The Whys and Hows of Active Exploration in Model-Based Reinforcement Learning

Alberto Caron
The Alan Turing Institute
London, UK
acaron@turing.ac.uk

Chris Hicks
The Alan Turing Institute
London, UK
c.hicks@turing.ac.uk

Vasilios Mavroudis
The Alan Turing Institute
London, UK
vmavroudis@turing.ac.uk

Abstract

In this work, we study the problem of sample efficient exploration in Model-Based Reinforcement Learning (MBRL). While most popular exploration methods in MBRL are 'reactive' in nature, and thus inherently sample inefficient, we discuss the benefits of an 'active' approach, where the agent selects actions to query novel states in a data-efficient way, provided that one can guarantee that regions of high epistemic, and not aleatoric, uncertainty are targeted. In order to ensure this, we consider popular exploration bonuses based on *Bayesian surprise*, and demonstrate their desirable properties under the assumption of a Gaussian Process model. We then introduce a novel exploration method, Bayesian Active Exploration, where the agent queries transitions based on a multi-step predictive search aimed at maximizing the expected information gain. Moreover, we propose alternative dynamics model specifications based on stochastic variational Gaussian Processes and deep kernels that allow for better scalability with sample size and state-action spaces, and accommodate non-tabular inputs by learning a latent representation, while maintaining good uncertainty-quantification properties.

1 Introduction

The exploration-exploitation trade-off is a long-standing problem in Reinforcement Learning (Sutton and Barto, 2018). Exploration in classic RL algorithms is often achieved via simple heuristics such as ϵ-greedy policy in Q-methods (Mnih et al., 2015), action noise injection (Lillicrap et al., 2015), or some form of policy entropy regularizers in policy gradient methods (Sutton et al., 1999; Kakade, 2001; Schulman et al., 2017). In some environments, these simple heuristics are enough to ensure sufficient exploration and learn the optimal policy; in others, such as 'sparse' rewards ones, they are prone to get stuck in sub-optimal policies instead.

Most existing methods satisfying the need for deeper exploration in such environments are **reactive** in nature (Ladosz et al., 2022). This means that when the agent encounters a new state, they assign it a higher internal, or 'intrinsic', exploration bonus reward r_t^i that encourages them to visit that state more often. Exploration bonuses can be *model-based* (Stadie et al., 2015; Osband et al., 2016; Pathak et al., 2017), where the agent learns a model of the environment dynamics and uses an output from it (prediction error, variance, etc.) to define r_t^i, or *count-based* (Bellemare et al., 2016; Ostrovski et al., 2017; Tang et al., 2017), where the intrinsic reward is defined as a function of the visitation frequency of a certain state-action pair. While reactive exploration can be sufficient to solve sparse reward tasks, it is intrinsically sample inefficient as it needs a large number of sampled transitions to ensure that the agent sees, and consequently moves to, novel state-action pairs. This is especially a problem in Markov Decision Processes (MDPs) (Puterman, 2014) where the agent is likely to be 'trapped' in certain pools of positive recurrent states characterized by small expected return times (Ortner, 2020).

17th European Workshop on Reinforcement Learning (EWRL 2024).

Active exploration (Kamthe and Deisenroth, 2018; Shyam et al., 2019; Tarbouriech and Lazaric, 2019; Tschantz et al., 2020; Sajid et al., 2021; Ball et al., 2020) instead represents a practical and data-efficient solution to exploration in settings characterized by expensive or budgeted data acquisition. Some real-world applications include material design and automated chemistry (Steiner et al., 2019; Burger et al., 2020), autonomous cybersecurity (Nguyen and Reddi, 2021; Andrew et al., 2022; Foley et al., 2022; Bates et al., 2023; Foley et al., 2023) and healthcare (Yu et al., 2021). Under the active exploration paradigm, the agent selects actions both for exploratory and exploitative purposes by planning to explore unknown trajectories. This leads to significant sample efficiency gains, provided that exploration is guided towards regions of epistemic (knowledge-based) rather than aleatoric (inherent) uncertainty (Itti and Baldi, 2009; Houthooft et al., 2016; Nikolov et al., 2019; Hüllermeier and Waegeman, 2021). To ensure that this requirement is met, we adopt a Bayesian view of the problem to demonstrate the desirable properties associated with information-theoretic based exploration bonuses (Cover, 1999; Houthooft et al., 2016; Mehta et al., 2022).

Related Work Model-based intrinsic rewards (or 'curiosity'-based approaches) (Bellemare et al., 2016; Osband et al., 2016; Pathak et al., 2017) are a popular class of methods that deal with sparse rewards environments. These are typically reactive and utilize exploration bonuses derived from a model of the dynamics, such as next-state prediction error (Stadie et al., 2015; Pathak et al., 2017) or variance (Sorg et al., 2010; Pathak et al., 2019). In a more Bayesian fashion, Houthooft et al. (2016) advocates for an intrinsic reward that captures 'Bayesian surprise' (Itti and Baldi, 2009), defined as the relative entropy (or information gain) (Lindley, 1956), between the old and the newly updated dynamics model's parameters after observing the next state s_{t+1}. The notion of active exploration is formalized in the early work of Carpentier et al. (2011) for the case of multi-armed bandits, and later in Tarbouriech and Lazaric (2019) for the case of purely exploratory tasks in the form of (non-Bayesian) Active Exploration MDPs. The model-based Bayesian RL literature (Ghavamzadeh et al., 2015) have come up with a Bayes-Adaptive version of MDPs (Strens, 2000; Duff, 2002; Ross et al., 2007) that we adopt, where exploitation and exploration are naturally balanced. Within this stream of literature, Shyam et al. (2019) have recently developed an active exploration method that uses deep ensembles and an information-theoretic exploration bonus, and focuses on purely exploratory tasks Tarbouriech and Lazaric (2019). Contributions including Deisenroth and Rasmussen (2011); Kamthe and Deisenroth (2018) advocate for the use of a Gaussian Processes (GPs) for Model Predictive Control (MPC) in RL (Draeger et al., 1995; Chua et al., 2018), but do not specifically consider the problem of active exploration. Similarly, Mehta et al. (2022) have recently proposed an information theoretic approach to active MPC in RL, again coupled with a GP model of the dynamics. Tangential to this work is also the vast literature on Bayesian Optimal Experimental Design (Pukelsheim, 2006; Foster et al., 2019, 2021; Foster, 2021; Rainforth et al., 2023) and on Bayesian Active Learning (Houlsby et al., 2011; Hanneke et al., 2014; Gal et al., 2017; Smith et al., 2023), that focus mostly on iid problems.

Contributions Within the rich literature on model-based Bayesian RL only few methods consider an active approach to exploration (Shyam et al., 2019; Mehta et al., 2022). However, these approaches either lack scalability to high-dimensional, non-tabular, state-action spaces or they require training deep ensemble, multi forward pass, models (Lakshminarayanan et al., 2017). In addition, none of the works above provide theoretical guarantees as to why information-theoretic exploration bonuses are desirable. Taking these into account, the contributions of this work can be summarized as follows. Firstly, by defining the problem via a Bayes-Adaptive MDP (BAMDP) framework (Strens, 2000; Duff, 2002; Ross et al., 2007; Ghavamzadeh et al., 2015), we prove that an information-theoretic exploration bonus such as information gain (Lindley, 1956) triggers the agent's 'epistemic curiosity' and has the desirable property of naturally converging to zero as the agent learns more about the environment dynamics. This property is not guaranteed for other curiosity-based methods that employ, e.g., prediction error or variance (Stadie et al., 2015; Pathak et al., 2017). Secondly, we present the concept of Bayesian Active Exploration, discussing how the information-theoretic bonus approach can be extended to settings where the agent actively queries trajectories using a surrogate, expected measure of information gain (Bernardo, 1979) that has the same desirable properties. Lastly, we propose two alternative, more scalable dynamics models based on: i) Stochastic Variational families of GPs (SVGP) (Titsias, 2009), which scale better with sample size; and ii) single forward pass deep kernels (Wilson et al., 2016a), which allows for high-dimensional state-action (and non-tabular) inputs while maintaining a fully Bayesian Active Exploration (BAE) approach.

2 Problem Setup

Consider the canonical definition of a **Bayes-Adaptive MDP** (BAMDP) (Duff, 2002; Ross et al., 2007). A BAMDP is a tuple $\langle \mathcal{H}_{\mathcal{S}}, \mathcal{A}, p_h, p_0, r^e, \gamma \rangle$, defined over an horizon of $t \in \{1, ..., T\}$ time steps where: i) $\mathcal{H}_{\mathcal{S}} = \mathcal{S} \times \Theta$ is the set of hyper-states, defined as the Cartesian product of the environment states space $\mathcal{S}$ and the space of parameters of the posterior transition dynamics Θ; ii) $\mathcal{A}$ is the action space; iii) $p_h(\cdot|h_s, a) \in \mathcal{P}_\tau$ is the hyper-state transition probability function that determines the next hyper-state (s', θ') given the current hyper-state (s_t, θ_t) and the action a_t, and is such that $p_h(s', \theta'|s, \theta, a) = p(s'|s, \theta, a)p(\theta'|s, \theta, a, s')$, where $p(\theta'|s, \theta, a, s')$ is the updated posterior given the new state s'; iv) $p_0 \in \mathcal{P}_0$ is the combination of the initial state probability function $s_0 \sim p_0(\cdot)$ and the prior probability on the dynamics parameters $\theta_0 \sim p(\theta)$; v) $r^e : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{C} \subset \mathbb{R}$ is a bounded extrinsic reward function; vi) $\gamma \in (0, 1)$ is a discount factor. The corresponding *Bayes-optimal* value function is then defined as:

$$V_t^*(s, \theta) = \max_{a \in \mathcal{A}} \left[r^e(s, a) + \gamma \int_{\mathcal{S}} \int_{\Theta} p(s'|s, \theta, a) V_{t-1}^*(s', \theta')\, ds' d\theta' \right]$$

Any policy $\pi^*(s, \theta)$ that maximizes the above is a Bayes-optimal policy. Notice that a Bayes-optimal policy is technically sub-optimal for the purely exploitative MDP task defined by the standard optimal value function $V_t^*(s)$ (Ghavamzadeh et al., 2015), as it intuitively represents a principled way of balancing out exploitation and exploration in MDPs. In a BAMDP, the transition dynamics is not known, and the agent only has a prior belief about it. The agent starts in a belief state corresponding to its prior on (s_0, θ_0), and updates his posterior belief by interacting with the environment and transitioning to different hyper-states (s', θ') (where θ' is the updated parameters).

2.1 Reactive Exploration

In order to achieve sufficient exploration in settings where the extrinsic rewards r^e are sparse, *curiosity-based* methods typically augment the rewards function with some notion of *intrinsic* rewards bonus r^i, such that $r_t = r_t^e + \eta_i(t) r_t^i$. The intrinsic reward r_t^i can be defined, e.g., as the next-state prediction error $r_t^i = \frac{\eta}{2} \| f_\theta(s_t, a_t) - s_{t+1} \|_p$, where $\|\cdot\|_p$ denotes the L space norm and $f_\theta(\cdot, \cdot)$ a predictive model for s_{t+1} (e.g., a neural network), or variance $r_t^i = \mathbb{E}\big[\| f_\theta(s_t, a_t) - \mathbb{E}_\theta[f_\theta(s_t, a_t)] \|_2^2 \big]$. Consistently with the goal outlined in BAMDPs, Houthooft et al. (2016) models the dynamics via Bayesian Neural Networks (Blundell et al., 2015) and proposes to use Information Gain (IG) (Lindley, 1956; Rainforth et al., 2023), or relative entropy, between the old and the updated models' parameters $\theta \in \Theta$ as Bayesian intrinsic reward (Kolter and Ng, 2009), defined as:

$$\begin{aligned} \mathrm{IG}_\theta(\xi_t, s_{t+1}) &= H[p(\theta|\xi_t)] - H[p(\theta|\xi_t, s_{t+1})] = \\ &= \mathbb{E}_{p(\theta|\xi_t, s_{t+1})}[\log p(\theta|\xi_t, s_{t+1})] - \mathbb{E}_{p(\theta|\xi_t)}[\log p(\theta|\xi_t)]\,, \end{aligned} \tag{1}$$

where $\xi_t = (s_t, a_t)$ is the Markovian history and $H[p(x)] = \mathbb{E}[-\log p(x)]$ is the Shannon entropy (Shannon, 1948). Here $p(\theta|\xi_t)$ represents the prior distribution on θ at time t, such that $p(\theta|\xi_0) = p(\theta)$ when $t = 0$, and $p(\theta|\xi_t, s_{t+1})$ is instead the posterior distribution that gets updated only after the next state s_{t+1} is revealed to the agent.

