EXPLORATION IN APPROXIMATE HYPER-STATE SPACE

Luisa Zintgraf *
University of Oxford

Leo Feng
University of Oxford

Maximilian Igl
University of Oxford

Kristian Hartikainen
University of Oxford

Katja Hofmann
Microsoft Research

Shimon Whiteson
University of Oxford

ABSTRACT

Bayes-optimal agents are those that optimally trade off exploration and exploitation under task uncertainty, i.e., maximise *online return* incurred while learning. Although computing such policies is intractable for most problems, recent advances in meta-learning and approximate variational inference make it possible to learn approximately Bayes-optimal behaviour for tasks from a given prior distribution. In this paper, we address the problem of exploration *during* meta-learning, i.e., gathering the data required for an agent to learn how to learn in an initially unknown task. Our approach uses reward bonuses that incentivise the agent to explore in *hyper-state space*, i.e., the joint state and belief space. On a sparse HalfCheetahDir task we show that our method can learn adaptation strategies for sparse tasks where existing meta-learning methods fail.

1 INTRODUCTION

In many situations for which we want to deploy autonomous agents, we care not only about final performance but about *online return*, i.e., how much reward the agent accrues while learning in an initially unknown environment. An agent that learns to drive a car has to learn to use the gas and brake pedals appropriately, while avoiding to crash the car. An agent playing StarCraft has to learn about the opponent and adjust its strategy, while minimising the risk of losing the game. In such settings, the agent has to choose actions with high expected return *under task uncertainty*.

By taking a Bayesian view on reinforcement learning, we can in principle compute an agent that does so optimally. This agent maintains a belief at every time step over which task it is in (more specifically, the reward and transition function), and conditions its actions on the joint environment and belief states, referred to as *hyper-states*. Formally, this is a Bayes-Adaptive Markov Decision Process (BAMDP, Duff & Barto (2002)), a solution to which is called a Bayes-optimal policy.

Unfortunately, solving a BAMDP exactly is hopelessly intractable for all but the smallest tasks. One way to address these challenges is with meta-learning, which can directly optimise the objective we care about: maximising online reward when learning in unknown environments. Instead of trying to solve the BAMDP directly, we let the agent learn how to learn under task uncertainty, by meta-learning how to do so on a set of related tasks (Ortega et al., 2019; Duan et al., 2016; Wang et al., 2016; Humplik et al., 2019; Zintgraf et al., 2020) (for more background on meta-learning approaches and how they explore when deployed in a new task, see Appendix A).

However, a key remaining challenge is how to *explore during meta-training* to gather data from which *Bayes-optimal exploration behaviour* can be learned. To make the distinction clear, we call this *pseudo exploration*, in contrast to the *deployed exploration* problem. Deployed exploration describes the exploration of an agent in a new, unknown task while maximizing online return. Since we care about this return, we want the exploration to be Bayes-optimal. On the other hand, pseudo exploration describes the exploration of the meta-learner which seeks data in order to learn such a Bayes-optimal strategy. This occurs before deployment, i.e., during meta-training. Contrary to deployed exploration, we do *not* care about the rewards incurred during pseudo exploration, but rather about gathering the data needed for meta-learning.

*luisa.zintgraf@cs.ox.ac.uk

The pseudo exploration problem in BAMDPs is related to exploration when learning in partially observable MDPs (see Appendix A), a topic mostly studied on small environments (e.g., Cai et al. (2009); Poupart & Vlassis (2008)) and some initial results using deep learning Yordanov (2019).

Pseudo exploration can be particularly difficult if rewards are sparse or not shaped to guide the agent towards good behaviour. For efficient exploration during meta-learning, the agent has to keep track of how much it has explored each task (since states can have different values per task), and learn about the shared structure between tasks to extract the information about how to perform deployed exploration. This means the agent cannot just explore each task independently, but has to try out different deployed exploration strategies in order to learn the Bayes-optimal one. As we show in this paper, existing methods can fail on even seemingly simple task distributions.

We address these challenges by incentivising the agent to explore in hyper-state space during meta-training, using exploration bonuses. We use the meta-learning method VariBAD (Zintgraf et al., 2020) since it maintains an explicit estimate of the belief which we can use to compute exploration bonuses. Adding the bonus to the *approximate* hyper-state is not trivial since the beliefs are wrong at the beginning of meta-training. We address this by adding an additional reward bonus solely on the state space, which guides the agent early on in training to new states that provide non-zero task reward. On a sparse HalfCheetahDir environment, we show that our approach can meta-learn approximately Bayes-optimal strategies for which existing meta-learning methods fail.

