On the Expressivity of Markov Reward (Extended Abstract)

David Abel[1] , **Will Dabney**[1] , **Anna Harutyunyan**[1] , **Mark K. Ho**[2] , **Michael L. Littman**[3] , **Doina Precup**[1] , **Satinder Singh**[1]

[1]DeepMind

[2]Princeton University

[3]Brown University

{dmabel, wdabney, harutyunyan, doinap, baveja}@deepmind.com, mho@princeton.edu, mlittman@cs.brown.edu,

Abstract

Reward is the driving force for reinforcement-learning agents. We here set out to understand the expressivity of Markov reward as a way to capture tasks that we would want an agent to perform. We frame this study around three new abstract notions of "task": (1) a set of acceptable behaviors, (2) a partial ordering over behaviors, or (3) a partial ordering over trajectories. Our main results prove that while reward can express many of these tasks, there exist instances of each task type that no Markov reward function can capture. We then provide a set of polynomial-time algorithms that construct a Markov reward function that allows an agent to perform each task type, and correctly determine when no such reward function exists.

1 Introduction

At the heart of artificial intelligence (AI) is the study of *agents* that learn to explore, plan, communicate, and pursue goals. The reinforcement-learning (RL) problem puts the study of such agents front and center under the assumption that we focus on agents that learn to maximize *reward*. In this sense, reward plays a significant role as a general purpose signal in RL: For any desired behavior, task, or other characteristic of agency, there must exist a reward signal that can incentivize an agent to learn to realize these desires. The expressivity of reward is taken as a backdrop assumption that frames RL, sometimes called the reward hypothesis [Littman, 2017; Christian, 2021]: "...all of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received scalar signal (reward)" [Sutton, 2004]. We here provide an extended abstract[1] that establishes first steps toward a systematic study of the reward hypothesis by examining the expressivity of reward as a signal, proceeding in three steps.

An Account of "Task". As rewards encode tasks, goals, or desires, we first ask, "what *is* a task?". We frame our study around a thought experiment (Figure 1) involving the interactions between a designer, Alice, and a learning agent, Bob, drawing inspiration from Ackley and Littman [1992], Sorg [2011], and Singh *et al.* [2009]. In this thought experiment, we draw a distinction between how Alice thinks of a task, and the means by which Alice incentivizes Bob to pursue this task. This distinction allows us to analyze the expressivity of reward as an answer to the latter question, conditioned on how we answer the former. Concretely, we consider three kinds of task in the context of finite Markov Decision Processes (MDPs): A task is either (1) a set of acceptable behaviors (policies), (2) a partial ordering over behaviors, or (3) a partial ordering over trajectories. Given these three task types, we then examine the *expressivity* of reward.

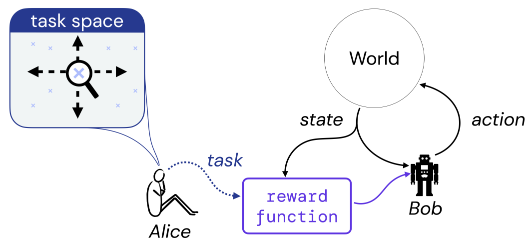


Figure 1: Alice, Bob, and the artifacts of task definition (blue) and task expression (purple).

Expressivity of Markov Reward. The core of our study asks whether there are tasks Alice would like to convey that admit no characterization in terms of a Markov reward function. Our emphasis on Markov reward functions, as opposed to arbitrary history-based reward functions, is motivated by several factors. First, disciplines such as computer science, psychology, biology, and economics typically rely on a notion of reward as a numerical proxy for the *immediate* worth of states of affairs (such as the fitness benefits of a phenotype). Given an appropriate way to describe states of affairs, Markov reward functions can represent immediate worth in an intuitive manner that also allows for reasoning about combinations, sequences, or re-occurrences of such states of affairs. Second, it is not clear that general history-based rewards are a reasonable target for learning as they suffer from the curse of dimensionality in the length of the history. Lastly, Markov reward functions are the standard in RL. A rigorous analysis of which tasks they can and cannot convey may pro-

[1]This is an extended abstract of a paper [Abel *et al.*, 2021] that won an outstanding paper award at NeurIPS 2022.

vide guidance into when it is necessary to draw on alternative formulations of a problem. In light of our focus on Markov rewards, we treat a reward function as accurately *expressing* a task just when the optimal value function it induces in an environment adheres to the constraints of the given task.