2.2 Why Information Gain Based Exploration Bonuses?

Employing information gain as an intrinsic reward $r_t^i = \mathrm{IG}_\theta(\xi_t, s_{t+1})$ has some desirable properties towards the solution of the BAMDP problem. Intuitively, $\mathrm{IG}_\theta(\cdot)$ captures epistemic uncertainty (Hüllermeier and Waegeman, 2021; Wimmer et al., 2023) associated with the dynamics parameters $\theta \in \Theta$. This can be seen from the decomposition of the Shannon entropy $H(\theta)$ (Cover, 1999), which is a measure of total uncertainty in θ, with respect to another variable Y: $H(\theta) = H(\theta|Y) + I(\theta; Y)$. Here, the conditional entropy $H(\theta|Y)$ quantifies the residual uncertainty after observing all the realization of Y, $H(\theta|Y) = -\sum_{k=1}^{K} Y_k \log Y_k$, i.e., aleatoric uncertainty. The mutual information component, $I(\theta; Y)$, measuring the expected information gained about one variable by observing the other, incorporates all the epistemic components. Conditioning everything on data $\mathcal{D}^n$, we have

$$I(\theta; Y|\mathcal{D}^n) = H(\theta|\mathcal{D}^n) - H(\theta|Y, \mathcal{D}^n) = \mathrm{IG}_\theta(Y, \mathcal{D}^n)\,.$$

Furthermore, $r_t^i = \mathrm{IG}_\theta(\xi_t, s_{t+1})$ is guaranteed to progressively reduce as the agent visits more (s_t, a_t) pairs and updates its model of the environment. To demonstrate this, assume for simplicity

$\mathcal{S}, \mathcal{A} \subseteq \mathbb{R}$ and that the transition distribution $p_{\theta_0}(s'|s,a)$ with true parameters $\theta_0 \in \Theta$ follows a zero-mean Markov dynamics with structural equation (Pearl, 2009):

$$S_{t+1} = f(S_t, A_t) + \varepsilon_t \,, \quad \text{where} \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2)\,, \tag{2}$$

where $\sigma^2 \in \mathbb{R}^+$. From a Bayesian modeling perspective, we want to perform inference on the parameters $\theta_0 = (f_0, \sigma_0^2)$ by placing a prior $p(\theta_0) \in \mathcal{P}$ and derive a posterior $p(\theta|\mathcal{D}^n) \triangleq p(\theta|(s,a)^n)$, used in the IG$_\theta$ formulation. Suppose we choose to model $f(\cdot)$ with a GP (Rasmussen et al., 2006) prior: $f|\omega \sim \mathcal{GP}\big(0, C(\cdot,\cdot|\omega)\big)$, where $l \in \mathbb{R}^+$ and $C(\cdot,\cdot|\omega)$ is a kernel covariance function with hyper-parameters ω. Then, under the assumptions of $\theta_0 \in$ KL-support$(p(\theta))$ and 'testability' (Schwartz, 1965; Ghosal et al., 1999) discussed in the proof, we have that:

Proposition 2.1 (Consistency)**.** *Assume true model (2) with $\theta_0 = (f_0, \sigma_0^2)$, prior $p(\theta) \in \mathcal{P}$ and posterior $p(\theta|(s,a)^n)$. Given conditions for weak posterior consistency (discussed in the proof), such that for $\epsilon > 0$,*

$$p\big(\theta \in \Theta : d\big((f.\sigma), (f_0, \sigma_0)\big) > \epsilon \mid (s,a)^n\big) \overset{P_{\theta_0}}{\to} 0\,,$$

as $n \to \infty$, then $r_t^i = IG_\theta(s_t, a_t, s_{t+1}) \overset{P_{\theta_0}}{\to} 0$ as $n \to \infty$.

Under the mild assumptions of Proposition 2.1, we can guarantee that the exploration incentive $r_t^i = \text{IG}_\theta(\cdot)$ naturally fades once the agent has sufficiently learnt $\theta \in \Theta$ via collecting samples of (s,a) pairs. Notice that this implies that the intrinsically sub-optimal value function $V^{*,augm}$ defined by the augmented rewards $r_t = r_t^e + \eta_i r_t^i$ converges to the optimal one defined only by r_t^e, $V^{*,augm} \overset{P}{\to} V^*$, as shown in the appendix. We emphasize also that the conditions of Proposition 2.1 are in fact satisfied for a larger set of Bayesian models other than just GPs, but GPs are flexible enough to leave the functional form of f unspecified. We can further derive contraction rates ϵ_n (Ghosal and Van der Vaart, 2017) at which $r_t^i = \text{IG}_\theta(\cdot)$ converges by imposing functional form restrictions on $f_0 \in \mathcal{F}$. If we assume that: $\mathcal{X} = \mathcal{S} \times \mathcal{A} = [0,1]^{|\mathcal{A}|+1}$; $f_0 \in C^\alpha(\mathcal{X}) \cap H^\alpha(\mathcal{X})$, where $\mathcal{C}^\alpha(\cdot)$ is the Hölder space and $H^\alpha(\cdot)$ is the Sobolev space of order α; $C(x,y) = \omega_1 \|x-y\|^\alpha K_\alpha(\omega_2\|x-y\|)$ in $f|\boldsymbol{\omega} \sim \mathcal{GP}\big(0, C(\cdot,\cdot|\boldsymbol{\omega})\big)$ is the Matérn kernel, then we have (Van Der Vaart and Van Zanten, 2011):

Proposition 2.2 (Contraction Rates)**.** *Under the same assumptions of 2.1, if $(f_0, \sigma_0) \in C^\alpha(\mathcal{X}) \cap H^\alpha(\mathcal{X}) \times [c,d]$, where $\mathcal{X} = [0,1]^{|\mathcal{A}|+1}$, and $f|\boldsymbol{\omega} \sim \mathcal{GP}\big(0, C(\cdot,\cdot|\boldsymbol{\omega})\big)$ where $C(\cdot,\cdot|\boldsymbol{\omega})$ is the Matérn kernel, then:*

$$IG_\theta(s_t, a_t, s_{t+1}) \overset{P_0^\infty}{\to} 0 \quad as \quad n \to \infty\,,$$

at the optimal minimax rate $\epsilon_n = n^{-\frac{1}{(2+|\mathcal{X}|/\alpha)}}$ (Yang and Barron, 1999).

The optimal minimax rate ϵ_n achieved in Proposition 2.2 typically pertains to a purely 'passive' sampling strategy, and can potentially be improved under certain conditions through active sampling (Willett et al., 2005; Castro and Nowak, 2008; Hanneke and Yang, 2015). We include a brief discussion on this in the supplementary material. Also, $V^{*,augm} \overset{P}{\to} V^*$ occurs at the same rate. Notice that, e.g., convergence is not guaranteed for r_t^i defined as prediction error $r_t^i = \frac{\eta}{2}\|f_\theta(s_t,a_t) - s_{t+1}\|_p$ (Stadie et al., 2015; Pathak et al., 2017) in stochastic environments. This phenomenon is also known as the "noisy TV problem" (Burda et al., 2018b), where the agent is erratically attracted by purely aleatoric noise when it cannot distinguish between epistemic and aleatoric uncertainty.

3 Bayesian Active Exploration

In reactive *curiosity*, the action at time t is sampled from a policy $a_t \sim \pi_\psi(s_t)$ whose parameters ψ are learnt according to updates based on the historical transitions $\{(s_k, a_k, r_k^e + r_k^i)\}_{k=0}^{t-1}$. In the active case instead (Shyam et al., 2019; Tschantz et al., 2020), a_t is sampled from a policy $\pi_\psi(s_t)$, whose parameters ψ are learnt using the future J transitions predicted via the dynamics model, $\{(\hat{s}_k, \hat{a}_k, \hat{r}_k^i)\}_{k=t}^{t+J}$. This means that in our information-theoretic setting, IG$_\theta(\xi_t, s_{t+1})$ cannot be computed in full form, and we have to resort to an expected value. From here onwards, we will assume for simplicity that the parameter σ^2 is fixed, i.e., estimated via maximum-likelihood, and will focus on posterior inference on $\theta = f(\cdot)$ only which is the main quantity of interest.

3.1 Expected Information Gain

As a predictive surrogate for IG_θ on $\theta = f(\cdot)$, one can make use of a type of *Expected Information Gain* (Bernardo, 1979; Pukelsheim, 2006; Foster, 2021; Rainforth et al., 2023) acquisition function, for a predictive relative entropy minimization search over the candidates s_{t+1} (Hernández-Lobato et al., 2014). Grouping again the observable Markov history $\xi_t = (s_t, a_t)$, EIG_θ can be defined as:

$$\begin{aligned}\text{EIG}_\theta(\xi_t) &= \mathbb{E}_{p_\theta(s_{t+1}|\xi_t)}\big[\text{IG}_\theta(\xi_t, s_{t+1})\big] = \mathbb{E}_{p(\theta|\xi_t)p(s_{t+1}|\xi_t,\theta)}\big[\log p(\theta|\xi_t, s_{t+1}) - \log p(\theta|\xi_t)\big] = \\ &= \mathbb{E}_{p(\theta|\xi_t)p(s_{t+1}|\xi_t,\theta)}\big[\log p(s_{t+1}|\xi_t, \theta) - \log p(s_{t+1}|\xi_t)\big] = \\ &= \mathbb{E}_{p(\theta|\xi_t)p(s_{t+1}|\xi_t,\theta)}\big[H[p(s_{t+1}|\xi_t)] - H[p(s_{t+1}|\xi_t, \theta)]\big]\end{aligned}$$

where we use the fact that $p(\theta|\xi_t, s_{t+1}) \propto p(\theta|\xi_t)p(s_{t+1}|\xi_t, \theta)$ by Bayes rule and $p(s_{t+1}|\xi_t) = p(s_{t+1}|\xi_t, \theta)$ by marginalization. Full derivation is provided in the appendix. In the active learning literature, EIG, in the form of the third equivalence above, is also known as Bayesian Active Learning by Disagreement (BALD) (Houlsby et al., 2011; Gal et al., 2017; Kirsch et al., 2019; Smith et al., 2023), and is often used as an acquisition function for new iid data batches. Notice that if $\text{IG}_\theta(\cdot) \overset{P}{\to} 0$ as per results in Section 2.2, then trivially $\text{EIG}_\theta(\cdot) \overset{P}{\to} 0$. We highlight also that in cases where $p_\theta(s_{t+1}|s_t, a_t)$ is approximated via a (multivariate) Gaussian distribution, as with a GP model (and differently than Shyam et al. (2019)), EIG can be further simplified for faster computation. This is because the entropy of a (multivariate) Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ can be reduced to $H[p(x)] = 0.5\,|\mathcal{S}|\,(1 + \log(2\pi)) + 0.5 \log(\det(\Sigma))$. Thus, if $p(s_{t+1}|\xi_t)$, where $\xi_t = (s_t, a_t)$, is Gaussian, $\text{EIG}_\theta(\xi_t)$ can be simplified to:

$$\begin{aligned}\text{EIG}_\theta(\xi_t) &= \mathbb{E}_{p_\theta(s_{t+1}|\xi_t)}\big[\text{IG}_\theta(\xi_t, s_{t+1})\big] = \mathbb{E}_{p(\theta|\xi_t)p(s_{t+1}|\xi_t,\theta)}\big[H[p(s_{t+1}|\xi_t)] - H[p(s_{t+1}|\xi_t, \theta)]\big] \\ &= \frac{1}{2}\Big(\log\det(\mathbb{V}[p(s_{t+1}|\xi_t)]) - \mathbb{E}_{p(\theta|\xi_t)}\big[\log\det(\mathbb{V}[p(s_{t+1}|\xi_t, \theta)])\big]\Big)\,, \qquad (3)\end{aligned}$$

where $\mathbb{V}[p(\cdot)]$ denotes the variance of distribution $p(\cdot)$. Full derivation is provided in the supplementary material.

3.2 Predictive Multi-Step Search

One outstanding issue with the use of EIG_θ as a transition acquisition function (Kearns et al., 2002) lies in the fact that computing EIG_θ in relation to the next state s_{t+1} only might result in short-term 'myopic' exploration, i.e., only looking at one-step-ahead predictive uncertainty potentially makes the agent miss out on higher uncertainty regions that are located further away from the current state s_t. One way to overcome this problem is via a multi-step predictive search (or a predictive Monte-Carlo Tree Search (Browne et al., 2012; Ghavamzadeh et al., 2015; Shyam et al., 2019; Fountas et al., 2020)). In our case, the predictive multi-step procedure consists in sampling trajectories from the posterior predictive $p_\theta(s_{t+1}|s_t, a_t)$ obtained from the dynamics model and compute the associated predicted intrinsic rewards $\hat{r}^i_t = \text{EIG}_\theta(\cdot)$. This procedure can be summarized as follows:

1. A sample of K actions is drawn from the current policy $a^{(k)}_t \sim \pi_\upsilon(s_t)$ (or initially at random)
2. Given s_t and each sampled action $a^{(k)}_t$, a sample of J-length trajectories $\tau_t = \{(s_{t+j}, a^{(k)}_{t+j}, s_{t+1+j}\}^J_{j=0}$ is drawn from the model's posterior $p_\theta(s_{t+1}|s_t, a_t)$
3. Cumulative $\sum_{t=T}^{T+J} \hat{r}^i_{\tau_t} = \sum_{t=T}^{T+J} \text{EIG}_\theta(\tau^{(k)}_t)$ is computed for the K sampled trajectories $\tau^{(k)}_t$ and the extrinsic rewards r^e_τ
4. Update policy π_υ using the trajectories $\tau^{(k)}_t$ via any policy gradient method (Sutton et al., 1999; Kakade, 2001; Schulman et al., 2017)

In denser reward environment, the procedure can also be coupled with a (extrinsic) reward model (Hafner et al., 2019; Sekar et al., 2020) $r^e_t = f_\psi(s_t, a_t)$, so that we can use $f_\psi(s, a)$ to predict also the future rewards r^e_{t+j} associated with the planning trajectories $(s_{t+j}, a^{(k)}_{t+j})^J_{j=0}$. To ease the computational burden of this multi-step procedure, we can control the predictive horizon J and/or introduce a decaying parameter $\zeta(t)$ to progressively diminish it. A pseudo-code algorithm of the full Bayesian Active Exploration procedure is given in the appendix.

4 Scalable Bayesian Dynamics Models

It is well-known that GPs tend to scale poorly both with sample size n and number of predictors $p = |\mathcal{S} \times \mathcal{A}|$ (in the dynamics model). The typical computational costs amount to $\mathcal{O}(n^3|\mathcal{S} \times \mathcal{A}|^3)$ for training and $\mathcal{O}(n^2|\mathcal{S} \times \mathcal{A}|^2)$ for test points (Quinonero-Candela and Rasmussen, 2005; Rasmussen et al., 2006). Thus, in order to make GPs scalable to larger samples and higher dimensional input, we propose two alternative models for the dynamics.