2 Problem Setting

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

Problem Setting. We consider a setting in which we have a distribution $p(M)$ over MDPs, with an MDP $M_i \sim p(M)$ defined by a tuple $M_i = (\mathcal{S}, \mathcal{A}, R_i, T_i, T_{i,0}, \gamma, H)$. Our objective is to maximise the online return achieved *during learning* in an unseen (test) task drawn from $p(M)$,

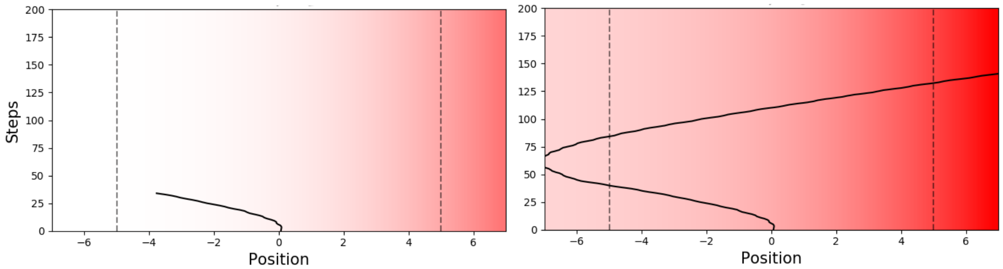
$$\max \ \mathbb{E}_{p(M)}\left[\mathcal{J}(\pi)\right]. \tag{1}$$

Maximising (1) requires a good deployed exploration strategy with respect to the initially unknown reward and transition functions, and exploiting the task information to adapt in this environment. The more an agent can make use of prior knowledge about p, the better it can perform this trade-off.

Bayesian RL. In principle, we can compute the optimal solution to (1) by reformulating the problem as a Bayes-Adaptive MDP (BAMDP, Duff & Barto (2002); for full definition see B.1). BAMDPs are a special type of belief MDP with hyper-state space $\mathcal{S}^+ = \mathcal{S} \times \mathcal{B}$, where $\mathcal{B}$ is the belief space. The solution to a BAMDP is called a Bayes-optimal policy $\pi(s_t, b_t)$, which conditions its actions on a belief $b_t = p(R, T|\tau_{:t}) \in \mathcal{B}$ that is the posterior over the reward and transition function given the agent's past experience $\tau_{:t} = (s_0, a_0, r_1, \ldots, s_t)$. Solving BAMDPs is generally intractable, but meta-learning on tasks sampled from the prior distribution p offers a scalable and flexible way of approaching this problem (Humplik et al., 2019; Zintgraf et al., 2020).

Meta-Learning. During meta-training, we sample batches of tasks $\mathbf{M} = \{M_i\}_{i=1}^N$ from $p(M)$ and interact with them. During this phase, we want to do good pseudo exploration. At meta-test time, the agent is evaluated based on the expected return it achieves *while learning*, in new tasks drawn from $p(M)$. This requires good deployed exploration strategies.

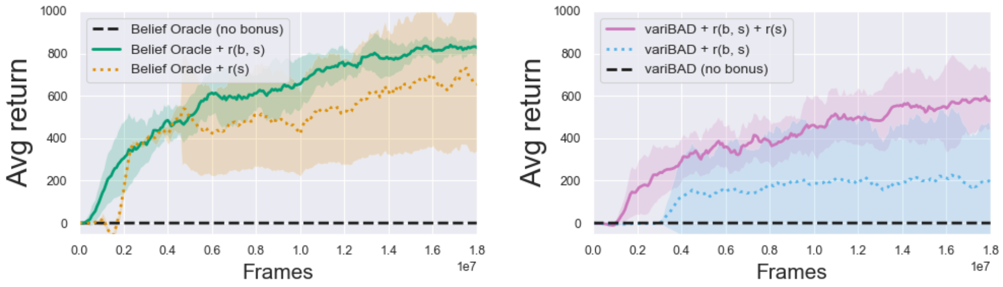
Example Environment. Throughout this paper, we use a sparse version of the HalfCheetahDir MuJoCo, the dense version of which is commonly used in prior meta-learning literature (Finn et al., 2017; Mishra et al., 2017; Rakelly et al., 2019). The prior distribution $p(M)$ is uniform over the tasks ”walk forward” and ”walk backward”, and the reward is the velocity in the correct direction. We create a sparse version by setting the reward to 0 within an interval $[-5, 5]$ around the agent's starting position. The true belief can be expressed as $b = [0.5, 0.5]$ for the prior, and updating this to the posterior $b = [0, 1]$ (left) or $b = [1, 0]$ (right) when the agent observes a single reward outside of $[-5, 5]$. The Bayes-optimal strategy is to walk outside the interval to one side, infer from the sdense reward what the task is, and walk into the correct direction from there on, as demonstrated in Fig 1b.



