BAYESIAN RESIDUAL POLICY OPTIMIZATION: SCALABLE BAYESIAN REINFORCEMENT LEARNING WITH CLAIRVOYANT EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Informed and robust decision making in the face of uncertainty is critical for robots that perform physical tasks alongside people. We formulate this as a Bayesian Reinforcement Learning problem over latent Markov Decision Processes (MDPs). While Bayes-optimality is theoretically the gold standard, existing algorithms do not scale well to continuous state and action spaces. We propose a scalable solution that builds on the following insight: in the absence of uncertainty, each latent MDP is easier to solve. We split the challenge into two simpler components. First, we obtain an ensemble of clairvoyant experts and fuse their advice to compute a baseline policy. Second, we train a Bayesian residual policy to improve upon the ensemble's recommendation and learn to reduce uncertainty. Our algorithm, Bayesian Residual Policy Optimization (BRPO), imports the scalability of policy gradient methods as well as the initialization from prior models. BRPO significantly improves the ensemble of experts and drastically outperforms existing adaptive RL methods.

1 INTRODUCTION

Robots operating in the real world must resolve uncertainty on a daily basis. Often times, a robot is uncertain about how the world around it evolves. For example, a self-driving car must drive safely around unpredictable actors like pedestrians and bicyclists. A robot arm must reason about occluded objects when reaching into a cluttered shelf. On other occasions, a robot is uncertain about the task it needs to perform. An assistive home robot must infer a human's intended goal by interacting with them. Both examples of uncertainty require simultaneous inference and decision making, which can be framed as Bayesian reinforcement learning (RL) over latent Markov Decision Processes (MDPs). Agents do not know which latent MDP they are interacting with, preventing them from acting optimally with respect to that MDP. Instead, *Bayes optimality* only requires that agents be optimal with respect to their current uncertainty over latent MDPs.

The Bayesian RL problem can be viewed as solving a large continuous belief MDP, which is computationally infeasible to solve directly (Ghavamzadeh et al., 2015). We build upon a simple yet recurring observation (Osband et al., 2013; Kahn et al., 2017; Choudhury et al., 2018): while solving the belief MDP may be hard, solving individual latent MDPs is much more tractable. Given exact predictions for all actors, the self-driving car can invoke a motion planner to find a collision-free path. The robot arm can employ an optimal controller to dexterously retrieve an object given exact knowledge of all objects. Once the human's intended goal is discovered, the robot can provide assistance. Hence, the overall challenge boils down to solving two (perhaps) simpler sub-challenges: solving the latent MDPs and combining these solutions to solve the belief MDP.

Let's assume we can approximately solve the latent MDPs to create an ensemble of policies as shown in Figure 1. We can think of these policies as *clairvoyant experts*, i.e., experts that think they know the latent MDP and offer advice accordingly. A reasonable strategy is to weigh these policy proposals by the belief and combine them into a single recommendation to the agent. While this recommendation is good for some regimes, it can be misleading when uncertainty is high. The onus then is on the agent to disregard the recommendation and explore the space effectively to collapse uncertainty. This leads to our key insight.

> Learning Bayesian corrections on top of clairvoyant experts is a scalable strategy for solving complex reinforcement learning problems.

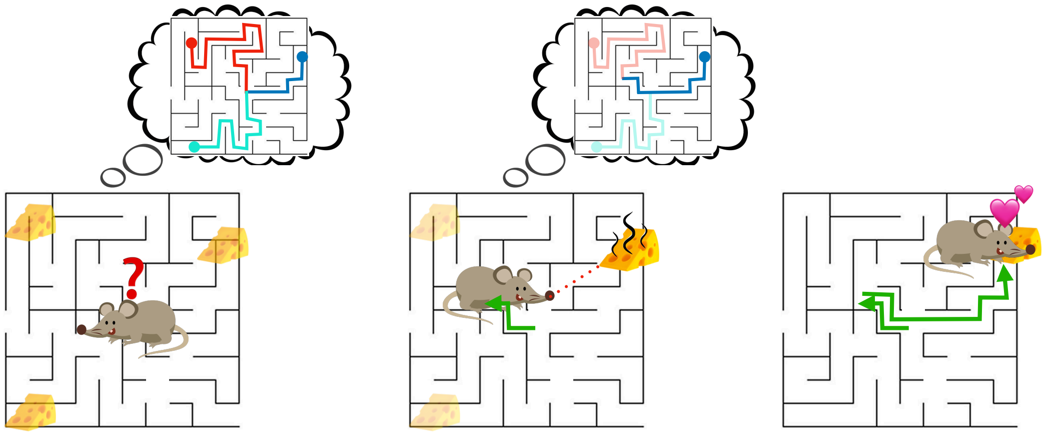


Figure 1: An overview of Bayesian Residual Policy Optimization. (a) Cheese location is unknown and expert proposals disagree about which direction to move in. (b) The Bayesian residual policy learns to smell for cheese, reducing uncertainty. (c) The experts' recommendation guides the agent to the cheese! BRPO balances exploration (Bayesian residual policy) with exploitation (expert recommendations).

