

---

# Expressing Non-Markov Reward to a Markov Agent

---

**David Abel**  
DeepMind  
London, UK  
dmabel@deepmind.com

**André Barreto**  
DeepMind  
London, UK  
andrebarreto@deepmind.com

**Michael Bowling**  
DeepMind  
London, UK  
bowlingm@deepmind.com

**Will Dabney**  
DeepMind  
London, UK  
wdabney@deepmind.com

**Steven Hansen**  
DeepMind  
London, UK  
stevenhansen@deepmind.com

**Anna Harutyunyan**  
DeepMind  
London, UK  
harutyunyan@deepmind.com

**Mark K. Ho**  
Princeton University  
Princeton, NJ  
mho@princeton.edu

**Ramana Kumar**  
DeepMind  
London, UK  
ramanakumar@deepmind.com

**Michael L. Littman**  
Brown University  
Providence, RI  
mlittman@cs.brown.edu

**Doina Precup**  
DeepMind  
London, UK  
doinap@deepmind.com

**Satinder Singh**  
DeepMind  
London, UK  
baveja@deepmind.com

## Abstract

Markov Decision Processes are the standard model of sequential decision-making problems in reinforcement learning. However, as noted by [Abel et al. \[1\]](#), for some environments, there exist choices of task that cannot be expressed as a reward function that is Markovian on the environment’s state space. We here address this limitation by studying a particular form of state-construction that is designed to systematically enrich the expressivity of reward. Concretely, we introduce the Split Markov Decision Process, a model of sequential decision-making problems with decoupled environment-state and reward-state, reminiscent of reward machines [2]. Using this model, we generalize one of the central questions of [Abel et al. \[1\]](#) regarding the expressivity of reward: given any task and Markovian environment, does there exist a reward function defined over some reward-state space that can express the task? Our main result answers this question in the affirmative for one type of case by offering a procedure that builds the realizing reward structures. We close by exploring basic aspects of reinforcement learning under these realizing reward structures in small-scale experiments, and call attention to open questions of interest.

**Keywords:** Reward, Reward Hypothesis, Expressivity, Markov Decision Process, MDP, Reinforcement Learning.

## Acknowledgements

The authors would like to thank Hado van Hasselt, Alex Turner, Jacob Buckman, and Ben Van Roy for helpful conversations.

# 1 Introduction

Discrete time Markov Decision Processes (MDPs) are the standard model of environments in reinforcement learning (RL). Naturally, restricting attention to MDPs limits the space of representable RL problems—for instance, we might instead consider cases in which there is no available description of state, that time is not discrete, or that either the transition or reward function depend partially on history. Indeed, [Abel et al. \[1\]](#) explore the expressivity of Markov reward as a possible limitation to the space of tasks we can represent. Given an environment modeled as a Controlled Markov Process (CMP: [Definition 1](#)), [Abel et al. \[1\]](#) suppose that a designer (Alice) forms preferences over the CMP that she will translate into a reward function to incentivize a learning agent (Bob) to realize the chosen preferences. [Abel et al. \[1\]](#) then ask: for any choice of such preferences and CMP, will there always exist a Markov reward function that captures Alice’s preferences?

One of the main results (Theorem 4.1) of [Abel et al. \[1\]](#) illustrates that there are two known types of failure cases in which the expressivity of Markov reward functions is lacking. The first they call the “steady state type” in which Alice holds preferences over events that occur with zero probability. For this reason, reward fails to elicit behavior that captures the given preferences, only because the behavior does not factor into the start-state value (or return) of the preferred behaviors. The second they call the “entailment type”, in which the value of the desired preferences is entangled. For example, consider the “always move in the same direction” task in a grid world. Here, the CMP is a grid world with environment state defined according to a typical  $(x, y)$  pair. However, there is no Markov reward function that can properly incentivize an agent to prefer the four “go the same direction” policies above all alternatives—such reward functions depend on knowledge of either history or the future.

**Results Overview.** We here provide one kind of remedy to entailment issues. Our construction also yields a new and potentially interesting model of sequential decision-making problems we call the Split Markov Decision Process (Split-MDP) that bears heavy resemblance to *reward machines* [2]: a Split-MDP follows from the insight that rewards need not be based on the same notion of state as the environment’s transitions. Concretely, a Split MDP is a CMP paired with an automaton-like structure we call a *reward bundle* ([Definition 3](#)) that produces reward. We use this model to provide a simple constructive proof that one aspect of the limited expressivity of Markov reward identified by [Abel et al. \[1\]](#) can be fixed by augmented state (or, said differently, that non-Markovian rewards do not suffer from entailment issues when encouraging Markov policies). Lastly, we conduct small-scale experiments that provide additional support to our findings, and highlight open questions of interest.

