j^{(2)} \frac{1}{\sigma_p^2} \|\boldsymbol{w}^* - \boldsymbol{w}_j\| + \frac{1}{2} d \log \frac{\sigma_p^2}{c} + \frac{1}{2} d \frac{c}{\sigma_p^2} \\
&\leq \frac{1}{\sigma_p^2 |\mathcal{W}_s|} \sum_j \|\boldsymbol{w}^* - \boldsymbol{w}_j\| + \frac{1}{2} d \log \frac{\sigma_p^2}{c} + \frac{1}{2} d \frac{c}{\sigma_p^2}
\end{aligned}$$

Notice that, from now on, one can simply apply the proof of Thm. 4 with $\sigma_{max} = \sigma_{min} = \sigma_p^2$ and $\frac{1}{2} \|\boldsymbol{w}^* - \boldsymbol{\mu}_p\|_{\boldsymbol{\Sigma}_p^{-1}} = \frac{1}{\sigma_p^2 |\mathcal{W}_s|} \sum_j \|\boldsymbol{w}^* - \boldsymbol{w}_j\|$. Thus, by changing the three constants to $c_1 = \frac{1}{2}\gamma^2\kappa^2\boldsymbol{\phi}_{max}^2 + \frac{\lambda d}{\sigma_p^2}$, $c_2 = \boldsymbol{w}_{max}^2 \boldsymbol{\phi}_{max}^2 (1+\gamma)^2 + \lambda d \log(\sigma_p^2 + \log N)$, $c_3 = 2\frac{R_{max}^2}{(1-\gamma)^2}$, and setting $c_4 = \frac{2\lambda}{\sigma_p^2}$, we can write that, with probability at least $1 - 2\delta$:

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

c_2 actually contains N. Should we make this more explicit?

□

B Additional Details on the Algorithms

B.1 Gaussian Variational Transfer

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

$$KL\left(q_{\boldsymbol{\xi}}(\boldsymbol{w}) \,||\, p(\boldsymbol{w})\right) = \frac{1}{2}\left(\log\frac{|\boldsymbol{\Sigma}_p|}{|\boldsymbol{\Sigma}|} + \text{Tr}\left(\boldsymbol{\Sigma}_p^{-1}\boldsymbol{\Sigma}\right) + (\boldsymbol{\mu}-\boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_p) - K\right) \tag{26}$$

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

$$\nabla_{\boldsymbol{\mu}} KL\left(q_{\boldsymbol{\xi}}(\boldsymbol{w}) \,||\, p(\boldsymbol{w})\right) = \boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_p) \tag{27}$$

$$\nabla_{\boldsymbol{L}} KL\left(q_{\boldsymbol{\xi}}(\boldsymbol{w}) \,||\, p(\boldsymbol{w})\right) = \boldsymbol{\Sigma}_p^{-1}\boldsymbol{L} - (\boldsymbol{L}^{-1})^T \tag{28}$$

Finally, the gradients w.r.t. the expected likelihood term of the variational objective (2) can be computed using the reparameterization trick (e.g., [12, 25]):

$$\nabla_{\boldsymbol{\mu}}\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{L}\boldsymbol{L}^T)}\left[||B(\boldsymbol{w})||_D^2\right] = \mathbb{E}_{\boldsymbol{v}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\nabla_{\boldsymbol{w}}||B(\boldsymbol{w})||_D^2\right] \;\text{ for }\; \boldsymbol{w} = \boldsymbol{L}\boldsymbol{v}+\boldsymbol{\mu} \tag{29}$$

$$\nabla_{\boldsymbol{L}}\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{L}\boldsymbol{L}^T)}\left[||B(\boldsymbol{w})||_D^2\right] = \mathbb{E}_{\boldsymbol{v}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})}\left[\nabla_{\boldsymbol{w}}||B(\boldsymbol{w})||_D^2 \cdot \boldsymbol{v}^T\right] \;\text{ for }\; \boldsymbol{w} = \boldsymbol{L}\boldsymbol{v}+\boldsymbol{\mu} \tag{30}$$

B.2 Mixture of Gaussian Variational Transfer

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

$$KL\left(q_{\boldsymbol{\xi}}(\boldsymbol{w}) \,||\, p(\boldsymbol{w})\right) \leq KL(\chi^{(2)}||\chi^{(1)}) + \sum_{i=1}^{C}\sum_{j=1}^{|\mathcal{W}_s|}\chi_{j,i}^{(2)}KL(\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \,||\, \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_j, \sigma_p^2\boldsymbol{I})) \tag{31}$$

And substituting 31 in the negative ELBO in 2 we get the following upper bound.

$$\begin{aligned}\mathcal{L}(\boldsymbol{\xi}) \leq \widetilde{\mathcal{L}}(\boldsymbol{\xi}) &= \mathbb{E}_{\boldsymbol{w}\sim q_{\boldsymbol{\xi}}}\left[\|B(\boldsymbol{w})\|_D^2\right] \\ &+ KL(\chi^{(2)}||\chi^{(1)}) + \frac{\lambda}{N}\sum_{i=1}^{C}\sum_{j=1}^{|\mathcal{W}_s|}\chi_{j,i}^{(2)}KL(\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \,||\, \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_j, \sigma_p^2\boldsymbol{I}))\end{aligned} \tag{32}$$