While learning corrections echoes the philosophy of *boosting* (Freund & Schapire, 1999), our agent goes one step beyond: it learns to take uncertainty-reducing actions that highlight which expert to boost.

Our algorithm, Bayesian Residual Policy Optimization (BRPO), augments a belief-space batch policy optimization algorithm (Lee et al., 2019) with clairvoyant experts (Figure 1). The agent observes the experts' recommendation, belief over the latent MDPs, and state. It returns a correction over the expert proposal, including uncertainty-reducing sensing actions that experts never need to take.

Our key contribution is the following:

- We propose a scalable Bayesian RL algorithm to solve problems with complex latent rewards and dynamics.
- We experimentally demonstrate that BRPO outperforms both the ensemble of experts and existing adaptive RL algorithms.

2 Related Work

Belief-Space RL Methods Bayesian reinforcement learning formalizes RL where one has a prior distribution over possible MDPs (Ghavamzadeh et al., 2015; Shani et al., 2013). However, the Bayes-optimal policy, which is the best one can do under uncertainty, is intractable to solve for and approximation is necessary (Hsu et al., 2008). One way is to approximate the value function, as done in SARSOP (Kurniawati et al., 2008) and PBVI (Pineau et al., 2003); however, they cannot deal with continous state actions. Another strategy is to resort to sampling, such as BAMCP (Guez et al., 2012), POMCP (Silver & Veness, 2010), POMCPOW (Sunberg & Kochenderfer, 2018). However, these approaches require a significant amount of online computation.

Online approaches forgo acting Bayes-optimally right from the onset, and instead aim to eventually act optimally. The question then becomes: how do we efficiently gain information about the test time MDP to act optimally? BEB (Kolter & Ng, 2009) and POMDP-lite (Chen et al., 2016) introduce an auxiliary reward term to encourage exploration and prove Probably-Approximately-Correct (PAC) optimality. This has inspired work on more general, non-Bayesian curiosity based heuristics for reward gathering (Achiam & Sastry, 2017; Burda et al., 2018; Pathak et al., 2017; Houthooft et al., 2016). Online exploration is also well studied in the bandit literature, and techniques such as posterior sampling (Osband et al., 2019) bound the learner's regret. UP-OSI (Yu et al., 2017) predicts the most likely MDP and maps that to an action. Gimelfarb et al. (2018) learns a gating over multiple expert value functions. However, online methods can over-explore and drive the agent to unsafe regimes.

Another alternative is to treat belief MDP problems as a large state space that must be compressed. Peng et al. (2018) use Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) to encode a history of observations to generate an action. Methods like BPO (Lee et al., 2019) explicitly utilize the belief distribution and compress it to learn a policy. The key difference between BRPO and BPO is that BRPO uses an expert, enabling it to scale to handle complex latent tasks that may require multimodal policies.

Meta-reinforcement Learning Meta-reinforcement learning (MRL) approaches train sample-efficient learners by exploiting structure common to a distribution of MDPs. For example, MAML (Finn et al., 2017) trains gradient-based learners while RL2 (Duan et al., 2016) trains memory-based learners. While meta-supervised learning has well established Bayesian roots (Baxter, 1998; 2000), it wasn't until recently that meta-reinforcement learning was strongly tied to Bayesian Reinforcement Learning (BRL) (Ortega et al., 2019; Rabinowitz, 2019). Nevertheless, even non-Bayesian MRL approaches address problems pertinent to BRL. MAESN (Gupta et al., 2018) learns structured noise for exploration. E-MAML (Stadie et al., 2018) adds an explicit exploration bonus to the MAML objective. GMPS (Mendonca et al., 2019) exploit availability of MDP experts to partially reduce BRL to IL. Our work is more closely related to Bayesian MRL approaches. MAML-HB (Grant et al., 2018) casts MAML as hierarchical Bayes and improves posterior estimates. BMAML (Yoon et al., 2018) uses non-parametric variational inference to improve posterior estimates. PLATIPUS (Finn et al., 2018) learns a parameter distribution instead of a fixed parameter. PEARL (Rakelly et al., 2019) learns a data-driven Bayes filter across tasks. In contrast to these approaches, we use experts at test time, learning only to optimally correct them.

Residual Learning Residual learning has its foundations in boosting (Freund & Schapire, 1999) where a combination of weak learners, each learning on the failures of previous, make a strong learner. It also allows for injecting priors in RL, by boosting off of hand-designed policies or models. Prior work has leveraged known but approximate models by learning the residual between the approximate dynamics and the discovered dynamics (Ostafew et al., 2014; 2015; Berkenkamp & Schoellig, 2015). There has also been work on learning residual policies over hand-defined ones for solving long horizon (Silver et al., 2018) and complex control tasks (Johannink et al., 2019). Similarly, our approach starts with a useful initialization (via experts) and learns to improve via Bayesian reinforcement learning.

3 Preliminaries: Bayesian Reinforcement Learning

In Bayesian reinforcement learning, the agent does not know the reward and transition functions but knows that they are determined by a latent variable $\phi \in \Phi$. Formally, the problem is defined by a tuple $\langle S, \Phi, A, T, R, P_0, \gamma \rangle$, where S is the observable state space of the underlying MDPs, Φ is the latent space, and A is the action space. T and R are the transition and reward functions parameterized by ϕ. The initial distribution over (s, ϕ) is given by $P_0 : S \times \Phi \to \mathbb{R}^+$, and γ is the discount.