## 1.1 Preliminaries: CMPs, Tasks, and the Split-MDP

We first introduce relevant concepts needed to make our study precise, though we adopt many of the conventions from [Abel et al. \[1\]](#). We begin with an environment described by a CMP, defined as follows.

**Definition 1.** A *Controlled Markov Process (CMP)* is a model of an environment,  $E = (\mathcal{S}_p, \mathcal{A}, p, \gamma, s_p^{(0)})$ , where:  $\mathcal{S}_p$  is a finite set of environment states,  $\mathcal{A}$  is a finite set of actions,  $p : \mathcal{S}_p \times \mathcal{A} \rightarrow \Delta(\mathcal{A})$  is a transition function,  $\gamma \in [0, 1)$  is a discount factor, and  $s_p^{(0)} \in \mathcal{S}_p$  is the environment start-state.

Then, a designer (Alice) inspects the environment and forms some set of preferences over desired outcomes. Following [Abel et al. \[1\]](#), we will be initially focused on *sets of acceptable policies* (SOAPs), defined as follows, though our main result extends to policy orderings (POs) and trajectory orderings (TOs) defined by [Abel et al. \[1\]](#) as well.

**Definition 2.** A *Set Of Acceptable Policies (SOAP)* is a non-empty subset of the deterministic policies,  $\Pi_G \subseteq \Pi$ , with  $\Pi$  the set of all deterministic mappings from  $\mathcal{S}_p$  to  $\mathcal{A}$  for a given  $E$ .

As discussed, one of the central questions of [Abel et al. \[1\]](#) asks: for a given  $(E, \Pi_G)$  pair, will there exist a reward function that is Markov on  $\mathcal{S}_p$  that can capture the given SOAP in  $E$ ? A reward function *captures* the SOAP just when the start-state value induced by the given reward function adheres to the constraints of the SOAP. That is, the good policies all have strictly higher start-state value than the bad policies (“range” SOAP), or the good policies are all *optimal* and have strictly higher start-state value than the bad (“equal” SOAP). We here generalize this question to the case where Alice can choose not just a reward function, but also a state space for the reward function to operate over.

**The Split Markov Decision Process.** We next introduce two new structures: (1) A reward bundle ([Definition 3](#)): A collection of structures that produce reward, reminiscent of reward machines [2]; and (2) the Split-MDP ([Definition 4](#)): An MDP formed by a CMP paired with a reward bundle. This model is motivated by the observation that the basic laws of an environment might depend on *different* information than the reward function, and thus, we can decouple environment-state from reward-state.

**Definition 3.** A *reward bundle* is a quadruple,  $\mathcal{R} = (\mathcal{S}_r, r, f, s_r^{(0)})$ , where:  $\mathcal{S}_r$  is a finite set of reward states,  $r : \mathcal{S}_r \times \mathcal{A} \times \mathcal{S}_p \rightarrow \mathbb{R}$  is a reward function,  $f : \mathcal{S}_r \times \mathcal{S}_p \times \mathcal{A} \rightarrow \mathcal{S}_r$  is a deterministic reward-state dynamics function, and  $s_r^{(0)} \in \mathcal{S}_r$  is the reward start-state.

We may combine a CMP with a reward bundle to form a Split-MDP as follows.

**Definition 4.** A *Split Markov Decision Process (Split-MDP)* is any MDP formed by pairing a CMP with a reward bundle,  $M = (E, \mathcal{R})$ , yielding  $\mathcal{S} = \mathcal{S}_p \times \mathcal{S}_r$ ,  $s^{(0)} = (s_p^{(0)}, s_r^{(0)})$ , and  $p_M : s_p \times s_r \times a \mapsto p(s_p, a), f(s_r, s_p, a)$ .

In cases where the reward-state space is unknown or unobservable to a learning algorithm, it might be natural to suppose an agent interacts with a Split-MDP but only observes the pair  $(s_p^{(t)}, r(s_r^{(t)}))$ . Such a setting induces a specific kind of partially observable MDP (POMDP) [3] in which the next observation,  $s_p^{(t+1)}$ , is predictable from  $s_p^{(t)}$  and  $a^{(t)}$  alone. We call such a model a *Split-POMDP*, and note that it defines a friendly class of POMDPs; we anticipate there are many interesting questions to pursue about Split-POMDPs beyond our scope.

