RL ALGORITHMS ARE INFORMATION-STATE POLICIES IN THE BAYES-ADAPTIVE MDP

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a new conceptual framework for understanding RL algorithms as ~~*hand-written information-state*~~ policies for the Bayes-Adaptive MDP, which augments the state space with the information gathered, making it straightforward to leverage powerful tools for analyzing policies in MDPs to analyze RL algorithms themselves. We demonstrate the utility of this framework by deriving a number of insights with practical implications for algorithm and reward shaping design. For instance, optimal policies for the BAMDP, i.e., ideal RL algorithms, should not necessarily converge to optimal policies for the underlying MDP—even though RL theory has typically regarded the latter property as essential. ~~We also apply the theory of potential-based reward shaping in the BAMDP to analyze valid forms of intrinsic motivation.~~ We can understand BAMDP Q-values as the sum of separate measures of the value gained from exploration and exploitation. We finally derive a direct relationship between an RL algorithm's shaping function in the MDP and its optimality in the BAMDP, and use these results to inform the design and explain the roles of reward shaping and intrinsic motivation functions.

1 INTRODUCTION

~~When designing a policy for a~~ Markov Decision Processes (MDPs) provide a clear problem specification for our policies – to maximize the discounted sum of rewards. As such, it has served as the foundation for the development of a large set of strong theoretical tools, such as bellman backups (Bellman, 1957), potential shaping (Ng et al., 1999), and regret bounds (Singh & Yee, 1994; Auer et al., 2008). However, there is no widely used analogous problem specification for Reinforcement Learning (RL) algorithms ~~is often unclear~~, resulting in much of RL theory being misdirected, and tempering its impact on RL practice. ~~The aim of RL in practice is to an create algorithm which, through interacting with its environment, learns to maximize a reward signal.~~ Much of the work in theoretical RL ~~operationalizes this by aiming instead~~ aims to find algorithms that eventually converge to the optimal MDP policy ~~given unlimited interactions~~. This may often result in fairly useful algorithms, for instance in episodic environments with arbitrarily many offline training episodes, and no further learning during deployment. However, such algorithms tend to over-explore, since achieving exact optimality generally requires additional exploration past the point of diminishing returns. Moreover, we want many other features from our policies which do not fall under this convergence criteria. For instance, for algorithms deployed in and adapting online to the real world, they must ~~we want efficient algorithms which~~ perform well throughout their interactions with the world, appropriately balancing collecting information with collecting rewards.

We propose specifying RL problems with the Bayes-Adaptive MDP (Duff, 2002; Ghavamzadeh et al., 2015) – which models the problem of learning to maximize reward in unknown domains as a Bayesian decision problem on ~~information~~ states augmented with the information gathered thus far. While prior work with BAMDPs attempts to *directly solve* them to find the optimal RL algorithm automatically (Zintgraf et al., 2019; Guez et al., 2012), we instead use it as a *conceptual framework* to cast all ~~manually programmed~~ RL algorithms as BAMDP policies. This is a powerful new perspective, making it straightforward to transfer tools developed for analyzing policies in MDPs to better understand RL algorithms themselves. This provides both fundamental principles and has practical implications for designing RL algorithms, reward shaping and intrinsic motivation functions.

~~Thus a natural solution concept for an RL algorithm is to maximize the discounted sum of rewards in the BAMDP. This line of inquiry is both complementary to and distinct from the problems of meta-learning, since regardless of how much meta-learning takes place, some algorithm must be written down eventually.~~

Our main contributions are:

- A new framework for analyzing RL algorithms and specifying RL problems by casting algorithms as policies in BAMDPs. ~~and clarifying that optimal learning does not imply convergence to the optimal MDP policy.~~
- A powerful implication for RL algorithm design, i.e., that algorithms should explore only when gathering further information is expected to maximize return, instead of until the optimal underlying MDP policy π^* is found. This is a radical departure from the mainstream view in RL of convergence to π^* as the gold standard.
- ~~Showing that the intrinsic rewards only preserve the behavior of optimal RL algorithms if they are potential shaping functions in the BAMDP.~~
- A simple but powerful model capturing the myopic behavior of many RL algorithms, represented as a policy $\bar{\Pi}^m$ in the BAMDP.
- A characterization of algorithmic regret in terms of BAMDP value via an application of prior work on suboptimality gaps (Yang et al., 2021; Simchowitz & Jamieson, 2019).
- A derivation of the precise dependency of optimal shaping rewards on the problem domain and the learning algorithm, providing practical insights into the design of intrinsic motivation and reward shaping functions.
- The decomposition of BAMDP value into value of information and value of opportunity, resulting in a new taxonomy clarifying the various roles of reward shaping terms, and a novel perspective on the empirical behavior of “Empowerment”-driven agents, more accurately describing the observed behaviors than the prior interpretation.

2 Background

Markov Decision Processes (MDPs) are defined by tuple $M = (\mathcal{S}, \mathcal{A}, R, T, T_0, \gamma)$ with $\mathcal{S}$ a set of states, $\mathcal{A}$ a set of actions, $R(r_t|s_t, a_t)$ a reward distribution (with $R(s_t, a_t)$ shorthand for expected reward), $T(s_{t+1}|s_t, a_t)$ a transition function, $T_0(s_0)$ an initial state distribution, and γ a discount factor. MDP policies map from current states to distributions over next actions: $\pi(a_t|s_t)$. The optimal policy for MDP M maximizes the expected discounted return: $\pi^* \in \arg\max_\pi \mathbb{E}_{M,\pi}[\sum_t \gamma^t R(s_t, a_t)]$.

RL Algorithms, which we denote by $\bar{\Pi}$, are methods which maintain state while interacting with an environment, selecting actions and receiving observations. Most often, these algorithms are designed to try to learn how to act to maximize return in uncertain environments; in this paper we focus on MDPs. Internally, they may select actions directly, or continuously update and sample actions from a learnt MDP policy.

RL Objective: We analyze the lifelong RL setting (Khetarpal et al., 2022)[1], where performance of algorithm $\bar{\Pi}$ in MDP M is measured by its expected discounted return *while learning*: $G_M(\bar{\Pi}) = \mathbb{E}_{M,\bar{\Pi}}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$. The objective for $\bar{\Pi}$ is to get high expected return over a distribution of MDPs: $\mathcal{J}(\bar{\Pi}) = \mathbb{E}_{p(M)}[G_M(\bar{\Pi})]$ (Duan et al., 2016; Singh et al., 2009).

Reward shaping is a common method for guiding RL algorithms (Dorigo & Colombetti, 1994). Shaping rewards are added to the original (also called *extrinsic*) reward at each time step, and the shaping function R^x can depend on not only the current transition but also the entire history of states, actions and rewards $h_t = s_1 a_1 r_1 s_2 ... a_{t-1} r_{t-1} s_t$, generating shaped reward signal: $r^x_t = r_t + R^x(s_t, a_t, h_t)$. *Potential-based shaping functions* (PBSFs) are a class of shaping functions that preserve the optimal policy of any MDP they're added to. They are of the form $R^x(s_t, s_{t+1}) = \gamma\phi(s_{t+1}) - \phi(s_t)$, where the potential function ϕ must be a function of only the state, e.g. in a goal-reaching task it could be negative distance from s_t to the goal state (Ng et al., 1999).

[1] Our theory is equally applicable to the case of episodic RL, as it is a special case of lifelong RL.

2.1 Formal Definition of the BAMDP

The BAMDP formulates RL problems as Markov Decision Processes, such that the optimal BAMDP policy is the Bayes-optimal RL algorithm for the problem. Our conventions are inspired by Zintgraf et al. (2019) and Guez et al. (2012).

Central to the definition of the BAMDP is the prior $p(M)$ over the underlying unknown MDP M that the RL agent is inside. When using the BAMDP to specify RL problems in practice, $p(M)$ represents the distribution of MDPs that the RL algorithm will encounter, e.g., in a navigation problem, $p(M)$ may correspond to the distribution of mazes created by a procedural maze generator. For disambiguation we call $p(M)$ the *task distribution*, from which *task MDPs* are sampled, and policies for task MDPs $\pi(a|s)$ are *task policies*. We use an overbar (e.g. $\bar{M}$) for objects at the BAMDP level. For clarity of exposition, we assume all task MDPs in each problem share the same $\mathcal{S}, \mathcal{A}, \gamma$, so only R, T, T_0 are initially unknown and vary across tasks[2]. We use $p(M|h_t)$ to denote the posterior over M after updating on evidence h_t using Bayes' rule, i.e. $p(M|h_t) \propto p(h_t|M)p(M)$.