Bayesian RL considers the long-term expected reward with respect to the uncertainty over ϕ rather than the true (unknown) value of ϕ. Uncertainty is represented as a *belief distribution* $b \in B$ over latent variables ϕ. The Bayes-optimal action value function is given by the Bellman equation:

$$Q(s, b, a') = R(s, b, a') + \gamma \sum_{s', b'} P(s'|s, b, a') P(b'|s, b, a') \max_{a''} Q(s', b', a'') \tag{1}$$

The Bayesian reward function is the expected reward $R(s, b, a') = \sum_{\phi \in \Phi} b(\phi) R(s, \phi, a')$. The Bayesian transition function is $P(s'|s, b, a') = \sum_{\phi \in \Phi} b(\phi) P(s'|s, \phi, a')$. The posterior update $P(b'|s, b, a')$ is computed recursively, starting from initial belief b_0.

$$b'(\phi'|s, b, a', s') = \eta \sum_{\phi \in \Phi} b(\phi) T(s, \phi, a', s', \phi') \tag{2}$$

where η is the normalizing constant, and the transition function is defined as $T(s, \phi, a', s', \phi') = P(s', \phi'|s, \phi, a') = P(s'|s, \phi, a') P(\phi'|s, \phi, a', s')$. At timestep t, the belief $b_t(\phi_t)$ is the posterior distribution over Φ given the history of states and actions, $(s_0, a_1, s_1, ..., s_t)$. When ϕ corresponds to physical parameters for an autonomous system, we often assume that the latent states are fixed.

Algorithm 1 Bayesian Residual Policy Optimization

Require: Bayes filter ψ, belief b_0, prior P_0, residual policy π_{θ_0}, expert π_e, horizon H, n_{itr}, n_{sample}

for $i = 1, 2, \cdots, n_{\text{itr}}$ **do**
    for $n = 1, 2, \cdots, n_{\text{sample}}$ **do**
        Sample latent MDP M: $(s_0, \phi_0) \sim P_0$
        $\tau_n \leftarrow \texttt{Simulate}(\pi_{\theta_{i-1}}, \pi_e, b_0, \psi, M, H)$
    Update policy: $\theta_i \leftarrow \texttt{BatchPolicyOptimization}(\theta_{i-1}, \{\tau_1, \cdots, \tau_{n_{\text{sample}}}\})$
return $\pi_{\theta_{best}}$

procedure SIMULATE($\pi_\theta, \pi_e, b_0, \psi, M, H$)
    for $t = 1, \cdots, H$ **do**
        $a_{e_t} \sim \pi_e(\cdot|s_{t-1}, b_{t-1})$ // Expert recommendation
        $a_t \leftarrow a_{r_t} + a_{e_t}, \quad a_{r_t} \sim \pi_\theta(s_{t-1}, b_{t-1}, a_{e_t})$ // Residual action
        Execute a_t on M, observe r_t, s_t
        $b_t \leftarrow \psi(s_{t-1}, b_{t-1}, a_t, s_t)$
    $\tau \leftarrow (s_0, b_0, a_{r_1}, r_1, s_1, b_1, \cdots, a_{r_H}, r_H, s_H, b_H)$ // Only residuals are recorded
    return τ

Our algorithm utilizes a black-box Bayes filter to produce a posterior distribution over the latent states. However, a Bayes filter can also be interpreted as a function that compresses the history of states and actions. Recent work suggests that Long Short-Term Memory (LSTM) cells (Hochreiter & Schmidhuber, 1997) can be meta-trained to compress history and predict subsequent states (Ortega et al., 2019). Such learned representations can be substituted for the belief distribution that we have chosen here.

4 Bayesian Residual Policy Optimization (BRPO)

In Bayesian Residual Policy Optimization, we first construct an ensemble of clairvoyant experts where each approximately solves a latent MDP. Expert proposals are gated with the belief over MDPs to compute a recommendation. We then train a Bayesian residual policy to correct the recommendation, resulting in an elegant exploration-exploitation tradeoff. The agent learns to produce smaller corrections when the recommendation is effective, i.e. when uncertainty is small or when all clairvoyant experts agree with a good recommendation. Otherwise, the agent overrides the recommendation and learns to explore effectively.

4.1 Ensemble of Clairvoyant Experts

For simplicity of exposition, assume the Bayesian RL problem consists of k underlying latent MDPs, $\phi_1, ..., \phi_k$. Clairvoyant experts π_i can be computed for each ϕ_i via single-MDP RL methods (or optimal control, if transition and reward functions are known). If $b(\phi_i)$ is the posterior belief over each MDP ϕ_i, we want to combine experts to construct a belief-aware recommendation that maps the state and belief to a distribution over actions $\pi_e : S \times B \rightarrow P(A)$.