Stochastic Variational GP Dynamics Model We first consider using a Stochastic Variational GP approximation (SVGP) (Titsias, 2009; Hensman et al., 2015). The main idea behind SVGP is that instead of using the whole sample n, one can learn an optimal, lower size, subset of $m < n$ inducing points, $\{z_i\}_{i=1}^m$, where $\{z_i\}_{i=1}^m = \{(s_i, a_i)\}_{i=1}^m$, with the minimum decrease in performance and similar uncertainty quantification and generalization properties. The way these inducing points, together with the other GP parameters, are learnt is by introducing some associated variational parameters $\phi \in \Phi$ and a variational distribution $q_\phi(\mathbf{z}) \triangleq \mathcal{N}(\mu_z, \Sigma_z)$ (Titsias, 2009), such that the GP prior $p(f)$ becomes $p(f|\mathbf{z};\phi)p(\mathbf{z}|\phi)$, and an approximation to the true posterior on f is then given by:

$$p(f|s,a) \approx q_\phi(f|s,a) = \int_{\mathcal{Z}} p(f|z) q_\phi(z)\, dz \,,$$

which is obtained by marginalizing over the inducing points $\mathbf{z}$. One can then derive a variational Evidence Lower Bound (ELBO) loss on the model's marginal likelihood $p(s'|s,a;f)$, which reads:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}\Big[\mathbb{E}_{q_\phi(z)}\big[\mathbb{E}_{p(f|z)}[\log p(s'|s,a;f)]\big] - \beta\, \text{KL}\big(q_\phi(z)\,||\,p_\theta(z)\big)\Big] \,, \tag{4}$$

where KL is the Kullback-Leibler divergence. The parameters of the SVGP are learnt by minimizing the surrogate loss in (4). SVGP is much more scalable than standard GP in that it scales by construction with a $\mathcal{O}(m^3|\mathcal{S} \times \mathcal{A}|^3)$ and $\mathcal{O}(m^2|\mathcal{S} \times \mathcal{A}|^2)$ training and test cost respectively (Hensman et al., 2013; Jankowiak et al., 2020), and still allows taking advantage of the computational simplification in (3). Moreover, we can prove that the convergence results pertaining to the 'reactive' use of $r_t^i = \text{IG}_\theta(\cdot)$ presented in Section 2.2 still hold for the SVGP case. Suppose we have variational SVGP posterior $q_\phi(\theta|(s,a)^n)$, full-sample covariance matrix $C_f(\cdot,\cdot)$, covariance matrix on the m inducing points $Q_f(\cdot,\cdot) = C_{fz}C_{zz}^{-1}C_{zf}$, and $\|A\|_2$ which denotes the spectral norm of matrix A. Then we can derive that (Nieman et al., 2022):

Proposition 4.1. *(Contraction SVGP) Assume that conditions such that $p(\theta|\xi_t^n) \to \delta_{\theta_0}$ at rate ϵ_n hold. If additionally the following holds:*

$$\mathbb{E}_x\|C_{ff} - Q_{ff}\| \le c, \;\; \textit{and} \;\; \mathbb{E}_x\, tr(C_{ff} - Q_{ff}) \le cn\epsilon_n^2$$

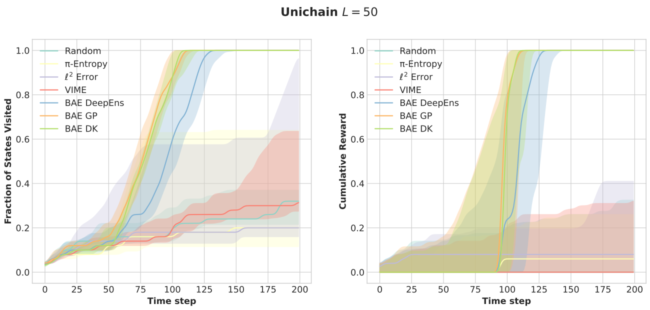
then the $IG_{\theta,q}(\cdot)$ associated with $q_\phi(\theta|(s,a)^n)$ is such that $IG_{\theta,q}(\cdot) \xrightarrow{P} 0$ at the same rate ϵ_n.

Additionally, as a corollary result from the equation above (Nieman et al., 2022), we can also show that:

Proposition 4.2. *(Matérn SVGP Contraction) Given $f_0 \in C^\alpha(\mathcal{X}) \cap H^\alpha(\mathcal{X})$ and $f \sim \mathcal{SVGP}\big(0, Q(\cdot,\cdot)\big)$ where $Q(\cdot,\cdot)$ is a Matérn kernel on $\mathcal{X} = [0,1]^{|\mathcal{S}\times\mathcal{A}|}$. The SVGP variational posterior's $IG_{\theta,q}(\xi_t, s_{t+1}) \to 0$ at the optimal rate $n^{-\frac{1}{(2+p/\alpha)}}$, for inducing points growing as $m = m_n \ge n^{p/(p+2\alpha)}$.*

Proofs and detailed assumptions are discussed in the appendix. There is one main drawback in the use of SVGPs, i.e., they still scale poorly with high-dimensional $\mathcal{S} \times \mathcal{A}$ and they do not allow for non-tabular inputs (e.g., images, graphs).

Deep Kernel Dynamics A way of coping with high dimensional $|\mathcal{S} \times \mathcal{A}|$ whilst keeping a Bayesian modeling approach and a Gaussian dynamics is via deep kernels (Wilson et al., 2016a). With deep kernels we can model the dynamics $p_\theta(s_{t+1}|s_t, a_t)$ by first mapping s_t, to a lower dimensional latent representation space h_t, $f_h : \mathcal{S} \to \mathcal{H}$, through a deep learning architecture. Then, the function to predict the next state, $f_s : \mathcal{H} \times \mathcal{A} \to \mathcal{S}$, can be assigned a SVGP prior, such that $p_\theta(s_{t+1}|s_t, a_t) = p_{\theta_2}(s_{t+1}|h_t)p_{\theta_1}(h_t|s_t, a_t)$; where $\theta_1 \in \Theta_1$ and $\theta_2 \in \Theta_2$ are the neural network and SVGP parameters, respectively (Bohn et al., 2019). The deep kernel parameters (θ_1, θ_2) are learnt through an end-to-end pipeline by minimizing a unique final ELBO loss on s_{t+1} similar



Model	States Fraction
Random	0.30 ± 0.02
π-Expl	0.37 ± 0.08
ℓ^2 Error	0.49 ± 0.09
VIME	0.43 ± 0.05
BAE DeepEns	0.99 ± 0.01
BAE GP	0.99 ± 0.01
BAE DK	0.99 ± 0.01

Model	Rewards
Random	0.15 ± 0.04
π-Expl	0.23 ± 0.08
ℓ^2 Error	0.28 ± 0.09
VIME	0.22 ± 0.06
BAE DeepEns	0.99 ± 0.01
BAE GP	0.99 ± 0.01
BAE DK	0.99 ± 0.01

Figure 1: Results over 20 replications on the $L = 50$ states noisy unichain environment, depicting the median and 75th-25th error bands of cumulative fraction of states visited (left plot) and of cumulative rewards $G_t = \sum_{t=0}^{T} r_t^e$ (middle plot). The tables report mean and 90% Monte Carlo standard error of the cumulative fraction of states visited and cumulative rewards at termination.

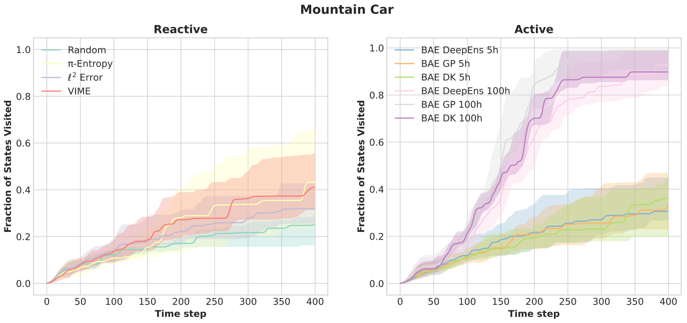
to (4) (Wilson et al., 2016b). Deep kernels necessitate only one forward pass, contrary to deep ensembles (Lakshminarayanan et al., 2017; Shyam et al., 2019), which require as many passes as the ensemble's dimension. In order to ensure that in the deep kernels architecture the desirable uncertainty quantification properties of GPs are not hindered by the over-confidence of neural nets (Guo et al., 2017; Ober et al., 2021) we incorporate a bi-Lipschitz constraint via spectral normalization on $f_h(\cdot)$, which we choose to be a fully-connected ResNet, that avoids feature collapse as in van Amersfoort et al. (2021). The last SVGP layer in the deep kernel model now scales at a $\mathcal{O}(m^3|\mathcal{H}|^3)$ training and $\mathcal{O}(m^2|\mathcal{H}|^2)$ test cost, where $|\mathcal{H}|$ is directly controllable. Contraction rates for deep kernels are notoriously difficult to prove due to the non-linearities introduced by the deep learning architecture, and thus present a challenging open research question.

5 Experiments

In this section we empirically validate some of the properties of Bayesian Active Exploration outlined in earlier sections. The environments we consider are generally characterized by sparse rewards, and thus would benefit from applying model-based exploration strategies and planning ahead. We employ Proximal Policy Optimization (PPO) (Schulman et al., 2017) as a common baseline RL algorithm underlying all the exploration methods we compare. The extensive pool of methods we compare throughout the experiments include: i) Purely random action exploration (**Random**); ii) Policy entropy $H(\pi_\psi(s_t))$ regularizer that upweights more stochastic policies, as part of the PPO loss function (**π-Entropy**); iii) Reactive ℓ^2 prediction error as intrinsic reward (Stadie et al., 2015; Pathak et al., 2017) (**ℓ^2 Error**); iv) Reactive IG$_\theta(\cdot)$ intrinsic reward, coupled with a Bayesian Neural Network dynamics model, as originally proposed in Houthooft et al. (2016) (**VIME**); v) Bayesian Active Exploration, coupled with an underlying deep ensemble dynamics model, as proposed by Shyam et al. (2019) (**BAE DeepEns**); vi) Bayesian Active Exploration coupled with SVGP prior described in Section 4 (Titsias, 2009; Hensman et al., 2013) (**BAE GP**); vii) Bayesian Active Exploration coupled with SV deep kernels, as described in Section 4 (Wilson et al., 2016b) (**BAE DK**). Note that in the case of **BAE DeepEns**, the inferred dynamics $p_\theta(s_{t+1}|s_t, a_t)$ is no longer Gaussian, but rather a mixture of Gaussians $p_\theta^{(m)}(s_{t+1}|s_t, a_t)$ indexed by the ensemble dimension $m \in \{1, ..., M\}$. As we can no longer use the simplification derived in Section 3.1, we need to resort to the non-parametric Jensen-Rényi Divergence to approximate EIG$_\theta(\cdot)$ (Shyam et al., 2019). Further information about the models employed can be found in the appendix.

5.1 Unichain Environment

The first environment we consider is a simple Markov unichain sequence of $L = 50$ states (Puterman, 2014), that allows us to showcase the properties of BAE. Versions of the unichain environment can also be found in other works (Osband et al., 2016; Shyam et al., 2019). We define the action space to be $\mathcal{A} = \{0, 1, 2\}$, where the discrete actions are {`go-left`, `stay`, `go-right`}. We use a two variables continuous representation for the discrete $\mathcal{S}$ and also introduce noise in the



Model	**Rewards**
Random	-399.0 ± 0.00
π-Entropy	-399.0 ± 0.00
ℓ^2 Error	-399.0 ± 0.00
VIME	-390.9 ± 88.1
BAE DeepEns 5h	-391.0 ± 88.1
BAE GP 5h	-399.0 ± 0.00
BAE DK 5h	-399.0 ± 0.00
BAE DeepEns 100h	-307.5 ± 88.2
BAE GP 100h	-152.9 ± 21.6
BAE DK 100h	-169.9 ± 19.7

Figure 2: Results on the discrete Mountain Car environment, over 20 replications. The plots report the median cumulative fraction of the car's visited positions per time step, together with the 75th and 25th error bands. The table reports the mean cumulative rewards at termination ($T = 400$), with the 90% Monte Carlo standard errors.

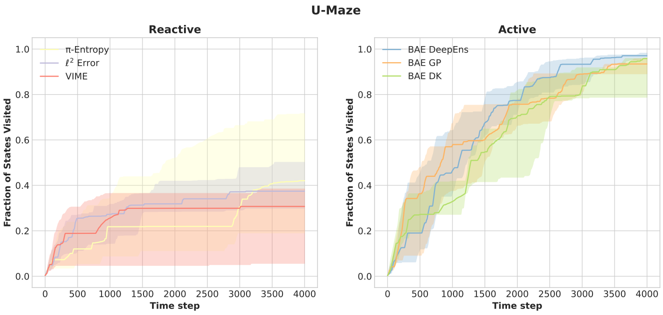
dynamics as $\varepsilon_t \sim \mathcal{N}(0, 0.01)$. The agent is initially spawned in state $s_0 = 2$, and the reward function $r^e : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is sparse, with state 1 paying off 0.001 (representing a 'reward trap'), last state 50 paying off 1, and the rest of the states 0. We provide more details about the unichain environment in the supplementary material, with with a visual representation and an additional experiment on $L = 100$ states. We set termination at $T = 200$ time steps or once the agent has reached the final state. We compare the pool of models detailed above. The BAE methods are run for $T_{warm} = 10$ steps at random initially, in order to gather enough data to estimate the posterior $p(\theta|(s_t, a_t))$ and consequently be able to compute the posterior predictive $p_\theta(s_{t+1}|s_t, a_t)$. For the predictive multi-step search, a collection of $K = 10$ different trees are grown, with a predictive horizon of $J = 200$ steps, to generate the trajectories $\{\tau_J^{(k)}\}_{k=1}^K$. Performance is measured via returns $G_t = \sum_{t=0}^T r_t^e$, and the cumulative fraction of states covered by the agent. Results on both these measures over 20 seeded replications of the experiment are reported in Figure 1's plots, where we plot the median plus 75th-25th error bands, and tables indicating the mean and 90% Monte Carlo standard errors of the two measures at termination. BAE methods are demonstrated to be able to solve the Unichain $L = 50$ task (indicated by $r_t^e = 1$) in just around 100 time steps, single episode, while reactive methods struggle due to their sample inefficiencies.

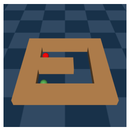
5.2 Mountain Car

The second environment we consider is the discrete version of Mountain Car (Towers et al., 2023), where $(\mathcal{S} \subseteq \mathbb{R}^2, \mathcal{A} = \{0, 1, 2\})$. Rewards are equal to $r_t^e = -1$ for each step t taken, while $r_t^e = 100$ is assigned when the agent reaches the top-right of the mountain. Using this setup we measure cumulative fraction of coverage of the possible positions of the car, at each time step t. In addition we also measure final cumulative rewards $G_t = \sum_{t=0}^T r_t^e$ at termination, which we set to $T = 400$, or when the car reaches the top of the mountain as the intended solution to the task. The warm up period for active methods is set to be the first $T_{warm} = 50$ of the total $T = 400$ steps.