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

$$\begin{aligned}\widetilde{\mathcal{L}}(\boldsymbol{\xi}) &= \frac{1}{C}\sum_{i=1}^{C}\mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)}\left[\|B(\boldsymbol{w})\|_D^2\right] \\ &+ KL(\chi^{(2)}||\chi^{(1)}) + \frac{\lambda}{N}\sum_{i=1}^{C}\sum_{j=1}^{|\mathcal{W}_s|}\chi_{j,i}^{(2)}KL(\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \,||\, \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_j, \sigma_p^2\boldsymbol{I}))\end{aligned} \tag{33}$$

that is easily differentiable with respect to $\boldsymbol{\xi} = (\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_C, \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_C)$ using the Eq. 27, 28, 29, 30 derived for the Gaussian case.

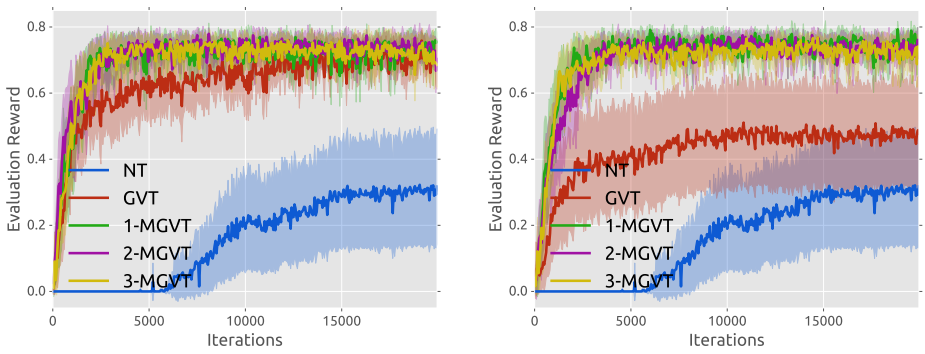


Figure 3: (a) Rooms Problem: Evaluation Reward, (b) Rooms Problem: Evaluation Reward in the generalization experiment

C Additional Details on the Experiments

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

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

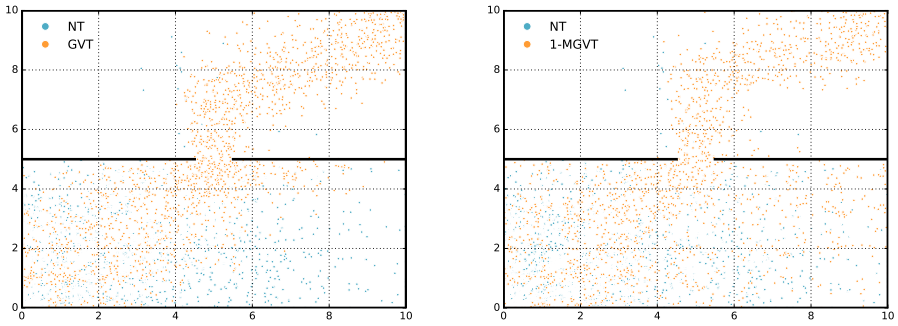
For the transfer algorithm GVT, we set a *Batch Size* of 50 and a *Buffer Size* of 10000. We use $\psi = 0.5$, $\lambda = 10^{-4}$ and 10 *weights* for approximating the expected TD error. For the learning rates we use $\alpha_\mu = 0.001$ for the mean and $\alpha_L = 0.0001$ to learn the Cholesky factor L. Furthermore, we restrict the minimum value reachable for the eigenvalues of these factors to be $\sigma^2_{min} = 0.0001$. 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 $\sigma^2_p = 10^{-5}$.

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 moment 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 for the Rooms Problem 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 algorithm 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 sources 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 is this more complicated setting, whereas GVT fails to adapt as MGVT. 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.



(a) (b)

Figure 4: (a) Rooms Problem: Evaluation Reward, (b) Rooms Problem: Evaluation Reward in the generalization experiment

Furthermore, we investigate how is the exploratory behavior of our transfer algorithms and how they differ. 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.