One choice for π_e is to select the maximum a posteriori (MAP) action.

$$a_{\text{MAP}} = \arg\max_a \sum_{i=1}^{k} b(\phi_i)\pi_i(a|s) \tag{3}$$

However, computing the MAP estimate may require optimizing a non-convex function, e.g., when the distribution is multimodal. We can instead maximize the lower bound using Jensen's inequality.

$$\log \sum_{i=1}^{k} b(\phi_i)\pi_i(a|s) \geq \sum_{i=1}^{k} b(\phi_i) \log \pi_i(a|s) \tag{4}$$

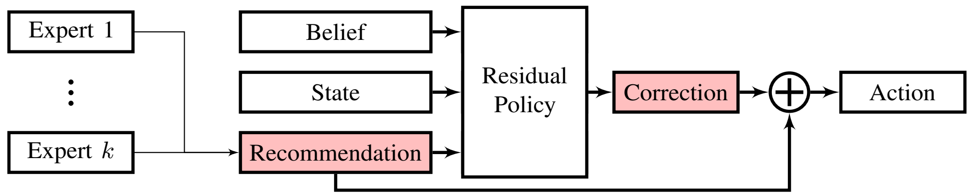


Figure 2: Bayesian residual policy network architecture.

This is much easier to solve, especially if $\log \pi_i(a|s)$ is convex. If each $\pi_i(a|s)$ is a Gaussian with mean μ_i and covariance Σ_i, e.g. from TRPO (Schulman et al., 2015), the resultant action is

$$a^* = \arg\max_a \sum_{i=1}^{k} b(\phi_i) \log \pi_i(a|s) = \left[\sum_{i=1}^{k} b(\phi_i)\Sigma_i^{-1}\right]^{-1} \sum_{i=1}^{k} b(\phi_i)\Sigma_i^{-1}\mu_i \tag{5}$$

When the belief has collapsed to one ϕ_i, the resulting ensemble recommendation follows the corresponding π_i exactly. Thus, as entropy reduces, the ensemble is more reliable.

There are other alternatives to consider. One choice for π_e is to directly use the mixture model $\sum_{i=1}^{k} b(\phi_i)\pi_i(a|s)$. This would be equivalent to posterior sampling (Osband et al., 2013).

While this belief-aware ensemble is easy to attain, it is not Bayes-optimal. In particular, since the clairvoyant experts do not take explicit uncertainty-reducing actions, the ensemble will not recommend to do so. Consider the `Maze4` example: each clairvoyant expert knows its corresponding latent MDP's hidden goal position, and thus navigates optimally without sensing. A Bayes-optimal agent, on the other hand, would take sensing actions to identify the latent goal.

4.2 Residual Policy Learning

In each training iteration, BRPO collects trajectories by simulating the current policy on several MDPs sampled from the prior distribution (Algorithm 1). At every timestep of the simulation, the ensemble is queried for an action recommendation, which is summed with the correction from the residual policy network (Figure 2) and executed. The Bayes filter updates the posterior after observing the resulting state. The collected trajectories (with only residual actions) are the input to a policy optimization algorithm (Schulman et al., 2015; 2017) which updates the residual policy network.

The residual policy does not solve the original belief MDP. Since its corrective actions are summed with the ensemble's recommendations, it in fact operates in a residual belief MDP (with respect to the recommendations). Actions are simply shifted by the recommendations. That is, for every residual action $a_{r'}$ and expert recommendation $a_{e'} \sim \pi_e(\cdot|s,b)$, we can define a new transition dynamics $\tilde{T}$ from the original T:

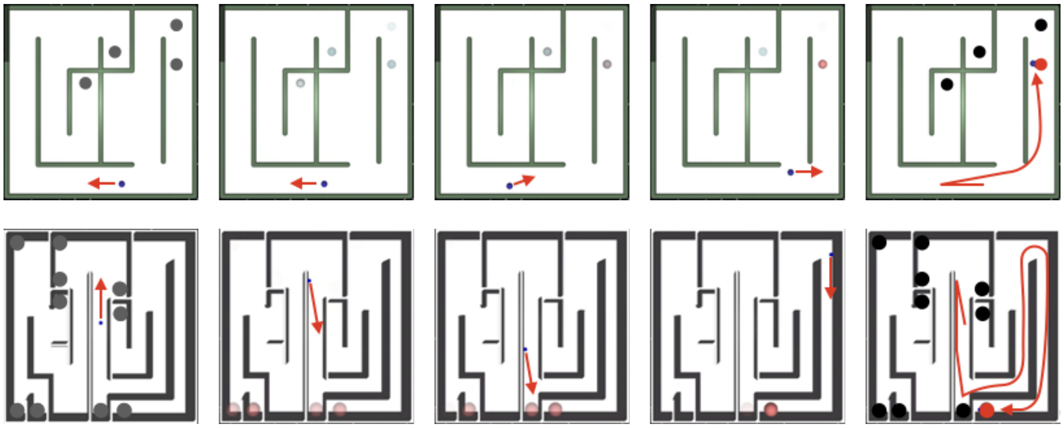
$$P_{\tilde{T}}(s', b', r'|s, b, a_{r'}) = \int_{a_{e'} \in A} P_T(s', b', r'|s', b', a_{r'} + a_{e'})\pi_e(a_{e'}|s, b) \tag{6}$$

This new transition function $\tilde{T}$ defines the residual belief MDP. Since the experts are fixed, the residual belief-MDP is also fixed during training and testing time. Thus, this strategy inherits all mathematical guarantees from the underlying policy optimization algorithm, such as monotonic improvement from the ensemble's baseline policy.