## 2 Results.

First, we inspect whether there is *always* a reward bundle to realize a given SOAP. As mentioned in the introduction, we are focused on the “entailment cases”, and set aside “steady state” issues by invoking the following assumption.

**Assumption 1.** All tasks only contain preferences over outcomes that occur with non-zero probability.

This assumption is just a concise way of limiting our attention to entailment cases, but we note that there are related arguments that go beyond this assumption that are out of scope for this paper.

**Proposition 1.** Under Assumption 1, for any choice of CMP  $E$  and SOAP  $\Pi_G$  (with policies defined on  $\mathcal{S}_p$ ), there exists a reward bundle  $\mathcal{R}$  such that the optimal policies in  $M = (E, \mathcal{R})$  are equivalent to those in  $\Pi_G$ .

### Proof of Proposition 1.

We are given a finite CMP,  $E = (\mathcal{S}_p, \mathcal{A}, p, \gamma, s_p^{(0)})$ , and a SOAP  $\Pi_G$  where each  $\pi_g \in \Pi_G$  is a mapping from  $\mathcal{S}_p$  to  $\mathcal{A}$ . We want to show that there is a reward bundle,  $\mathcal{R} = (\mathcal{S}_r, r, f, s_r^{(0)})$ , such that in the resulting Split-MDP  $M = (E, \mathcal{R})$ , the optimal policies,  $\Pi_M^* = \arg \max_{\pi \in \Pi} V_M^\pi(s^{(0)})$ , agree with the SOAP on each  $s_p$ ,

$$(i) \quad \pi_M(s_p, s_r) = \pi_g(s_p), \quad \forall s_p, s_r \in \mathcal{S}_p \times \mathcal{S}_r \quad \forall \pi_M \in \Pi_M^* \quad \exists \pi_g \in \Pi_G, \quad (1)$$

$$(ii) \quad |\Pi_M^*| = |\Pi_G|. \quad (2)$$

We proceed by constructing such a reward bundle:

- *Reward state space.* Let  $\mathcal{S}_r = \mathcal{P}(\Pi_G)$ , where  $\mathcal{P}(X)$  denotes the powerset of  $X$ .
- *Reward start-state.* Let  $s_r^{(0)}$  be  $\Pi_G$ .
- *Reward state dynamics.* We define  $f$  to remove any policy from  $\Pi_G$  that is inconsistent with the state-action taken, and to repopulate the full SOAP when all policies have been removed:

$$f(s_r, s_p, a) = \begin{cases} s_r \setminus \Pi_{\neq}(s_r, s_p, a) & |s_r| > 0 \\ \Pi_G & \text{otherwise.} \end{cases} \quad (3)$$

where  $\Pi_{\neq}(s_r, s_p, a)$  is the set of all policies in  $s_r$  that do *not* take  $a$  in  $s_p$ .

- *Reward function.* Lastly, let  $r(s_r^{(t-1)}, a^{(t)}, s_r^{(t)}) = \mathbb{I}\{\exists \pi \in s_r : \pi(s_p^{(t)}) = a^{(t)}\}$ .

There are two key facts about this reward bundle. First, the reward function provides +1 reward to those state action pairs that *agree* with one of the acceptable policies. Second, the reward state dynamics function will *remove* any policy from the reward state that is inconsistent with an action taken. Thus, at any point in time, the *only* policies that remain in the reward state are those consistent with every action taken thus far.  $\square$

In some situations, we might imagine that the reward-state is either unknown or unobservable to the learning agent, thus inducing a Split-POMDP. We next highlight the fact that, by direct consequence of the previous construction, there will exist optimal policies in the Split-POMDP that only depend on environment-state.

**Corollary 1.** *Consider a Split-MDP constructed from an  $(E, \Pi_G)$  pair according to the procedure outlined in the proof of Proposition 1. When viewed as a POMDP with  $s_r$  the hidden state and  $s_p$  the observation, there will always exist an optimal deterministic policy that only depends on environment-state,  $s_p$ .*

Thus, when the agent only observes environment-state, there is still a *representable* optimal policy, unlike traditional POMDPs in which memoryless policies are known to be arbitrarily sub-optimal [6, 4]. We explore this consequence further in our experiments (subsection 2.1).

Furthermore, we find that the *trajectory ordering* and the *policy ordering* cases considered by Abel et al. [1] can be captured by a similar construction.