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

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

C.2 Classic Control

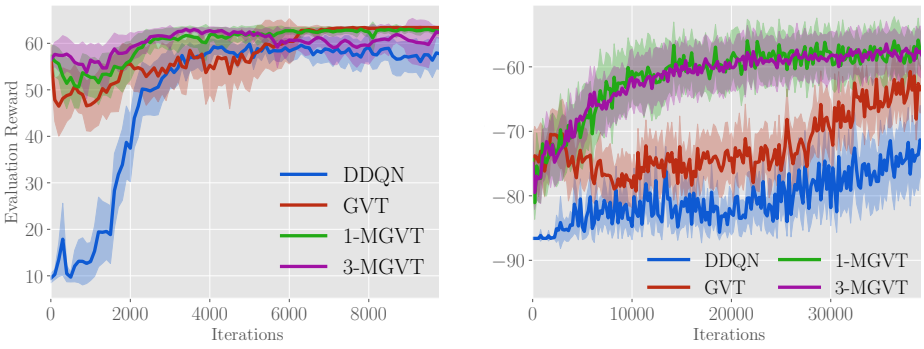
C.2.1 Cartpole

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 and *Exploration Fraction* of 0.35. 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 gradient of the TD error to 5, $\lambda = 0.001$ and $\psi = 0.5$. We use $\alpha_\mu = 0.001$ as the learning rate for the means. For the Cholesky factor L we use $\alpha_L = 0.0001$ and set the a limit that the minimum eigenvalue may reach $\sigma^2_{min} = 0.0001$. Additionally, for MGVT we set the variance of the prior components $\sigma^2_p = 10^{-5}$.

In Fig. 5a, 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 of the Q function. 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.



(a) (b)

Figure 5: (a) Cartpole: Evaluation Reward, (b) Mountain Car: Evaluation Reward

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. 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 2 actions.

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 $\sigma^2_{min} = 0.0001$. In the case of MGVT, additionally, we set $\sigma_p^2 = 10^{-5}$.

In Fig. 5b, 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 current parameters of the network in the case of DDQN and the mean value of the posterior in the cases of GVT and MGVT.

C.3 Maze Navigation

For the maze navigation task presented in Sec. 5.3, here we enumerate the mazes that were designed to realize the experiments. In Fig. 6, 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 used as an approximator an MLP with two hidden layers of 32 neurons with ReLU as activation functions, an input layer receiving the state of the MDP and an output layer that outputs the value of each 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.

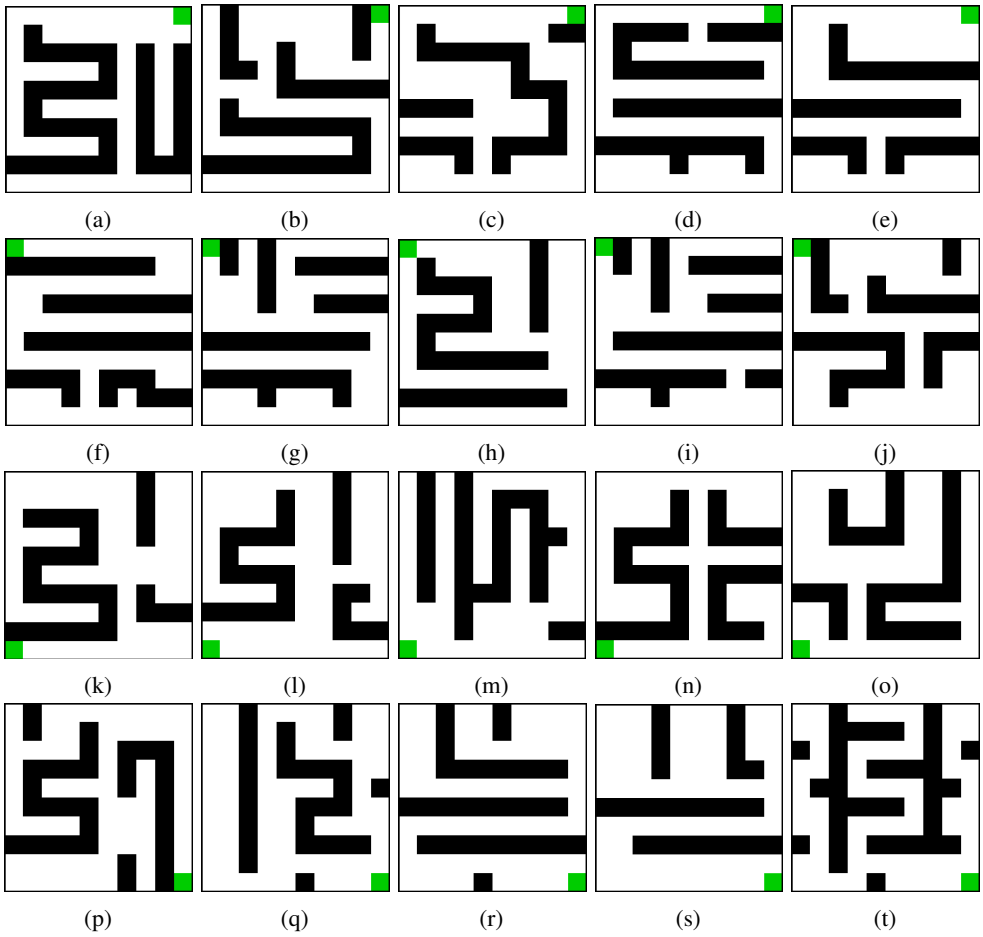


Figure 6: Set of mazes for the Maze Navigation task