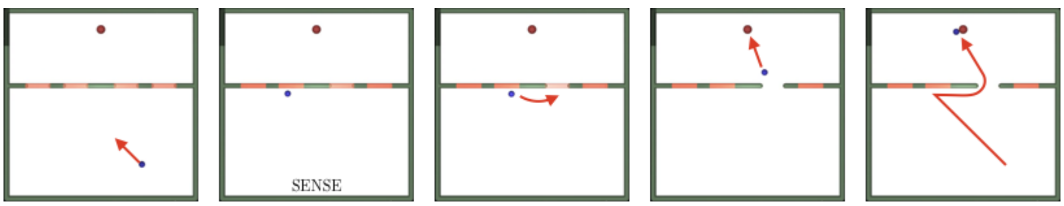
5 Experimental Results

We choose problems that highlight common challenges in robotics:

- Explicit or implicit sensing actions are required to infer the latent MDP.
- Sensing is costly, and different sensing actions may have different costs.
- Solutions for each latent MDP are significantly different.



(a) Latent goal mazes with four (**Maze4**) and ten (**Maze10**) possible goals. The agent senses as it navigates, changing its direction as goals are deemed less likely (more transparent). We have marked the true goal with red in the last frame for clarity.



(b) **Door4**. The agent senses only when it is near the wall with doors, where sensing is most accurate. The transparency of the red bars indicates the posterior probability that the door is blocked. With sensing, the third door becomes more likely to be open while the others become more likely to be closed.

Figure 3: BRPO policy keyframes. Best viewed in color.

In all domains that we consider, BRPO improves on the ensemble's recommendation, learning to sense in a cost-effective manner. As seen in `Door4`, BRPO also can develop more effective strategies for both sensing and control.

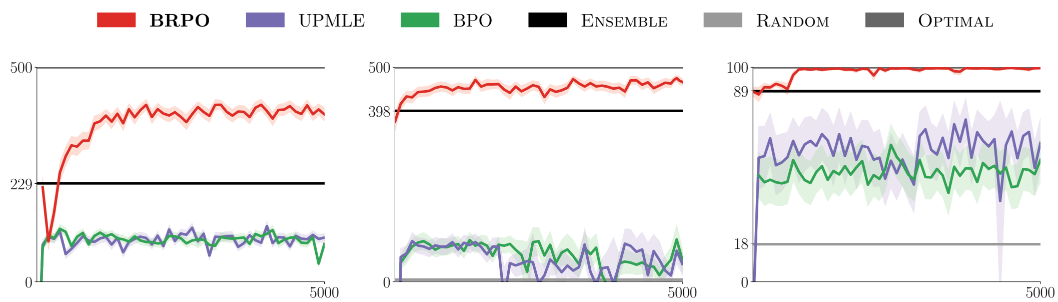
Latent Goal Mazes In the **Maze4** and **Maze10** environments, the agent must identify which latent goal is active. This is an example of where the dynamics are the same across all latent MDPs, but the task must be inferred.

At the beginning of each episode, the latent goal is set to one of four goals (`Maze4`) or ten goals (`Maze10`). This problem requires explicit sensing to distinguish the goal. Sensing can happen simultaneously as the agent moves, but costs -1; the agent receives a noisy measurement of the distance to the goal, with noise proportional to the true distance. This motivates the agent to minimize sensing and sense when closer to goals to obtain better measurements.

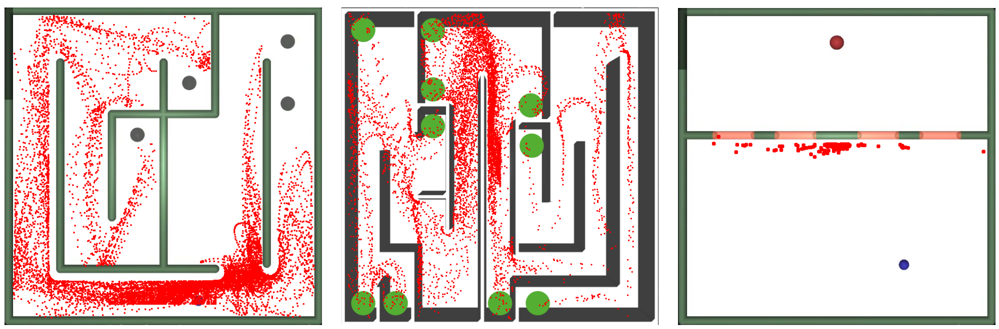
After each action, the agent observes its current position, velocity, and distance to all latent goals. If sensing is invoked, it also observes the noisy distance to the goal. In addition, the agent observes the categorical belief distribution over the latent goals and the ensemble's recommendation. Each expert proposes an action (computed via motion planning) that navigates to the corresponding goal.