A BAMDP is a tuple $\bar{M} = (\bar{\mathcal{S}}, \mathcal{A}, \bar{R}, \bar{T}, \bar{T}_0, \gamma)$ where:

- Augmented state space $\bar{\mathcal{S}} = \mathcal{S} \times \mathcal{H}$, with $\mathcal{H}$ the set of possible histories, so $\bar{s} = \langle s, h \rangle$. This encapsulates all the information $\bar{\Pi}$ could use when choosing an action- though typically it maintains a lossy memory of h_t which we denote by $b^{\bar{\Pi}}(h_t)$.
- For the lifelong learning setting we study, $\mathcal{A}, \gamma$ are the same as the task MDPs, although $\bar{\Pi}$ may choose its action from a learnt task policy $\pi_t = \bar{\Pi}(\bar{s}_t)$; for ease of notation we use π_t and a_t interchangeably, since π_t is only used at step t to output a_t
- $\bar{R}(\langle s_t, h_t \rangle, a_t) = \mathbb{E}_{p(M|h_t)}[R(s_t, a_t)]$, the expected reward under the current posterior.
- Similarly, $\bar{T}(\bar{s}_{t+1}|\bar{s}_t, a_t) = \mathbb{E}_{p(M|h_t)}[T(s_{t+1}|s_t, a_t)R(r_t|s_t, a_t)\mathbb{1}[h_{t+1} = h_t a_t r_t s_{t+1}]]$.
- Initial state distribution $\bar{T}_0(\langle s_0, h_0 \rangle) = \mathbb{E}_{p(M)}[T_0(s)]\mathbb{1}[h_0 = s_0]$.

For example, in the caterpillar problem depicted in Figure 1, $p(M)$ represents the fact that butterflies typically lay their eggs on the best food source in the vicinity, but 10% of the time there is a better source nearby. The bush's reward varying across M manifests as initially stochastic BAMDP dynamics for the action staying at s_b. But once the reward has been observed (trajectories A and B), $p(M|h_t)$ collapses onto the underlying task MDP and all dynamics become deterministic.

3 Optimality of RL Algorithms

An RL algorithm's expected BAMDP return is equal to its expected performance on the problem:

$$\mathbb{E}_{\bar{M},\bar{\Pi}}[\sum_t \gamma^t \bar{R}(\bar{s}_t, a_t)] = \mathbb{E}_{p(M)}[\mathbb{E}_{M,\bar{\Pi}}[\sum_t \gamma^t R(s_t, a_t)]] = \mathcal{J}(\bar{\Pi}), \tag{1}$$

thus an algorithm that calculates the optimal action in the BAMDP and executes it in the task MDP is Bayes-optimal for that problem. For instance in the caterpillar problem, for large enough γ the Bayes-optimal algorithm would first go to s_b, then if it found food it would stay forever (trajectory A), otherwise it would return to s_w and use the information in h_t to never go back (trajectory B)[3].

This also means that the optimality of exploratory actions can be determined directly by their impact on BAMDP return. Observe that with time discounting, **exploring enough to converge to the optimal task policy is not generally Bayes-optimal** (originally discovered when Gittins (1979) showed optimal policies for bandit problems don't always converge to the optimal arm). For instance, if γ were sufficiently small, $\bar{\Pi}^*$ would never explore s_b because immediate expected reward would dominate the expected return. Thus, maximizing $\mathcal{J}(\bar{\Pi})$ corresponds to finding the optimal exploration-exploitation trade-off.

Implications for RL Algorithm Design This perspective implies that RL practitioners should *not* design algorithms to converge to the optimal task policy, instead they should be designed to explore exactly when gathering further information is expected to maximize overall return. This could

[2] This formulation can be extended to POMDPs and for distributions over $\mathcal{S}, \mathcal{A}, \gamma$ without any conceptual changes - the agent receiving observations o_t, and calculating expectations over additional variables as needed.

[3] See appendix A.6.1 for the full calculations.

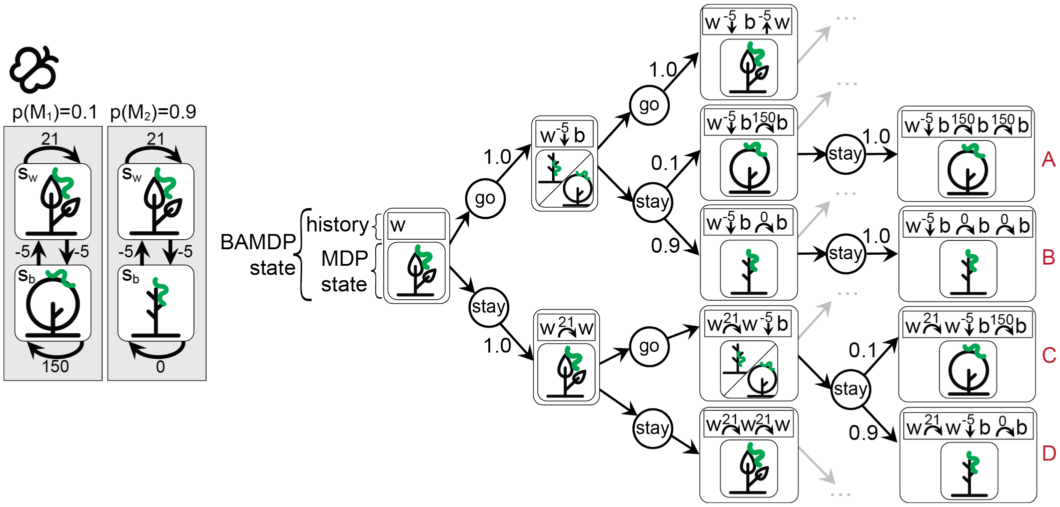

Figure 1: MDP prior (left) and truncated BAMDP transition diagram (right) for the caterpillar problem. Numbers on arrows represent transition probabilities. $\mathcal{S}$ consists of a weed s_w and a bush s_b. The caterpillar hatches at s_w, and can either *stay* or expend energy to *go* (-5 reward). Each step staying to eat the weed gains 21 reward, and staying to eat the bush gains 150, but in 90% of tasks the bush is dead and yields 0 reward. $\gamma = 0.95$.

be done by tracking how useful exploration has been, using this to predict the value of further knowledge gain, and exploiting more aggressively once this is no longer worth the opportunity cost. Taking epsilon-greedy exploration as a simple concrete example, instead of decaying epsilon with a hard-coded schedule, it could be decreased whenever expected return from purely greedy behavior exceeds the extrapolated gains from continuing with the epsilon noise.

Potential-based Reward Shaping We can immediately apply Ng et al. (1999)'s result on optimal policies in MDPs to the BAMDP, to prove that intrinsic rewards only preserve the behavior of optimal RL algorithms if they are BAMDP potential-based shaping functions – see appendix A.4 for the proof and discussion.

4 RL Algorithms

We now show how existing RL algorithms can be explicitly expressed as BAMDP policies, which will allow further analysis and insights about reward shaping.

4.1 The Bayes-optimal RL Algorithm

We start by expressing the Bayes-optimal RL algorithm, i.e. the optimal BAMDP policy, in terms of the optimal BAMDP Q Value. The Q value or state-action value of any RL algorithm is its expected return over its future trajectory through the BAMDP, which includes future histories h_{t+i}:

$$\bar{Q}^{\bar{\Pi}}(\bar{s}_t, a) = \mathbb{E}_{\bar{T},\bar{\Pi}}\left[\sum\nolimits_{i=0}^{\infty} \gamma^i \bar{R}\left(\langle s_{t+i}, h_{t+i}\rangle, a_{t+i}\right) | a_t = a\right]. \tag{2}$$

The optimal BAMDP Q value, $\bar{Q}^*$, is the maximum over all $\bar{\Pi}$, which is maximized by the Bayes-optimal algorithm $\bar{\Pi}^*$. At each step $\bar{\Pi}^*$ chooses the optimal action:

$$\bar{Q}^*(\bar{s}_t, a) = \max\nolimits_{\bar{\Pi}} \bar{Q}^{\bar{\Pi}}(\bar{s}_t, a); \qquad \bar{\Pi}^*(\bar{s}_t) \in \arg\max\nolimits_a \bar{Q}^*(\bar{s}_t, a). \tag{3}$$

Note that $\bar{Q}^*(\bar{s}_t, a)$ is analogous to the Gittins Index for arm a, representing the combined value of its expected payoff and the information gained by pulling it, given the history of all arm pulls h_t.

4.2 The Myopic RL Algorithm

It is generally unrealistic to assume algorithms can compute $\bar{Q}^*$ exactly, since for most interesting problems $p(M)$ is difficult to specify in closed form and the Bayesian posterior update is intractable.

