VariBAD

Zintgraf, Shiarlis, Igl, Schulze, Gal, Hofmann, Whiteson

Presented by – Homanga Bharadhwaj

The exploration-exploitation dilemma in RL

Since the environment is unknown, an RL agent needs to balance:

- Exploration: Searching for high reward regions of the state-space
- Exploitation: Exploiting the promising regions found

Balancing this trade-off is key to maximising expected return during learning. A Bayes-optimal policy does so optimally

Bayes-optimal policy

In principle,

- A Bayes-optimal policy can be calculated using the framework of Bayes-adaptive MDP (BAMDP)
- In a BAMDP, an agent maintains a belief distribution over possible environments
- A Bayes-optimal agent systematically seeks out new data to reduce uncertainty, but only insofar as doing so helps maximize expected return.
- Performance bounded above by the optimal policy of the corresponding MDP that does not need to take exploratory actions (knows the MDP dynamics)

RL: MDP

- We define a Markov decision process (MDP) as a tuple $M = (\mathcal{S}, \mathcal{A}, R, T, T_0, \gamma, H)$ with $\mathcal{S}$ a set of states, $\mathcal{A}$ a set of actions, $R(r_{t+1}|s_t, a_t, s_{t+1})$ a reward function, $T(s_{t+1}|s_t, a_t)$ a transition function, $T_0(s_0)$ an initial state distribution, γ a discount factor, and H the horizon.
- In the standard RL setting, we want to learn a policy π that maximises $\mathcal{J}(\pi) = \mathbb{E}_{T_0, T, \pi}\left[\sum_{t=0}^{H-1} \gamma^t R(r_{t+1}|s_t, a_t, s_{t+1})\right]$, the expected return.

Bayesian RL: BAMDP

- In the Bayesian formulation of RL, we assume that the transition and reward functions are distributed according to a prior $b_0 = p(R, T)$. Since the agent does not have access to the true reward and transition function, it can maintain a belief $b_t(R, T) = p(R, T|\tau_{:t})$, which is the posterior over the MDP given the agent's experience $\tau_{:t} = \{s_0, a_0, r_1, s_1, a_1, \ldots, s_t\}$ up until the current timestep. This is often done by maintaining a distribution over the model parameters.
- To allow the agent to incorporate the task uncertainty into its decision-making, this belief can be augmented to the state, resulting in hyper-states $s_t^+ \in \mathcal{S}^+ = \mathcal{S} \times \mathcal{B}$, where $\mathcal{B}$ is the belief space.

Bayesian RL: BAMDP

- These transition according to

$$
\begin{aligned}
T^+(s_{t+1}^+ | s_t^+, a_t, r_t) &= T^+(s_{t+1}, b_{t+1} | s_t, a_t, r_t, b_t) \\
&= T^+(s_{t+1} | s_t, a_t, b_t)\ T^+(b_{t+1} | s_t, a_t, r_t, b_t, s_{t+1}) \\
&= \mathbb{E}_{b_t}\left[T(s_{t+1} | s_t, a_t)\right]\ \delta(b_{t+1} = p(R, T | \tau_{:t+1}))
\end{aligned} \tag{1}
$$

- The reward function on hyper-states is defined as the expected reward under the current posterior (after the state transition) over reward functions,

$$
R^+(s_t^+, a_t, s_{t+1}^+) = R^+(s_t, b_t, a_t, s_{t+1}, b_{t+1}) = \mathbb{E}_{b_{t+1}}\left[R(s_t, a_t, s_{t+1})\right]. \tag{2}
$$

This results in a BAMDP $M^+ = (\mathcal{S}^+, \mathcal{A}, R^+, T^+, T_0^+, \gamma, H^+)$

Bayesian RL: BAMDP

- The agent's objective is now to maximise the expected return in the BAMDP,

$$\mathcal{J}^{+}(\pi) = \mathbb{E}_{b_0, T_0^{+}, T^{+}, \pi}\left[\sum_{t=0}^{H^{+}-1} \gamma^{t} R^{+}(r_{t+1} | s_t^{+}, a_t, s_{t+1}^{+})\right], \quad (3)$$

- Solving this BAMDP exactly is *extremely* intractable.

Bayesian RL: BAMDP

The main challenges are as follows.

- We typically do not know the parameterisation of the true reward and/or transition model,
- The belief update (computing the posterior $p(R, T|\tau_{:t})$) is often intractable, and
- Even with the correct posterior, planning in belief space is typically intractable.

Proposed : Bayes-Adaptive Deep RL via Meta Learning

- In the typical meta-learning setting, the reward and transition functions that are unique to each MDP are unknown, but also share some structure across the MDPs M_i in $p(M)$.
- There exists a true i which represents either a task description or task ID, but we do not have access to this information.