In `Maze4`, reaching the active goal provides a terminal reward of 500, while reaching an incorrect goal gives a penalty of -500. The task ends when the agent receives either the terminal reward or penalty, or after 500 timesteps. In `Maze10`, the agent receives a penalty of -50 and continues to explore after reaching an incorrect goal.

Figure 3a demonstrates rollouts by the trained BRPO agents on `Maze4` and `Maze10`. In `Maze10`, goals that are near each other have drastically different paths to them, making task inference even



(a) Training curves. BRPO dramatically outperforms agents that do not leverage expert knowledge, and significantly improves the ensemble of experts.



(b) Sensing locations. In `Maze4` and `Maze10`, sensing is dense around the starting regions (the bottom row in `Maze4` and center in `Maze10`) and in areas where multiple latent goals are nearby. The agent sometimes reroutes before reaching an incorrect goal. In `Door4`, BRPO only senses when close to the doors, where the sensor is most accurate.

Figure 4: BRPO performance on `Maze4`, `Maze10`, and `Door4` (left to right).

more important. For both `Maze4` and `Maze10`, we see that the agent reroutes itself (multiple times in `Maze10`) while it invokes sensing to get a better belief.

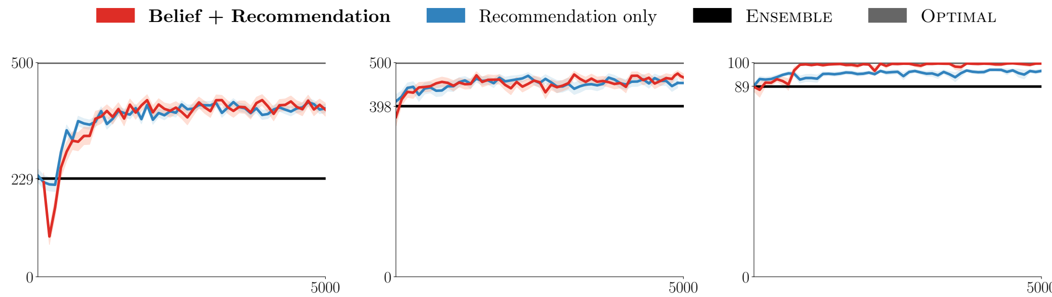
Doors In this more classical POMDP problem, there are 4 possible doors to the goal in the next room. At the beginning of each episode, each door is opened or closed with 0.5 probability. To check the doors, the agent can either sense (-1) or crash into them (-10). As with the mazes, the sense action can be taken simultaneously as the agent moves. Sensing returns a noisy binary vector for all four doors, with exponentially decreasing accuracy proportional to the distance to each door. Crashing returns an accurate indicator of the door it crashed into. At every step, the agent observes its position, velocity, distance to goal, and whether it crashed or passed through a door. In addition, the agent observes the categorical distribution over the $2^4 = 16$ possible door configurations (from the Bayes filter) and the ensemble's recommendation. The agent receives a terminal reward of 100 if it reaches the goal within 300 timesteps.

We observe that BRPO's learned policy is quite different from any of the experts. Each expert navigates directly through the closest open door. BRPO gets very close to the wall (to minimize sensor noise) and senses while sliding along the wall, before identifying an open door and navigating through it.

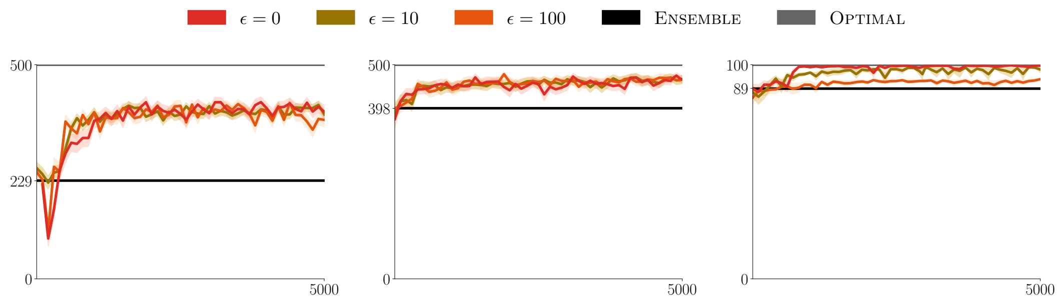
5.1 BRPO Outperforms Adaptive RL Methods

We compare BRPO to adaptive RL algorithms that consider the belief over latent states: BPO (Lee et al., 2019) and UP-MLE, a modification to Yu et al. (2017) that uses the most likely estimate from the Bayes filter[1]. We also compare with the ensemble of experts baseline, with one key difference.

[1]This was originally introduced in Lee et al. (2019).



(a) Including both belief and recommendation as policy inputs results in faster learning in `Door4`.



(b) Information-gathering reward bonuses (Equation 7) are unnecessary with BRPO. Large bonuses can cause the agent to ignore penalties from the environment, resulting in suboptimal performance (`Door4`).

Figure 5: Residual policy input and reward bonus experiments on `Maze4`, `Maze10`, and `Door4` (left to right).

The ensemble will not take any sensing actions (as discussed in Section 4), so we strengthen it by sensing with probability 0.5 at each timestep. This is equivalent to the initial BRPO policy, which adds random noise to the ensemble recommendation.