Many practical algorithms, from policy-based methods like policy gradient (Sutton et al., 1999; Schulman et al., 2017) to value-based methods like Q-Learning (Watkins, 1989; Mnih et al., 2015), aim to estimate *task policy return* instead. We formalize this objective with the Myopic RL algorithm $\bar{\Pi}^m$.

Definition 4.1 (Myopic RL Algorithm)**.** At each step, the Myopic Algorithm $\bar{\Pi}^m$ follows the task policy that maximizes its estimated expected task return under current knowledge[4]:

$$\bar{\Pi}^m(\langle s_t, h_t \rangle) \in \arg\max\nolimits_\pi \mathbb{E}_{b^m(h_t)}[V^\pi(s_t)], \tag{4}$$

where $b^m(\cdot)$ denotes how $\bar{\Pi}^m$ interprets its experience, which could be anything from a distribution over world models maintained by updating a conjugate prior, to a point estimate of Q^* maintained by training a randomly initialized neural net on batches sampled from h_t (Mnih et al., 2015).

For example, policy gradient algorithms like Reinforce sample actions from a task policy π_θ, i.e.,

$$\bar{\Pi}^{\text{REINFORCE}}(\langle s_t, h_t \rangle) = \pi_\theta(s_t). \tag{5}$$

π_θ is learnt by gradient updates towards maximizing the expected returns $\mathcal{R}(\tau)$ of the trajectories τ that it generates, i.e. $J(\theta) = E_{\tau\sim\pi_\theta}[\mathcal{R}(\tau)]$. The algorithm estimates $J(\theta)$ from environment interactions so far, i.e. h_t, so $\hat{J}(\theta) = \mathbb{E}_{b^m(h_t)}[E_{\tau\sim\pi_\theta}[\mathbb{R}(\tau)]]$ where $b^m(h_t)$ is concentrated on a point estimate of $J(\theta)$. If, as a model, we assume that policy gradient were to maximize $J(\theta)$ between each interaction, then we find that it matches the behavior of $\bar{\Pi}^m$, or more precisely:

$$\arg\max\nolimits_\theta \hat{J}(\theta) = \arg\max\nolimits_\theta \mathbb{E}_{b^m(h_t)}[E_{\tau\sim\pi_\theta}[R(\tau)]] = \arg\max\nolimits_\pi \mathbb{E}_{b^m(h_t)}[V^\pi(s)]. \tag{6}$$

Returning to the problem in Figure 1, if $\bar{\Pi}^m$ knew $p(M)$ it would pick π_w that stays at the weed, because other π (either going to stay at the bush or alternating states forever) get lower expected return. This undervalues staying at s_b, because $\bar{\Pi}^m$ *can update* π. Formally, the Myopic algorithm estimates the following return for an action:

$$\hat{\bar{Q}}^m(\bar{s}_t, a) = \max_\pi \mathbb{E}_{b^m(h_t)}[Q^\pi(s_t, a)] = \max_\pi \mathbb{E}_{b^m(h_t),\pi}\left[\sum\nolimits_{i=0}^{\infty} \gamma^i R(s_{t+i}, a_{t+i}) | a_t = a\right], \tag{7}$$

which assumes $\bar{\Pi}^m$ follows π forever, and often *underestimates* its true value $\bar{Q}^m(\bar{s}_t, a)$ because it misses the value often gained from updating on new evidence. For instance, if $\bar{\Pi}^m$ started at s_b instead, it would go straight back to s_w because its value estimate for staying assumes it would always follow π_w back to s_w at the next step anyway, even if it found the bush alive[5].

5 Analyzing Algorithms

Formulating RL algorithms as BAMDP policies in the previous section made it clear how Myopic RL algorithms' assumption of fixed behavior leads them to undervalue information gathering actions in their value estimates $\hat{\bar{Q}}^m$, which is why injected stochasticity (e.g. ϵ-greedy) or reward shaping is needed in many RL algorithms. To see how they can be fixed with reward shaping, we now introduce tools to understand Bayesian regret and value in the BAMDP.

5.1 Regret in terms of Myopic Value Estimates

First, we formalize the relationship between $\bar{\Pi}^m$'s Bayesian regret and its value misestimation.

Theorem 5.1. *The Bayesian regret of the Myopic algorithm can be expressed as the discounted sum of BAMDP suboptimality gaps over its trajectory:*

$$\bar{V}^*(\bar{s}_0) - \bar{V}^m(\bar{s}_0) = \mathbb{E}_{\bar{s}_t\sim\bar{\Pi}^m}\big[\sum\nolimits_t \gamma^t\big(\bar{V}^*(\bar{s}_t) - \bar{Q}^*(\bar{s}_t, \arg\max\nolimits_a \hat{\bar{Q}}^m(\bar{s}_t, a))\big)\big]. \tag{8}$$

This can be shown by application of a prior result on the regret of policies in MDPs to the regret of RL algorithms in BAMDPs; see Appendix A.3 for the proof.

[4] This can be described as the *certainty equivalent* solution (Simon, 1956), or *best reactive policy* with respect to $\bar{\Pi}^m$'s beliefs, a concept introduced by Duff (2002) to lower-bound $\bar{\Pi}^*$'s value given the same prior.

[5] See appendix A.6.2 for the calculations.

5.2 BAMDP Value Decomposition

Theorem 5.1 tells us that to minimize $\bar{\Pi}^m$'s regret, we must align its value estimate $\hat{\bar{Q}}^m$ with the optimal value $\bar{Q}^*$ at the states it visits. This could be achieved by adding shaping rewards to modify $\bar{\Pi}^m$'s perceived values, but $\bar{Q}^*$ is intractable to compute, so we now break it into components that capture different types of value $\bar{\Pi}^m$ misses in different situations, and which may be easier to design approximations to signal as needed. Naturally, these components align with two distinct effects that reward shaping functions are often designed to measure: the information gain of an action, and the inherent utility of the action to the task. We call these the Incremental Value of Information $\bar{\mathcal{I}}$ and the Value of Opportunity $\bar{Q}_O$.

Definition 5.2 (Incremental Value of Information)**.** The $\bar{\mathcal{I}}$ to $\bar{\Pi}$ from taking a in state $\bar{s}_t$ is the increase in the expected return $\bar{\Pi}$ will achieve due to the information gained in the resulting transition:

$$\bar{\mathcal{I}}^{\bar{\Pi}}(\langle s_t, h_t\rangle, a) = \gamma \mathbb{E}_{\bar{T},\bar{R}}[\bar{V}^{\bar{\Pi}}(\langle s_{t+1}, h_{t+1}\rangle) - \bar{V}^{\bar{\Pi}}(\langle s_{t+1}, h_t\rangle)|\bar{s}_t, a]. \tag{9}$$

The $\bar{V}$ are evaluated at s_{t+1} because the information is only actionable from the *next* time step.

E.g., in the caterpillar problem in Figure 1, for $\bar{\Pi}$ that starts knowing ~~prior~~ the actual egg-laying distribution $p(M)$, $\bar{\mathcal{I}}^{\bar{\Pi}}$ of staying at s_w is 0, since $p(M)$ already determines the reward and transition for staying, so no information would be gained and thus no change in behavior. The $\bar{\mathcal{I}}$ of staying for the first time at s_b is $\bar{\Pi}$'s increase in expected return from knowing s_b's reward at the next step.

Definition 5.3 (Value of Opportunity)**.** The $\bar{Q}_O$ to $\bar{\Pi}$ from taking action a in state $\bar{s}_t$ is the expected inherent utility of that decision, i.e.:

$$\bar{Q}_O^{\bar{\Pi}}(\langle s_t, h_t\rangle, a) = \mathbb{E}_{\bar{T},\bar{R}}[r_{t+1} + \gamma \bar{V}^{\bar{\Pi}}(\langle s_{t+1}, h_t\rangle)|\bar{s}_t, a] \tag{10}$$

E.g., $\bar{Q}_O$ of staying at s_b for the first time is the expectation over $p(M)$ of the reward at s_b plus the discounted value of being at s_b at the next step, albeit with no memory of its reward[6].

Lemma 5.4 (Decomposition of Value)**.** *The BAMDP state-action value of any RL algorithm $\bar{\Pi}$ can be decomposed into the sum of the Incremental Value of Information and the Value of Opportunity:*

$$\bar{Q}^{\bar{\Pi}}(\bar{s}_t, a) = \bar{\mathcal{I}}^{\bar{\Pi}}(\bar{s}_t, a) + \bar{Q}_O^{\bar{\Pi}}(\bar{s}_t, a) \tag{11}$$

Thus, the Bayes-optimal $\bar{Q}^*$ can be decomposed into $\bar{\mathcal{I}}^{\bar{\Pi}^*}$ and $\bar{Q}_O^{\bar{\Pi}^*}$, abbreviated by superscript $*$:

$$\bar{Q}^*(\bar{s}_t, a) = \bar{\mathcal{I}}^*(\bar{s}_t, a) + \bar{Q}_O^*(\bar{s}_t, a). \tag{12}$$

Remark 5.5. $\bar{\mathcal{I}}^*$ can never be negative, but $\bar{\Pi}$ that are irrational or have misspecified priors can get negative $\bar{\mathcal{I}}^{\bar{\Pi}}$ for the same reason that giving ignorant or irrational people partial information can mislead them to make worse decisions.

