OFFLINE META LEARNING OF EXPLORATION

Ron Dorfman
Department of Electrical Engineering
Technion
rdorfman@campus.technion.ac.il

Idan Shenfeld
Department of Electrical Engineering
Technion
idanshen@campus.technion.ac.il

Aviv Tamar
Department of Electrical Engineering
Technion
avivt@technion.ac.il

ABSTRACT

Consider the following instance of the Offline Meta Reinforcement Learning (OMRL) problem: given the complete training logs of N conventional RL agents, trained on N different tasks, design a meta-agent that can quickly maximize reward in a new, unseen task from the same task distribution. In particular, while each conventional RL agent explored and exploited its own different task, the meta-agent must identify regularities in the data that lead to effective exploration/exploitation in the unseen task. Here, we take a Bayesian RL (BRL) view, and seek to learn a Bayes-optimal policy from the offline data. Building on the recent VariBAD BRL approach, we develop an off-policy BRL method that learns to plan an exploration strategy based on an adaptive neural belief estimate. However, learning to infer such a belief from offline data brings a new identifiability issue we term MDP ambiguity. We characterize the problem, and suggest resolutions via data collection and modification procedures. Finally, we evaluate our framework on a diverse set of domains, including difficult sparse reward tasks, and demonstrate learning of effective exploration behavior that is qualitatively different from the exploration used by any RL agent in the data[1].

1 INTRODUCTION

A central question in reinforcement learning (RL) is how to learn quickly (i.e., with few samples) in a new environment. Meta-RL addresses this issue by assuming a distribution over possible environments, and having access to a large set of environments from this distribution during training (Duan et al., 2016; Finn et al., 2017). Intuitively, the meta-RL agent can learn regularities in the environments, which allow quick learning in any environment that shares a similar structure.

One possible formulation of quick RL is Bayesian RL (BRL, Ghavamzadeh et al., 2016). In BRL, the environment parameters are treated as unobserved variables, with a known prior distribution. Consequentially, the standard problem of maximizing expected returns (taken with respect to the posterior distribution) *explicitly accounts for the environment uncertainty*, and its solution is a *Bayes-optimal* policy, wherein actions optimally balance exploration and exploitation. Recently, Zintgraf et al. (2020) showed that meta-RL is in fact an instance of BRL, where the meta-RL environment distribution is simply the BRL prior. They proposed the VariBAD algorithm – an implementation of this approach that uses a variational autoencoder (VAE) for adaptive belief estimation and deep neural network policies that take as input both state and posterior belief over the environment parameters.

Most meta-RL studies, including VariBAD, have focused on the *online* setting, where, during training, the meta-RL policy is updated using data collected from running it in the training environments. In domains where data collection is expensive, such as robotics and healthcare , online training is a limiting factor. In this work we investigate the *offline approach to meta-RL* (OMRL). Figure 1 illustrates our problem: in this navigation task, each RL agent in the data learned to find its own goal, and converged to a behavior that quickly navigates toward it. The meta-RL agent, on the other hand, needs to learn a completely different behavior that effectively *searches* for the unknown goal position.

[1]The full version of this paper is available on arXiv (Dorfman et al., 2020).

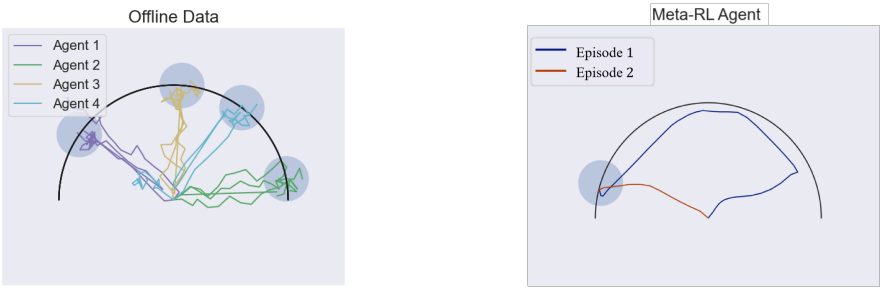


Figure 1: Offline meta-RL on the Semi-Circle domain: The reward is sparse (light-blue), and the offline data (left) contains training logs of conventional RL agents trained to find individual goals. The meta-RL agent (right) needs to find a policy that quickly finds the unknown goal, here, by searching across the semi-circle in the first episode, and directly reaching it the second – a completely different strategy from the dominant behaviors in the data.

Our method for solving OMRL is an off-policy variant of the VariBAD algorithm, based on replacing the on-policy policy gradient optimization in VariBAD with an off-policy Q-learning based approach. The offline setting, however, brings about a challenge – when the agent visits different parts of the state space in different environments, learning to identify the correct environment and obtain an accurate belief estimate becomes challenging, a problem we term *MDP ambiguity*. We formalize this problem and propose a general data collection strategy that can mitigate it. Further, when ambiguity is only due to reward differences, we show that a simple reward relabelling trick suffices, without changing data collection. We collectively term our data collection/relabelling and off-policy algorithm as *Bayesian Offline Reinforcement Learning (BOReL)*.

2 Problem Definition

We follow the standard meta-RL formulation with distribution over MDP rewards and transitions $p(\mathcal{R}, \mathcal{P})$. We are provided training data of an agent interacting with N different MDPs sampled from the distribution, $\{\mathcal{R}_i, \mathcal{P}_i\}_{i=1}^{N} \sim p(\mathcal{R}, \mathcal{P})$. We assume that each interaction is organized as M trajectories of length H, $\tau^{i,j} = s_0^{i,j}, a_0^{i,j}, r_1^{i,j}, s_1^{i,j} \ldots, r_H^{i,j}, s_H^{i,j}, \quad i \in 1, \ldots, N, j \in 1, \ldots, M$, where the rewards satisfy $r_{t+1}^{i,j} = \mathcal{R}_i(s_t^{i,j}, a_t^{i,j})$, the transitions satisfy $s_{t+1}^{i,j} \sim \mathcal{P}_i(\cdot|s_t^{i,j}, a_t^{i,j})$, and the actions are chosen from an arbitrary data collection policy, π_β^i. To ground our work in a specific context, we sometimes assume that the trajectories are obtained from running a conventional RL agent in each one of the MDPs (i.e., the complete RL training logs), which implicitly specifies the data collection policy. In Appendix F we investigate implications of this assumption, but we emphasize that this is merely an illustration, and our approach does not place any such constraint – the trajectories can also be collected differently. Our goal is to use the data for learning a Bayes-optimal policy (Ghavamzadeh et al., 2016), i.e., a policy π that maximizes $\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$, where the expectation is taken with respect to *both the uncertainty in state-action transitions* $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$, $a_t \sim \pi$, *and the uncertainty in the MDP parameters* $\mathcal{R}, \mathcal{P} \sim p(\mathcal{R}, \mathcal{P})$.

3 Identifiability Problems in OMRL

We developed an off-policy variant of the VariBAD algorithm, detailed in the appendix. The main idea is to replace the on-policy policy gradient algorithm with an off-policy method, and to show that when training data is collected as in Section 2, the resulting data tuples can be seen as coming from the correct belief-MDP in expectation. However, while in principle, it is possible to simply run off-policy VariBAD on the offline data, we claim that in many problems this may not work well. The reason is that the VariBAD belief update should reason about the uncertainty in the MDP parameters, which requires to effectively distinguish between the different possible MDPs. Training the VAE to distinguish between MDPs, however, *depends on the offline data*, and might not always

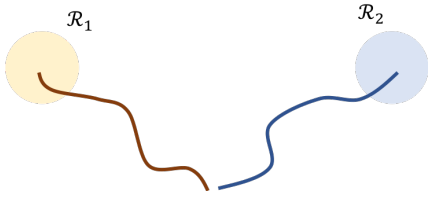


Figure 2: Reward ambiguity: from the two trajectories, it is impossible to know if there are two MDPs with different rewards (blue and yellow circles), or one MDP with rewards at both locations.

be possible. This problem, which we term *MDP ambiguity*, is illustrated in Figure 2: consider two MDPs, one with rewards in the blue circle, and the other with rewards in the yellow circle. If the data contains trajectories similar to the ones in the figure, it is impossible to distinguish between having two different MDPs with the indicated rewards, or a single MDP with rewards at both the blue and yellow circles. Accordingly, we cannot expect to learn a meaningful belief update. In the following, we formalize MDP ambiguity, and how it can be avoided.