Proposed : Bayes-Adaptive Deep RL via Meta Learning

- The authors represent this value using a learned stochastic latent variable m_i. So, for a given MDP M_i we can then write

$$R_i(r_{t+1}|s_t, a_t, s_{t+1}) \approx R(r_{t+1}|s_t, a_t, s_{t+1}; m_i), \quad (4)$$

$$T_i(s_{t+1}|s_t, a_t) \approx T(s_{t+1}|s_t, a_t; m_i), \quad (5)$$

  where R and T are shared across tasks.

- Since we do not have access to the true task description or ID, we need to *infer* m_i given the agent's experience up to time step t collected in M_i,

$$\tau_{:t}^{(i)} = (s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_{t-1}, a_{t-1}, r_t, s_t), \quad (6)$$

Proposed : Bayes-Adaptive Deep RL via Meta Learning

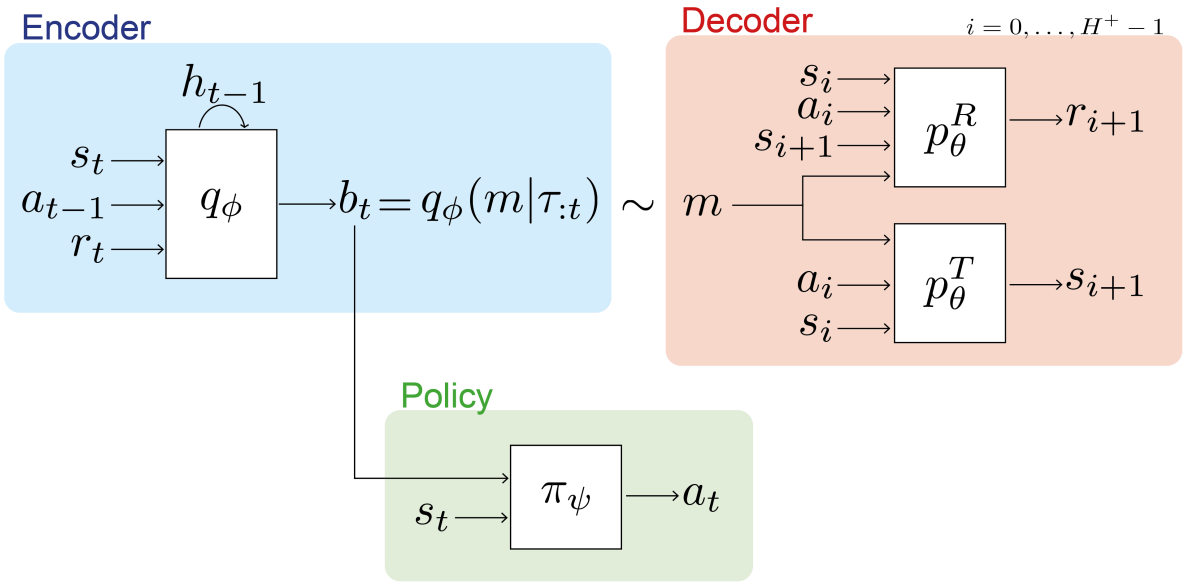


Figure 1: VariBAD architecture: A trajectory of states, actions and rewards is processed online using an RNN to produce the posterior over task embeddings, $q_\phi(m|\tau_{:t})$. The posterior is trained using a decoder which attempts to predict past and future states and rewards from current states and actions. The policy conditions on the posterior in order to act in the environment and is trained using RL.

Proposed: Bayes-Adaptive Deep RL via Meta Learning

- Computing the exact posterior is typically not possible: we do not have access to the MDP (and hence the transition and reward function), and marginalising over tasks is computationally infeasible
- We need to learn a model of the environment $p_\theta(\tau_{:H^+}|a_{:H^+-1})$, parameterised by θ, together with an amortised inference network $q_\phi(m|\tau_{:t})$, parameterised by ϕ, which allows fast inference at runtime *at each timestep* t.
- At any given time step t, the model learning objective is thus to maximise

$$\mathbb{E}_{\rho(M,\tau_{:H^+})}\left[\log p_\theta(\tau_{:H^+}|a_{:H^+-1})\right], \tag{7}$$

where $\rho(M, \tau_{:H^+})$ is the trajectory distribution induced by the policy

Proposed: Bayes-Adaptive Deep RL via Meta Learning

Optimizing the previous objective is intractable, so the following tractable lowed bound with a learned approximate posterior $q_\phi(m|\tau_{:t})$ which can be estimated by MC sampling is optimized :

- $$\begin{aligned}\mathbb{E}_{\rho(M,\tau_{:H^+})}\left[\log p_\theta(\tau_{:H^+})\right] &\geq \mathbb{E}_\rho[\mathbb{E}_{q_\phi(m|\tau_{:t})}[\log p_\theta(\tau_{:H^+}|m)] && (8)\\ &\quad - KL(q_\phi(m|\tau_{:t})||p_\theta(m))] && (9)\\ &= ELBO_t.\end{aligned}$$

- The term $\mathbb{E}_q[\log p(\tau_{:H^+}|m)]$ is often referred to as the reconstruction loss, and $p(\tau_{:t}|m)$ as the decoder.
- The term $KL(q(m|\tau_{:t})||p_\theta(m))$ is the KL-divergence between our variational posterior q_ϕ and the prior over the embeddings $p_\theta(m)$.
- We set the prior to our previous posterior, $q_\phi(m|\tau_{:t-1})$, with initial prior $q_\phi(m) = \mathcal{N}(0, I)$.