Figure 4a compares the training performance of all algorithms across the three environments. (Note that OPTIMAL is unachievable, since it requires full knowledge of the latent MDP.) In RANDOM, the agent randomly chooses one of the clairvoyant experts to follow for the entire episode.

BRPO agents dramatically outperforms BPO and UP-MLE agents. In fact, we have trained BPO and UP-MLE with an additional boost to encourage information-gathering (Section 5.3); without such bonuses, they did not learn to take any meaningful behavior. Even with the bonus, these agents learns to solve the task only partially. In `Maze4` and `Maze10`, they only reach some of the goals. In `Door4`, they only learn to navigate through one of the first two doors and will occasionally crash.

Examining where sensing has happened (Figure 4b), we see that the BRPO agent learns to sense when goals must be distinguished, and uses the belief to reroute itself in `Maze4` and `Maze10`. Qualitatively, we find that UP-MLE relies exclusively on crashing into doors to reduce uncertainty, which is extremely costly. The BRPO agent avoids crashing in almost all scenarios.

5.2 RESIDUAL POLICY INPUTS

The BRPO policy takes the belief distribution, state, and ensemble recommendation as inputs (Figure 2). However, since the ensemble recommendation implicitly includes the belief, the belief may not be a necessary input to the policy if the recommendation is already provided.

The results show that providing both belief and recommendation as inputs to the policy are important (Figure 5a). Although BRPO with only the recommendation performs comparably to BRPO with both inputs on `Maze4` and `Maze10`, the one with both inputs produce faster learning on `Door4`.

5.3 Information-Gathering Reward Bonuses

Because BRPO maximizes the Bayesian Bellman equation (Equation 1), exploration is incorporated into its long-term objective. As a result, auxiliary rewards to encourage exploration are unneeded. However, existing work that does not explicitly consider the belief has suggested various auxiliary reward terms that encourage exploration, such as intrinsic rewards (Pathak et al., 2017) or surprisal rewards (Achiam & Sastry, 2017). To investigate whether such rewards benefit the BRPO agent, we augment the reward function with the following auxiliary bonus:

$$\tilde{r}(s_t, b_t, a_t) = r(s_t, b_t, a_t) + \epsilon \cdot \mathbb{E}_{b_{t+1}} \left[\|b_t - b_{t+1}\|_1 \right] \tag{7}$$

where $\|b_t - b_{t+1}\|_1 = \sum_{i=1}^{k} |b_t(\phi_i) - b_{t+1}(\phi_i)|$ rewards change in belief.[2]

Figure 5b summarizes the performance of BRPO when training with $\epsilon = 0, 10, 100$. Too much emphasis on information-gathering causes the agent to over-explore and therefore underperform. In `Door4` with $\epsilon = 100$, we qualitatively observe that the agent crashes into the doors more often. This is because crashing significantly changes the belief for that door; the huge reward bonus outweighs the penalty of crashing from the environment.

We find that BPO and UP-MLE are unable to learn without an exploration bonus. We used $\epsilon = 1$ for `Maze4` and `Door4`, and $\epsilon = 100$ for `Maze10`. With the bonus, both BPO and UP-MLE learn to sense initially but struggle to solve the challenging latent MDPs.

6 Discussion and Future Work

In the real world, robots must deal with uncertainty, either due to complex latent dynamics or task specifics. Because uncertainty is an inherent part of these tasks, we can at best aim for optimality under uncertainty, i.e., Bayes optimality. Existing BRL algorithms or POMDP solvers do not scale well to problems with complex latent MDPs or a large (continuous) set of MDPs.

We decompose BRL problems into two parts: solving each latent MDP and being Bayesian over the solutions. Our algorithm, Bayesian Residual Policy Optimization, operates on the residual belief-MDP space given an ensemble of experts. BRPO focuses on learning to explore, relying on the experts for exploitation. BRPO is capable of solving complex problems, outperforming existing BRL algorithms and improving on the original ensemble of experts.

Although out of scope for this work, a few key challenges remain. First is an efficient construction of an ensemble of experts, which becomes particularly important for continuous latent spaces with infinitely many MDPs. Infinitely many MDPs do not necessarily require infinite experts, as many may converge to similar policies. An important future direction is subdividing the latent space and computing a qualitatively diverse set of policies (Liu et al., 2016). Another challenge is developing an efficient Bayes filter, which is an active research area. In certain occasions, the dynamics of the latent MDPs may not be accessible, which would require a learned Bayes filter. Combined with a tractable, efficient Bayes filter and an efficiently computed set of experts, we believe that BRPO will provide an even more scalable solution for BRL problems.

References

Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017.

Jonathan Baxter. Theoretical models of learning to learn. In *Learning to learn*, pp. 71–94. Springer, 1998.

Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12: 149–198, 2000.

Felix Berkenkamp and Angela P Schoellig. Safe and robust learning control with gaussian processes. In *2015 European Control Conference (ECC)*, pp. 2496–2501. IEEE, 2015.

[2] An analogous term has been introduced in Chen et al. (2016).

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Min Chen, Emilio Frazzoli, David Hsu, and Wee Sun Lee. POMDP-lite for robust robot planning under uncertainty. In *IEEE International Conference on Robotics and Automation*, 2016.