**Corollary 2.** *Under Assumption 1, for any choice of Split-CMP  $E$  and trajectory ordering  $L_{\tau, N}$  or policy ordering  $L_{\Pi}$  (with trajectories and policies defined over  $S_p$ ), there exists a reward bundle  $\mathcal{R}$  such that the start-state return of the trajectories of  $L_{\tau, N}$  or start-state value of the ordering  $L_{\Pi}$  adheres to the constraints specified by the given task.*

We exclude the full proof due to space constraints, but note that the idea is a straightforward extension of the earlier proof. In the case of trajectory orderings, we define  $\mathcal{S}_r$  to be the space of length  $[1 : N]$  trajectories, and define  $r : s_r \mapsto \mathbb{1}\{|s_r| = N\} \times (\text{RMAX} - \text{rank}(s_r))$  so that end-of-trajectory reward ranks the trajectories from best to worst. In the case of a policy ordering, let  $\mathcal{S}_r = \mathcal{P}(L_{\Pi})$ , and provide reward each time-step proportional to the inverse-rank of the best policy still consistent with the behavior. Both constructions ensure that each ordering is respected.

## 2.1 Experiments

We conduct experiments with a simple SOAP and CMP reminiscent of the XOR problem from Abel et al. [1]. The CMP has three environment-states and two actions, pictured in Figure 1a. The desired SOAP contains four policies, each requiring that the agent take *different* actions across two of the three environment-states. We construct a reward bundle with four-reward states (pictured in Figure 1b) according to the constructive procedure described by Proposition 1, and note that, as a consequence, there will exist four optimal policies over *just*  $s_p$ . Our experiments are intended to explore two questions. First, we examine learning curves of a variety of agents interacting with this environment to study the relationship between the representability of a good policy and its learnability—when viewed as a Split-POMDP, we know typical learning algorithms can *represent* each of the policies in the SOAP (by Corollary 1), but do not know whether learning these policies is also feasible (building on the results in Section 4.2 by McCallum [5]). Second, we inspect whether the proposed reward bundles can in fact incentivize learning algorithms to discover *any* of the acceptable policies, rather than just learn a single desired behavior. For simplicity, we experiment with tabular Q-learning of two variations: (1) When viewing the problem as an MDP, so  $(s_p, s_r, r(s_r))$  are given as input each time step, and (2) When viewing the problem as a POMDP, so  $(s_p, r(s_r))$  are given as input each time step. We further vary the initialization of  $Q$  between all zeros and uniform random from the interval  $[0, 1]$ , as well as different settings of the exploration parameter ( $\epsilon$ , no annealing) used in  $\epsilon$ -greedy action selection. We set the learning rate  $\alpha = 0.05$  and discount factor  $\gamma = 0.95$ .

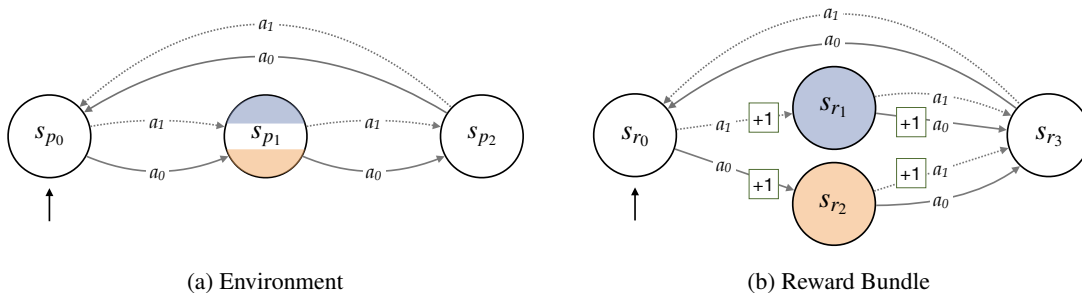


Figure 1: The XOR-like environment used in the experiments (left), with the reward bundle (right) constructed by the procedure described in the proof of Proposition 1. The desired SOAP contains all policies that disagree on action choice across  $s_{p_0}$  and  $s_{p_1}$ . That is,  $\Pi_G = \{\pi_{010}, \pi_{100}, \pi_{011}, \pi_{101}\}$ , where  $\pi_{010}$  denotes  $\{s_{p_0} \mapsto a_0 \mid s_{p_1} \mapsto a_1 \mid s_{p_2} \mapsto a_0\}$ .