Meanwhile, $\hat{\bar{Q}}^m$ can be understood as the expected return assuming $\bar{\Pi}^m$ will act like $\bar{\Pi}^\pi$, which always outputs task policy π. $\bar{\mathcal{I}}^{\bar{\Pi}^\pi}$ is always 0, since $\bar{\Pi}^\pi$ ignores new information. Thus, given accurate prior $b^m(h_t) = p(M|h_t)$, $\bar{\Pi}^m$ estimates only a $\bar{Q}_O$ term:

$$\hat{\bar{Q}}^m(\bar{s}_t, a) = \max\nolimits_\pi \mathbb{E}_{b^m(h_t)}[Q^\pi(s_t, a)] = \max\nolimits_\pi \bar{Q}^{\bar{\Pi}^\pi}(\bar{s}_t, a) = \max\nolimits_\pi \bar{Q}_O^{\bar{\Pi}^\pi}(\bar{s}_t, a) \tag{13}$$

So with an accurate prior, $\hat{\bar{Q}}^m$ is likely to be more aligned with $\bar{Q}^*$ if $\bar{Q}_O^*$ has more influence. E.g., when exploration is unnecessary because $p(M|h_t)$ already determines π^*, $\bar{\mathcal{I}}^* = 0$ and $\bar{\Pi}^*(\bar{s}_t) = \bar{\Pi}^m(\bar{s}_t) = \pi^*(s_t)$. Or, less trivially, in *dense reward* problems where predictable rewards lead to even more rewards (e.g., coins enticing players forward in 2D platform games, or mice following crumbs to a fallen cookie). Here, $\bar{\Pi}^*$ maximizes $\bar{Q}_O^*$ by going to the predictable rewards - positive reward is received and the agent gets closer to the jackpot - while scant information is available at most steps. Meanwhile, $\hat{\bar{Q}}^m$ is also highest for going to known rewards, so $\bar{\Pi}^m$ does the same.

However, $\hat{\bar{Q}}^m$ is often misaligned, and we show how reward shaping can align it in the next section.

[6] See Appendix A.6.3 for calculations.

6 Effects of Reward Shaping

Given the tools we have developed in the prior sections, we can explain how and under what assumptions of the *task distribution* and the *RL algorithm* itself reward shaping terms are beneficial, shedding light on how to select or design the appropriate reward shaping for a given problem. ~~Specifically, how such terms help Myopic $\bar{\Pi}$ through estimating the Incremental Value of Information or the Value of Opportunity of its decisions.~~

6.1 How Reward Shaping Affects the Myopic Algorithm's Regret

To be explicit about how reward shaping affects the RL algorithm, we analyze its effect as modifying the observed history instead of the BAMDP itself. Denoting shaped history h_t by h'_t, so $\bar{\Pi}^m$'s beliefs become $b^m(h'_t)$, we can express its shaped value estimate $\hat{\bar{Q}}^{m\prime}$ as the sum of separate estimates of value from extrinsic and shaping rewards, Q^π_e and Q^π_x:

$$\hat{\bar{Q}}^{m\prime}(\langle s_t, h'_t\rangle, a) = \max_\pi \mathbb{E}_{b^m(h'_t)}[Q^\pi_e(s_t, a) + Q^\pi_x(s_t, a)]. \tag{14}$$

We can combine theorem 5.1 and equation 14 to conclude that reward shaping can minimize a Myopic algorithm's Bayesian regret by maximizing the following quantity:

$$\mathcal{J}(R^x) = \mathbb{E}_{\bar{M},\bar{\Pi}^m}\big[\sum_{t=0}^{\infty} \gamma^t \bar{Q}^*\big(\bar{s}_t, \arg\max_a \max_\pi \mathbb{E}_{b^m(h'_t)}[Q^\pi_e(s_t, a) + Q^\pi_x(s_t, a)]\big)\big] \tag{15}$$

Equation 15 formalizes the dependence of the optimal R^x on the *task distribution*, via $\bar{Q}^*$ and $\bar{M}$, and the properties of the *learning algorithm*, via b^m. R^x combined with the extrinsic rewards ideally create a **natural curriculum** for $\bar{\Pi}^m$, so at each step the action expected under $b^m(h'_t)$ to maximize return with a fixed π also maximizes the value of the optimal learning algorithm $\bar{Q}^*$.

We can decompose $\bar{Q}^*$ into Incremental Value of Information and Value of Opportunity, using our findings from section 5.2, and categorize and explain many popular shaping functions by which of these components they signal (see table 1). We analyze a subset in more depth in the remainder of this section.

Table 1: Shaping functions grouped by value signalled; bolded terms described in detail below, starred terms in Appendix A.1.

	No $\bar{Q}^*_O$ Signal	Attractive $\bar{Q}^*_O$ Signal	Repulsive $\bar{Q}^*_O$ Signal
No $\bar{\mathcal{I}}^*$ Signal		• **Goal proximity** • Subgoal reaching	• **Negative surprise**
Attractive $\bar{\mathcal{I}}^*$ Signal	• **Prediction error** • **Entropy regularization** • Skill discovery* • Information gain	• Unlocking subtasks*	• **Empowerment**

6.2 Pure $\bar{Q}^*_O$ Signal

These shaping functions help when $\bar{Q}^*_O$ has more influence on $\bar{Q}^*$ but $\hat{\bar{Q}}^m$ is misaligned with $\bar{Q}^*_O$. This often happens when $p(M)$ is very informative of the relative values of reaching states in M (the $\gamma\hat{V}(\langle s_{t+1}, h_t\rangle)$ term in $\bar{Q}^*_O$), but $\hat{\bar{Q}}^m$ is unaware of this information. Thus R^x is often based on just the immediate MDP state transition and is very problem-specific, to correct the perceived value of certain s_i according to the ~~true $p(M)$~~ actual task distribution of the problem.

6.2.1 Attractive $\bar{Q}^*_O$ Signal

These shaping functions often help where $\hat{\bar{Q}}^m$ *underestimates* the value of getting to certain states, by rewarding the agent for reaching them. A common example is *goal proximity*-based shaping (Ng et al., 1999; Ghosh et al., 2018; Lee et al., 2021; Ma et al., 2022); which rewards each step of progress towards a goal in problems where extrinsic reward is only at the goal itself. The goal location varies across tasks but is fully observable from the initial state, therefore taking one step towards it yields no $\bar{\mathcal{I}}^*$ but is Bayes-optimal because it maximizes $\bar{Q}^*_O$. $\bar{\Pi}^m$ knowing $p(M)$ would also approach the goal, since that would also maximize its estimated MDP return, but often its prior is uninformative (e.g. a randomly initialized neural net) so it wouldn't prioritize that behavior. Shaping compensates for this, making $\bar{\Pi}^m$ predict approaching the goal will maximize $\hat{Q}^m$. Another common example with the same underlying mechanism is rewarding points scored in points-based victory games like Pong. But if winning is not purely points-based, this is not necessarily good signal for $\bar{Q}^*_O$; e.g., Clark & Amodei (2016) found an agent learned to crash itself to maximize points, when the true goal was to place first in the race.

6.2.2 Repulsive $\bar{Q}^*_O$ Signal

Shaping functions based on repulsive $\bar{Q}^*_O$ signal help in RL problems where $\bar{\Pi}^m$ takes suboptimal actions because it overestimates their Value of Opportunity, again often due to misspecified priors, by penalizing behavior that goes to states with lower $\bar{Q}^*_O$. A prime example is *negative prediction error* or *surprise*-based reward shaping (Berseth et al., 2019; Eysenbach et al., 2021) which give negative rewards based on the unpredictability of the states and transitions experienced. This is beneficial, assuming:

1. A task distribution where unpredictable situations are undesirable, e.g. for driverless cars where it is dangerous to drive near other erratic vehicles, or robotic surgery, where it is dangerous to use unreliable surgical techniques with highly variable outcomes.
2. An RL algorithm that a priori does not expect danger in unpredictable states and thus overestimates the $\bar{Q}_O$ of exploring them; it could learn to avoid them by getting into some accidents and receiving negative extrinsic rewards, but obviously this is incredibly costly to run in the real world.