Myopic Horizon. Beside the default set of methods detailed above, we also compare two sets of different BAE methods specifications. Namely, one specification where we set predictive horizon $J = 100$ (100h), and a more 'short-sighted' one where the horizon is reduced to just $J = 5$ steps ahead (5h). The number of predictive trees/independent trajectories sampled $\{\tau_J^{(k)}\}_{k=1}^K$ is kept equal instead ($K = 10$).

Results over $B = 20$ replications on cumulative coverage fraction of the car's position are reported in Figure 2's plots on the left, while the table on the right reports cumulative rewards $G_t = \sum_{t=0}^T r_t^e$ at termination. As demonstrated already in the unichain experiment, BAE methods outperform reactive ones. However, this is true only for the sufficiently deep predictive horizon of $J = 100$. The short-sighted $J = 5$ horizon specification, whilst more computationally appealing, fails to reach epistemic uncertainty regions further apart from the spawn location, $s_0 \sim \text{Uniform}([-0, 6, -0.4])$. Note that BAE methods with $J = 100$ solve the task in approximately 250-300 time steps in a single episode which demonstrates their high sample efficiency.





Model	States Fraction
No Expl	0.44 ± 0.09
ICM	0.41 ± 0.05
VIME	0.33 ± 0.10
MAX	0.92 ± 0.03
BAE-GP	0.84 ± 0.07
BALE	0.82 ± 0.09

Figure 3: Results on the U-Maze continuous control environment over 15 seeded replications. The plots report the median cumulative fraction of the agent's (green ball) visited positions per time step, together with the 75th and 25th error bands. In the table we report the mean cumulative fraction of maze coverage at termination ($T = 4000$), with the 90% Monte Carlo standard errors. The top right corner features a visual rendition of the U-Maze structure, where the red ball is the task objective.

5.3 Maze Environments

The last set of experiments features two different types of 2D Gymnasium Robotics maze structures (Fu et al., 2020; de Lazcano et al., 2023), where the agent's task is to explore the maze environment and collect a goal object located somewhere in the maze. These are continuous control environments with ($\mathcal{S} \subseteq \mathbb{R}^4, \mathcal{A} \subseteq \mathbb{R}^2$). The reward function is sparse, as the agents receive $r_t^e = 1$ only when they reach the objective, otherwise they receive $r_t^e = 0$. As these maze environments are considerably more complex, termination is set to $T = 4000$, or once the agent collects the goal, and the warm-up period for active methods is set to $T_{warm} = 500$. The number of predictive multi-step trajectories is set to $K = 10$, and their length to $J = 20$, to ease the computational burden.

U-Maze. The first maze is a simple U-shaped maze, where the agent (green ball) is spawned at one end of the maze and must reach the other end of the maze to collect the goal item (red ball) (see Figure 3). Similarly to the previous experiments, we measure cumulative fraction of maze coverage at every time step and at termination. These results, computed over 15 seeded replications, are reported in Figure 3. Again, we observe how reactive methods are orders of magnitude slower in exploring the maze as they barely reach 40% cumulative maze coverage approximately at termination $T = 4000$. BAE methods all approach nearly 90% of the maze coverage at termination even with a 'medium-sighted' predictive horizon of $J = 20$.

Medium Maze. We consider a second 2D medium-size, squared open maze featuring random obstacles in the space (i.e., columns and wall ledges). The agent starts in one of corners and has to reach the opposite corner to gather the objective (see Figure 8 in the appendix). Results over 10 runs are reported in Table 1, where performance is measured as the average number of steps required to reach the goal and the end-state rewards, where $s_T = 1.0$ indicates task is solved (termination is at $T = 5000$). BAE methods (with $J = 20$) again outperform reactive ones, as they are able to solve the task in all the different runs, with BAE DK being on average the first methods to reach the goal.

6 Conclusions

In this work, we studied the problem of data-efficient exploration in MBRL, by formalizing the paradigm of *Bayesian Active Exploration*. We proved, both theoretically and empirically, the associated benefits in environments where exploration is costly and where rewards are sparse. We also proposed two alternative models for the dynamics, SVGP and DK, that guarantee better scalability. We note that BAE trades off extremely data-efficient exploration and planning with higher computational costs. While, as our results show, this is incredibly beneficial in settings where exploration is costly, other less sample efficient methods may be better suited for cases where data acquisition is cheap.

Model	**# Steps to Goal**	**s_T Rewards**
π-Entropy	3371.0 ± 508.3	0.60 ± 0.21
ℓ^2 Error	2821.6 ± 566.4	0.64 ± 0.20
VIME	3662.0 ± 481.9	0.36 ± 0.20
BAE DeepEns	1211.9 ± 334.4	1.00 ± 0.00
BAE GP	1304.2 ± 606.4	1.00 ± 0.00
BAE DK	863.8 ± 360.8	1.00 ± 0.00

Table 1: Results over 10 runs on Medium Maze, in terms of average number of steps needed to reach the goal object and end state s_T rewards, with 90% standard errors.

Acknowledgements

Research funded by the Defence Science and Technology Laboratory (Dstl) which is an executive agency of the UK Ministry of Defence providing world class expertise and delivering cutting-edge science and technology for the benefit of the nation and allies. The research supports the Autonomous Resilient Cyber Defence (ARCD) project within the Dstl Cyber Defence Enhancement programme.

References

Alvarez, M. A., Rosasco, L., Lawrence, N. D., et al. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266.

Andrew, A., Spillard, S., Collyer, J., and Dhir, N. (2022). Developing optimal causal cyber-defence agents via cyber security simulation. *arXiv preprint arXiv:2207.12355*.

Ball, P., Parker-Holder, J., Pacchiano, A., Choromanski, K., and Roberts, S. (2020). Ready policy one: World building through active learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 591–601.

Barron, A., Schervish, M. J., and Wasserman, L. (1999). The consistency of posterior distributions in nonparametric problems. *The Annals of Statistics*, 27(2):536–561.

Bates, E., Mavroudis, V., and Hicks, C. (2023). Reward shaping for happier autonomous cyber security agents. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pages 221–232.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. (2016). Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29.

Bernardo, J. M. (1979). Expected information as expected utility. *the Annals of Statistics*, pages 686–690.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR.

Bohn, B., Rieger, C., and Griebel, M. (2019). A representer theorem for deep kernel learning. *The Journal of Machine Learning Research*, 20(1):2302–2333.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012). A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43.

Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., and Efros, A. A. (2018a). Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. (2018b). Exploration by random network distillation. In *International Conference on Learning Representations*.

Burger, B., Maffettone, P. M., Gusev, V. V., Aitchison, C. M., Bai, Y., Wang, X., Li, X., Alston, B. M., Li, B., Clowes, R., et al. (2020). A mobile robotic chemist. *Nature*, 583(7815):237–241.

Carpentier, A., Lazaric, A., Ghavamzadeh, M., Munos, R., and Auer, P. (2011). Upper-confidence-bound algorithms for active learning in multi-armed bandits. In *International Conference on Algorithmic Learning Theory*, pages 189–203.

Castro, R. M. and Nowak, R. D. (2008). Minimax bounds for active learning. *IEEE Transactions on Information Theory*, 54(5):2339–2353.

Chua, K., Calandra, R., McAllister, R., and Levine, S. (2018). Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31.

Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

de Lazcano, R., Andreas, K., Tai, J. J., Lee, S. R., and Terry, J. (2023). Gymnasium robotics.

Deisenroth, M. and Rasmussen, C. E. (2011). Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472.

Draeger, A., Engell, S., and Ranke, H. (1995). Model predictive control using neural networks. *IEEE Control Systems Magazine*, 15(5):61–66.

Duff, M. O. (2002). *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. University of Massachusetts Amherst.

Foley, M., Hicks, C., Highnam, K., and Mavroudis, V. (2022). Autonomous network defence using reinforcement learning. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pages 1252–1254.

Foley, M., Wang, M., Hicks, C., Mavroudis, V., et al. (2023). Inroads into autonomous network defence using explained reinforcement learning. *arXiv preprint arXiv:2306.09318*.

Foster, A., Ivanova, D. R., Malik, I., and Rainforth, T. (2021). Deep adaptive design: Amortizing sequential bayesian experimental design. In *International Conference on Machine Learning*, pages 3384–3395. PMLR.

Foster, A., Jankowiak, M., Bingham, E., Horsfall, P., Teh, Y. W., Rainforth, T., and Goodman, N. (2019). Variational bayesian optimal experimental design. *Advances in Neural Information Processing Systems*, 32.

Foster, A. E. (2021). *Variational, Monte Carlo and policy-based approaches to Bayesian experimental design*. PhD thesis, University of Oxford.

Fountas, Z., Sajid, N., Mediano, P., and Friston, K. (2020). Deep active inference agents using monte-carlo methods. *Advances in neural information processing systems*, 33:11662–11675.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. (2020). D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Gal, Y., Islam, R., and Ghahramani, Z. (2017). Deep bayesian active learning with image data. In *International conference on machine learning*, pages 1183–1192. PMLR.

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

Ghosal, S., Ghosh, J. K., and Ramamoorthi, R. (1999). Posterior consistency of dirichlet mixtures in density estimation. *The Annals of Statistics*, 27(1):143–158.

Ghosal, S. and Van der Vaart, A. (2017). *Fundamentals of nonparametric Bayesian inference*, volume 44. Cambridge University Press.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. (2019). Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*.

Hanneke, S. et al. (2014). Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309.

Hanneke, S. and Yang, L. (2015). Minimax analysis of active learning. *J. Mach. Learn. Res.*, 16(1):3487–3602.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 282–290.

Hensman, J., Matthews, A., and Ghahramani, Z. (2015). Scalable variational gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360. PMLR.

Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. (2014). Predictive entropy search for efficient global optimization of black-box functions. *Advances in neural information processing systems*, 27.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. (2016). Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29.

Hüllermeier, E. and Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506.

Itti, L. and Baldi, P. (2009). Bayesian surprise attracts human attention. *Vision research*, 49(10):1295–1306.

Jankowiak, M., Pleiss, G., and Gardner, J. (2020). Parametric gaussian process regressors. In *International Conference on Machine Learning*, pages 4702–4712. PMLR.

Jarrett, D., Tallec, C., Altché, F., Mesnard, T., Munos, R., and Valko, M. (2023). Curiosity in hindsight: Intrinsic exploration in stochastic environments. In *International Conference on Machine Learning, ICML 2023*, volume 202, pages 14780–14816.

Kakade, S. M. (2001). A natural policy gradient. *Advances in neural information processing systems*, 14.

Kamthe, S. and Deisenroth, M. (2018). Data-efficient reinforcement learning with probabilistic model predictive control. In *International conference on artificial intelligence and statistics*, pages 1701–1710. PMLR.

Kearns, M., Mansour, Y., and Ng, A. Y. (2002). A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine learning*, 49:193–208.

Kirsch, A., Van Amersfoort, J., and Gal, Y. (2019). Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32.

Kolter, J. Z. and Ng, A. Y. (2009). Near-bayesian exploration in polynomial time. In *Proceedings of the 26th annual international conference on machine learning*, pages 513–520.

Ladosz, P., Weng, L., Kim, M., and Oh, H. (2022). Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lindley, D. V. (1956). On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005.

Mehta, V., Paria, B., Schneider, J., Neiswanger, W., and Ermon, S. (2022). An experimental design perspective on model-based reinforcement learning. In *International Conference on Learning Representations*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Nguyen, T. T. and Reddi, V. J. (2021). Deep reinforcement learning for cyber security. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):3779–3795.

Nieman, D., Szabo, B., and van Zanten, H. (2022). Contraction rates for sparse variational approximations in gaussian process regression. *Journal of Machine Learning Research*, 23(205):1–26.

Nikolov, N., Kirschner, J., Berkenkamp, F., and Krause, A. (2019). Information-directed exploration for deep reinforcement learning. In *International Conference on Learning Representations*.

Ober, S. W., Rasmussen, C. E., and van der Wilk, M. (2021). The promises and pitfalls of deep kernel learning. In *Uncertainty in Artificial Intelligence*, pages 1206–1216. PMLR.

Ortner, R. (2020). Regret bounds for reinforcement learning via markov chain concentration. *Journal of Artificial Intelligence Research*, 67:115–128.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. (2016). Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29.

Ostrovski, G., Bellemare, M. G., Oord, A., and Munos, R. (2017). Count-based exploration with neural density models. In *International conference on machine learning*, pages 2721–2730. PMLR.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32.

Parr, T. and Friston, K. J. (2017). Uncertainty, epistemics and active inference. *Journal of the Royal Society Interface*, 14(136):20170376.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR.

Pathak, D., Gandhi, D., and Gupta, A. (2019). Self-supervised exploration via disagreement. In *International conference on machine learning*, pages 5062–5071. PMLR.

Pearl, J. (2009). *Causality*. Cambridge university press.

Pukelsheim, F. (2006). *Optimal design of experiments*. SIAM.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Quinonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959.

Rahaman, R. et al. (2021). Uncertainty quantification and deep ensembles. *Advances in Neural Information Processing Systems*, 34:20063–20075.

Rainforth, T., Cornish, R., Yang, H., Warrington, A., and Wood, F. (2018). On nesting Monte Carlo estimators. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4267–4276.

Rainforth, T., Foster, A., Ivanova, D. R., and Smith, F. B. (2023). Modern bayesian experimental design. *arXiv preprint arXiv:2302.14545*.

Rasmussen, C. E., Williams, C. K., et al. (2006). *Gaussian processes for machine learning*, volume 1. Springer.

Ross, S., Chaib-draa, B., and Pineau, J. (2007). Bayes-adaptive pomdps. *Advances in neural information processing systems*, 20.

Sajid, N., Ball, P. J., Parr, T., and Friston, K. J. (2021). Active inference: demystified and compared. *Neural computation*, 33(3):674–712.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Schwartz, L. (1965). On bayes procedures. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 4:10–26.

Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., and Pathak, D. (2020). Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pages 8583–8592. PMLR.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Shyam, P., Jaśkowski, W., and Gomez, F. (2019). Model-based active exploration. In *International conference on machine learning*, pages 5779–5788. PMLR.