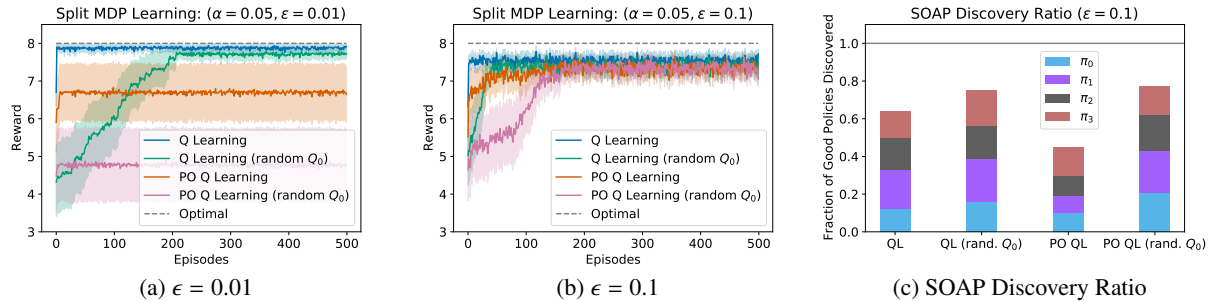


Figure 2: The first two figures present results from Q-learning interacting with an environment as a Split-MDP (blue) compared to a Split-POMDP (orange, “PO” prefix), when the rewards are generated from a well-constructed reward bundle as per Proposition 1. We further contrast performance relative to a randomly initialized Q function in both the MDP (green) and POMDP (pink, “PO” prefix) variations. The y-axis displays the mean, per-episode reward, averaged over 100 runs of the experiment with 95% confidence intervals, with optimal performance shown in the grey dashed line. The third plot illustrates the fraction of the four acceptable policies discovered by each learning algorithm in the final 50 episodes of learning across *all* runs of the experiment.

Results are presented in Figure 2. In Figure 2a and Figure 2b, we plot learning curves of all four agent varieties. We observe that in both zero-initialized and randomly-initialized learning for the standard MDP case (shown in blue and green, respectively), Q-learning can reliably discover optimal behavior, corroborating Proposition 1. In the POMDP case (orange and pink), the results suggest that, depending on  $\epsilon$ , PO Q-learning will either achieve the *same level of performance* as its MDP counterpart (as in  $\epsilon = 0.1$ ), or that there is a statistically significant gap separating the performance of the two (as in  $\epsilon = 0.01$ ). This suggests that learning in a Split-POMDP is sometimes feasible; a natural direction for future work will further clarify the precise conditions under which effective learning is always possible. In Figure 2c, we visualize the fraction of time that each agent’s greedy policy at the end of the episode is one of the SOAP policies (in just the final 50 episodes, averaged over 100 runs of the experiment). These results demonstrate that *all four* learning algorithms can reliably recover each one of the acceptable policies during learning, in roughly equal proportion. These results indicate that well-constructed reward bundles can in fact enhance what is learnable, even when the agent does not have access to the reward-state. A key direction for future work will identify simple learning procedures that can automatically construct agent-state to discover any kind of behavior expressible by reward.

## 2.2 Discussion

This work presents a simple state construction procedure that can enrich the expressivity of reward. Our results patch one of the holes identified by Abel et al. [1], and show that we can produce reward that is often conducive to learning. Lastly, we introduce the Split-MDP and Split-POMDP, which we believe may offer useful perspectives for studying state-construction, learning under partial-observability, and task complexity in RL.

## References

- [1] David Abel, Will Dabney, Anna Harutyunyan, Mark K. Ho, Michael L. Littman, Doina Precup, and Satinder Singh. On the expressivity of Markov reward. In *Advances in Neural Information Processing Systems*, 2021.
- [2] Rodrigo Toro Icarte, Toryn Klassen, Richard Valenzano, and Sheila McIlraith. Using reward machines for high-level task specification and decomposition in reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2018.
- [3] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998.
- [4] Michael L. Littman. Memoryless policies: Theoretical limitations and practical results. In *Proceedings of the International Conference on Simulation of Adaptive Behavior*, 1994.
- [5] Andrew K. McCallum. *Reinforcement learning with selective perception and hidden state*. PhD thesis, University of Rochester, 1996.
- [6] Satinder Singh, Tommi Jaakkola, and Michael I. Jordan. Learning without state-estimation in partially observable Markovian decision processes. In *Proceedings of the International Conference on Machine Learning*. 1994.