These rewards decrease the Bayesian regret of RL algorithms by decreasing their expected value $\hat{Q}^m$ of going to these dangerous states, better aligning it with the optimal task distribution-aware $\bar{Q}^*$, so they return to safety *before* getting into accidents.

More formally, negative surprise shaping works under the assumption that *on the distribution of trajectories the agent actually experiences*, surprise almost always correlates well with negative outcomes. An example where this assumption wouldn't hold is if Times Square were a popular and safe destination for the driverless taxi, but the unpredictability of all the adverts were included in the surprise penalty. This measure of surprise correlates poorly with negative outcomes in trajectories through Times Square, so it would increase regret by making the agent unnecessarily reroute around it. In this problem, the signal for $\bar{Q}^*_O$ must be more specific- only penalizing surprise with respect to things that could cause accidents, such as the positions of other vehicles.

6.3 Attractive $\bar{\mathcal{I}}^*$ Signal Shaping Functions

Many reward shaping functions, often called 'Intrinsic Motivation', are intended to reward behavior that gains valuable experience. Because $\hat{Q}^m$ ignores the value of gaining information, R^x that signal $\bar{\mathcal{I}}^*$ are often very helpful in problems where $\bar{\mathcal{I}}^*$ has significant influence on $\bar{Q}^*$. This commonly holds for *sparse reward* problems where $p(M|h_t)$ is uninformative about where the few rewarding states could be (e.g., random mazes each with just one rewarding goal state). Here, $\bar{Q}^*_O$ is about the same for all actions because most steps get no reward and are just as likely to be getting closer *or* further from the rewarding states. Thus, $\bar{\Pi}^*$ just maximizes $\bar{\mathcal{I}}^*$ by visiting novel states, ruling out possible rewarding states until the goal is found. But without shaping, $\bar{\Pi}^m$ would estimate virtually no value for all actions so wouldn't bother exploring novel states. Thus, these R^x aim to reward the agent for reaching states with more informative h_t, to make $\hat{Q}^{m\prime}$ believe collecting information is inherently rewarding.

- *Prediction error*-based R^x (Schmidhuber, 1991; Pathak et al., 2017; Burda et al., 2018) rewards experiences that are predicted poorly by models trained on h_t. This helps when unpredictability given h_t is good signal for the Incremental Value of Information gained from the observation- thus, for R^x based on dynamics models there must be *minimal stochasticity* (stochastic transitions are always unpredictable but yield no information) and in general *most information must be task-relevant* (so the information gained has value). For example, Burda et al. (2018) observed dynamics-based R^x failing in the 'noisy TV' problem, where a TV that changes channels randomly maximizes R^x despite providing no information. This motivated their design of *RND*, which only predicts features of the current state. However, RND would still fail in problems that don't meet the second criterion, e.g. an 'infinite TV' problem where the TV has infinite unique channels that provide useless information.

- *Entropy regularization* R^x is proportional to the entropy of the task policy's action distribution (Szepesvári, 2010; Mnih et al., 2016; Haarnoja et al., 2017). This increases the estimated return of more stochastic task policies, so it can be understood as adding in the value of exploring a wider range of actions. This helps when $\bar{\Pi}^m$ gets stuck in local maxima, but breaks down if the scale of R^x is too high, because overly random behavior is unlikely to reach interesting states. Therefore it must be carefully balanced with the scale and frequency of the task rewards (Hafner et al., 2023).

6.4 Composite Value Signals

Finally, we can analyze more complex rewards that signal a combination of both $\bar{\mathcal{I}}$ and $\bar{Q}_O$. As an example, we provide a novel interpretation of Empowerment-based intrinsic motivation. The *Empowerment* of an agent is typically measured as the mutual information $I(s';a|s)$ between its actions and their effect on the environment (Klyubin et al., 2005; Gregor et al., 2016). In prior work, this was generally understood as motivating agents to move to the states of "maximum influence" (Salge et al., 2014; Mohamed & Jimenez Rezende, 2015), e.g., the center of the room, or the junction of intersecting hallways. However, this does not always explain the full story. Mohamed & Jimenez Rezende (2015) found that in problems with predators chasing or lava flowing toward the agent, Empowerment motivates it to barricade itself from the lava or avoid the predators- even when this requires holing up in a tiny corner of the room. We can understand this by decomposing Empowerment into the sum of attractive $\bar{\mathcal{I}}$ and repulsive $\bar{Q}_O$ signals:

$$I(s';a|s) = H(a|s) - H(a|s,s'), \tag{16}$$

where we can view $H(a|s)$ as adding attractive signal for $\bar{\mathcal{I}}^*$, similar to Entropy Regularization, encouraging the exploration of different actions such as the barricade-placing action. Meanwhile $-H(a|s,s')$ adds repulsive signal for $\bar{Q}^*_O$, similar to Negative Surprise, signalling that states where the agent dies (which crucially resets it to a random location, i.e., death is a highly unpredictable transition) have low value, and thus should be avoided. Empowerment intrinsic motivation has mostly been tested in small finite environments, but this decomposition suggests its potential for lifelong learning in open-ended worlds, where it can encourage the exploration of a wide range of possibilities while staying out of danger.

7 Related Work

Formal Specifications for RL Problems Abel et al. (2023) recently proposed a formalism where agents (analogous to the RL algorithm) act on histories of experience, but in their definition an RL problem involves only one environment rather than a distribution of environments $p(M)$. The *Epistemic POMDP* (Ghosh et al., 2021) also uses a Bayesian lens to formalize RL algorithms operating under uncertainty over the MDP; however it focuses on zero-shot generalization in offline learning- where there is a training-test split and performance is measured by a single test episode, whereas we study *exploration* in online learning, where performance is measured by return throughout all interactions. This setting is more naturalistic and of practical importance, because agents deployed in the real world must continuously adapt (Jiang et al., 2022).

Reward Shaping can be incredibly effective in some environments and counterproductive in others, but useful theory is still limited (Burda et al., 2018). Aubret et al. (2023) recently propose using mutual information to understand intrinsic rewards falling into 3 categories: mutual information between a learnt model and the observed transitions, between states and their learnt representation, and between self-assigned goals and corresponding trajectories, but this framework ignores the learning algorithm and distribution of MDPs so is less helpful for understanding when and how to use them effectively. Eck et al. (2016) introduce an extension of PBSFs to online POMDP planning, allowing ϕ to be defined over POMDP belief states. They propose a categorization of potential functions that shares similarities with our shaping function taxonomy (specifically their Domain-dependent and Domain-independent categories, corresponding loosely to $\bar{Q}_O$ and $\bar{\mathcal{I}}$ signal respectively) and similarly observe that negative entropy of the belief state can be a potential function for information gain. Like us, Singh et al. (2009) propose that to maximize expected performance, reward shaping should account for properties of both the distribution of MDPs and how $\bar{\Pi}$ learns from experience. This idea also has parallels in bounded rationality;Simon (1955; 1956) argued that rational strategies must be adapted to both structure in the environment and one's cognitive limitations. However, prior works did not derive as direct a relationship between these factors as ours in equation 15.

Value of Information The classical notion of the Value of Information originates in decision theory (Howard, 1966). Early work in metareasoning considers the utility of the information resulting from a computation, applied to tree search (Russell & Wefald, 1989; 1991). The concept was first applied to reinforcement learning by Dearden et al. (1998), who upper bound the 'myopic value of information" for exploring action a by the expected Value of Perfect Information, i.e. the expected gain in return due to learning the true value of the task MDP's $Q^*(s,a)$ given prior beliefs- which doesn't consider the impact the information could have on beliefs about other Q values. Chalkiadakis & Boutilier (2003) proposed viewing BAMDP Q values as involving two main components: an expected value with respect to current beliefs, and a value of the change in beliefs quantified by its impact on subsequent decisions, calling the latter the *expected value of information* of an action. However, they do not derive expressions for each component, only ever expressing the combined value. Ryzhov & Powell (2011) define value of information of pulling a bandit arm as the expected resulting increase in the believed mean reward of the best arm, and derive an exact expression for bandits with exponentially distributed rewards. This value, which they also call the Knowledge Gradient, is related but clearly not equal to the increase in expected return due to the knowledge.

See Appendix A.5 for more related work that was omitted due to space constraints.