Smith, F. B., Kirsch, A., Farquhar, S., Gal, Y., Foster, A., and Rainforth, T. (2023). Prediction-oriented bayesian active learning. In *International Conference on Artificial Intelligence and Statistics*, pages 7331–7348. PMLR.

Sorg, J., Singh, S., and Lewis, R. L. (2010). Variance-based rewards for approximate bayesian reinforcement learning. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pages 564–571.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Stadie, B. C., Levine, S., and Abbeel, P. (2015). Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*.

Steiner, S., Wolf, J., Glatzel, S., Andreou, A., Granda, J. M., Keenan, G., Hinkley, T., Aragon-Camarasa, G., Kitson, P. J., Angelone, D., et al. (2019). Organic synthesis in a modular robotic system driven by a chemical programming language. *Science*, 363(6423):eaav2211.

Strens, M. (2000). A bayesian framework for reinforcement learning. In *ICML*, volume 2000, pages 943–950.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Tang, H., Houthooft, R., Foote, D., Stooke, A., Xi Chen, O., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. (2017). # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in neural information processing systems*, 30.

Tarbouriech, J. and Lazaric, A. (2019). Active exploration in markov decision processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 974–982. PMLR.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR.

Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE.

Towers, M., Terry, J. K., Kwiatkowski, A., Balis, J. U., Cola, G. d., Deleu, T., Goulão, M., Kallinteris, A., KG, A., Krimmel, M., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J. J., Shen, A. T. J., and Younis, O. G. (2023). Gymnasium.

Tschantz, A., Baltieri, M., Seth, A. K., and Buckley, C. L. (2020). Scaling active inference. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

van Amersfoort, J., Smith, L., Jesson, A., Key, O., and Gal, Y. (2021). On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint arXiv:2102.11409*.

Van Der Vaart, A. and Van Zanten, H. (2011). Information rates of nonparametric gaussian process methods. *Journal of Machine Learning Research*, 12(6).

van der Vaart, A. W. and van Zanten, J. H. (2008). Rates of contraction of posterior distributions based on Gaussian process priors. *The Annals of Statistics*, 36(3):1435 – 1463.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Y., Balakrishnan, S., and Singh, A. (2018). Optimization of smooth functions with noisy observations: Local minimax rates. *Advances in Neural Information Processing Systems*, 31.

Willett, R., Nowak, R., and Castro, R. (2005). Faster rates in regression via active learning. *Advances in Neural Information Processing Systems*, 18.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. (2016a). Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR.

Wilson, A. G., Hu, Z., Salakhutdinov, R. R., and Xing, E. P. (2016b). Stochastic variational deep kernel learning. *Advances in neural information processing systems*, 29.

Wilson, A. G. and Izmailov, P. (2020). Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708.

Wimmer, L., Sale, Y., Hofman, P., Bischl, B., and Hüllermeier, E. (2023). Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In *Uncertainty in Artificial Intelligence*, pages 2282–2292. PMLR.

Yang, Y. and Barron, A. (1999). Information-theoretic determination of minimax rates of convergence. *Annals of Statistics*, pages 1564–1599.

Yu, C., Liu, J., Nemati, S., and Yin, G. (2021). Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36.

A Proofs of Propositions

In order to prove the convergence properties of $\mathrm{IG}_\theta(\cdot)$ presented in the main body's propositions, we need to define a couple of necessary building blocks concepts (Ghosal and Van der Vaart, 2017). For this section, let us rename the prior distribution as $\pi(\theta)$, or shortly π, and the posterior as $\pi\big(\theta|(\xi_t)^n\big)$ for simplicity, where ξ_t^n is a sequence of samples $n \in \{1, 2, ...\}$ following likelihood distribution $\xi^n \sim p(\xi^n|\theta)$ with $\theta \in \Theta$ and $\theta \in \pi(\theta)$. The posterior distribution associated with n-th sample, $\pi_n\left(B \mid D^n\right)$, is obtained via Bayes theorem as

$$\pi_n\left(B \mid \xi^n\right) = \frac{\int_B p_n\left(\xi^n \mid \theta\right) d\pi_n(\theta)}{\int_\Theta p_n\left(\xi^n \mid \theta\right) d\pi_n(\theta)}, \qquad \text{where } B \subset \Theta,$$

as we assume that with increasing n, the data ξ^n provide more information about $\theta \in \Theta$.

A.1 Proof of Proposition 2.1

We start with the notion of posterior consistency. Posterior consistency loosely means that for $n \to \infty$, the posterior $\pi\big(\theta|(\xi_t)^n\big)$ converges weakly to the Dirac measure δ_{θ_0} in Θ, and this depends on the topology of Θ. More formally, let us assume that Θ is a metric space equipped with metric $d(\cdot, \cdot)$, then:

Definition A.1 (Posterior Consistency)**.** Given a prior distribution $\pi(\theta) \in \Pi$, a posterior distribution $\pi\big(\theta|(\xi_t)^n\big)$ is said to be consistent w.r.t. a true parameter $\theta_0 \in (\Theta, d)$ if

$$\pi\big(\theta \in \Theta : d(\theta, \theta_0) > \epsilon \mid \xi_t^n\big) \overset{P_{\theta_0}}{\to} 0 \quad \text{as} \quad n \to \infty\,.$$

One important consequence of posterior consistency that we are going to make use of relates to the consistency of estimators $\hat{\theta}_n = T(\xi^n)$ defined from the posterior, and it is stated as follows:

Proposition A.2 (Estimators Consistency)**.** *Assume $\Theta^* \subset \Theta$ is a subset such that posterior consistency holds for $\theta_0 \in \Theta^*$. The following facts are true:*

i) There exists $\exists\, \hat{\theta}_n = T(\xi^n)$ an estimator that is consistent for $\theta_0 \in \Theta^*$, i.e., $d(\hat{\theta}_n, \theta_0) \overset{P_{\theta_0}}{\to} 0$ when $\xi^n \sim p(\xi^n|\theta_0)$

ii) If Θ is convex and $d(\cdot, \cdot)$ is a bounded and convex distance function, then $\hat{\theta}_n = T(\xi^n)$ can be the posterior mean $\hat{\theta}_n = T(\xi^n) = \int \theta d\pi_n(\theta|\xi^n)$

iii) If $g : \Theta \to \Theta$ is a function continuous at θ_0, then $d\big(g(\hat{\theta}_n), g(\theta_0)\big) \overset{P_{\theta_0}}{\to} 0$.

Convexity of Θ indeed holds for the non-parametric regression problem defined by $Y_i = \theta(X_i) + \varepsilon_i$, where $f = \theta \in \Theta = C(\mathcal{X})$ and $X \in \mathcal{X}$. As for what requirements are needed for posterior consistency, we need intuitively that prior $\pi(\theta)$ do not exclude θ_0 from its support. Define the 'model' as the likelihood density function that generates samples $\xi_t^n \sim p_0 \in \mathcal{P}$, where $\mathcal{P}$ is a probability measure, and $p_0 \in \mathcal{P}$ being the true density. Schwartz (1965) derived conditions for posterior consistency based on whether the true data generating model p_0 belongs to the KL support of the prior π. In particular, we define

Definition A.3 (KL-support of $p(\theta)$)**.** By denoting as $d_{KL}(p, q)$ the KL-divergence between distributions p and q, p_0 is said to belong to the KL support of the prior π, written $p_0 \in \text{KL}(\pi)$, if

$$\forall \epsilon > 0, \quad \pi\big(p : d_{KL}(p_0, p) < \epsilon\big) > 0\,.$$

Now, define $U \subset \mathcal{P}$ as a open neighborhood of p_0 according to metric $d(\cdot, \cdot)$, then we intuitively have that the posterior $\pi(\cdot|\xi_t^n)$ is also consistent if and only if for every open neighborhood U of p_0, $\pi(U^c|\xi_t^n) \to 0$ (i.e., if all neighborhood around p_0 collapse to 0). This is formalized as follows (Schwartz, 1965; Ghosal and Van der Vaart, 2017). Given a neighborhood $U \subset \mathcal{P}$, then we can test hypotheses $H_0 : p = p_0$ versus $H_1 : p \in U$. Assume these exist a real function $\varphi_n = \varphi(\xi_1, ..., \xi_n) : \Xi \to [0, 1]$, representing the probability of rejecting H_0, such that $\mathbb{E}_{p_0}[\varphi_n] \to 0$ and $\sup_{p \in U} \mathbb{E}_p[1 - \varphi_n] \to 0$ (i.e., probability of rejecting H_0 goes to zero if $p = p_0$, and conversely probability of rejecting H_1 goes to zero when $p \neq p_0$). The idea of Schwartz (1965) theorem is that posterior consistency is guaranteed if the prior π assigns mass that is 'arbitrarily close' to the true model $p_0 \in \mathcal{P}$ and as n grows we can more correctly classify H_0 vs H_1. In addition, define P_0^∞ as the joint density of $\xi^n = (\xi_1, \xi_2, ...)$ under data generating model p_0. In full form then the weak consistency theorem reads:

Theorem A.4 (Schwartz (1965))**.** *Assume that $p_0 \in KL(\pi)$, and that for neighborhoods $U_n \subset \mathcal{P}$ of p_0 there are test functions φ_n satisfying the following requirements:*

$$\mathbb{E}_{p_0}\varphi_n \leq Be^{-bn}, \qquad \sup_{p \in U_n^c} \mathbb{E}_p\,(1 - \varphi_n) \leq Be^{-bn}$$

for some constants $b, B > 0$, then $\pi(U_n|\xi^n) \overset{P_0^\infty}{\to} 0$, or equivalently, $\pi(\theta|\xi_t^n) \overset{P_0^\infty}{\to} \delta_{\theta_0}$ via corollary result.

Now we revert back to our specific dynamics model case, having true parameters $\theta_0 = (f_0, \sigma_0^2) \in (\Theta, d)$. Assuming the data generating model p_0 described by equation (2), and assuming that

$p_0 \in KL(\pi)$, we have by Schwartz's theorem that $\pi(\theta|\xi_t^n) \overset{P_{\theta_0}}{\to} \delta_{\theta_0}$. As stated in Proposition 2.1 this translate into

$$p\big(\theta \in \Theta : d\big((f.\sigma), (f_0, \sigma_0)\big) > \epsilon \mid (s, a)^n\big) \overset{P_{\theta_0}}{\to} 0 \quad \text{as} \quad n \to \infty,$$

in our case. As per Proposition A.2 reported above then we know that we can construct an estimator $g(\hat{\theta}_n) = T(\xi^n)$ from the posterior $\pi(\theta|\xi^n))$ that is consistent for the parameter $g(\theta_0) \in \Theta$. Arbitrarily, we can pick $g(\hat{\theta}_n)$ to be exactly $T(\xi^n) = \text{IG}_\theta(\xi^n, s_{t+1})$. Then, according to Proposition A.2, we have $d\big(\text{IG}_\theta(\xi^n, s_{t+1}), \text{IG}_{\theta_0}(\xi^n, s_{t+1})\big) \overset{P_{\theta_0}}{\to} 0$ or

$$Pr\big(\theta \in \Theta : d\big(\text{IG}_\theta(\xi^n, s_{t+1}), \text{IG}_{\theta_0}(\xi^n, s_{t+1})\big) > \epsilon\big) \overset{P_{\theta_0}}{\to} 0 \quad \text{as} \quad n \to \infty\,.$$

However, since at convergence we have $\pi(\theta|\xi_t^n) \overset{P_0^\infty}{\to} \delta_{\theta_0}$, then the equivalence $p(\theta|\xi_t) = p(\theta|\xi_t, s_{t+1})$ holds (i.e., new datum s_{t+1} does not convey any additional information on $\theta \in \Theta$). This implies that $\text{IG}_{\theta_0}(\xi^n, s_{t+1})\big) = H[\delta_{\theta_0}] - H[\delta_{\theta_0}] = 0$, thus the convergence result is $H[p(\theta|\xi_t)] - H[p(\theta|\xi_t, s_{t+1})] \overset{P_{\theta_0}}{\to} H[\delta_{\theta_0}] - H[\delta_{\theta_0}] = 0$, or similarly $d\big(\text{IG}_\theta(\xi^n, s_{t+1}), 0\big) \overset{P_{\theta_0}}{\to} 0$.