For an MDP defined by $\{\mathcal{R}, \mathcal{P}\}$, denote by $P_{\mathcal{R},\mathcal{P},\pi}(s, a, r, s')$ and $P_{\mathcal{P},\pi}(s, a)$ the distribution over (s, a, r, s') and (s, a), respectively, induced by a policy π.

Definition 1 (MDP Ambiguity)**.** *Consider data coming from a set of N different MDPs $M = \{\mathcal{R}_i, \mathcal{P}_i\}_{i=1}^N \subset \mathcal{M}$, where $\mathcal{M}$ is an hypothesis set of possible MDPs, and corresponding data collection policies $\{\pi_\beta^i\}_{i=1}^N$, resulting in N different data distributions $D = \{P_{\mathcal{R}_i,\mathcal{P}_i,\pi_\beta^i}(s, a, r, s')\}_{i=1}^N$. We say that the data is ambiguous if there is an MDP $\{\mathcal{R}, \mathcal{P}\} \in \mathcal{M}$ and two policies π and π' such that $P_{\mathcal{R}_i,\mathcal{P}_i,\pi_\beta^i}(s, a, r, s') = P_{\mathcal{R},\mathcal{P},\pi}(s, a, r, s')$ and $P_{\mathcal{R}_j,\mathcal{P}_j,\pi_\beta^j}(s, a, r, s') = P_{\mathcal{R},\mathcal{P},\pi'}(s, a, r, s')$, for some $i \neq j$. Otherwise, the data is termed identifiable.*[2]

The essence of identifiability, as expressed in Definition 1, is that there is no single MDP in the hypothesis set that can explain data from multiple MDPs in the data, as in this case it will be impossible to learn an inference model that accurately distinguishes MDPs, even with infinite data.

Note that MDP ambiguity is special to offline meta-RL; in online meta-RL, the agent may be driven by the online adapting policy (or guided explicitly) to explore states that reduce its ambiguity. We also emphasize that this problem is not encountered in standard (non-meta) offline RL, as the problem here concerns the *identification of the MDP*, which in standard RL is unique.

Proposition 1. *Consider the setting described in Definition 1. For a pair of MDPs i and j, we define the identifying state-action pairs as the state-action pairs that satisfy $\mathcal{R}_i(s, a) \neq \mathcal{R}_j(s, a)$ and/or $\mathcal{P}_i(\cdot|s, a) \neq \mathcal{P}_j(\cdot|s, a)$. If for every $i \neq j$ there exists an identifying state-action pair that has positive probability under both i and j, i.e., $P_{\mathcal{P}_i,\pi_\beta^i}(s, a), P_{\mathcal{P}_j,\pi_\beta^j}(s, a) > 0$, then the data is identifiable.*

Thus, if the agent has data on identifying state-actions *obtained from different MDPs*, it has the capability to identify which data samples belong to which MDP, regardless of the hypothesis set $\mathcal{M}$.

How can one collect data to mitigate MDP ambiguity? We present a simple, general modification to the data collection scheme we term **policy replaying**, which, under mild conditions on the original data collection policies, guarantees that the resulting data will be identifiable. We importantly note that changing the data collection method in-hindsight is not suitable for the offline setting. Therefore, the proposed scheme should be viewed as *a guideline for effective OMRL data collection*. For each MDP, we propose collecting data in the following manner: randomly draw a data collection policy from $\{\pi_\beta^i\}_{i=1}^N$, collect a trajectory following that policy, and repeat. After this procedure, the new data distributions are all associated with *the same* data collection policy, which we denote π_r.

Proposition 2. *For every $i \neq j$, denote the set of identifying state-action pairs by $\mathcal{I}_{i,j}$. If for every i and every j exists $(s_{i,j}, a_{i,j}) \in \mathcal{I}_{i,j}$ such that $P_{\mathcal{P}_i,\pi_\beta^i}(s_{i,j}, a_{i,j}) > 0$, then replacing π_β^i with π_r for all i results in identifiable data.*

Note that the requirement on identifying state-actions in Proposition 2 is minimal – without it, the original data collecting policies π_β^i are useless, as they do not visit any identifying state-action pair.

When the tasks only differ in their reward function, and the reward functions for the training environments are known, policy replaying can be implemented in hindsight, *without changing the data collection process*. This technique, which we term **Reward Relabelling (RR)**, is applicable under the offline setting, and described next. In RR, we replace the rewards in a trajectory from some MDP i in the data with rewards from another randomly chosen MDP $j \neq i$. That is, for each $i \in 1, \ldots, N$, we add to the data K trajectories $\hat{\tau}^{i,k} = (s_0^{i,k}, a_0^{i,k}, \hat{r}_1^{i,k}, s_1^{i,k} \ldots, \hat{r}_H^{i,k}, s_H^{i,k}), \quad k \in 1, \ldots, K$, where the relabelled rewards $\hat{r}$ satisfy $\hat{r}_{t+1}^{i,k} = \mathcal{R}_j(s_t^{i,k}, a_t^{i,k})$. Thus, our relabelling effectively runs π_β^i on MDP j, which is equivalent to performing policy replaying (in hindsight).

[2] $P(\cdot) = P'(\cdot)$ means equality almost everywhere.

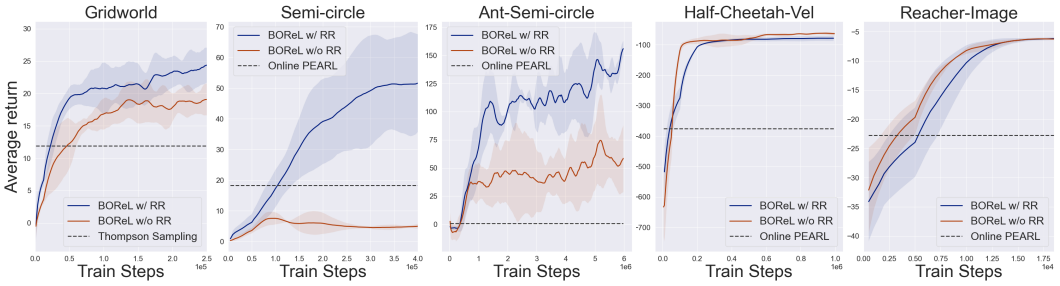


Figure 3: Performance on evaluated domains. We compare BOReL with and without reward relabeling (blue and red, respectively) with Thompson sampling baselines – calculated exactly in the Gridworld domain, and using online PEARL for the other domains. We plot the best performance for PEARL.

4 Experiments

In our experiments, we aim to demonstrate: (1) Learning approximately Bayes-optimal policies in the offline setting; and (2) The severity of MDP ambiguity, and the effectiveness of our proposed resolutions. We compare our offline results with online methods based on Thompson sampling, which are not Bayes-optimal, and aim to show that the performance improvement due to being approximately Bayes-optimal gives an advantage, *even under the offline data* restriction. We further describe our method, domains, evaluation metric, and data collection process in the supplementary.

Main Results: In Figure 3 we compare our offline algorithm with Thompson sampling based methods, and also with an ablation of the reward relabelling method. For Gridworld, the Thompson sampling method is computed exactly, while for the continuous environments, we use online PEARL (Rakelly et al., 2019) – a strong baseline that is *not affected by our offline data limitation*. Note that **we significantly outperform Thompson sampling based methods, demonstrating our claim of learning non-trivial exploration from offline data**. These results are further explained qualitatively by observing the exploration behavior of our learned agents. In Figure 1 and in Figure 4, we visualize the trajectories of the trained agents in the Semi-circle and Ant-Semi-circle domains, respectively. An approximately Bayes-optimal behavior is evident: in the first episode, the agents search for the goal along the semi-circle, and in the second episode, the agents maximize reward by moving directly towards the already found goal. In contrast, a Thompson sampling based agent will never display such search behavior, as *it does not plan* to proactively reduce uncertainty. Instead, such an agent will randomly choose an un-visited possible goal at each episode and directly navigate towards it (cf. Figure 1 in Li et al. 2020). We further emphasize that the approximately Bayes-optimal search behavior is different from the training data, in which the agents learned to reach specific goals.