(a) Behaviour and $r_h(s^+)$ when $b = [0.5, 0.5]$.

(b) Behaviour and $r_h(s^+)$ when $b = [0, 1]$.

Figure 1: Example rollout of a belief oracle policy for the task ”go right”, mid-way through training. The background (red gradient) visualises the reward bonus: darker means more. (a) The bonus for state-belief pairs with the prior belief. (b) The bonus for state-belief pairs with the updated belief.



(a) Belief Oracle.

(b) VariBAD

Figure 2: Learning curves for the Belief Oracle (a) and variBAD (b), with and without reward bonus.

3 Method and Results

To learn Bayes-optimal behaviour via meta-learning, the agent must seek out the necessary data during training. For problems with dense and structured rewards, the reward signal is often sufficient to guide the agent. While in principle, a sparse reward signal should be enough to solve the task, in practice existing meta-learning methods fail in such setting. For example, in the sparse HalfCheetahDir environment, both a Belief Oracle (with access to the true belief) and variBAD (with access to an approximate, learned belief) fail to learn, as shown in Figure 2 (dashed lines).

We argue that learning Bayes-optimal behaviour via meta-learning requires (A) exploring the state space sufficiently, while also considering that states can have different values across tasks, and (B) trying out different strategies that ultimately allow the agent to learn the Bayes-optimal one. In this work we show that this can be achieved via meta-exploration in the hyper-state space of a BAMDP, i.e., by incentivising the agent to explore the joint state and belief space during meta-training.

In order to demonstrate the meta-exploration setting, we begin by assuming that we have access to the true hyper state $s_t^+ = (s_t, b_t)$, i.e., including the true posterior distribution $b_t(R, T|\tau_{:t})$ over the reward and transition function of the underlying MDP. We later lift this assumption and discuss the challenges that arise when dealing with approximate and non-stationary beliefs.

3.1 Exploration in (Exact) Hyper-State Space

With access to the true belief $b_t = p(R, T|\tau_{:t})$, we can utilise commonly used exploration bonuses directly on the hyperstates $s_t^+ = (s_t, b_t)$ to incentivise the agent to explore during meta-training. In this paper, we use random network distillation (see Appendix B.3) given the method's empirical successes in standard RL problems (Osband et al., 2017; 2018; Burda et al., 2019) and theoretical justifications for deep neural networks (Pearce et al., 2018; Ciosek et al., 2020).

To compute a reward bonus, a predictor network $g_h(s^+)$ is trained to predict the outputs of a fixed, randomly initialised prior network $f_h(s^+)$, on all hyper-states s^+ visited by the agent so far during meta-training. The mismatch between those predictions is low for frequently visited states and high for novel states. Formally we define the reward bonus for a hyper-state $s_t^+ = (s_t, b_t)$ as

$$r_h(s_t^+) = ||f_h(s_t^+) - g_h(s_t^+)||^2. \tag{2}$$

We train the predictor network g_h together with a policy trained using PPO on the sum of external environment rewards and reward bonus.

Results. Figure 2a shows the performance of the Belief Oracle (a policy with access to the ground-truth belief at each timestep), with and without reward bonuses. Without the bonus, even the Belief Oracle policy completely fails the task with an average return of 0. By adding an exploration bonus $r_h(b, s)$ on the hyper-state, the policy learns an approximately Bayes-optimal behaviour. This is visualised in Figure 1: the policy walks outside of the zero-reward interval, infers which task it is in, and either proceeds further or (like in this case) turns around to walk in the opposite direction. We also show the performance of a policy trained with a reward bonus only on the state, $r(s)$, and observe it performing worse. Upon inspection of the learned policies, we see that some agents do turn around if the direction was wrong, but then stay in the non-reward zone (see Appendix C.1).