A.1.1 Convergence to Optimal Value Function

As a corollary of Proposition 2.1 proved above, we can straightforwardly show that the value function $V^{*,augm}$ defined by the augmented rewards $r_t = r_t^e + \eta_i r_t^i$ is such that $V^{*,augm} \overset{P}{\to} V^*$. More formally, we define $V^{*,augm}$ as follows

$$V_t^{*,augm}(s, \theta) = \max_{a \in \mathcal{A}} \left[r^e(s, a) + \eta_i r^i(s, a; \theta) + \gamma \int_{\mathcal{S}, \Theta} p(s'|s, \theta, a) V_{t-1}^*(s', \theta') ds' d\theta' \right] \tag{5}$$

while the optimal value function associated with the original MDP's extrinsic rewards only is simplified to

$$V_t^*(s; \theta_0) = \max_{a \in \mathcal{A}} \left[r^e(s, a) + \gamma \int_{\mathcal{S}} p(s'|s, a; \theta_0) V_{t-1}^*(s'; \theta_0) ds' \right]$$

Wrapping this up in a corollary statement:

Corollary A.5 ($V^{*,augm}$ Convergence)**.** *Under the conditions for posterior consistency of Proposition 2.1, it is easy to see from Eq. (5) that we have $V_t^{*,augm}(s, \theta) \overset{P_{\theta_0}}{\to} V_t^*(s; \theta_0)$, as $r_t^i = IG_\theta(\cdot) \overset{P_{\theta_0}}{\to} 0$ and $\theta \overset{P_{\theta_0}}{\to} \theta_0$.*

A.2 Proof of Proposition 2.2

The intuition behind Proposition 2.2 is simply that, following the statement of Proposition 2.1, if we can prove a contraction rate ϵ_n holds for posterior convergence $\pi(\theta|\xi^n) \to \delta_{\theta_0}$, then in light of the proves in the above section, both $d\big(\text{IG}_\theta(\xi^n, s_{t+1}), 0\big) \overset{P_{\theta_0}}{\to} 0$ and $d(V^{augm,*}, V^*) \overset{P_{\theta_0}}{\to} 0$ happen at the same rate ϵ_n. To prove Proposition 2.2, we begin by defining contraction rates for a consistent posterior $\pi(\theta|\xi_t^n)$, where the true $\theta_0 \in (\Theta, d)$. Notice that posterior consistency always implies a contraction rate ϵ_n (Ghosal and Van der Vaart, 2017). We define

Definition A.6 (Contraction Rate)**.** A posterior $\pi(\theta|\xi_t^n)$ is said to contract to δ_{θ_0} at the rate $\epsilon_n \to 0$, if for every constant $\forall M > 0$:

$$\pi\big(\theta \in \Theta : d(\theta, \theta_0) > M\epsilon_n \mid \xi_n\big) \overset{P_{\theta_0}}{\to} 0 \quad \text{when} \quad \xi_n \sim p_n(\xi|\theta_0)\,.$$

A corollary of Proposition A.2 then straightforwardly holds, with respect to posterior estimators $\hat{\theta}_n = T(\xi^n)$ convergence rate, and states:

Corollary A.7 (Estimator Contraction)**.** *If posterior $\pi(\theta|\xi_t^n)$ contracts at rate ϵ_n (or faster) to $\theta_0 \in \Theta_0 \subset \Theta$, then there exists an estimator $\hat{\theta}_n = T(\xi^n)$ that is consistent for $\theta \in \Theta_0$ and converges at least as fast as ϵ_n.*

Assume now the dynamics model presented in Section 2.2 holds, that is:

$$S_{t+1} = f(S_t, A_t) + \varepsilon_t \,, \quad \text{where } \varepsilon_t \sim \mathcal{N}(0, \sigma^2) \,, \text{ and } \sigma^2 \in \mathbb{R}^+$$

and assume that $(f_0, \sigma_0) \in \mathcal{C}^\alpha(\mathcal{X}) \times [c, d]$, where: $\mathcal{X} = \mathcal{S} \times \mathcal{A}$ is a compact subset of $\mathbb{R}$; $\mathcal{C}^\alpha(\cdot)$ is the class of continuous functions with finite Hölder norm of order α; and $[c, d] \subset \mathbb{R}^+$. Notice that we assumed that $\mathcal{S}, \mathcal{A} \subseteq \mathbb{R}$, so that the target variable s_{t+1} is single-task continuous $S_{t+1} \in \mathbb{R}$, but the following would hold also for multi-task settings, just individually and independently for each task. Finally, assume that $f \sim \mathcal{GP}\big(0, C(\cdot, \cdot)\big)$, a GP prior where $C(\cdot, \cdot)$ is the isotropic squared exponential kernel, i.e., $C(x, y) = \exp\{-l^2 \|x - y\|^2\}$ with length parameter $l \in \mathbb{R}^+$. Then, one can prove that posterior $\pi(f, \sigma | \xi^n)$ contracts at the optimal minimax rate up to a log constant, that is:

Theorem A.8 (van der Vaart and van Zanten (2008)). *Assume $(f_0, \sigma_0) \in \mathcal{C}^\alpha(\mathcal{X}) \times [c, d]$, where $\mathcal{X}$ is a compact subset of $\mathbb{R}$, the dynamics model in (2), and $f \sim \mathcal{GP}\big(0, C(\cdot, \cdot)\big)$ prior where $C(\cdot, \cdot)$ is the squared exponential kernel. Then, denoting $p = |\mathcal{S} \times \mathcal{A}|$:*

$$\pi\left(\{(f, \sigma) : d\big((f.\sigma), (f_0, \sigma_0)\big) > \epsilon_n\} \mid \xi_t^n\right) \overset{P_0^\infty}{\to} 0 \quad \textit{as} \quad n \to \infty$$

where $\epsilon_n = n^{-\frac{1}{(2+p/\alpha)}} (\log n)^t$ with $t = 1 - \frac{1}{(2+4\alpha/p)}$.

Notice that with $(\log n)^t = 1$, the above is equal to the minimax rate (best rate of estimation) for functions in the class $\mathcal{C}^\alpha(\mathcal{X})$ (Yang and Barron, 1999). Since posterior consistency implies the existence of a contraction rate ϵ_n, then given that conditions for posterior consistency hold in Proposition 2.1, and given result in Corollary A.7, we have that $\text{IG}_\theta(\xi^n, s_{t+1}) \to 0$ at the same rate ϵ_n. Same hold for $V_t^{*,augm}(s, \theta) \to V_t^*(s; \theta_0)$, by extending Corollary A.5. Thus, this implies that under the condition of Theorem A.8 above, we have that $\text{IG}_\theta(\xi^n, s_{t+1}) \to 0$ and $V_t^{*,augm}(s, \theta) \to V_t^*(s; \theta_0)$ at same rate $\epsilon_n = n^{-\frac{1}{(2+p/\alpha)}} (\log n)^t$.

Moreover, Van Der Vaart and Van Zanten (2011) show that the optimal minimax rate $\epsilon_n = n^{-\frac{1}{(2+p/\alpha)}}$ (Yang and Barron, 1999) for posterior contraction can be also achieve instead by imposing a further restriction on f_0 and assuming the covariance function $C(\cdot, \cdot|\cdot)$ is a Matérn kernel. Define $H^\alpha(\mathcal{X})$ as the Sobolev space, then:

Theorem A.9 (Van Der Vaart and Van Zanten (2011)). *Given $f_0 \in C^\alpha(\mathcal{X}) \cap H^\alpha(\mathcal{X})$ and $f \sim \mathcal{GP}\big(0, C(\cdot, \cdot|\cdot)\big)$ where $C(\cdot, \cdot|\cdot)$ is a Matérn kernel on $\mathcal{X} = [0, 1]^p$, then the posterior $\pi(f|\cdot)$ contracts at rate $\epsilon_n = n^{-\frac{1}{(2+p/\alpha)}}$.*

Thus, under the assumptions specified by Theorem A.9, also $\text{IG}_\theta(\xi^n, s_{t+1}) \to 0$ and $V_t^{*,augm}(s, \theta) \to V_t^*(s; \theta_0)$ happen at the same rate $\epsilon_n = n^{-\frac{1}{(2+p/\alpha)}}$.

A.3 Proof of Proposition 4.1 and 4.2

Results of Proposition 2.1 and 2.2 above pertains to exact GP models of the dynamics, but these can be shown to hold also for Stochastic Variational GPs and their implied variational posterior $q_\phi(\theta|\xi^n)$, given assumptions on how well this approximate the full posterior $\pi(\theta|\xi^n)$. In particular, as recently shown by Nieman et al. (2022), one does not necessarily need that $q_\phi(\theta|\xi^n) \to \pi(\theta|\xi^n)$, or equivalently $d\big(q_\phi(\theta|\xi^n), \pi(\theta|\xi^n)\big) \to 0$, as implied by when number of inducing points $m \to n$. Instead, we need conditions on how 'distant' are the true covariance matrix $C_f(\cdot, \cdot) = \text{cov}_\pi(f, f)$ and the inducing points one $Q_f(\cdot, \cdot) = C_{fu} C_{uu}^{-1} C_{uf}$. Suppose the quantity $C_f(\cdot, \cdot) - Q_f(\cdot, \cdot)$ determines how well the sub-sample of inducing points approximate the prior distribution. Let $\|A\|_2$ be the spectral norm of matrix A and $\text{tr}(A)$ the trace of A.

Theorem A.10 (Nieman et al. (2022)). *Suppose that, under the necessary assumptions seen above (i.e., minimally $f_0 \in C^\alpha(\mathcal{X})$), the posterior distribution contracts $\pi(\theta|\xi_t^n) \to \delta_{\theta_0}$ at rate ϵ_n. Then if the following hold*

$$\mathbb{E}_x \|C_{ff} - Q_{ff}\| \leq C, \quad \textit{and} \quad \mathbb{E}_x \, \textit{tr}(C_{ff} - Q_{ff}) \leq C n \epsilon_n^2$$

the variational posterior contracts $q_\phi(\theta|\xi_t^n) \to \delta_{\theta_0}$ at the same rate ϵ_n.

Denote with $\text{IG}_{\theta,\phi}(\cdot)$, $V_t^{*,augm}(s, \theta, \phi)$ and $V_t^*(s, \theta, \phi)$ the IG and optimal value functions associated with the variational SVGP posterior $q_\phi(\theta|\xi^n)$. Assuming again the same dynamics of Eq. (2), but this time placing a $f \sim \mathcal{SVGP}\big(0, C(\cdot,\cdot|\cdot)\big)$ prior, then given Proposition 2.1 and 2.2 and the conditions detailed by Thm A.10, we have that $r^i_\phi = \text{IG}_{\theta,\phi}(\cdot) \to 0$ and $V_t^{*,augm}(s,\theta,\phi) \to V_t^*(s;\theta_0,\phi)$ at the full posterior convergence rate ϵ_n. Now, one can show that the conditions on $C_{ff} - Q_{ff}$ can essentially be translated into conditions on the rate of growth on the number of inducing points (to keep the approximation Q_{ff} arbitrarily close to C_{ff}). In particular Nieman et al. (2022) have shown that, for example, in the case described by Proposition 2.2 above (Matèrn kernel and $f_0 \in C^\alpha(\mathcal{X}) \cap H^\alpha(\mathcal{X})$), the optimal minimax contraction rate (Yang and Barron, 1999) is achieved if the number of inducing variables m scales at least as $n^{p/(p+2\alpha)}$ with full sample size. That is

Corollary A.11 (Nieman et al. (2022))**.** *Given $f_0 \in C^\alpha(\mathcal{X}) \cap H^\alpha(\mathcal{X})$ and $f \sim \mathcal{SVGP}\big(0, C(\cdot,\cdot|\cdot)\big)$ where $C(\cdot,\cdot|\cdot)$ is a Matérn kernel on $\mathcal{X} = [0,1]^p$. The variational posterior $\Psi(\theta|\xi^n_t) \to \delta_{\theta_0}$ contracts at the optimal rate $n^{-\frac{1}{(2+p/\alpha)}}$, for inducing points growing as $m = m_n \geq n^{p/(p+2\alpha)}$.*

This again implies that, under the same assumptions, $r^i_\phi = \text{IG}_{\theta,\phi}(\cdot) \to 0$ and $V_t^{*,augm}(s,\theta,\phi) \to V_t^*(s;\theta_0,\phi)$ contracts at the same rate $n^{-\frac{1}{(2+p/\alpha)}}$.

A.4 Brief Discussion on Active vs Passive Learning Rates

The posterior contraction rates ϵ_n derived in the results above all pertains to the classical case of 'passive' learning (Yang and Barron, 1999). We note that these can be improved under some conditions with an active sampling strategy (Hanneke et al., 2014; Hanneke and Yang, 2015), which is what the work ultimately advocates for the realm of data-efficient exploration in MBRL. In Willett et al. (2005), the authors show that the minimax convergence rates for regression cases where f_0 is a piecewise constant function can be strictly reduced from $\epsilon_n = n^{-1/|\mathcal{X}|}$ (passive) to $\epsilon_n = n^{-1/(|\mathcal{X}|-1)}$ (active). Castro and Nowak (2008) instead show that for classification problems with similar Hölder smooth decision boundaries the minimax lower bound convergence rate can be tightly improved from $\epsilon_n = n^{\kappa/(\kappa+\rho-1)}$ (passive) to $\epsilon_n = n^{\kappa/(\kappa+\rho-2)}$ (active), where $\rho = (|\mathcal{X}|-1)/\alpha$ and κ is the highest integer such that $\kappa < \alpha$. As a final example, results in Wang et al. (2018) show that if f_0 is strongly smooth and convex (e.g., as in Example 2 in their paper), with $\alpha = 2$, one can achieve a much better rate of $\epsilon_n = n^{-1/2}$ compared to the passive minimax, which in that example's case is $\epsilon_n = n^{-2/4+|\mathcal{X}|}$.

B Information Gain Derivations

In this second appendix section we include the full derivations of the Expected Information Gain (EIG) (Lindley, 1956; Bernardo, 1979; Rainforth et al., 2023) and the simplified version of EIG under a Gaussian dynamics model, encountered in Section 3.1. We assume the setup is the one described in the MDP definition of Section 2 in the main paper. We group again the Markovian history of the time step $t \to t+1$ in the transition $\tau \in \mathcal{T}$, represented by the current (s_t, a_t) pair, in the auxiliary variable $\xi_t = (s_t, a_t)$ for simplicity. The Information Gain (IG) of the full transition $t \to t+1$ composed by (s_t, a_t, s_{t+1}) is defined as

$$\begin{aligned} \text{IG}_\theta(\xi_t, s_{t+1}) &= H[p(\theta|\xi_t)] - H[p(\theta|\xi_t, s_{t+1})] = \\ &= \mathbb{E}_{p(\theta|\xi_t,s_{t+1})}[\log p(\theta|\xi_t, s_{t+1})] - \mathbb{E}_{p(\theta|\xi_t)}[\log p(\theta|\xi_t)]\,, \end{aligned} \tag{6}$$

where $p(\theta|s_{t+1}, \xi_t) \propto p(\theta)p(s_{t+1}|\theta, \xi_t)$, and where $H[p(x)] = \mathbb{E}[-\log p(x)]$ is the Shannon entropy (Shannon, 1948) and $\theta \in \Theta$ the set of dynamics parameters. As stated in main paper, the issue associated with computing $\text{IG}_\theta(\xi_t, s_{t+1})$ is that it can be done only in a reactive setting where s_{t+1} is actually revealed to the agent. Thus in an active setting, we have to resort to an expected value surrogate version of it, $\text{EIG}_\theta(\cdot)$.