Proposed : Bayes-Adaptive Deep RL via Meta Learning

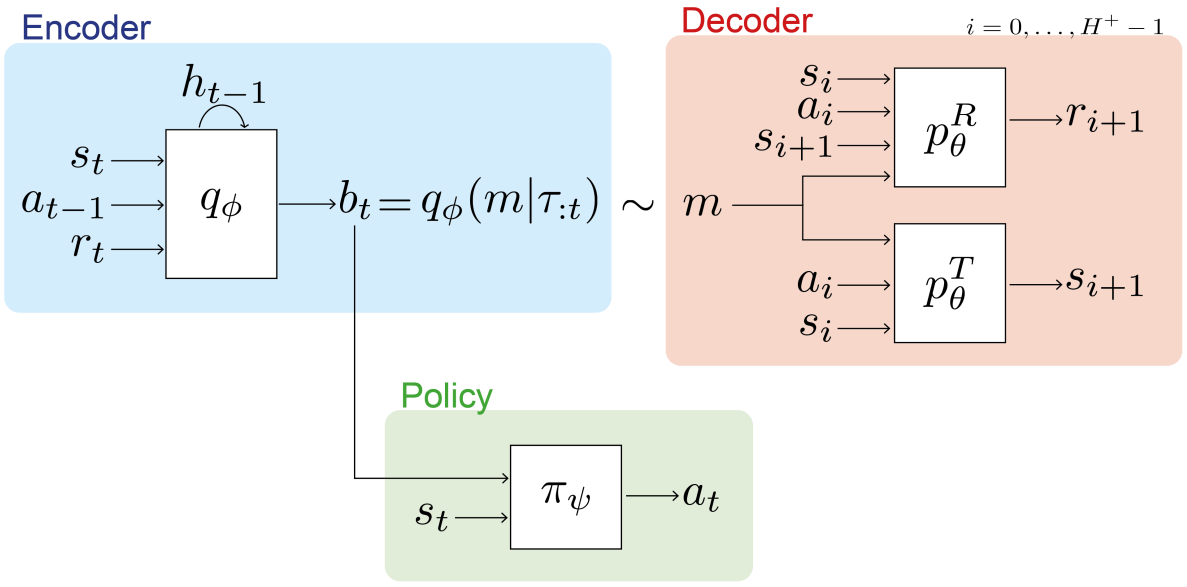


Figure 2: VariBAD architecture: A trajectory of states, actions and rewards is processed online using an RNN to produce the posterior over task embeddings, $q_\phi(m|\tau_{:t})$. The posterior is trained using a decoder which attempts to predict past and future states and rewards from current states and actions. The policy conditions on the posterior in order to act in the environment and is trained using RL.

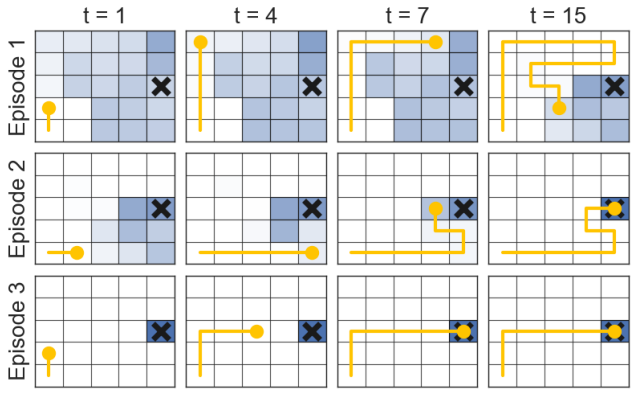
Proposed : Bayes-Adaptive Deep RL via Meta Learning