3.2 Exploration in (Approximate) Hyper-State Space

So far, we assumed having access to the true belief b_t. However, for most interesting problems, computing the true belief is hopelessly intractable. Instead, we have to resort to approximation. Zintgraf et al. (2020) propose variBAD, a method to meta-learn approximately Bayes-optimal behaviour for a given training task distribution. To achieve this, they train a VAE-type architecture where the latent state represents the agent's belief over the current task, $\hat{b}_t(m|\tau_{:t})$, with m a low-level task representation. During meta-training, this model is trained together with the policy and learns how to perform approximate inference. The belief $\hat{b}$ is thus nonstationary, i.e., the estimated posterior for the *same* history $\tau_{:t}$ changes over time as the VAE learns.

This introduces additional difficulties, especially in the beginning of training when the belief does not carry any information about the task. This is problematic, because (A) the agent might prematurely stop visiting states that contain task information, if this was not reflected in the belief state; and (B) the agent can exploit the reward bonus on the approximate hyper-states $\hat{s}^+ = (s, \hat{b})$ by exploring state-action pairs that cause the VAE latent state to fluctuate a lot but not guide the agent towards novel states with meaningful task rewards.

We therefore add an additional reward bonus that is only dependent on the environment state,

$$r_s(s_t) = ||f_s(s_t) - h_s(s_t)||^2, \tag{3}$$

and add this to the rewards with which the agent is trained. At the beginning of training, this helps the agent to focus on reaching states with non-zero rewards. This in turn allows the VAE to learn represent the approximate posterior in its belief state. It is then that the agent can start exploring in hyper-state space.

The overall reward on which the agent is trained with is thus

$$R(r_t|s_{t-1}, a_{t-1}, s_t) + \lambda_h r_h(\hat{s}_t^+) + \lambda_s r_s(s_t), \tag{4}$$

where λ_h and λ_s are scalar weights for the reward bonuses.

Results. Figure 2b shows the results with and without reward bonuses for training the variBAD agent. We first note that variBAD without a bonus does not learn at all. With an added reward bonus on the hyper-state $r(s, \hat{b})$, variBAD learns poorly. Finally, an additional bonus on the states alone, $r(\hat{b}, s) + r(s)$ enables variBAD to also learn an approximately Bayes-optimal solution.

Future Work. In the given toy example, we found that adding a reward bonus $r(s)$ for variBAD is enough to enable the agent to meta-learn approximately Bayes-optimal strategies, even when the weights λ_h and λ_s are held fixed. In the future, we plan to extend this work to more challenging environments. We expect that in this case, we have to trade off different reward bonuses in a smarter way. E.g., by annealing the weight of the state reward bonus λ_s or using the uncertainty *over* the beliefs to trade off the reward bonuses. These could be obtained using, for example, an ensemble of encoders or using the reconstruction loss of the VAE as a proxy for VAE model uncertainty.

Acknowledgements

We thank Wendelin Böhmer and Tabish Rashid for helpful discussions on exploration bonuses. Luisa Zintgraf is supported by the Microsoft Research PhD Scholarship Program. Kristian Hartikainen is funded by the EPSRC. Maximilian Igl is supported by the UK EPSRC CDT in Autonomous Intelligent Machines and Systems. This work was supported by a generous equipment grant from NVIDIA. This project has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation programme (grant agreement number 637713).

References

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

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pp. 1471–1479, 2016.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representation (ICLR)*, 2019.

Chenghui Cai, Xuejun Liao, and Lawrence Carin. Learning to explore and exploit in pomdps. In *Advances in Neural Information Processing Systems*, pp. 198–206, 2009.

Luca Carlone, Jingjing Du, Miguel Kaouk Ng, Basilio Bona, and Marina Indri. An application of kullback-leibler divergence to active slam and exploration with particle filters. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 287–293. IEEE, 2010.

Kamil Ciosek, Vincent Fortuin, Ryota Tomioka, Katja Hofmann, and Richard Turner. Conservative uncertainty estimation by fitting prior networks. In *International Conference on Learning Representation (ICLR)*, 2020.

Finale Doshi, Joelle Pineau, and Nicholas Roy. Reinforcement learning with limited reinforcement: Using bayes risk for active learning in pomdps. In *Proceedings of the 25th international conference on Machine learning*, pp. 256–263, 2008.

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 *International Conference on Machine Learning (ICML)*, 2017.

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.

Héctor H González-Banos and Jean-Claude Latombe. Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research*, 21(10-11):829–848, 2002.

Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. In *Advances in Neural Processing Systems (NeurIPS)*, 2018.

Swaminathan Gurumurthy, Sumit Kumar, and Katia Sycara. Mame: Model-agnostic meta-exploration. *arXiv preprint arXiv:1911.04024*, 2019.