8 Discussion

We make contributions to the theoretical understanding of BAMDPs and show how casting RL algorithms as BAMDP policies is a powerful and widely applicable analysis tool. This new perspective implies that RL algorithms should be designed to explore only when the expected value of further information outweighs the value of pure exploitation, rather than exploring until guaranteed convergence to the optimal policy for the underlying MDP. By decomposing RL algorithms' Q values into the Incremental Value of Information and the Value of Opportunity, we provide principles for how to tailor reward shaping functions to the properties of the problems and algorithms they're applied to, and derive a new perspective on the empirical behavior of "Empowerment"-driven agents, more accurately describing the observed behaviors than the prior interpretation. We also demonstrate that existing MDP theory can be easily reused at the BAMDP level, by leveraging results on suboptimality gaps to characterize algorithmic regret, and results on potential-based shaping to derive principles for designing general-purpose intrinsic rewards.

References

David Abel, André Barreto, Hado van Hasselt, Benjamin Van Roy, Doina Precup, and Satinder Singh. On the convergence of bounded agents. *arXiv preprint arXiv:2307.11044*, 2023.

Arthur Aubret, Laetitia Matignon, and Salima Hassas. An information-theoretic perspective on intrinsic motivation in reinforcement learning: a survey. *Entropy*, 25(2):327, 2023.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21, 2008.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pp. 679–684, 1957.

Glen Berseth, Daniel Geng, Coline Devin, Nicholas Rhinehart, Chelsea Finn, Dinesh Jayaraman, and Sergey Levine. Smirl: Surprise minimizing reinforcement learning in unstable environments. *arXiv preprint arXiv:1912.05510*, 2019.

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

Georgios Chalkiadakis and Craig Boutilier. Coordination in multiagent reinforcement learning: A bayesian approach. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pp. 709–716, 2003.

Jack Clark and Dario Amodei. Faulty reward functions in the wild, 2016. URL `https://openai.com/research/faulty-reward-functions`.

Richard Dearden, Nir Friedman, and Stuart Russell. Bayesian q-learning. *Aaai/iaai*, 1998:761–768, 1998.

Marco Dorigo and Marco Colombetti. Robot shaping: Developing autonomous agents through learning. *Artificial intelligence*, 71(2):321–370, 1994.

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

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

Adam Eck, Leen-Kiat Soh, Sam Devlin, and Daniel Kudenko. Potential-based reward shaping for finite horizon online pomdp planning. *Autonomous Agents and Multi-Agent Systems*, 30:403–445, 2016.

Ben Eysenbach, Russ R Salakhutdinov, and Sergey Levine. Robust predictable control. *Advances in Neural Information Processing Systems*, 34:27813–27825, 2021.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

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

Dibya Ghosh, Abhishek Gupta, and Sergey Levine. Learning actionable representations with goal-conditioned policies. *arXiv preprint arXiv:1811.07819*, 2018.

Dibya Ghosh, Jad Rahme, Aviral Kumar, Amy Zhang, Ryan P Adams, and Sergey Levine. Why generalization in rl is difficult: Epistemic pomdps and implicit partial observability. *Advances in Neural Information Processing Systems*, 34:25502–25515, 2021.

John C Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 41(2):148–164, 1979.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

Arthur Guez, David Silver, and Peter Dayan. Efficient bayes-adaptive reinforcement learning using sample-based search. *Advances in neural information processing systems*, 25, 2012.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. *arXiv preprint arXiv:2109.06780*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Mikael Henaff, Minqi Jiang, and Roberta Raileanu. A study of global and episodic bonuses for exploration in contextual mdps. *arXiv preprint arXiv:2306.03236*, 2023.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Ronald A Howard. Information value theory. *IEEE Transactions on systems science and cybernetics*, 2(1):22–26, 1966.

Minqi Jiang, Tim Rocktäschel, and Edward Grefenstette. General intelligence requires rethinking exploration. *arXiv preprint arXiv:2211.07819*, 2022.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476, 2022.

Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. Empowerment: A universal agent-centric measure of control. In *2005 ieee congress on evolutionary computation*, volume 1, pp. 128–135. IEEE, 2005.

Youngwoon Lee, Andrew Szot, Shao-Hua Sun, and Joseph J Lim. Generalizable imitation learning from observation via inferring goal proximity. *Advances in neural information processing systems*, 34:16118–16130, 2021.

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

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PMLR, 2016.

Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 28, 2015.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287. Citeseer, 1999.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Martin Riedmiller, Jost Tobias Springenberg, Roland Hafner, and Nicolas Heess. Collect & infer-a fresh look at data-efficient reinforcement learning. In *Conference on Robot Learning*, pp. 1736–1744. PMLR, 2022.

Stuart Russell and Eric Wefald. On optimal game-tree search using rational meta-reasoning. In *IJCAI*, pp. 334–340. Citeseer, 1989.

Stuart Russell and Eric Wefald. Principles of metareasoning. *Artificial intelligence*, 49(1-3):361–395, 1991.

Ilya O Ryzhov and Warren B Powell. The value of information in multi-armed bandits with exponentially distributed rewards. *Procedia Computer Science*, 4:1363–1372, 2011.

Christoph Salge, Cornelius Glackin, and Daniel Polani. Empowerment–an introduction. *Guided Self-Organization: Inception*, pp. 67–114, 2014.

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

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE transactions on autonomous mental development*, 2(3):230–247, 2010.

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

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pp. 8583–8592. PMLR, 2020.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. Model-based active exploration. In *International conference on machine learning*, pp. 5779–5788. PMLR, 2019.

Max Simchowitz and Kevin G Jamieson. Non-asymptotic gap-dependent regret bounds for tabular mdps. *Advances in Neural Information Processing Systems*, 32, 2019.

Herbert A Simon. A behavioral model of rational choice. *The quarterly journal of economics*, pp. 99–118, 1955.

Herbert A Simon. Dynamic programming under uncertainty with a quadratic criterion function. *Econometrica, Journal of the Econometric Society*, pp. 74–81, 1956.

Satinder Singh, Richard L Lewis, and Andrew G Barto. Where do rewards come from. In *Proceedings of the annual conference of the cognitive science society*, pp. 2601–2606. Cognitive Science Society, 2009.

Satinder P Singh and Richard C Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16:227–233, 1994.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Csaba Szepesvári. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103, 2010.

David Warde-Farley, Tom Van de Wiele, Tejas Kulkarni, Catalin Ionescu, Steven Hansen, and Volodymyr Mnih. Unsupervised control through non-parametric discriminative rewards. *arXiv preprint arXiv:1811.11359*, 2018.

Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. *PhD thesis, Cambridge University, Cambridge, England*, 1989.

Kunhe Yang, Lin Yang, and Simon Du. Q-learning with logarithmic regret. In *International Conference on Artificial Intelligence and Statistics*, pp. 1576–1584. PMLR, 2021.

Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. *arXiv preprint arXiv:1910.08348*, 2019.

A Appendix

A.1 Additional Examples of Reward Shaping Value Signalling

A.1.1 Attractive $\bar{\mathcal{I}}^*$ Signal

- *Mutual Information-based skill discovery* (Sharma et al., 2019; Warde-Farley et al., 2018; Eysenbach et al., 2018) rewards the agent based on the mutual information between the skills (temporally correlated sequences of actions) it learns and the resulting states. The higher this mutual information, the more diverse and controllable the skill set, so it signals the value of the agent's experience honing its skills. It's a good signal for $\bar{\mathcal{I}}^*$ in RL problems where mutual information is a good measure for how useful the set of skills is for maximizing return, which depends on the choice of representation used for the skills and states.
- *Information Gain* is a measure of the amount of information gained about the environment (Lindley, 1956). Info gain-based shaping has led to successful exploration in RL (Sekar et al., 2020; Houthooft et al., 2016; Shyam et al., 2019); it is a good signal for $\bar{\mathcal{I}}^*$ in problems where all information about the MDP is useful for maximizing return. But it could be a distraction in environments with many irrelevant dynamics to learn about, since the quantity of information gained would not always align with the value of that information.

A.2 Combined $\bar{\mathcal{I}}^*$ and $\bar{Q}^*_O$ Signal

Rewards for unlocking new necessary subtasks, e.g. successfully chopping wood for the first time as used in Crafter (Hafner, 2021), adds both the value of discovering how to complete the subtask (because more wood will be needed) and the value of being one step closer to mining the diamond (one less woodblock needed to build a pick-axe). This helps Myopic RL algorithms that lack the prior knowledge that in all initializations of the world permitted under $p(M)$, wood is always a prerequisite for diamonds.

A.3 Characterizing Algorithmic Regret with BAMDP Suboptimality Gaps

Proof. We prove this using the observation of Yang et al. (2021) in their Equation (18) that regret can be expressed in terms of the value missed with each action, i.e. the *Suboptimality Gap*[7].