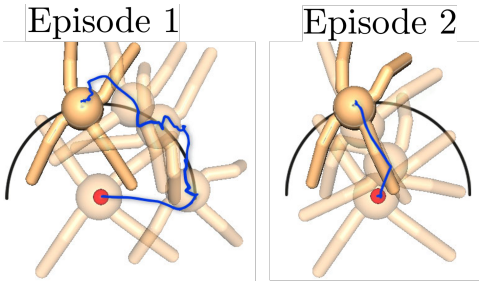


Figure 4: Ant-Semi-circle – a challenging modification of the popular Ant-Goal task (Finn et al., 2017), where a sparse reward is located somewhere on a semi-circle: trajectories from trained policy on a new goal. Note that in the first episode, the ant searches for the goal, and in the second one it directly moves toward the goal it has previously found. This search behavior is different from the goal-reaching behaviors that dominate the training data.

Our results also signify the severity of MDP ambiguity, and align with the theory in Sec. 3. In domains with non-overlapping identifying states (Gridworld, Semi-circle, and Ant-Semi-circle), as expected, performance without reward relabelling is poor, while in domains with overlapping identifying states reward relabelling has little effect. In Figure 6 in the supplementary, we provide further insight into these results, by plotting the belief update during the episode rollout for Semi-circle: the belief starts as uniform on the semi-circle, and narrows in on the target as the agent explores the semi-circle. With reward relabelling ablated, however, we show that the belief does not update correctly.

5 Conclusion and Future Work

We presented the first offline meta-RL algorithm that is approximately Bayes-optimal, allowing to solve problems where efficient exploration is crucial. The connection between Bayesian RL and meta

learning allows to reduce the problem to offline RL on belief-augmented states. However, learning a neural belief update from offline data is prone to the MDP ambiguity problem. We formalized the problem, and proposed a simple data collection protocol that guarantees identifiability. In the particular case of tasks that differ in their rewards, our protocol can be implemented in hindsight, for arbitrarily offline data. Our results show that this solution is effective on several challenging domains.

It is intriguing whether other techniques can mitigate MDP ambiguity. For example, designing data collection policies that diversify the data or injecting prior knowledge by controlling the hypothesis set of the neural belief update.

REFERENCES

Anthony R Cassandra, Leslie Pack Kaelbling, and Michael L Littman. Acting optimally in partially observable stochastic domains. In *AAAI*, volume 94, pp. 1023–1028, 1994.

Ron Dorfman, Idan Shenfeld, and Aviv Tamar. Offline meta learning of exploration. *arXiv preprint arXiv:2008.02598*, 2020.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Michael O'Gordon Duff and Andrew Barto. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massachusetts at Amherst, 2002.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. Bayesian reinforcement learning: A survey. *arXiv preprint arXiv:1609.04436*, 2016.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.

Jiachen Li, Quan Vuong, Shuang Liu, Minghua Liu, Kamil Ciosek, Henrik Iskov Christensen, and Hao Su. Multi-task batch reinforcement learning with metric learning. *arXiv preprint arxiv:1909.11373*, 2020.

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):529–533, 2015.

Kate Rakelly, Aurick Zhou, Deirdre Quillen, Chelsea Finn, and Sergey Levine. Efficient off-policy meta-reinforcement learning via probabilistic context variables. *arXiv preprint arXiv:1903.08254*, 2019.

Harm Van Seijen, Hado Van Hasselt, Shimon Whiteson, and Marco Wiering. A theoretical and empirical analysis of expected sarsa. In *2009 ieee symposium on adaptive dynamic programming and reinforcement learning*, pp. 177–184. IEEE, 2009.

Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. In *International Conference on Learning Representation (ICLR)*, 2020.

A EXTENDED BACKGROUND

We recapitulate BRL and the VariBAD algorithm.

Bayesian Reinforcement Learning: The goal in BRL is to find the optimal policy π in an MDP, when the transitions and rewards are not known in advance. Similar to meta-RL, we assume a prior over the MDP parameters $p(\mathcal{R}, \mathcal{P})$, and seek to maximize the expected discounted return,

$$\mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right], \tag{1}$$

where the expectation is taken with respect to *both the uncertainty in state-action transitions* $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$, $a_t \sim \pi$, *and the uncertainty in the MDP parameters* $\mathcal{R}, \mathcal{P} \sim p(\mathcal{R}, \mathcal{P})$. Key here is that this formulation naturally accounts for the exploration/exploitation tradeoff – an optimal agent must plan its actions to reduce uncertainty in the MDP parameters, if such leads to higher rewards.

One way to approach the BRL problem is to model $\mathcal{R}, \mathcal{P}$ as unobserved state variables in a partially observed MDP (POMDP, Cassandra et al., 1994), reducing the problem to solving a particular POMDP instance where the unobserved variables do not change in time. The *belief* at time t, b_t, denotes the posterior probability over $\mathcal{R}, \mathcal{P}$ given the history of state transitions and rewards observed until this time $b_t = P(\mathcal{R}, \mathcal{P}|h_{:t})$, where $h_{:t} = \{s_0, a_0, r_1, s_1 \ldots, r_t, s_t\}$ (note that we denote the reward after observing the state and action at time t as $r_{t+1} = r(s_t, a_t)$). The belief can be updated iteratively according to Bayes rule, where $b_0(\mathcal{R}, \mathcal{P}) = p(\mathcal{R}, \mathcal{P})$, and: $b_{t+1}(\mathcal{R}, \mathcal{P}) = P(\mathcal{R}, \mathcal{P}|h_{:t+1}) \propto P(s_{t+1}, r_{t+1}|h_{:t}, \mathcal{R}, \mathcal{P}) b_t(\mathcal{R}, \mathcal{P})$.

Similar to the idea of solving a POMDP by representing it as an MDP over belief states, the state in BRL can be augmented with the belief to result in the Bayes-Adaptive MDP model (BAMDP, Duff & Barto, 2002). Denote the augmented state $s_t^+ = (s_t, b_t)$ and the augmented state space $\mathcal{S}^+ = \mathcal{S} \times \mathcal{B}$, where $\mathcal{B}$ denotes the belief space. The transitions in the BAMDP are given by: $P^+(s_{t+1}^+|s_t^+, a_t) = \mathbb{E}_{b_t}\left[\mathcal{P}(s_{t+1}|s_t, a_t)\right] \delta\left(b_{t+1} = P(\mathcal{R}, \mathcal{P}|h_{:t+1})\right)$, and the reward in the BAMDP is the expected reward with respect to the belief: $R^+(s_t^+, a_t) = \mathbb{E}_{b_t}\left[\mathcal{R}(s_t, a_t)\right]$. The Bayes-optimal agent seeks to maximize the expected discounted return in the BAMDP, and the optimal solution of the BAMDP gives the optimal BRL policy. As in standard MDPs, the optimal action-value function in the BAMDP satisfies the Bellman equation: $\forall s^+ \in \mathcal{S}^+, a \in \mathcal{A}$ we have that

$$Q(s^+,a) = R^+(s^+, a) + \gamma \mathbb{E}_{s^{+'} \sim P^+}\big[\max_{a'} Q(s^{+'}, a')\big]. \tag{2}$$

Computing a Bayes-optimal agent amounts to solving the BAMDP, where the optimal policy is a function of the augmented state. For most problems this is intractable, as the augmented state space is continuous and high-dimensional, and the posterior update is also intractable in general.