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.

Jan Humplik, Alexandre Galashov, Leonard Hasenclever, Pedro A Ortega, Yee Whye Teh, and Nicolas Heess. Meta reinforcement learning as task inference. *arXiv preprint arXiv:1905.06424*, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representation (ICLR)*, 2014.

Mikko Lauri and Risto Ritala. Planning for robotic exploration based on forward simulation. *Robotics and Autonomous Systems*, 83:15–31, 2016.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

Pedro A Ortega, Jane X Wang, Mark Rowland, Tim Genewein, Zeb Kurth-Nelson, Razvan Pascanu, Nicolas Heess, Joel Veness, Alex Pritzel, Pablo Sprechmann, et al. Meta-learning of sequential strategies. *arXiv preprint arXiv:1905.03030*, 2019.

Ian Osband, Daniel Russo, Z Wen, and B Van Roy. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 2017.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8617–8629, 2018.

Georg Ostrovski, Marc G Bellemare, Aäron van den Oord, and Rémi Munos. Count-based exploration with neural density models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2721–2730. JMLR. org, 2017.

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.

Tim Pearce, Mohamed Zaki, Alexandra Brintrup, Nicolas Anastassacos, and Andy Neely. Uncertainty in neural networks: Approximately bayesian ensembling. *arXiv preprint arXiv:1810.05546*, 2018.

Pascal Poupart and Nikos Vlassis. Model-based bayesian reinforcement learning in partially observable domains. In *Proc Int. Symp. on Artificial Intelligence and Mathematics,*, pp. 1–2, 2008.

Kate Rakelly, Aurick Zhou, Deirdre Quillen, Chelsea Finn, and Sergey Levine. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning (ICML)*, 2019.

Stephane Ross, Brahim Chaib-draa, and Joelle Pineau. Bayes-adaptive pomdps. In *Advances in neural information processing systems*, pp. 1225–1232, 2008.

Stéphane Ross, Joelle Pineau, Brahim Chaib-draa, and Pierre Kreitmann. A bayesian approach for learning and planning in partially observable markov decision processes. *Journal of Machine Learning Research*, 12(May):1729–1770, 2011.

Jonas Rothfuss, Dennis Lee, Ignasi Clavera, Tamim Asfour, and Pieter Abbeel. Promp: Proximal meta-policy search. In *International Conference on Learning Representation (ICLR)*, 2019.

Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pp. 222–227, 1991.

Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Robotics: Science and Systems*, volume 2, pp. 65–72, 2005.

Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

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. In *Advances in Neural Processing Systems (NeurIPS)*, 2018.

Susanne Still and Doina Precup. An information-theoretic approach to curiosity-driven reinforcement learning. *Theory in Biosciences*, 131(3):139–148, 2012.

Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in neural information processing systems*, pp. 2753–2762, 2017.

Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. In *Annual Meeting of the Cognitive Science Community (CogSci)*, 2016.

Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, pp. 146–151. IEEE, 1997.

Yordan Yordanov. Using instrinsic motivation for exploration in partially observable environments. Master's thesis, University of Oxford, Oxford, UK, 2019.

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.

Exploration in Approximate Hyper-State Space

Supplementary Material

A Related Work

Exploration Bonuses. Deep reinforcement learning has been successful on many tasks, and if the reward is dense and structured it is often sufficient to perform exploration via ϵ-greedy action selection or by having a stochastic policy from which we can sample. For hard exploration tasks with sparse rewards however, these myopic exploration strategies perform poorly. A range of exploration strategies have been proposed in the literature, such as count-based exploration (Strehl & Littman, 2008; Bellemare et al., 2016; Ostrovski et al., 2017; Tang et al., 2017) or reward bonuses that rely on errors in predicting the environment dynamics (Achiam & Sastry, 2017; Burda et al., 2018; Pathak et al., 2017; Schmidhuber, 1991; Stadie et al., 2015). Other work has looked at exploration via quantifying uncertainty (Houthooft et al., 2016; Still & Precup, 2012).

In this paper, we use random network distillation (RNDs) (Osband et al., 2018; Burda et al., 2019) as an exploration bonus on the hyper-state space. RNDs can be used to obtain uncertainty estimates at single data points (as opposed to global uncertainty) (Burda et al., 2019; Ciosek et al., 2020) and have previously been applied to exploration problems in reinforcement learning (Osband et al., 2017; Burda et al., 2019).