B.1 Expected Information Gain

As s_{t+1} is not revealed to the agent at time t, we can use $\text{EIG}_\theta(\xi_t) = \mathbb{E}_{p_\theta(s_{t+1}|\xi_t)}\big[\text{IG}_\theta(\xi_t, s_{t+1})\big]$, which essentially marginalizes over possible next $s_{t+1} \in \mathcal{S}$, defined in full form as:

$$\begin{aligned} \text{EIG}_\theta(\xi_t) &= \mathbb{E}_{p(s_{t+1}|\xi_t)}\big[\text{IG}_\theta(\xi_t, s_{t+1})\big] = \\ &= \mathbb{E}_{p(s_{t+1}|\xi_t)}\big[\mathbb{E}_{p(\theta|\xi_t,s_{t+1})}[\log p(\theta|\xi_t, s_{t+1})] - \mathbb{E}_{p(\theta|\xi_t)}[\log p(\theta|\xi_t)]\big]\,. \end{aligned} \tag{7}$$

Now we note that we can re-write $p(s_{t+1}|\xi_t) = \int p(s_{t+1}|\xi_t, \theta)p(\theta|\xi_t)\, d\theta$ by marginalization, and that following Bayes theorem:

$$\begin{aligned} p(\theta|\xi_t) &= \frac{p(s_{t+1}|\xi_t, \theta)p(\xi_t, \theta)}{\int_\Theta p(s_{t+1}|\xi_t, \theta)p(\xi_t, \theta)\, d\theta} = \\ &= \frac{p(s_{t+1}|\xi_t, \theta)p(\xi_t|\theta)p(\theta)}{\int_\Theta p(s_{t+1}|\xi_t, \theta)p(\xi_t|\theta)p(\theta)\, d\theta} \,. \end{aligned}$$

Using these two facts and applying them to (7) we get:

$$\begin{aligned} &\mathbb{E}_{p(s_{t+1}|\xi_t)}\left[\mathbb{E}_{p(\theta|\xi_t, s_{t+1})}[\log p(\theta|\xi_t, s_{t+1})] - \mathbb{E}_{p(\theta|\xi_t)}[\log p(\theta|\xi_t)]\right] = \\ &= \mathbb{E}_{p(\theta)p(s_{t+1}|\theta,\xi_t)}\left[\log \frac{\frac{p(s_{t+1}|\xi_t,\theta)p(\xi_t|\theta)p(\theta)}{\int_\Theta p(s_{t+1}|\xi_t,\theta)p(\xi_t|\theta)p(\theta)\, d\theta}}{\frac{p(\xi_t|\theta)p(\theta)}{p(\xi_t)}}\right] = \\ &= \mathbb{E}_{p(\theta)p(s_{t+1}|\theta,\xi_t)}\left[\log \frac{\frac{p(s_{t+1}|\xi_t,\theta)p(\xi_t|\theta)p(\theta)}{p(s_{t+1}|\xi_t)p(\xi_t)}}{\frac{p(\xi_t|\theta)p(\theta)}{p(\xi_t)}}\right] = \\ &= \mathbb{E}_{p(\theta)p(s_{t+1}|\xi_t,\theta)}\left[\log p(s_{t+1}|\xi_t, \theta) - \log p(s_{t+1}|\xi_t)\right] \\ &= \mathbb{E}_{p(\theta)p(s_{t+1}|\xi_t,\theta)}\left[H[p(s_{t+1}|\xi_t)] - H[p(s_{t+1}|\xi_t, \theta)]\right] \,, \end{aligned}$$

by cancelling terms out and re-ordering. Thus, we have now obtained an equivalent specification of $\text{EIG}_\theta(\xi_t)$ that can computed using only the posterior predictive distribution $p_\theta(s_{t+1}|s_t, a_t)$, which marginalizes over parameters $\theta \in \Theta$ according to their posterior, i.e., $p_\theta(s_{t+1}|s_t, a_t) = \int_\Theta p_\theta(s_{t+1}|s_t, a_t, \theta)p(\theta|s_t, a_t)\, d\theta$. $\text{EIG}_\theta(\xi_t)$ in this form can be interpreted as follows: it represents the expected reduction in the predictive uncertainty over the next state s_{t+1} obtained from observing a different set of parameters θ.

B.2 EIG under Gaussian dynamics

If the predictive posterior distribution is a (multivariate) Gaussian, we can simplify $\text{EIG}_\theta(\xi_t)$ calculations even more, as stated at the end of Section 3.1. This is because if $p(\mathbf{x}) \triangleq \mathcal{N}(\mu, \Sigma)$, the entropy $H[p(\mathbf{x})]$ can be simplified (Cover, 1999) as:

$$\begin{aligned} H[p(\mathbf{x})] &= -\int_{-\infty}^{\infty} \mathcal{N}(\mu, \Sigma)\, \log \mathcal{N}(\mu, \Sigma)\, d\mathbf{x} = \\ &= \frac{D}{2}\log 2\pi + \frac{1}{2}\log\det(\Sigma) + \frac{1}{2}\mathbb{E}\left[(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)\right] = \\ &= \frac{D}{2}\log 2\pi + \frac{1}{2}\log\det(\Sigma) + \frac{1}{2}D = \\ &= \frac{1}{2}\log\det(\Sigma) + \frac{D}{2}(1 + \log 2\pi) \,, \end{aligned}$$

where D is the dimensionality of the multivariate normal, that in our case corresponds to $D = |\mathcal{S}|$, and where the term $\mathbb{E}\left[(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)\right]$ is simplified as follows:

$$\begin{aligned} \mathbb{E}\left[(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)\right] &= \mathbb{E}\left[tr((\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu))\right] = \\ &= \mathbb{E}\left[tr(\Sigma^{-1}(\mathbf{x}-\mu)(\mathbf{x}-\mu)^\top)\right] = \\ &= tr(\mathbb{E}\left[\Sigma^{-1}(\mathbf{x}-\mu)^\top(\mathbf{x}-\mu)\right]) = \\ &= tr(\Sigma^{-1}\Sigma) = \\ &= tr(I_D) = \\ &= D \end{aligned}$$

Taking this simplification into account, we can in turn reduce $\text{EIG}_\theta(\xi_t)$ calculations to the following:

$$\begin{aligned} &\mathbb{E}_{p(\theta)p(s_{t+1}|\xi_t,\theta)}\left[H[p(s_{t+1}|\xi_t)] - H[p(s_{t+1}|\xi_t, \theta)]\right] = \\ &= \mathbb{E}_{p(\theta)p(s_{t+1}|\xi_t,\theta)}\left[\frac{1}{2}\log\det(\mathbb{V}[p(s_{t+1}|\xi_t)]) - \frac{1}{2}\log\det(\mathbb{V}[p(s_{t+1}|\xi_t, \theta)])\right] = \\ &= \frac{1}{2}\Big(\log\det(\mathbb{V}[p(s_{t+1}|\xi_t)]) - \mathbb{E}_{p(\theta)}\left[\log\det(\mathbb{V}[p(s_{t+1}|\xi_t, \theta)])\right]\Big) \,, \end{aligned}$$

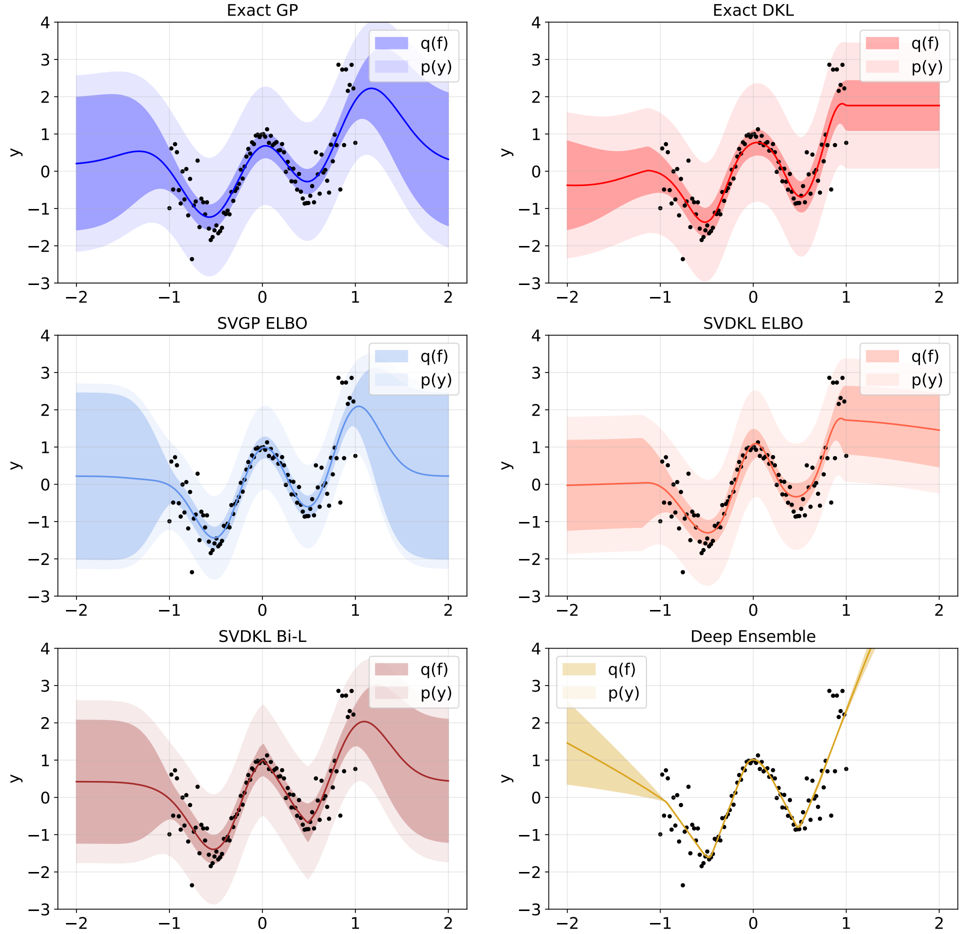


Figure 4: Comparison of uncertainty quantification models on a simple one dimensional regression example. The sample size of generated data points is $n = 100$. The models considered are, starting from the top left corner towards the bottom right corner: i) Exact GP, trained on the whole sample; ii) Exact DKL, trained on the whole sample; iii) Stochastic Variational GP, trained on the ELBO loss over a subset of 20 data points; iv) SV DKL, trained the same way as SVGP; v) SV DKL coupled with dropout fully-connected ResNet structure and Bi-Lipschitz constraints to avoid feature collapse; vi) Deep ensembles made of 5 neural networks models.

where $\mathbb{V}[p(\cdot)]$ denotes the variance of distribution $p(\cdot)$.

C Additional Information on the BAE procedure

This section is intended to clarify some of the main components featuring in the Bayesian Active Exploration procedure that we introduce in the main paper. In particular, we describe the choice of the environment dynamics model and the predictive MCTS algorithm, with posterior predictive sampling.

C.1 Models for the Environment Dynamics

As for the first fundamental component of the BAE algorithm, that is, a Bayesian model for the environment dynamics, we have primarily considered three classes of models, namely (Stochastic Variational) Gaussian Processes (Quinonero-Candela and Rasmussen, 2005; Rasmussen et al., 2006; Hensman et al., 2013, 2015), Deep Kernels (Wilson et al., 2016a,b; Ober et al., 2021) and deep ensembles (Lakshminarayanan et al., 2017; Rahaman et al., 2021).

C.1.1 Gaussian Processes

GPs place a prior on the functional form of $f(\cdot) \sim \mathcal{GP}\big(m(\cdot), k(\cdot,\cdot)\big)$, such that $p(f) \triangleq \mathcal{N}\big(f \,|\, m(\cdot), k(\cdot,\cdot)\big)$, where $m(\cdot)$ is a mean function (e.g., constant, linear, etc.) and $k(\cdot,\cdot)$ a kernel function (e.g., linear, squared-exponential, matérn, etc.). Prior knowledge about $f \in \mathcal{F}$ can be efficiently conveyed through the choice of $m(\cdot)$ and especially $k(\cdot,\cdot)$, which governs main features of the function approximator $\hat{f}(\cdot)$ such as smoothness and sparsity. The joint density of $(\mathbf{y}, \mathbf{f})$ and the marginal likelihood, which is used for training, in a GP take the following form

$$p(\mathbf{y}, \mathbf{f}|X) = p(\mathbf{y}|\mathbf{f}, \sigma^2)p\big(\mathbf{f}|X; m(\cdot), k(\cdot,\cdot)\big) \quad \text{and} \quad p(\mathbf{y}|X) = \int_{\mathcal{F}} p(\mathbf{y}|\mathbf{f}, \sigma^2)p(\mathbf{f}|X)\, d\mathbf{f}\ ,$$

where all the quantities of interest can be computed analytically as the distributions are Gaussian. From this point of view, GPs have the advantage of being complementary to the simplified version of $\text{EIG}_\theta(\xi_t)$ derived above, as their predictive posterior distribution on s_{t+1} is in fact Gaussian as well. Besides GPs are well-known for their excellent uncertainty quantification properties. Their main drawback lies in their poor scalability, which is why we consider a Stochastic Variational approach where we learn a sub-sample of inducing points $\mathbf{u} = \{u_i\}_{i=1}^m$ (Hensman et al., 2013, 2015). We also considered using multitask/multioutput kernel learning (Alvarez et al., 2012) but did not notice any significant improvements, while training costs were higher for the higher number of parameters involved in the multi-task kernels.

Throughout the experiments presented in the work, we utilize a GP with constant prior mean function, $m(\cdot) = 0$ and with base squared exponential kernel $k(x, y) = C_{ff}(x, y) = \exp\{-l^2\|x - y\|^2\}$.

C.1.2 Deep Kernels

Deep kernels (Wilson et al., 2016a,b) are a generalization of GPs where inputs (s_t, a_t), or just s_t, are first mapped to a (potentially lower dimensional) latent representation space h_t, $f_h : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{H}$ or $f_h : \mathcal{S} \rightarrow \mathcal{H}$, through a deep learning architecture. Then, the function to predict the next state, $f_s : \mathcal{H} \rightarrow \mathcal{S}$ or $f_s : \mathcal{H} \times \mathcal{A} \rightarrow \mathcal{S}$, can be assigned any GP prior (e.g., SVGP). The advantage of using deep kernels lies in the fact that they are better suited for dealing with high-dimensional state-action spaces thanks to their deep architecture, while they retain good function approximation and uncertainty quantification properties of GPs. On this matter, we specifically employ a version of deep kernels that uses a fully-connected ResNet neural net architecture (with dropout) for $f_h(\cdot)$ and incorporates a bi-Lipschitz constraint on the transformations in $f_h(\cdot)$ van Amersfoort et al. (2021), in order to avoid the tendency of neural nets of exhibiting feature collapse (Guo et al., 2017; Ober et al., 2021). Note that the last GP layer in a deep kernel model scales by construction at a $\mathcal{O}(m^3|\mathcal{H}|^3)$ training and $\mathcal{O}(m^2|\mathcal{H}|^2)$ test cost, where $|\mathcal{H}|$ is directly controllable. Lastly, notice that a (SV) deep kernel still allow one to use the simplified version of $\text{EIG}_\theta(\xi_t)$ on the latent space $\mathcal{H}$, i.e., $\text{EIG}_\theta(h_t)$, as $p(s_{t+1}|h_t)$ is modelled as a Gaussian, as any component coming after the neural network block that models $p(s_{t+1}|h_t)$ behaves exactly as a Gaussian Process.