Sanjiban Choudhury, Mohak Bhardwaj, Sankalp Arora, Ashish Kapoor, Gireeja Ranade, Sebastian Scherer, and Debadeepta Dey. Data-driven planning via imitation learning. *The International Journal of Robotics Research*, 37(13-14):1632–1672, 2018.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

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.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817*, 2018.

Yoav Freund and Robert Schapire. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, Aviv Tamar, et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.

Michael Gimelfarb, Scott Sanner, and Chi-Guhn Lee. Reinforcement learning with multiple experts: A bayesian model combination approach. In *Advances in Neural Information Processing Systems*, pp. 9528–9538, 2018.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018.

Arthur Guez, David Silver, and Peter Dayan. Efficient Bayes-adaptive reinforcement learning using sample-based search. In *Advances in Neural Information Processing Systems*, 2012.

Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. In *Advances in Neural Information Processing Systems*, pp. 5302–5311, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pp. 1109–1117, 2016.

David Hsu, Wee S Lee, and Nan Rong. What makes some pomdp problems easy to approximate? In *Advances in neural information processing systems*, pp. 689–696, 2008.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6023–6029. IEEE, 2019.

Gregory Kahn, Tianhao Zhang, Sergey Levine, and Pieter Abbeel. Plato: Policy learning using adaptive trajectory optimization. In *IEEE International Conference on Robotics and Automation*, pp. 3342–3349. IEEE, 2017.

Zico Kolter and Andrew Ng. Near-Bayesian exploration in polynomial time. In *International Conference on Machine Learning*, 2009.

Hanna Kurniawati, David Hsu, and Wee Sun Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems*, 2008.

Gilwoo Lee, Brian Hou, Aditya Mandalika, Jeongseok Lee, Sanjiban Choudhury, and Siddhartha S. Srinivasa. Bayesian policy optimization for model uncertainty. In *International Conference on Learning Representations*, 2019.

Yao Liu, Zhaohan Guo, and Emma Brunskill. Pac continuous state online multitask reinforcement learning with identification. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pp. 438–446. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

Russell Mendonca, Abhishek Gupta, Rosen Kralev, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Guided meta-policy search. *arXiv preprint arXiv:1904.00956*, 2019.

Pedro A. Ortega, Jane X. Wang, Mark Rowland, Tim Genewein, Zeb Kurth-Nelson, Razvan Pascanu, Nicolas Heess, Joel Veness, Alexander Pritzel, Pablo Sprechmann, Siddhant M. Jayakumar, Tom McGrath, Kevin Miller, Mohammad Gheshlaghi Azar, Ian Osband, Neil C. Rabinowitz, András György, Silvia Chiappa, Simon Osindero, Yee Whye Teh, Hado van Hasselt, Nando de Freitas, Matthew Botvinick, and Shane Legg. Meta-learning of sequential strategies. *arXiv preprint arXiv:1905.03030*, 2019.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, 2013.

Ian Osband, Benjamin Van Roy, Daniel J. Russo, and Zheng Wen. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019. URL http://jmlr.org/papers/v20/18-339.html.

Chris J Ostafew, Angela P Schoellig, and Timothy D Barfoot. Learning-based nonlinear model predictive control to improve vision-based mobile robot path-tracking in challenging outdoor environments. In *IEEE International Conference on Robotics and Automation*, pp. 4029–4036. IEEE, 2014.

Chris J Ostafew, Angela P Schoellig, and Timothy D Barfoot. Conservative to confident: treating uncertainty robustly within learning-based control. In *IEEE International Conference on Robotics and Automation*, pp. 421–427. IEEE, 2015.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2017.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation*, 2018.

Joelle Pineau, Geoff Gordon, Sebastian Thrun, et al. Point-based value iteration: An anytime algorithm for POMDPs. In *International Joint Conference on Artificial Intelligence*, 2003.

Neil C. Rabinowitz. Meta-learners' learning dynamics are unlike learners'. *arXiv preprint arXiv:1905.01320*, 2019.

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.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Guy Shani, Joelle Pineau, and Robert Kaplow. A survey of point-based POMDP solvers. *Journal on Autonomous Agents and Multiagent Systems*, 27(1):1–51, 2013.

David Silver and Joel Veness. Monte-carlo planning in large POMDPs. In *Advances in Neural Information Processing Systems*, 2010.

Tom Silver, Kelsey Allen, Josh Tenenbaum, and Leslie Kaelbling. Residual policy learning. *arXiv preprint arXiv:1812.06298*, 2018.

Bradly C. Stadie, Ge Yang, Rein Houthooft, Xi Chen, Yan Duan, Yuhuai Wu, Pieter Abbeel, and Ilya Sutskever. Some considerations on learning to explore via meta-reinforcement learning. *arXiv preprint arXiv:1803.01118*, 2018.

Zachary Sunberg and Mykel Kochenderfer. Online algorithms for POMDPs with continuous state, action, and observation spaces. In *International Conference on Automated Planning and Scheduling*, 2018.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pp. 7332–7342, 2018.

Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Robotics: Science and Systems*, 2017.