The VariBAD Algorithm: VariBAD (Zintgraf et al., 2020) approximates the Bayes-optimal solution by combining a model for the MDP parameter uncertainty, and an optimization method for the corresponding BAMDP. The MDP parameters are represented by a vector $m \in \mathbb{R}^d$, corresponding to the latent variables in a parametric generative model for the state-reward trajectory distribution conditioned on the actions $P(s_0, r_1, s_1 \ldots, r_H, s_H|a_0, \ldots, a_{H-1}) = \int p_\theta(m) p_\theta(s_0, r_1, s_1 \ldots, r_H, s_H|m, a_0, \ldots, a_{H-1}) dm$. The model parameters θ are learned by a variational approximation to the maximum likelihood objective, where the variational approximation to the posterior $P(m|s_0, r_1, s_1 \ldots, r_H, s_H, a_0, \ldots, a_{H-1})$ is chosen to have the structure $q_\phi(m|s_0, a_0, r_1, s_1 \ldots, r_t, s_t) = q_\phi(m|h_{:t})$. That is, the approximate posterior is conditioned on the history up to time t. The evidence lower bound (ELBO) in this case is $ELBO_t = \mathbb{E}_{m \sim q_\phi(\cdot|h_{:t})}\left[\log p_\theta(s_0, r_1, s_1 \ldots, r_H, s_H|m, a_0, \ldots, a_{H-1})\right] - D_{KL}(q_\phi(m|h_{:t})||p_\theta(m))$. The main claim of Zintgraf et al. (2020) is that $q_\phi(m|h_{:t})$ can be taken as an approximation of the belief b_t. In practice, $q_\phi(m|h_{:t})$ is represented as a Gaussian distribution $q(m|h_{:t}) = \mathcal{N}(\mu(h_{:t}), \Sigma(h_{:t}))$, where μ and Σ are learned recurrent neural networks.

To approximately solve the BAMDP, Zintgraf et al. (2020) exploit the fact that an optimal BAMDP policy is a function of the state and belief, and therefore consider neural network policies that take the augmented BAMDP state as input $\pi(a_t|s_t, q_\phi(m|h_{:t}))$, where the posterior is practically represented by the distribution parameters $\mu(h_{:t}), \Sigma(h_{:t})$. The policies are trained using policy gradients, optimizing

$$J(\pi) = \mathbb{E}_{\mathcal{R}, \mathcal{P}} \mathbb{E}_{\pi}\left[\sum_{t=0}^{H}\gamma^t r(s_t, a_t)\right]. \tag{3}$$

The expectation over MDP parameters in (3) is approximated by averaging over training environments, and the RL agent is trained online, alongside the VAE.

B OMRL AND OFF-POLICY VARIBAD

The online VariBAD algorithm updates the policy using *trajectories sampled from the current policy*, and thus cannot be applied to our offline setting. Our first step is to modify VariBAD to work off-policy. We start with an observation about the use of the BAMDP formulation in VariBAD, which will motivate our subsequent development.

Does VariBAD really optimize the BAMDP? Recall that a BAMDP is in fact a reduction of a POMDP to an MDP over augmented states $s^+ = (s, b)$, and with the rewards and transitions given by R^+ and P^+. Thus, an optimal Markov policy for the BAMDP exists in the form of $\pi(s^+)$. The VariBAD policy, as described above, similarly takes as input the augmented state, and is thus capable of representing an optimal BAMDP policy. However, ***VariBAD's policy optimization in Eq. (3) does not make use of the BAMDP parameters*** R^+ and P^+! While at first this may seem counterintuitive, Eq. (3) is in fact a sound objective for the BAMDP, as we now show .

Proposition 3. *Let $\tau = s_0, a_0, r_1, s_1 \ldots, r_H, s_H$ denote a random trajectory from a fixed history dependent policy π, generated according to the following process. First, MDP parameters $\mathcal{R}, \mathcal{P}$ are drawn from the prior $p(\mathcal{R}, \mathcal{P})$. Then, the state trajectory is generated according to $s_0 \sim P_{init}$, $a_t \sim \pi(\cdot|s_0, a_0, r_1, \ldots, s_t)$, $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$ and $r_{t+1} \sim \mathcal{R}(s_t, a_t)$. Let b_t denote the posterior belief at time t, $b_t = P(\mathcal{R}, \mathcal{P}|s_0, a_0, r_1, \ldots, s_t)$. Then*

$$P(s_{t+1}|s_0, a_0, r_1, \ldots, r_t, s_t, a_t) = \mathbb{E}_{\mathcal{R},\mathcal{P}\sim b_t}\mathcal{P}(s_{t+1}|s_t, a_t),$$

and,

$$P(r_{t+1}|s_0, a_0, r_1, \ldots, s_t, a_t) = \mathbb{E}_{\mathcal{R},\mathcal{P}\sim b_t}\mathcal{R}(r_{t+1}|s_t, a_t).$$

Proof. For the transitions, we have that,

$$\begin{aligned}
P(s_{t+1}|s_0, a_0, r_0, \ldots, r_t, s_t, a_t) &= \int P(s_{t+1}, \mathcal{R}, \mathcal{P}|s_0, a_0, r_0, \ldots, r_t, s_t, a_t) d\mathcal{R} d\mathcal{P} \\
&= \int P(s_{t+1}|\mathcal{R}, \mathcal{P}, s_0, a_0, r_0, \ldots, r_t, s_t, a_t) \\
&\qquad P(\mathcal{R}, \mathcal{P}|s_0, a_0, r_0, \ldots, r_t, s_t, a_t) d\mathcal{R} d\mathcal{P} \\
&= \mathbb{E}_{\mathcal{R},\mathcal{P}}[P(s_{t+1}|\mathcal{R}, \mathcal{P}, s_0, a_0, r_0, \ldots, r_t, s_t, a_t)|\, s_0, a_0, r_0, \ldots, r_t, s_t, a_t] \\
&= \mathbb{E}_{\mathcal{R},\mathcal{P}}\left[\mathcal{P}(s_{t+1}|s_t, a_t)|\, s_0, a_0, r_0, \ldots, r_t, s_t, a_t\right] \\
&= \mathbb{E}_{\mathcal{R},\mathcal{P}\sim b_t}\mathcal{P}(s_{t+1}|s_t, a_t).
\end{aligned}$$

The proof for the rewards proceeds similarly. □

For on-policy VariBAD, Proposition 3 shows that the rewards and transitions in each trajectory can be seen as sampled from a distribution that **in expectation** is equal to R^+ and P^+, and therefore maximizing Eq. 3 is valid.[3] However, off-policy RL does not take as input trajectories, but tuples of the form $(s, a, r, s') \equiv (state, action, reward, next\ state)$, where states and actions can be sampled **from any distribution**. For an arbitrary distribution of augmented states, we must replace the rewards and transitions in our data with R^+ and P^+, which can be difficult to compute. Fortunately, Proposition 3 shows that when collecting data by sampling complete trajectories, this is not necessary, as in expectation, the rewards and transitions are correctly sampled from the BAMDP. In the following, we therefore focus on settings where data can be collected that way, for example, by collecting logs of RL agents trained on the different training tasks.

Based on Proposition 3, we can use a state augmentation method similar to VariBAD, which we refer to as **state relabelling**. Consider each trajectory in our data $\tau^{i,j} = s_0^{i,j}, a_0^{i,j}, r_1^{i,j}, \ldots, s_H^{i,j}$, as defined above. Recall that the VariBAD VAE encoder provides an estimate of the belief given the state history $q(m|h_{:t}) = \mathcal{N}(\mu(h_{:t}), \Sigma(h_{:t}))$. Thus, we can run the encoder on every partial t-length history $\tau_{:t}^{i,j}$ to obtain the belief at each time step. Following the BAMDP formulation, we define the augmented state $s_t^{+,i,j} = (s_t^{i,j}, b_t^{i,j})$, where $b_t^{i,j} = \mu(\tau_{:t}^{i,j}), \Sigma(\tau_{:t}^{i,j})$. We next replace each state in our data $s_t^{i,j}$ with $s_t^{+,i,j}$, effectively transforming the data to as coming from a BAMDP. After applying state relabelling, any off-policy RL algorithm can be applied to the modified data, for learning a Bayes-optimal policy. In our experiments we used DQN (Mnih et al., 2015) for discrete action domains, and soft actor critic (SAC, Haarnoja et al., 2018) for continuous control.