- The overall objective is to maximise

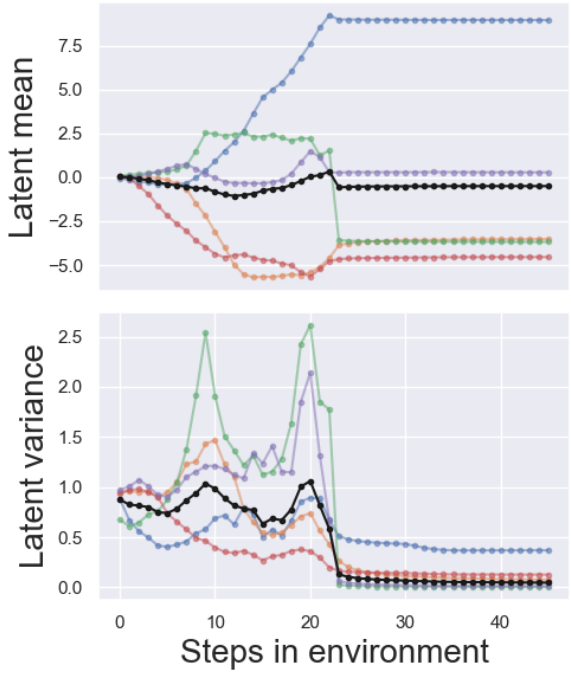
$$\mathcal{L}(\phi, \theta, \psi) = \mathbb{E}_{p(M)} \left[\mathcal{J}(\psi, \phi) + \lambda \sum_{t=0}^{H^+} \mathit{ELBO}_t(\phi, \theta) \right]. \quad (10)$$

- Equation (10) is trained end-to-end, and λ weights the supervised model learning objective against the RL loss.

Results



(a) Example Rollout



(b) Latent Space

Figure 3: Behaviour of VariBAD in the gridworld environment. **(a)** Hand-picked but representative example test rollout. The blue background indicates the posterior probability of receiving a reward at that cell. **(b)** Visualisation of the latent space; each line is one latent dimension, the black line is the average.

Results

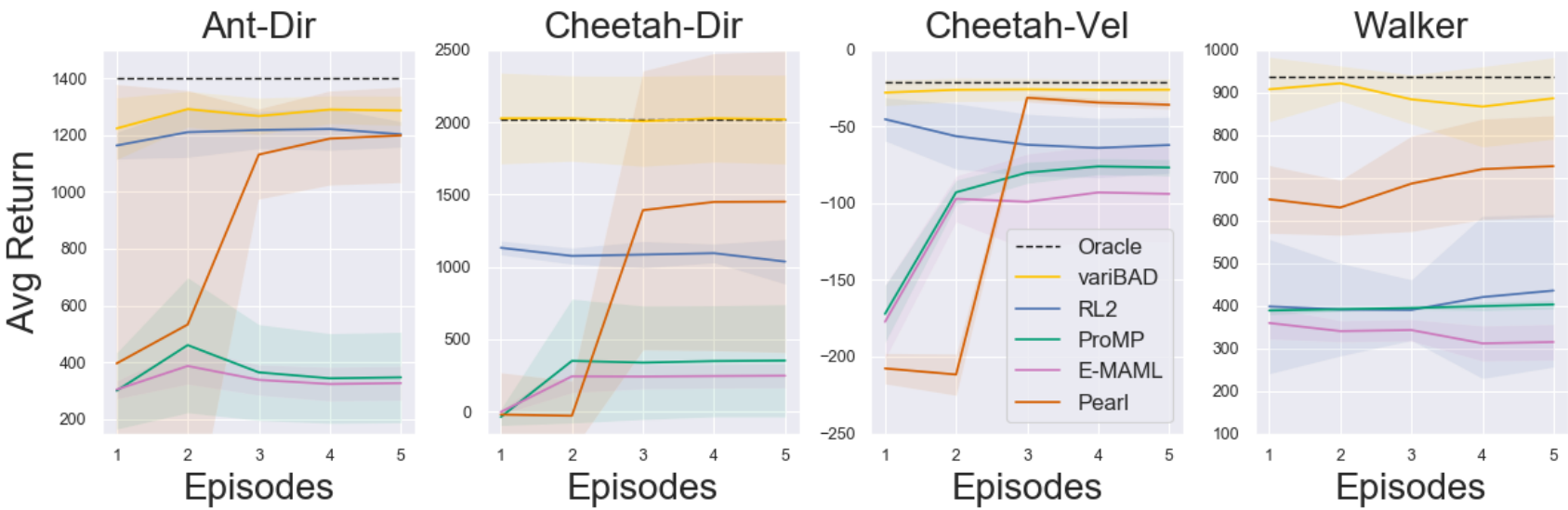


Figure 4: Average test performance for the first 5 rollouts of MuJoCo environments (using 5 seeds).

Summary, Limitations, Future Work

- This papers attempts to learn an approximation to the Bayes-optimal policy in a BAMDP.
- The authors formulate the problem as a Meta Learning setting and through VI learn embeddings corresponding to different MDPs as a proxy for the posterior over belief.
- Training and test distributions are the same. How good is OOD generalization?
- What happens when the problem cannot be conveniently formulated as Meta Learning?
- How to scale it to high dimensional observations (images)?