Definition A.1 (Suboptimality Gap, Simchowitz & Jamieson (2019))**.** Given any $(\bar{s}, a) \in \bar{\mathcal{S}} \times \mathcal{A}$, the *Suboptimality Gap* is defined as the decrease in the value for $\bar{\Pi}^*$ from taking action a at state $\bar{s}$:

$$\bar{\Delta}(\bar{s}, a) = \bar{V}^*(\bar{s}) - \bar{Q}^*(\bar{s}, a). \tag{17}$$

For each step taken, expected regret increases by the additional value missed by choosing that action at that state, and thus total regret is the discounted sum of Suboptimality Gaps (see Yang et al. (2021) for the full proof):

Theorem A.2. *The regret of RL algorithm $\bar{\Pi}$ compared to the Bayes-optimal algorithm $\bar{\Pi}^*$ is equal to the expected discounted sum of Suboptimality Gaps along its trajectory:*

$$\bar{V}^*(\bar{s}_0) - \bar{V}^{\bar{\Pi}}(\bar{s}_0) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \bar{\Delta}(\bar{s}_t, a_t) | a_t = \bar{\Pi}(\bar{s}_t)] \tag{18}$$

Theorem A.2 can be applied to characterize the regret of the Myopic algorithm, which gives us 5.1 as a corollary. □

For example, in our caterpillar problem from Figure 1, at the first step $\bar{\Pi}^*$ would go to s_b, but $\bar{\Pi}^m$ knowing $p(M)$ would stay at s_w. Staying just delays whatever reward $\bar{\Pi}^*$ would eventually get, so the Suboptimality Gap is the loss in value from discounting, $(1-\gamma)\bar{V}^*(\bar{s}_0)$. Staying at s_w forever accumulates these value losses, summing to a total regret equal to the full expected Bayes-optimal return. See section A.7 for an example with the full calculations.

[7] Corollary 2 from Singh & Yee (1994) can also be applied to bound regret in terms of the error of the $\bar{Q}^*$ estimate, but we can get more precise characterization from Yang et al. (2021)'s result

A.4 Potential Based Shaping in the BAMDP

A potential-based shaping function (PBSF) in the BAMDP is of the form $\gamma\phi(\bar{s}_{t+1}) - \phi(\bar{s}_t)$. The potential-based shaping theorem Ng et al. (1999) also applies in BAMDPs, telling us that BAMDP potential-based shaping functions preserve the behavior of Bayes-optimal RL algorithms. Although we would typically use reward shaping to change the behavior of a non-optimal Rl algorithm, a guarantee that it doesn't affect the optimal algorithm is a nice property to have.

Because BAMDP state includes h_t, many intrinsic motivation (IM) functions based on accumulating experience can be valid BAMDP PBSFs, e.g. *information gain* (Houthooft et al., 2016) corresponds to $\phi(\bar{s}_t) = -H(\hat{p}(T|h_t))$ i.e. the certainty of the algorithm's belief over the task dynamics $\hat{p}(T)$ after updating on h_t:

$$H(\hat{p}(T|h_t)) - \gamma H(\hat{p}(T|h_{t+1})) \;\widehat{=}\; \gamma\phi(\bar{s}_{t+1}) - \phi(\bar{s}_t) \quad | \quad \phi(\bar{s}_t) = -H(\hat{p}(T|h_t)) \tag{19}$$

Similarly, *novelty* or *count-based* IM functions (Bellemare et al., 2016; Schmidhuber, 2010) correspond to PBSFs where $\phi(\bar{s}_t)$ is the number of unique task MDP states in h_t.

Theorem A.3. *For a reward shaping function to guarantee that the Bayes-optimal algorithm for the shaped RL problem is also Bayes-optimal for the original problem (and vice versa), it is necessary and sufficient condition for it to be a BAMDP PBSF.*

The key insight is that $\bar{\Pi}^*$ maximizes the infinite disounted sum of rewards; the contribution of PBS rewards to this sum is a constant [8].

Proof. Recall that the Bayes-optimal algorithm $\bar{\Pi}^*$ maximizes the discounted sum of rewards in the BAMDP:

$$\bar{\Pi}^* = \arg\max_{\bar{\Pi}} \mathbb{E}_{\bar{M},\bar{\Pi}}[\sum_t \gamma^t \bar{R}(\bar{s}_t, a_t)] \tag{20}$$

Denote the shaped RL problem as $\bar{M}'$ and the optimal algorithm for it $\bar{\Pi}^{*'}$. It thus maximizes the following expression:

$$\bar{\Pi}^{*'} = \arg\max_{\bar{\Pi}} \mathbb{E}_{\bar{M}',\bar{\Pi}}[\sum_t \gamma^t \bar{R}'(\bar{s}_t, a_t)] \tag{21}$$

The transition function in $\bar{M}'$ is the same, modulo the shaped history h_t^x containing rewards shifted by the shaping function:

$$\bar{T}'(\bar{s}_{t+1}|\bar{s}_t, a_t) = \mathbb{E}_{p(\bar{M}|h_t)}[T(s_{t+1}|s_t, a_t)R(r_t|s_t, a_t)\mathbb{1}[h_{t+1} = h_t^x a_t(r_t + \gamma\phi(\bar{s}_{t+1}) - \phi(\bar{s}_t))s_{t+1}]] \tag{22}$$

Similarly, the expected reward in $\bar{M}'$:

$$\mathbb{E}_{\bar{M}'}[\bar{R}'(\bar{s}_t, a_t)] = \mathbb{E}_{\bar{M}'}[\mathbb{E}_{p(\bar{M}'|h_t)}[R(s_t, a_t)]] = \mathbb{E}_{\bar{M}}[\mathbb{E}_{p(M|h_t)}[R(s_t, a_t)] + \gamma\phi(\bar{s}_{t+1}) - \phi(\bar{s}_t)] \tag{23}$$

The shaping rewards cancel out in the infinite discounted sum:

$$\begin{aligned} \sum_{t\,=0}^{\infty} \gamma\phi(\bar{s}_{t+1}) - \phi(\bar{s}_t) &= \gamma\phi(\bar{s}_1) - \phi(\bar{s}_0) + \gamma^2\phi(\bar{s}_2) - \gamma\phi(\bar{s}_1) + \gamma^3\phi(\bar{s}_3) - \gamma^2\phi(\bar{s}_2) + \ldots \\ &= -\phi(\bar{s}_0) + \gamma(\phi(\bar{s}_1) - \phi(\bar{s}_1)) + \gamma^2(\phi(\bar{s}_2) - \phi(\bar{s}_2)) + \ldots \\ &= -\phi(\bar{s}_0) \end{aligned} \tag{24}$$

[8] BAMDP PBSFs *can* affect *non-Bayes-optimal* algorithms' behavior, e.g., encouraging exploration, because they do not calculate this full discounted return.

Plugging this in:

$$\begin{aligned}
\bar{\Pi}^{*'} &= \arg\max_{\bar{\Pi}} \mathbb{E}_{\bar{M}',\bar{\Pi}}[\sum_{t=0}^{\infty} \gamma^t \bar{R}(\bar{s}_t, a_t)] \\
&= \arg\max_{\bar{\Pi}} \mathbb{E}_{\bar{M}',\bar{\Pi}}[\sum_{t=0}^{\infty} \gamma^t \bar{R}(\bar{s}_t, a_t) + \gamma\phi(\bar{s}_{t+1}) - \phi(\bar{s}_t)] \\
&= \arg\max_{\bar{\Pi}} \mathbb{E}_{\bar{M}',\bar{\Pi}}[\sum_{t=0}^{\infty} \gamma^t \bar{R}(\bar{s}_t, a_t) + \sum_{t=0}^{\infty} \gamma\phi(\bar{s}_{t+1}) - \phi(\bar{s}_t)] \\
&= \arg\max_{\bar{\Pi}} \mathbb{E}_{\bar{M},\bar{\Pi}}[\sum_{t=0}^{\infty} \gamma^t \bar{R}(\bar{s}_t, a_t) - \phi(\bar{s}_0)] \\
&= \bar{\Pi}^*
\end{aligned} \tag{25}$$

□

A.5 More Related Work

Formulating Exploration Problems: Riedmiller et al. (2022) separate RL into collecting data by interacting with the environment, and inferring knowledge about the environment from the collected data. This does not aim to optimize task performance during data collection i.e. the ”learning” phase; performance is assessed in a separate ”deployment” phase. We believe that it is important to consider the setting where performance during data collection matters too. Jiang et al. (2022) reconceptualize exploration as a search process over the space of MDPs, involving active discovery and invention of new MDPs to keep learning more. We consider the more basic problem of exploring effectively in a single MDP sampled from a given distribution, which is still a major challenge in RL.