[3] To further clarify, if we could calculate R^+, replacing all rewards in the trajectories with R^+ will result in a lower variance policy update, similar to expected SARSA (Van Seijen et al., 2009).

C Extended Definitions and Proofs for Section 3

For the proofs of identifiability, we start by elaborating the formal definition of our setting. For simplicity, we assume that the MDPs $\{\mathcal{R}_i, \mathcal{P}_i\}_{i=1}^N$ are defined over finite state-action spaces ($|\mathcal{S}|, |\mathcal{A}| < \infty$). For every $i = 1, \ldots, N$, let π_β^i be a general stationary, stochastic, history-dependent policy[4]. The initial state distribution P_{init} is the same across all MDPs.[5]

We assume that data is collected from trajectories of length at most $T_{\max}$. This is a convenient assumption that holds in every practical scenario, and allows us to side step issues of defining visitation frequencies when $t \to \infty$.

For some $0 \leq t \leq T_{\max}$, denote by $P_{\mathcal{P}_i,\pi_\beta^i,t}(s,a) = P_{\mathcal{P}_i,\pi_\beta^i}(s_t = s, a_t = a)$ the probability of visiting the state-action pair (s,a) at time t by running the policy π_β^i on MDP with transition function $\mathcal{P}_i$ and initial state distribution P_{init}. Now, we define:

$$P_{\mathcal{P}_i,\pi_\beta}(s,a) = P_{\mathcal{P}_i,\pi_\beta^i}\Big(\bigcup_{t\in\{0,\ldots,T_{\max}\}} \{s_t = s, a_t = a\}\Big),$$

that is, $P_{\mathcal{P}_i,\pi_\beta}(s,a)$ is the probability of observing state-action (s,a) in the data from MDP i. Similarly, we define

$$P_{\mathcal{P}_i,\mathcal{R}_i,\pi_\beta}(s,a,r,s') = P_{\mathcal{P}_i,\pi_\beta}(s,a)\mathcal{P}_i(s'|s,a)P_{\mathcal{R}_i}(r|s,a),$$

the probability of observing the tuple (s,a,r,s') in the data from MDP i.

A trajectory from the replay policy π_r in MDP i is generated as follows. Let x be a discrete random variable defined on $1, \ldots, N$ with probability $P_x(\cdot)$ that satisfies $P_x(x = k) > 0$ for every $k = 1, \ldots, N$. First, we draw x. Then, we sample a trajectory from MDP i using policy π_β^x.

For ease of reading, we copy here the propositions from the main text.

Proposition 1. *Consider the setting described in Definition 1. For a pair of MDPs i and j, we define the identifying state-action pairs as the state-action pairs that satisfy $\mathcal{R}_i(s,a) \neq \mathcal{R}_j(s,a)$ and/or $\mathcal{P}_i(s'|s,a) \neq \mathcal{P}_j(s'|s,a)$. If for every $i \neq j$ there exists an identifying state-action pair that has positive probability under both i and j, i.e., $P_{\mathcal{P}_i,\pi_\beta^i}(s,a), P_{\mathcal{P}_j,\pi_\beta^j}(s,a) > 0$, then the data is identifiable.*

Before we prove Proposition 2, we present the following lemma, which will be used later in the proof.

Lemma 1. *Consider a pair of MDPs $(\mathcal{R},\mathcal{P})$ and $(\mathcal{R}',\mathcal{P}')$, and two policies π and π'. If there exists an identifying state-action pair of the MDPs $(\bar{s},\bar{a})$ that has positive probability under both $(\mathcal{P},\pi)$ and $(\mathcal{P}',\pi')$, i.e., $P_{\mathcal{P},\pi}(\bar{s},\bar{a}), P_{\mathcal{P}',\pi'}(\bar{s},\bar{a}) > 0$, then $P_{\mathcal{R},\mathcal{P},\pi}(s,a,r,s') \neq P_{\mathcal{R}',\mathcal{P}',\pi'}(s,a,r,s')$.*

Proof. Assume to the contrary that $P_{\mathcal{R},\mathcal{P},\pi}(s,a,r,s') = P_{\mathcal{R}',\mathcal{P}',\pi'}(s,a,r,s')$. Marginalizing over r and s', we obtain:

$$\sum_{r,s'} P_{\mathcal{R},\mathcal{P},\pi}(s,a,r,s') = \sum_{r,s'} P_{\mathcal{R},\mathcal{P}',\pi'}(s,a,r,s')$$
$$P_{\mathcal{P},\pi}(s,a) = P_{\mathcal{P}',\pi'}(s,a), \quad \forall (s,a).$$

Specifically, we have $P_{\mathcal{P},\pi}(\bar{s},\bar{a}) = P_{\mathcal{P}',\pi'}(\bar{s},\bar{a})$. Since $P_{\mathcal{R},\mathcal{P},\pi}(\bar{s},\bar{a},r,s') = P_{\mathcal{R},\mathcal{P}}(r,s'|\bar{s},\bar{a})P_{\mathcal{P},\pi}(\bar{s},\bar{a})$ for every r and s', and $P_{\mathcal{P},\pi}(\bar{s},\bar{a}) = P_{\mathcal{P}',\pi'}(\bar{s},\bar{a}) > 0$, it holds that $P_{\mathcal{R},\mathcal{P}}(r,s'|\bar{s},\bar{a}) = P_{\mathcal{R}',\mathcal{P}'}(r,s'|\bar{s},\bar{a})$ for every r and s'. By marginalizing over s' we get that

$$\sum_{s'} P_{\mathcal{R},\mathcal{P}}(r,s'|\bar{s},\bar{a}) = \sum_{s'} P_{\mathcal{R}',\mathcal{P}'}(r,s'|\bar{s},\bar{a})$$
$$P_{\mathcal{R}}(r|\bar{s},\bar{a}) = P_{\mathcal{R}'}(r|\bar{s},\bar{a}).$$

Similarly, by marginalizing over r, we get $\mathcal{P}_i(s'|\bar{s},\bar{a}) = \mathcal{P}_j(s'|\bar{s},\bar{a})$. Overall, both reward and transition function do not differ in $(\bar{s},\bar{a})$, which contradicts the fact that $(\bar{s},\bar{a})$ is an identifying state-action pair. □

[4] We consider stationary policies for notation simplicity, although similar analysis can be made for non-stationary policies.

[5] The idea of policy replaying can be extended to MDP with different initial state distributions by randomly selecting the state distribution along with the policy. For simplicity, we do not consider this case, although a similar analysis holds for it.

We now prove Proposition 1.

Proof. Consider some $i \neq j$. Let $(s_{i,j}, a_{i,j})$ be an identifying state-action pair that has positive probability under both i and j. Assume to the contrary that there exists an MDP $\{\mathcal{R}, \mathcal{P}\} \in \mathcal{M}$ and two policies π and π' such that $P_{\mathcal{R}_i,\mathcal{P}_i,\pi_\beta^i}(s,a,r,s') = P_{\mathcal{R},\mathcal{P},\pi}(s,a,r,s')$ and $P_{\mathcal{R}_j,\mathcal{P}_j,\pi_\beta^j}(s,a,r,s') = P_{\mathcal{R},\mathcal{P},\pi'}(s,a,r,s')$.

Since $(s_{i,j}, a_{i,j})$ has positive probability under $(\mathcal{P}_i, \pi_\beta^i)$ and $P_{\mathcal{R}_i,\mathcal{P}_i,\pi_\beta^i}(s,a,r,s') = P_{\mathcal{R},\mathcal{P},\pi}(s,a,r,s')$, then $(s_{i,j}, a_{i,j})$ must also have positive probability under $(\mathcal{P}, \pi)$ (otherwise, there are r and s' for which $P_{\mathcal{R}_i,\mathcal{P}_i,\pi_\beta^i}(s_{i,j}, a_{i,j}, r, s') > 0$, while $P_{\mathcal{R},\mathcal{P},\pi}(s_{i,j}, a_{i,j}, r, s') = 0$). Now, since $(s_{i,j}, a_{i,j})$ has positive probability under both $(\mathcal{P}_i, \pi_\beta^i)$ and $(\mathcal{P}, \pi)$, and $P_{\mathcal{R}_i,\mathcal{P}_i,\pi_\beta^i}(s,a,r,s') = P_{\mathcal{R},\mathcal{P},\pi}(s,a,r,s')$, according to Lemma 1, it cannot be an identifying state-action pair of $(\mathcal{R}_i, \mathcal{P}_i)$ and $(\mathcal{R}, \mathcal{P})$. Therefore, the MDP $\{\mathcal{R}, \mathcal{P}\}$ must satisfy $\mathcal{P}(\cdot|s_{i,j}, a_{i,j}) = \mathcal{P}_i(\cdot|s_{i,j}, a_{i,j})$ and $\mathcal{R}(s_{i,j}, a_{i,j}) = \mathcal{R}_i(s_{i,j}, a_{i,j})$.