Meta-Learning and Exploration. How to explore efficiently when faced with a new task is key in order for an agent to adapt quickly and cost effectively to a new task. MAML Finn et al. (2017) does not directly account for how the initial data distribution influences the gradient update, and hence cannot directly learn how to explore in a completely new task. This is not a problem if the rewards are sufficiently dense. Several extensions, (Rothfuss et al., 2019; Stadie et al., 2018) address this problem by explicitly accounting for the effect which the pre-update data distribution has on the post-update performance. Gurumurthy et al. (2019) learn a separate exploration policy for pre-update data gathering, and one for post-adaptation behaviour.

Gupta et al. (2018) and Rakelly et al. (2019) meta-learn structured exploration strategies that are based on sampling. In particular, the latter method exhibits behaviour that is akin to posterior sampling. Even though this is, by definition, a suboptimal exploration strategy (the optimal being Bayes-adaptive action selection), these approaches have the advantage that they could be tested on sparse reward environments (while being meta-trained on dense versions of the environment).

When the task is unknown, the optimal exploration strategy takes into account this uncertainty and maintains a belief over the possible tasks. Such a policy is called Bayes-optimal. Meta-learning methods that rely on recurrent policies (Duan et al., 2016; Wang et al., 2016) learn how to explore within the dynamics of the policy, and can be seen as implicitly maintaining a belief over tasks Ortega et al. (2019). Humplik et al. (2019) and Zintgraf et al. (2020) develop methods that represent this belief more explicitly, by meta-learning to perform inference either using privileged task information during training such as the task ID or ground-truth description Humplik et al. (2019), or by meta-learning to perform inference in an unsupervised way Zintgraf et al. (2020).

So far, we have only seen first successes on a simple 2D environment with sparse rewards. Gupta et al. (2018) and Rakelly et al. (2019) solve this environment by training on dense rewards but evaluating on sparse rewards. The only notable exception are Humplik et al. (2019), who manage to meta-train an agent on the sparse version of this environment. However, in turn they rely on privileged information during meta-training such as, in this case, the ground-truth task description.

Exploration in POMDPs. The distinction between the *pseudo exploration problem* we face during meta-training and the *deployed exploration problem* we aim to solve at test time is also encountered when reasoning about exploration in POMDPs. This is unsurprising as a BAMDP, which we are aiming to solve using meta-learning, is a special case of a POMDP in which the environment state is fully observable, and the environmental parameters (reward and transition function) are not observable. In POMDPs, the *deployed* exploration is part of the policy, trading off information-gathering actions against optimizing the reward. *Pseudo* exploration is required during training of said policy.

Encouraging *pseudo* exploration during policy optimization in POMDPs is not widely explored for deep policies. To the best of our knowledge, only Yordanov (2019) provide some initial results on a simple environment by applying random network distillation (Burda et al., 2019) to the problem, proposing various approaches to deal with the non-stationarity of the latent embedding such as computing the reward bonuses not on the learned latent state of the policy, but on the output of a random recurrent network which aggregates past trajectories. We expect that when scaling our method on harder tasks than the spars HalfCheetahDir environment, the non-stationarity of the VAE latent space will become more of a problem and will require dealing with this more explicitly.

Similar to us, Cai et al. (2009) incentivize exploration in under-explored regions of belief space, slowly decreasing the exploration probability over time. However, they are using two separate policies for exploration and exploitation and rely on Bayesian learning to update the policies, preventing them to scale beyond small discrete state spaces.

Several authors (Poupart & Vlassis, 2008; Ross et al., 2008; Doshi et al., 2008; Ross et al., 2011) explore model-based Bayesian reinforcement learning in partially observable domains. However, by relying on approximate value iteration to solve the planning problem, they are also restricted to small environments.

Lastly, an important area for robotics is to learn to explore, e.g. (Lauri & Ritala, 2016; Carlone et al., 2010; Stachniss et al., 2005; Yamauchi, 1997; González-Banos & Latombe, 2002). In contrast to our work, exploration here is the goal of the agent and not part of the learning process of the policy. As such, it does not need to be traded off against exploitation as exploration is the optimal behaviour.

B Background

B.1 Bayes-Adaptive MDPs.