We have also considered implementing the MC dropout (Gal and Ghahramani, 2016) technique and apply it to the deep neural network layers of deep kernels in order to get better generalization properties, but did not notice any further improvement. MC Dropout consists in re-sampling a pre-trained neural network with dropout layers (Srivastava et al., 2014) K times, at test time, such that each prediction $\{(\mathbf{y}|X)\}_{i=1}^K$ represents a sample from the predictive distribution $q(\cdot \mid \cdot)$, with different weights $\omega = \{W\}_{i=1}^L$ and biases $\mathbf{b} = \{b\}_{i=1}^L$:

$$q(\mathbf{y}^* \mid \mathbf{x}^*) = \int p(\mathbf{y}^* \mid \mathbf{x}^*, \mathbf{W}, \mathbf{b})\ p(\mathbf{W}, \mathbf{b})\ d\mathbf{W} d\mathbf{b}.$$

MC Dropout approximates sampling from the Bayesian posterior predictive distribution, marginalizing out $\mathbf{W}, \mathbf{b}$. Perhaps the fact that MC dropout does not improve uncertainty quantification has to do with the fact that using a sub-sample of input points through the SV inducing points procedure already prevents the deep kernels model from being over-confident and collapsing to the mean (Rahaman et al., 2021), resulting into better generalization properties than standard deep kernels. However, we leave this topic to be investigated as part of future research.

Throughout the experiments, we consider a deep kernel model where the neural network architecture consists of two hidden layers of [32, 32] units, and a GP with constant mean and squared exponential kernel.

```
Algorithm 1 Bayesian Active Exploration
  Initialize: Transition Buffer $\mathcal{D}$, Bayesian Dynamics Model $p(f)$
  while Not Done do
      Agent $\mathcal{M} \leftarrow$ PPO$(\mathcal{S}, \mathcal{A}, \epsilon_{clip}, \gamma)$          ▷ $\gamma$ = learning rate
      if $t > \bar{t}$ then                                                    ▷ $\bar{t}$ = warm-up
          Sample $\{\hat{a}_t^{(k)}\}_{k=1}^{K} \sim \pi_{explore,\psi}$ and
          Sample $J$-length trajectories $\hat{\tau}_t = \{(s_{t+j}, \hat{a}_{t+j}^{(k)}, \hat{s}_{t+1+j}\}_{j=0}^{J}$
          Compute $u_t(s_t, \hat{a}_t^{(k)}) = \mathrm{EIG}_\theta(s_t, \hat{a}_t^{(k)})$
          Solve $\pi_{explore,\psi} \leftarrow$ PPO.update$(\hat{\tau}_t, u_t)$
          Sample actual $a_{t+1} \sim \pi_{explore,\psi}$
      end if
      Add $\mathcal{D}_{new} \leftarrow \mathcal{D} \cup \{s_t, a_t, s_{t+1}\}$          ▷ $\pi_{explore,\psi}(a_t)$ random if $t < \bar{t}$
      Train dynamics model $p(f|\mathcal{D}_{new})$
      Get new posterior $p_{\theta'}(s_{t+1}|s_t, a_t)$
  end while
```

Figure 5: Pseudo-code representation of the Bayesian Active Exploration (BAE) algorithm. With the ˆ sign we denote estimated quantities, sampled from parametrized models (policy and environment), e.g., $\hat{a}^{(k)}$.

C.1.3 Deep Ensembles

Deep ensembles (Lakshminarayanan et al., 2017) are arguably the most popular standard tool for uncertainty quantification in neural networks, thanks to their very good generalization properties compared to other sampling techniques such as MC dropout. The main difference with MC dropout is that deep ensembles requires to separately train (so they operate at train time) a pool of different neural network independently on the same data. Deep ensembles have been shown to have good out of sample uncertainty coverage (Wilson and Izmailov, 2020) in some examples, although they rarely match that of Gaussian Processes.

Some of the models, and their variations, considered for the environment dynamics are compared in the one dimensional regression example depicted in Figure 4. We use this simple example to depict each candidate model's generalization and uncertainty quantification properties. Description of each model is enclosed in the figure's caption. We can see how the best models emerging from this small scale study are the Exact GP, SVGP and SVDKL. In particular, SV DKL trains over 15 times faster than Exact DKL, and we notice that reducing the training sample to the 20 inducing points result in better uncertainty coverage in SV DKL versus Exact DKL.

Throughout the experiments in this work, we consider a deep ensemble of 10 neural networks made of two-hidden layer of [32, 32] units.

C.2 Posterior predictive MCTS sampling

In this section we briefly discuss the posterior predictive MCTS sampling procedure and provide a high-level pseudo-code representation of the Bayesian Active Exploration algorithm, in Figure 5.

The fundamental difference between the posterior MCTS sampling we use and the standard random MCTS, is that we use the posterior predictive distribution over the next state $p_\theta(s_{t+1}|s_t, a_t)$ that we obtain from our agent's Bayesian model of the environment to sample trajectories (Ghavamzadeh et al., 2015). This leads to more efficient trajectories sampling as we take into account more likely agent's moves, a posteriori.

There is a connection between the predictive MCTS procedure described in BAE (and in other methods such as in Shyam et al. (2019)) and the concept of agent's 'imagination', as in Hafner et al. (2019). The trajectories sampled via this predictive MCTS procedure can be referred to as

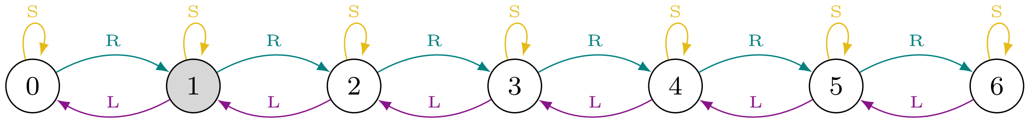
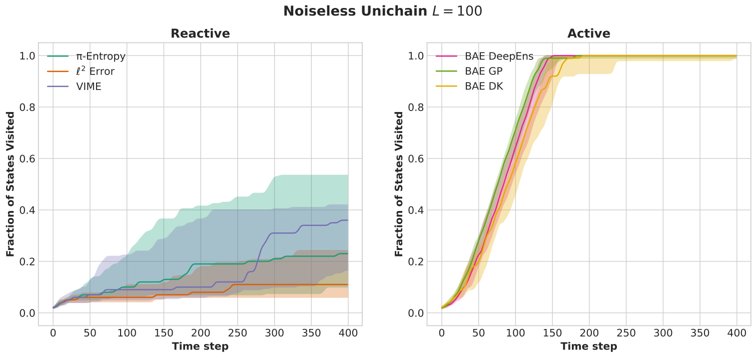


Figure 6: Example of unichain environment made of $L = 7$ states. Arrows represent the possible actions in each state, state one is shaded to indicate that the agent is spawned there as $s_0 = 1$.



Model	States Fraction
π-Expl	0.36 ± 0.07
ℓ^2 Error	0.22 ± 0.05
VIME	0.34 ± 0.05
BAE DeepEns	0.99 ± 0.01
BAE GP	0.97 ± 0.04
BAE DK	0.99 ± 0.01

Figure 7: Results on the $L = 100$ states unichain environment exploration task. The plots show the median cumulative fraction of states visited (solid line) together with the 75th and 25th error bands, computed over 20 replications. The table instead reports the average fraction of states visited at termination (i.e., after 400 steps), together with the 95% confidence intervals.

'imaginary', and the resulting collection of K trajectories is used to solve a so-called 'ImaginaryMDP' with .

C.2.1 Exploit-to-Explore

In the predictive MCTS procedure, a policy gradient (Kakade, 2001) algorithm is employed to solve the 'ImaginaryMDP', formed by the trajectories obtained via the predictive posterior $p_\theta(s_{t+1}|s_t, a_t)$. In this work, we choose the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017), but note that any other policy gradient algorithm serves the purpose (e.g., SAC). As in BA(L)E we are already learning an exploration policy, we modify the PPO algorithm to make it fully exploitative by removing any exploration bonus heuristics. Recall that the standard PPO loss reads:

$$\mathcal{L}_t^{\text{PPO}}(\psi) = \mathbb{E}\big[\mathcal{L}_t^{\text{Clip}}(\psi) - c_1\mathcal{L}_t^{\text{Value}}(\psi) + c_2 H[\pi_\psi(s_t)]\big]$$

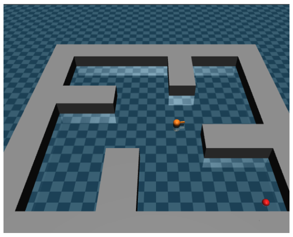
where $\mathcal{L}_t^{\text{Clip}}$ is the clipped surrogate objective, $\mathcal{L}_t^{\text{Value}}$ is a squared loss on the value function $V(\cdot)$ and $H[\pi_\psi(s_t)]$ is a policy entropy term used to ensure sufficient exploration (for more details see Schulman et al. (2017)). Thus, to make PPO fully exploitative we remove the policy entropy bonus term, $c_2 = 0$, which is only needed when learning exploitative policies.

D Additional Experiments

D.1 The Unichain Environment

In this last supplementary material section we include a brief description of the Unichain enviroment, plus additional results on it where we consider a longer chain of $L = 100$ states, without noise this time.

As outlined in the main paper, the unichain environment consists in a simple sequence of L Markov sequentially connected states, where the goal is usually to explore all the possible states and reach the final one (Puterman, 2014; Osband et al., 2016). We consider a discrete action space defined as $\mathcal{A} = \{\text{go left}, \text{stay}, \text{go right}\}$, where each action's consequence is self-explanatory. If the agent is in the first or last state and tries to go left or right, it automatically hit a wall and remains in that state. The agent is initially spawned in the second state $s_0 = 1$ (state counter starting from 0). The reward



Model	**# Steps to Goal**	s_T **Rewards**
π-Entropy	3371.0 ± 508.3	0.60 ± 0.21
ℓ^2 Error	2821.6 ± 566.4	0.64 ± 0.20
VIME	3662.0 ± 481.9	0.36 ± 0.20
BAE DeepEns	1211.9 ± 334.4	1.0 ± 0.0
BAE GP	1304.2 ± 606.4	1.0 ± 0.0
BAE DK	863.8 ± 360.8	1.0 ± 0.0

Figure 8: On the left, a visual 3D rendition of a type of Medium Maze with wall ledges only. On the right, a table reporting results of the different exploration methods in terms of average number of steps required to reach the goal object and average reward at terminal state ($s_t = 1.0$ indicates that the task is solved for all the runs), together with 90% Monte Carlo standard errors.

function $r^e : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is sparse and assigns 0.001 to visiting the first state, 1 to visiting the last state, and 0 otherwise, i.e.:

$$r^e(s_t, a_t) = \begin{cases} 0.001 & \text{if } s_t = 0 \\ 1 & \text{if } s_t = L \\ 0 & \text{otherwise .} \end{cases}$$

The sub-optimal reward associated with state 0 represents a 'reward trap' for algorithms based solely on extrinsic reward exploitation, as it acts as disincentive for the agent to move elsewhere and explore. Thus, strong exploration bonus is needed to move it away from there. A simple visual representation of a $L = 7$ state unichain environment, together with the possible actions is depicted in Figure 6. The shaded state $s_0 = 1$ represents the spawning location.

D.2 Additional Experiments on the 100 States Unichain Environment

In this subsection, we present the additional results on the $L = 100$ unichain environment, but without added noise this time. The task is again purely exploratory, and the methods compared are a subset of those compared in the main paper: i) PPO policy entropy $H(\pi_\psi(s_t))$ regularizer (Schulman et al., 2017) (***pi*-Entropy**); ii) PPO with ℓ^2 prediction error as reactive intrinsic reward (Stadie et al., 2015; Pathak et al., 2017) (**ℓ^2 Error**); iii) PPO with VIME, i.e., $\text{IG}_\theta(\cdot)$ as a reactive intrinsic reward coupled with Bayesian Neural Network dynamics (Houthooft et al., 2016) (**VIME**); iv) Bayesian Active Exploration with Deep Ensembles (Shyam et al., 2019); v) Bayesian Active Exploration with SVGP (**BAE GP**); vi) Bayesian Active Exploration with Deep Kernels (**BAE DK**).

We measure performance of the methods again via the cumulative fraction of visited states at each time step t, and the final fraction of coverage at episodic termination, which we set to be after 400 steps in this case, as the Markov chain of states is longer. Active methods are left running for 10 steps initially in order to gather enough data to estimate $p_\theta(s_{t+1}|s_t, a_t)$. Results over 20 seeded replication of the experiment are reported in Figure 7's plots and table. Similarly to the unichain $L = 50$ states environment results presented in the main paper, active methods consistently outperform reactive ones, as they require approximately 150 steps only to reach complete coverage of the environment and solve the task.

D.3 Additional Details on the Medium Sized Maze

Finally, we report here a few extra information on the Maze environments, and results on the Medium Maze version of it. The Maze environments are 2D, and their state space $\mathcal{S}$ features x and y coordinates of the agent and x and y coordinates linear velocity of the agent. The continuous action space $\mathcal{A}$ instead include coordinate's linear force in the x and y directions.

Figure 8 above reports a visual 3D rendition of a prototypical type of open medium maze (with wall ledges only); while the table reports results of the different exploration methods in terms of average number of steps required to reach the goal object and average reward at terminal state ($s_T = 1.0$ indicates that the agent has solved the task in all the different runs), together with 90% Monte Carlo standard errors.