The same argument can be made for $(\mathcal{R}_j, \mathcal{P}_j, \pi_\beta^j)$ and $(\mathcal{R}, \mathcal{P}, \pi')$, resulting in $\mathcal{P}(\cdot|s_{i,j}, a_{i,j}) = \mathcal{P}_j(\cdot|s_{i,j}, a_{i,j})$ and $\mathcal{R}(s_{i,j}, a_{i,j}) = \mathcal{R}_j(s_{i,j}, a_{i,j})$. Overall, we get $\mathcal{P}_i(\cdot|s_{i,j}, a_{i,j}) = \mathcal{P}(\cdot|s_{i,j}, a_{i,j}) = \mathcal{P}_j(\cdot|s_{i,j}, a_{i,j})$ and $\mathcal{R}_i(s_{i,j}, a_{i,j}) = \mathcal{R}(s_{i,j}, a_{i,j}) = \mathcal{R}_j(s_{i,j}, a_{i,j})$, which is a contradiction, as $(s_{i,j}, a_{i,j})$ is an identifying state-action pair of MDPs i and j. □

Proposition 2. *For every $i \neq j$, denote the set of identifying state-action pairs by $\mathcal{I}_{i,j}$. If for every i and every j exists $(s_{i,j}, a_{i,j}) \in \mathcal{I}_{i,j}$ such that $P_{\mathcal{P}_i,\pi_\beta^i}(s_{i,j}, a_{i,j}) > 0$, then replacing π_β^i with π_r for all i results in identifiable data.*

Proof. Consider some $i \neq j$. We observe that by the construction of π_r, for every (s,a) pair that satisfies $P_{\mathcal{P}_i,\pi_\beta^i}(s,a) > 0$, we also have $P_{\mathcal{P}_i,\pi_r}(s,a) > 0$. In particular, we have $P_{\mathcal{P}_i,\pi_r}(s_{i,j}, a_{i,j}) > 0$.

We will show that either $(s_{i,j}, a_{i,j})$ also has positive probability under $(\mathcal{P}_j, \pi_r)$ or there must exist some other state-action pair that has positive probability under both $(\mathcal{P}_i, \pi_r)$ and $(\mathcal{P}_j, \pi_r)$. This, according to Proposition 1, will result in identifiability of the data.

We define the following sets of state-action pairs:

$$\Sigma_t^i = \{(s,a) : P_{\mathcal{P}_i,\pi_r,t}(s,a) > 0\}, \quad t = 0, 1, \ldots, T_{\max},$$
$$\Sigma_t^{i,j} = \left\{(s,a) : P_{\mathcal{P}_i,\pi_r,t}(s,a) = P_{\mathcal{P}_j,\pi_r,t}(s,a) > 0\right\}, \quad t = 0, 1, \ldots, T_{\max}.$$

Note that $\Sigma_0^i = \Sigma_0^{i,j}$, as the initial state distribution P_{init} and π_r are fixed across all MDPs.

First, consider the case where for every $t = 0, 1, \ldots, T_{\max}$ we have $\Sigma_t^i = \Sigma_t^{i,j}$. Given that $(s_{i,j}, a_{i,j})$ has positive probability under $(\mathcal{P}_i, \pi_r)$, there exists some t for which $(s_{i,j}, a_{i,j}) \in \mathcal{I}_{i,j} \cap \Sigma_t^i$. Since $\Sigma_t^i = \Sigma_t^{i,j}$, we have $(s_{i,j}, a_{i,j}) \in \mathcal{I}_{i,j} \cap \Sigma_t^{i,j}$, which means $(s_{i,j}, a_{i,j})$ also has positive probability under $(\mathcal{P}_j, \pi_r)$.

Next, consider the case where there exists some $t \in \{1, \ldots, T_{\max}\}$ for which $\Sigma_t^i \neq \Sigma_t^{i,j}$ and let $\hat{t} = \min\{t : \Sigma_t^i \neq \Sigma_t^{i,j}\}$. Note that $\hat{t} > 0$, since we have already shown that $\Sigma_0^i = \Sigma_0^{i,j}$. Thus, for every $t < \hat{t}$ we have $\Sigma_t^i = \Sigma_t^{i,j} = \Sigma_t^j$, and for $\hat{t}$ it holds that $P_{\mathcal{P}_i,\pi_r,\hat{t}}(s,a) \neq P_{\mathcal{P}_j,\pi_r,\hat{t}}(s,a)$. If there exits a $t' < \hat{t} - 1$ and $(s,a) \in \Sigma_{t'}^i$ such that $\mathcal{P}_i(\cdot|s,a) \neq \mathcal{P}_j(\cdot|s,a)$, then we are done as $\Sigma_{t'}^i = \Sigma_{t'}^{i,j}$, which means that (s,a) is an identifying state-action pair that has positive probability under both $(\mathcal{P}_i, \pi_r)$ and $(\mathcal{P}_j, \pi_r)$. Therefore, consider the case where for every $t < \hat{t} - 1$ and every $(s,a) \in \Sigma_t^i$ we have $\mathcal{P}_i(\cdot|s,a) = \mathcal{P}_j(\cdot|s,a)$. We will show that there exists $(s,a) \in \Sigma_{\hat{t}-1}^i$ such that $\mathcal{P}_i(\cdot|s,a) \neq \mathcal{P}_j(\cdot|s,a)$.

Assume to the contrary that for every $(s,a) \in \Sigma_{\hat{t}-1}^i$ we have $\mathcal{P}_i(\cdot|s,a) = \mathcal{P}_j(\cdot|s,a)$, i.e., the transition function is also equivalent for $t = \hat{t} - 1$. Let $h_{\hat{t}} = (x, s_0, a_0, \ldots, s_{\hat{t}}, a_{\hat{t}})$ be the state-action history up to time $\hat{t}$, including the random variable x that was used to choose a policy. We next

consider the probability of observing a history under $(\mathcal{P}_i, \pi_r)$,

$$\begin{aligned}P_{\mathcal{P}_i,\pi_r}(h_{\hat{t}}) =& P_{init}(s_0)P_x(x)\pi_r(a_0|x,s_0)P_{\mathcal{P}_i,\pi_r}(s_1|x,s_0,a_0)\pi_r(a_1|x,s_0,a_0,s_1)\cdots \\ & \cdots P_{\mathcal{P}_i,\pi_r}(s_{\hat{t}}|x,s_0,a_0,\ldots,s_{\hat{t}-1},a_{\hat{t}-1})\pi_r(a_{\hat{t}}|x,s_0,a_0,\ldots,s_{\hat{t}}) \\ =& P_{init}(s_0)P_x(x)\pi_r(a_0|x,s_0)\prod_{t=1}^{\hat{t}}\mathcal{P}_i(s_t|s_{t-1},a_{t-1})\pi_r(a_t|x,s_0,a_0,\ldots,s_t),\end{aligned}$$

where the last equality holds according to the Markov property, $P_{\mathcal{P}_i,\pi_r}(s_t|s_0,a_0,\ldots,s_{t-1},a_{t-1}) = \mathcal{P}_i(s_t|s_{t-1},a_{t-1})$. Since $\pi_r(a_t|x,s_0,a_0,\ldots,s_t)$ is the same replaying policy for all MDPs, and for every $t \leq \hat{t}-1$ and $(s,a) \in \Sigma_t^i$ we have $\mathcal{P}_i(\cdot|s,a) = \mathcal{P}_j(\cdot|s,a)$, then $P_{\mathcal{P}_i,\pi_r}(h_{\hat{t}}) = P_{\mathcal{P}_j,\pi_r}(h_{\hat{t}})$. By marginalizing over $x, s_0, a_0, \ldots, s_{\hat{t}-1}, a_{\hat{t}-1}$ we obtain:

$$\sum_{x,s_0,a_0,\ldots,s_{\hat{t}-1},a_{\hat{t}-1}} P_{\mathcal{P}_i,\pi_r}(x,s_0,a_0,\ldots,s_{\hat{t}},a_{\hat{t}}) = \sum_{x,s_0,a_0,\ldots,s_{\hat{t}-1},a_{\hat{t}-1}} P_{\mathcal{P}_j,\pi_r}(x,s_0,a_0,\ldots,s_{\hat{t}},a_{\hat{t}})$$

$$P_{\mathcal{P}_i,\pi_r}(s_{\hat{t}},a_{\hat{t}}) = P_{\mathcal{P}_j,\pi_r}(s_{\hat{t}},a_{\hat{t}}),$$

which means that $\Sigma_{\hat{t}}^i = \Sigma_{\hat{t}}^{i,j}$, which contradicts the definition of $\hat{t}$.

□

D Environments Description

In this section we describe the evaluation metric, data collection process, and the details of the domains we experimented with.

Evaluation Metric and Data collection

To evaluate performance, we measure average reward in the first few episodes on unseen tasks – this is where efficient exploration makes a critical difference. For Gridworld, we measure average reward in the first 4 episodes, for Wind only the first episode reward is measured, and for the rest of the domains, measure average performance in the first 2 episodes.

For data collection, we used off-the-shelf DQN (Gridworld) and SAC (continuous domains) implementations. The tasks are episodic, but we want the agent to maintain its belief between episodes, so that it can continually improve performance (see Figure 1). We follow Zintgraf et al. (2020), and aggregate k consecutive episodes of length H to a long trajectory of length $k \times H$, and we do not reset the hidden state in the VAE recurrent neural network after episode termination. For reward relabelling, we replace either the first or last $k/2$ trajectories with trajectories from a randomly chosen MDP, and relabel their rewards. For policy replay, we replace trajectories by sampling a new trajectory using the trained RL policy of another MDP.

Domains with Varying Reward Function

Gridworld: A 5×5 gridworld environment as in Zintgraf et al. (2020). The task distribution is defined by the location of a goal, which is unobserved and can be anywhere but around the starting state at the bottom-left cell. For each task, the agent receives a reward of -0.1 on non-goal cells and $+1$ at the goal, i.e.,

$$r_t = \begin{cases} 1, & s_t = g \\ -0.1, & \text{else,} \end{cases}$$

where s_t is the current cell and g is the goal cell.
Similarly to Zintgraf et al. (2020), the horizon for this domain is set to 15 and we aggregate $k = 4$ consecutive episodes to form a trajectory of length 60.

Semi-circle: A continuous 2D environment as in Figure 1, where the agent must navigate to an unknown goal, randomly chosen on a semi-circle of radius 1 (Rakelly et al., 2019). For each task, the agent receives a reward of $+1$ if it is within a small radius $r = 0.2$ of the goal, and 0 otherwise,

$$r_t = \begin{cases} 1, & \|x_t - x_{\text{goal}}\|_2 \leq r \\ 0, & \text{else,} \end{cases}$$

where x_t is the current 2D location. Action space is 2-dimensional and bounded: $[-0.1, 0.1]^2$.
We set the horizon to 60 and aggregate $k = 2$ consecutive episodes to form a trajectory of length 120.

MuJoCo:

1. **Half-Cheetah-Vel:** In this environment, a half-cheetah agent must run at a fixed target velocity. Following recent works in meta-RL (Finn et al., 2017; Rakelly et al., 2019; Zintgraf et al., 2020), we consider velocities drawn uniformly between 0.0 and 3.0. The reward in this environment is given by

$$r_t = -|v_t - v_{\text{goal}}| - 0.05 \cdot \|a_t\|_2^2$$

where v_t is the current velocity, and a_t is the current action. The horizon is set to 200 and we aggregate $k = 2$ consecutive episodes.

2. **Ant-Semi-circle:** In this environment, an ant needs to navigate to an unknown goal, randomly chosen on a semi-circle, similarly to the Semi-circle task above.

When collecting data for this domain, we found that the standard SAC algorithm (Haarnoja et al., 2018) was not able to solve the task effectively due to the sparse reward (which is described later), and did not produce trajectories that reached the goal. We thus modified the reward **only during data collection** to be dense, and inversely proportional to the distance from the goal,

$$r_t^{\text{dense}} = -\|x_t - x_{\text{goal}}\|_1 - 0.1 \cdot \|a_t\|_2^2$$

where x_t is the current 2D location and a_t is the current action. After collecting the data trajectories, we replaced all the dense rewards in the data with the sparse rewards that are given by

$$r_t^{\text{sparse}} = -0.1 \cdot \|a_t\|_2^2 + \begin{cases} 1, & \|x_t - x_{\text{goal}}\|_2 \leq 0.2 \\ 0, & \text{else.} \end{cases}$$

We note that Rakelly et al. (2019) use a similar approach to cope with sparse rewards in the online setting.
The horizon is set to 200 and we aggregate $k = 2$ consecutive episodes.

3. **Reacher-Image:** In this environment, a two-link planar robot needs to reach an unknown goal, randomly chosen on a quarter circle. The robot receives dense reward which is given by

$$r_t = -\|x_t - x_{\text{goal}}\|_2$$

where x_t is the location of the robot's end effector. The agent observes single-channel images of size 64×64 of the environment. The horizon is set to 100 and we aggregate $k = 2$ consecutive episodes.

DOMAINS WITH VARYING TRANSITION FUNCTION

Wind: A continuous 2D domain with varying transitions, where the agent must navigate to a fixed (unknown) goal within a distance of $D = 1$ from its initial state (the goal location is the same for all tasks). Similarly to Semi-circle, the agent receives a reward of $+1$ if it is within a radius $r = 0.2$ of the goal, and 0 otherwise,

$$r_t = \begin{cases} 1, & \|s_t - s_{\text{goal}}\|_2 \leq r \\ 0, & \text{else.} \end{cases}$$

For each task in this domain, the agent is experiencing a different 'wind', which results in a shift in the transitions, such that when taking action $a_t \in [-0.1, 0.1]^2$ from state s_t in MDP $\mathcal{M}$, the agent transitions to a new state s_{t+1}, which is given by

$$s_{t+1} = s_t + a_t + w_{\mathcal{M}},$$

where $w_{\mathcal{M}}$ is a task-specific wind, which is randomly drawn for each task from a uniform distribution over $[-0.05, 0.05]^2$. To navigate correctly to the goal and stay there, the agent must take actions that cancel the wind effect.
We set the horizon to 25 and evaluate the performance in terms of average return within the **first** episode of interaction on test tasks ($k = 1$).

Escape-Room: A continuous 2D domain where the agent must navigate outside a circular room of radius $R = 1$ through an opening, whose location is unknown. For all tasks, the central angle of the opening is $\pi/8$. The tasks differ by the location of the opening – the center point of the opening is sampled uniformly from $[0, \pi]$. The reward function is sparse, task-independent, and given by

$$r_t = \begin{cases} 1, & \|s_t\|_2 > R \\ 0, & \text{else.} \end{cases}$$

The transition function, however, is task-dependent and given by

$$s_{t+1} = \begin{cases} \frac{s_t + a_t}{\|s_t + a_t\|_2}, & \text{if } \textit{intersection occurs} \\ s_t + a_t, & \text{else,} \end{cases}$$

where *intersection occurs* means that the line that connects s_t and $s_t + a_t$ and the wall of the circular room intersects. To solve a task, the agent must search for the opening by moving along the wall until he finds it. We set the horizon to 60 and aggregate $k = 2$ consecutive episodes to a form a trajectory of length 120.