Given a distribution $p(M)$ over possible MDPs which our agent could find itself in, we define a Bayes-Adaptive MDP (BAMDP) as $M^+ = \big(\mathcal{S}^+, \mathcal{A}, R^+, T^+, T_0^+, \gamma, H^+\big)$. Here, $\mathcal{S}^+ = \mathcal{S} \times \mathcal{B}$ is the hyper-state space, consisting of the underlying MDP environment state space $\mathcal{S}$ and a belief space $\mathcal{B}$ with its elements being beliefs over the MDP. This belief is typically expressed as a distribution over the reward and transition function $b_t(R, T) = p(R, T|\tau_{:t})$, where $\tau_{:t}$ is the agent's experience up until the current time step t (in the current environment). The transition function T^+ in a BAMDP is defined as

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

and the reward function R^+ as

$$R^+(s^+_t, a_t, s^+_{t+1}) = \mathbb{E}_{b_{t+1}}\left[R(s_t, a_t, s_{t+1})\right]. \tag{5}$$

$T_0^+(s^+)$ is the initial hyper-state distribution, and H^+ is the horizon in the BAMDP.

A policy $\pi(s^+)$ acting in a BAMDP conditions its actions not only on the environment state s, but also on the belief b. This way, it can take task uncertainty into account when making decisions.

The agent's objective is to maximise the expected return in an initially unknown environment, while learning, within the horizon H^+:

$$\mathcal{J}^+(\pi) = \mathbb{E}_{b_0, T_0^+, T^+, \pi}\left[\sum_{t=0}^{H^+-1} \gamma^t R^+(r_{t+1}|s^+_t, a_t, s^+_{t+1})\right]. \tag{6}$$

An agent that optimises this objective optimally trades off exploration and exploitation in order to maximise expected cumulative return. How to optimally trade off exploration and exploitation depends on the horizon H^+, i.e., how much time the agent has left affects whether information-gathering actions are worth it.

For an in-depth introduction to BAMDPs we refer the reader to Duff & Barto (2002) or Ghavamzadeh et al. (2015).

B.2 VariBAD

The BAMDP setting allows us, in theory, to compute the Bayes-optimal exploration strategy. In practice however, this is intractable for all but the easiest tasks for a number of reasons. As summarised in Zintgraf et al. (2020), we might not have a concise representation of the transition and reward function over which we want to maintain a belief (we instead approximate them using deep neural networks); the inference required to maintain a belief might be intractable; and planning in hyper-state space might be intractable.

Zintgraf et al. (2020) propose a method to meta-learn Bayes-optimal policies for a given task distribution, in which the belief is approximated by a (modified) variational auto encoder (VAE; Kingma & Welling (2014)). We build on this method because it gives us access to an approximate hyper-state, consisting of the environment state given by the MDP and the approximate belief given by the VAE model.

VariBAD jointly trains a policy $\pi_\psi(s_t, b_t)$, an encoder $q_\theta(m|\tau_{:t})$ and a decoder for the reward $p(r_{i+1}|s_i, a_i, s_{i+1}, m_t)$ and the transitions $p(s_{i+1}|s_i, a_i, m_t)$, with $m \sim b_t$ a sample from the belief distribution at time step t. The overall objective is

$$\mathcal{L}(\phi, \theta, \psi) = \mathbb{E}_{p(M)} \left[\mathcal{J}(\psi, \phi) + \lambda \sum_{t=0}^{H^+} ELBO_t(\phi, \theta) \right]. \tag{7}$$

with

$$ELBO_t = \mathbb{E}_{p(M)} \left[\mathbb{E}_{q_\phi(m|\tau_{:t})} \left[\log p_\theta(\tau_{:H^+}|m) \right] - KL(q_\phi(m|\tau_{:t}) || q_\phi(m|\tau_{:t-1})) \right],$$

with prior $q_\phi(m) = \mathcal{N}(0, I)$. For details we refer the reader to the original paper, but essentially the objective maximises jointly a reinforcement learning loss $\mathcal{J}$ (where the agent is conditioned on the state s_t and the approximate belief b_t) and an evidence lower bound (ELBO) that consists of a reconstruction term for rewards and transitions, and a KL term to the previous posterior.

B.3 Random Network Distillation

In reinforcement learning, we can use the fact that unseen states can be seen as out-of-distribution data of a model that is trained on all data the agent has seen so far. Getting uncertainty estimates on states can thus quantify our uncertainty about the value of a state and in turn whether we have explored these states sufficiently. We can think about why exploration purely in the state space $\mathcal{S}$ (which is shared across tasks) is not enough: if the agent has explored a state many times in one task and is certain of its value, it should not exploit this knowledge in a different task, because this same state could have a completely different value. We cannot view these as separate exploration problems however, since we also have to try out different deployed exploration strategies and combine the information to meta-learn Bayes-optimal behaviour.