Henaff et al. (2023) study exploration bonuses in contextual MDPs, where the dynamics are sampled from a distribution at the start of *every episode,* and the goal is to learn *one* policy that performs well across all contexts. We instead study the setting where the RL algorithm learns a different policy for each MDP, and the goal is to design an algorithm that can learn effectively across a distribution of MDPs. Our goal is to be good at learning in general, their goal is to learn one policy well. They find that global novelty bonuses work when the contexts are more similar, and episodic novelty bonuses work when the contexts are more different. We can explain this in the BAMDP framework by thinking of a CMDP as a lifelong infinite-sized MDP where the end of an episode corresponds to transitioning to a new part of the state space and resetting the episodic novelty counter. The more similar the contexts, the lower the $\bar{\mathcal{I}}$ of an experience that already happened in a previous context, and thus the better signal a global novelty bonus will provide over episodic.

A.6 Caterpillar Problem Analysis

A.6.1 Bayes-Optimal Policy Values

In section 3 we describe the behavior for the Bayes-optimal algorithm: for large enough γ, $\bar{\Pi}^*$ should check s_b first, then stay forever if it's alive, otherwise return to s_w forever. Let's look at the $\bar{Q}^*$ values in this case, with $\gamma = 0.95$. First, the value of going to s_b:

$$\bar{Q}^*(\bar{s}_0, go) = -5 + \gamma\mathbb{E}_{p(M)}[\bar{V}^*(\langle s_b, h_1^b \rangle)], \tag{26}$$

where the first term is the energy cost of travelling. Now the value from h_1^b is the weighted sum of the values in the presence and absence of food at s_b:

$$\mathbb{E}_{p(M)}[\bar{V}^*(\langle s_b, h_1^b \rangle)] = 0.1\frac{150}{1-\gamma} + 0.9(-5\gamma + \frac{21\gamma^2}{1-\gamma}) = 637, \tag{27}$$

where the first term is the return from eating at s_b forever, and the second is from going back to eat at s_w forever. Plugging this in, we get $\bar{Q}^*(\bar{s}_0, go) = 600$.

Now, the Q value for staying at s_w:

$$\bar{Q}^*(\bar{s}_0, stay) = 21 + \gamma \mathbb{E}_{p(M)}[\bar{V}^*(\langle s_w, h_1^w \rangle)]. \tag{28}$$

Since h_1^w contains no more information than h_0, $\bar{\Pi}^*(\langle s_w, h_1^w \rangle)$ would make the same choice as $\bar{\Pi}^*(\bar{s}_0)$ i.e. to check s_b, so $E_{p(M)}[V^*(\langle s_w, h_1^w \rangle)] = \bar{Q}^*(\bar{s}_0, go) = 600$. This gives us:

$$\bar{Q}^*(\bar{s}_0, stay) = 21 + 600\gamma = 591 < \bar{Q}^*(\bar{s}_0, go), \tag{29}$$

and thus $\bar{\Pi}^*$ would first go to s_b.

A.6.2 Myopic Algorithm Values

In section 4.2 we describe how the Myopic RL algorithm would act in the caterpillar MDP example. Here we go through the full calculations.

Algorithm $\bar{\Pi}^m$, assuming it had the correct prior $p(M)$, would estimate the values of following various π as follows:

- π_b goes to the bush and stays there: $E_{p(M)}[V^{\pi_b}(s_w)] = -5 + 0.1 \times 150\frac{\gamma}{1-\gamma} = 280$
- π_{alt} alternates between the plants: $E_{p(M)}[V^{\pi_{alt}}(s_w)] = -5\frac{1}{1-\gamma} = -100$
- π_w stays at the weed forever: $E_{p(M)}[V^{\pi_w}(s_w)] = 21\frac{1}{1-\gamma} = 420$; and it would go from s_b so $E_{p(M)}[V^{\pi_w}(s_b)] = -5 + \gamma 420 = 394$
- π_{stay} always stays wherever it is, so $E_{p(M)}[V^{\pi_{stay}}(s_w)] = 21\frac{1}{1-\gamma} = 420$ and $E_{p(M)}[V^{\pi_{stay}}(s_b)] = 0.1 \times 150\frac{1}{1-\gamma} = 300$

Because π_w gets the highest estimated value, $\bar{\Pi}^m$ would choose to follow it, thus never learning about the bush and staying at the weed forever.

As an example of $\bar{\Pi}^m$ underestimating its own value, take its estimate of its value of staying from $\langle s_b, h_0 \rangle$, i.e. if s_0 was actually at the bush. It assumes it would follow the best task policy under current information at the next step no matter what it found at s_b, which is still π_w, giving estimate:

$$\hat{\bar{Q}}^m(\langle s_b, h_0 \rangle, stay) = E_{p(M)}[R(s_w, stay) + \gamma V^{\pi_w}(s_b)] = 0.1 \times 150 + 394\gamma = 369 \tag{30}$$

However, this is very wrong. If $\bar{\Pi}^m$ stayed at s_b and then found no food, it would update to π_w to go and stay at s_w, and if it did find food it would update to a π that continues staying at s_b. This behavior corresponds to this much higher true value:

$$\bar{Q}^m(\langle s_b, h_0 \rangle, stay) = 0.1\frac{150}{1-\gamma} + 0.9(-5\gamma + \frac{21\gamma^2}{1-\gamma})) = 637 \tag{31}$$

A.6.3 Caterpillar Problem Value Decomposition

For any $\bar{\Pi}$ that starts out ~~with the true prior~~ knowing the actual egg laying distribution $p(M)$ and does Bayesian updating, its $\bar{\mathcal{I}}$ of staying at the weed is 0 because there would be no new information gained and thus no change in behavior. The $\bar{Q}_O$ is $21 + \gamma\bar{V}^{\bar{\Pi}}(\langle s_w, h_0 \rangle)$, the payout from eating weeds and the discounted value of starting the next step at s_w.

The $\bar{\mathcal{I}}$ of going to s_b at the first step is also 0 because no new information is revealed yet. The $\bar{Q}_O^{\bar{\Pi}}$ of going to s_b at the first step is the energy cost of going plus the discounted value from starting the next step at s_b, which is $-5 + \gamma\bar{V}^{\bar{\Pi}}(\langle s_b, h_0 \rangle)$.

The $\bar{\mathcal{I}}$ of staying at s_b for the first time is the expected added value to $\bar{\Pi}$ of knowing s_b's payout when starting the next step at s_b (denoted by h^b), which is $\gamma(\bar{V}^{\bar{\Pi}}(\langle s_b, h^b \rangle) - \bar{V}^{\bar{\Pi}}(\langle s_b, h_0 \rangle))$. The $\bar{Q}_O$ of staying at s_b for the first time is the expected reward plus the discounted value of starting the next step at s_b without knowing anything new, which is $0.1 \times 150 + \gamma\bar{V}^{\bar{\Pi}}(\langle s_b, h_0 \rangle))$.

A.7 Bear and Treasure Chest

Take a problem where the agent starts in front of a closed door with a 50-50 chance of either treasure (reward 50) or a bear (reward -100) on the other side. The agent can stay, go up to peek at the keyhole, or go through the door. At the first timestep, the Bayes optimal algorithm would peek at the keyhole, but a greedy algorithm might stay in front of the door. This delays whatever reward $\bar{\Pi}^*$ would eventually get, so the Suboptimality Gap is the loss in value due to the discount factor, $(1-\gamma)\bar{V}^*(\bar{s}_0)$. An RL algorithm that stays in front of the door forever would accumulate these value losses, summing to a total regret equal to the full expected Bayes optimal return $\gamma 50$.

By staying, $\bar{\Pi}^m$'s $\bar{Q}_O$ Suboptimality Gap is 0 but its $\bar{\mathcal{I}}$ Suboptimality Gap is the entire added value to $\bar{\Pi}^*$ of knowing which door contains the treasure. If it knows which room contains the treasure, it enters and gains $r = 50$, otherwise it peeks and enters at the next step, so $\bar{V}^*(\langle s_0, h_1\rangle) = 50$ and $\bar{V}^*(\langle s_0, h_0\rangle) = \gamma 50$, and thus the $\bar{\mathcal{I}}$ residual is:

$$\gamma(\bar{V}^*(\langle s_0, h_1\rangle) - \bar{V}^*(\langle s_0, h_0\rangle)) = \gamma 50(1-\gamma) \tag{32}$$

Both $\bar{Q}_O$ and $\bar{\mathcal{I}}$ Suboptimality Gaps are constant because $\bar{\Pi}^m$'s state remains at s_0 and the beliefs after staying are constant, and thus $\bar{\Pi}^m$'s overall regret is:

$$E_{p(M)}[G(\bar{\Pi}^*) - G(\bar{\Pi}^m)] = \gamma 50(1-\gamma)(1+\gamma+\gamma^2+...) = \gamma 50(1-\gamma)\frac{1}{1-\gamma} = \gamma 50 \tag{33}$$

as expected.