In Figure 5 we compare our offline algorithm with PEARL (Rakelly et al., 2019), and also with an ablation of the policy replaying method. In the Escape-Room domain, where the identifying states do not overlap, policy replaying indeed improves performance, while in Wind, where the identifying states do overlap, policy replaying has little effect.

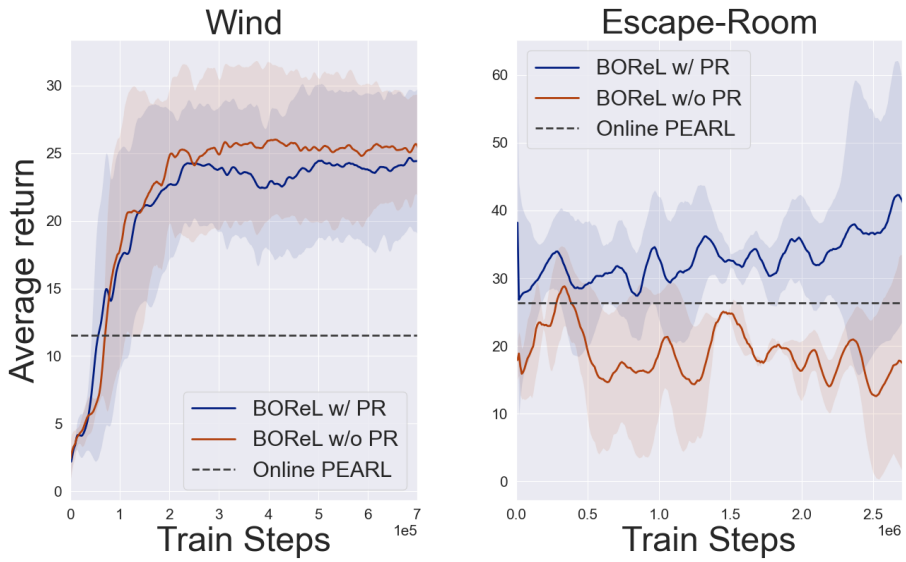


Figure 5: Offline performance on domains with varying transitions. We compare BOReL instantiated with and without policy replaying (blue and red, respectively) with online PEARL.

E Learned Belief Visualization

In this section we visualize the learned belief states in Semi-circle domain, in order to get more insight into the decision making process of the agent during interaction.

In Figure 6, we plot the reward belief (obtained from the VAE decoder) at different steps during the agent's interaction in the Semi-circle domain. Note how the belief starts as uniform over the semi-circle, and narrows in on the target as more evidence is collected. Also note that without reward relabelling, the agent fails to find the goal. In this instance of the MDP ambiguity problem, the training data for the meta-RL agent consists of trajectories that mostly reach the goal, and as a result, the agent believes that the reward is located at the first point it reaches on the semi-circle.

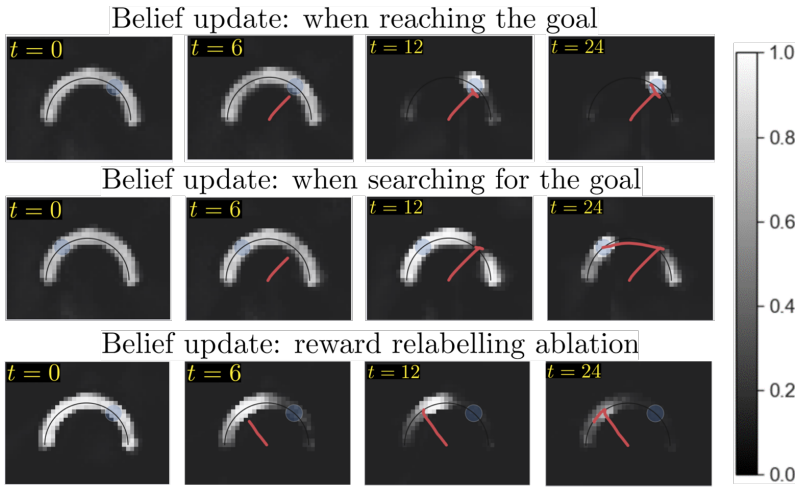
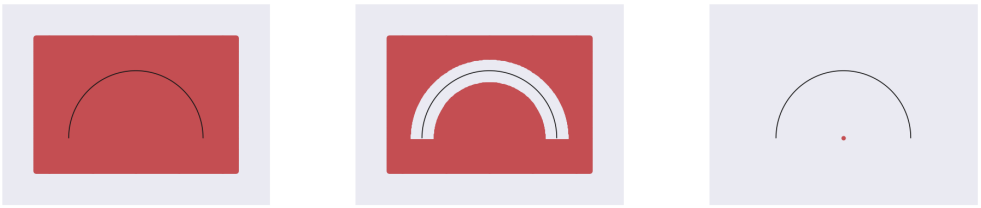


Figure 6: Semi-circle belief visualization. The plots show the reward belief over the 2-dimensional state space (obtained from the VAE) at different stages of interacting with the system. The red line marks the agent trajectory, and the light blue circle marks the true reward location. **Top:** Once the agent finds the true goal, it reduces the belief over other possible goals from the task distribution. **Middle:** As long as the agent doesn't find the goal, it explores efficiently, reducing the uncertainty until the goal is found. **Bottom:** Without reward relabelling, the agent doesn't learn to differentiate between different MDPs, and therefore fails to identify the goal.

F Data Quality Ablation

In our data quality ablative study, we consider the Ant-Semi-circle domain for which we modify the initial state distribution during the data collection phase. We diversified the offline dataset by modifying the initial state distribution P_{init} to either (1) uniform over a large region, (2) uniform over a restricted region (excluding state on the semi-circle), or (3) fixed to a single position. At meta-test time, only the single fixed position is used. The initial state distributions we consider are visualized in Figure 7.



(a) Uniform (b) Excluding s.c. (c) Fixed

Figure 7: Initial state distributions. Red locations indicate non-zero sampling probability.

We report results for the 3 different data collection strategies described above, summarized in Table 1. As expected, data diversity is instrumental to offline training. However, as we qualitatively show in Figure 8, even on the low-diversity datasets, our agents learned non-trivial exploration strategies that search for the goal. This is especially remarkable for the fixed-distribution dataset, where it is unlikely that any training trajectory traveled along the semi-circle.

	BOReL	BOReL+CQL
Uniform	171.8 ± 7.0	176.0 ± 10.2
Excluding s.c.	102.8 ± 32.7	116.6 ± 19.9
Fixed	99.2 ± 27.4	112.4 ± 31.3

Table 1: Average return in Ant-Semi-circle for different initial state distributions during offline data collection: **Uniform** distribution, uniform distribution excluding states on the semi-circle (**Excluding s.c.**), and fixed initial position (**Fixed**).

One may ask whether OMRL presents the same challenge as standard offline RL, and whether recent offline RL advances can mitigate the dependency on data diversity. To investigate this, we also

compare our method with a variant that uses CQL (Kumar et al., 2020) – a state-of-the-art offline RL method – to train the critic network of the meta-RL agent. Interestingly, while CQL improved results (Table 1), the data diversity is much more significant. Together with our results on MDP ambiguity, our investigation highlights the particular challenges of the OMRL problem.

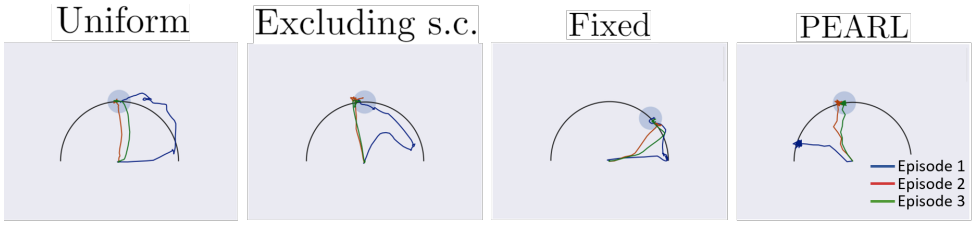


Figure 8: Ant-Semi-circle: trajectories of trained agents for different offline datasets and for PEARL.