Therefore, we want to incentivise the agent to explore in the hyper-state space $\mathcal{S}^+ = \mathcal{S} \times \mathcal{B}$. Only if an environment state together with a specific belief has been observed sufficiently often to determine its value should the agent. This therefore amounts to exploration in a BAMDP state space, which essentially means trying out different exploration strategies in the environments of the training distribution.

We use random network distillation (Osband et al., 2018; Burda et al., 2019; Ciosek et al., 2020) to obtain such uncertainty estimates and review them using the formulation of Ciosek et al. (2020) in the following.

Assume we are given a set of training data $\mathcal{D} = \{s_i\}_{i=1}^N$ of all states the agent has observed. To get uncertainty estimates, we first fit B predictor networks $g_j(s)$ ($j = 1, \dots, B$) to a random prior process $f_j(s)$ each (a network with randomly initialised weights, which is fixed and never updated). We then estimate the uncertainty for a state s_* as,

$$\sigma^2(s_*) = \max(0,\ \sigma_\mu^2(s_*) + \beta v_\sigma(s_*) - \sigma_A^2), \tag{8}$$

where $\sigma_\mu^2(s_*)$ is the sample mean of the squared error between the B predictor networks and the prior processes, $v_\sigma(s_*)$ is the sample variance of the squared error. The first quantifies our uncertainty,

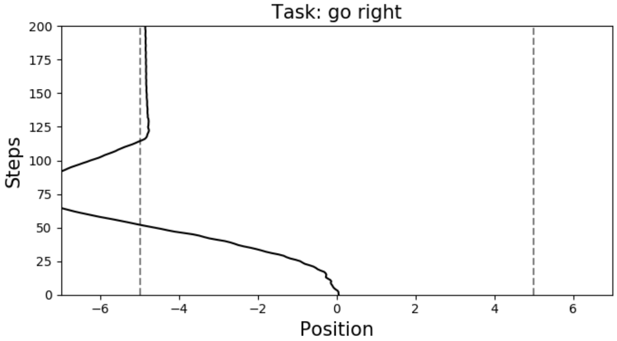


Figure 3: Behaviour of a policy which failed to learn Bayes-optimal behaviour. We observe such behaviour often when training a variBAD policy with only reward bonus on the hyper-states, $r_h(b, s)$ (and no additional bonus on the environment state), and when training a Belief Oracle policies trained with a reward bonys only on the environment state, $r_s(s)$ (instead of on the hyper-states).

whereas the second quantifies our uncertainty over what our uncertainty is. In practice, $B = 1$ is typically sufficient and the second term disappears (Ciosek et al., 2020). The term σ_A^2 is the aleatoric noise inherent in the data which is an irreducible constant. In theory, this can be learned as well and depends on how much information can be extracted about the value of states and actions from the data. In practice, we just set this term to 0. The reward bonus can be added to the normal environment reward, $\hat{r}(s_t) = r(s_t) + \sigma^2(s_t)$ and used with any RL algorithm.

Given a hyper-state $s_t^+ = (s_t, b_t)$, an ensemble of B prior networks $\{f^i(s^+)\}_{i=1}^B$ and corresponding predictor networks $\{h^i(s^+)\}_{i=1}^B$, the reward bonus is defined as

$$r_c(s_t^+) = \max(0,\ \sigma_m u^2(s_t^+) + \beta v_\sigma(s_t^+) - \sigma_A^2) \tag{9}$$

where $\sigma_m u^2(s_t^+)$ is the sample mean of the squared error between prior and predictor networks and $v_\sigma(s_t^+)$ is the sample variance of that error. When necessary, we can expand this definition to do exploration in hyper-state-action space.

C Experiments

We use the algorithm variBAD as described in Zintgraf et al. (2020). They train their policy using PPO, and we just add the intrinsic bonus rewards to the extrinsic environment reward and use the sum when learning with PPO. We normalise the intrinsic and extrinsic separately by dividing by a rolling estimate of the standard deviation. For the bonus rewards, we set $\lambda_h = 10$ and $\lambda_s = 10$.

The learning curves shown in Figure 2 are averaged across 3 seeds, with 95% confidence intervals.

C.1 Additional Results

Figure 3 shows example behaviour of a suboptimal policy at test time. The agent back into the zero-reward zone after realising that the task was not ”go left”, but stays in there instead of doing the optimal thing, which is going further to the right and into the dense reward area beyond the sparse interval border.