Main Results. We find that, for all three task types, there are environment–task pairs for which there is no Markov reward function that realizes the task (Theorem 3.1). In light of this finding, we design polynomial-time algorithms that can determine, for any given task and environment, whether a reward function exists in the environment that captures the task (Theorem 3.2). When such reward functions do exist, the algorithms also return one of them. In this sense, we take the perspective that these algorithms can be used to identify when the given representation of state might be insufficient for learning to solve the desired task. We take these findings to shed light on the nature of reward maximization as a principle, and highlight pathways for further investigation.

Background. RL defines the problem facing an agent that learns to improve its behavior over time by interacting with its environment. We make the typical assumption that the RL problem is well modeled by an agent interacting with a finite Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, R, T, \gamma, s_0)$. An MDP gives rise to deterministic behavioral policies, $\pi : \mathcal{S} \rightarrow \mathcal{A}$, and the value, $V^\pi : \mathcal{S} \rightarrow \mathbb{R}$, and action–value, $Q^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, functions that measure their quality. We will refer to a Controlled Markov Process (CMP) as an MDP without a reward function, which we denote E for environment. We assume that all reward functions are deterministic, and may be a function of either state, state-action pairs, or state-action-state triples, but *not* history. Henceforth, we simply use "reward function" to refer to a deterministic Markov reward function for brevity, but note that more sophisticated settings beyond MDPs and deterministic Markov reward functions are important directions for future work. For more on MDPs or RL, see the books by Puterman [2014] and Sutton and Barto [2018] respectively. We also note that there is considerable relevant literature that focuses on understanding reward, and refer readers to Abel *et al.* [2021] for a full exposition of related work.

2 Three Task Types: SOAPs, POs, and TOs

Consider an onlooker, Alice, and a learning agent, Bob, engaged in the interaction pictured in Figure 1. Suppose that Alice has a particular task in mind that she would like Bob to learn to solve, and that Alice constructs a reward function to incentivize Bob to pursue this task. Here, Alice is playing the role of "all of what we mean by goals and purposes" for Bob to pursue, with Bob playing the role of the standard reward-maximizing RL agent. That is, we suppose Alice might think of one of three kinds of task: 1) A set of acceptable policies, 2) A partial ordering over policies, or 3) A partial ordering over trajectories. We adopt these three as they can capture many kinds of task while also allowing a great deal of flexibility in the level of detail of the specification.

Set Of Acceptable Policies. A classical view of the equivalence of two reward functions is based on the optimal policies they induce. For instance, Ng *et al.* [1999] develop potential-based reward shaping by inspecting which shaped reward signals will ensure that the optimal policy is unchanged. Extrapolating, it is natural to say that for any environment E, two reward functions are equivalent if the optimal policies they induce in E are the same. In this way, a task is viewed as a choice of optimal policy. However, this notion of task fails to allow for the specification of the quality of other behaviors. For this reason, we generalize task-as-optimal-policy to a *set of acceptable policies*, defined as follows.

Definition 2.1. *A set of acceptable policies (SOAP) is a non-empty subset of the deterministic policies, $\Pi_G \subseteq \Pi$, with Π the set of all mappings from $\mathcal{S}$ to $\mathcal{A}$ for a given E.*

With one task type defined, it is important to address what it means for a reward function to properly *realize* or *express* a task in a given environment. We offer the following account.

Definition 2.2. *A reward function is said to realize a task $\mathscr{T}$ in an environment E just when the start-state value (or trajectory-return) induced by the reward function exactly adheres to the constraints of $\mathscr{T}$.*

Precise conditions for the realization of each task type are provided alongside each task definition, with a summary presented in column four of Table 1. For SOAPs, we take the start-state value $V^\pi(s_0)$ to be the mechanism by which a reward function realizes a SOAP. That is, for a given E and Π_G, a reward function R is said to *realize* Π_G in E when the start-state value function is optimal for all good policies, and strictly higher than the start-state value of all other policies. It is clear that SOAP strictly generalizes a task in terms of a choice of optimal policy, as captured by the SOAP $\Pi_G = \{\pi^*\}$.

We note that there are two natural ways for a reward function to realize a SOAP: First, each $\pi_g \in \Pi_G$ has *optimal* start-state value and all other policies are sub-optimal. We call this type *equal*-SOAP, or just SOAP for brevity. Alternatively, we might only require that the acceptable policies are each *near*-optimal, but are allowed to differ in start-state value so long as they are all better than *every* bad policy $\pi_b \in \Pi_B$. That is, in this second kind, there exists an $\epsilon \geq 0$ such that every $\pi_g \in \Pi_G$ is ϵ-optimal in start-state value, $V^*(s_0) - V^{\pi_g}(s_0) \leq \epsilon$, while all other policies are worse. We call this second realization condition *range*-SOAP. We note that the range realization generalizes the equal one: Every equal-SOAP is a range-SOAP (by letting $\epsilon = 0$). However, there exist range-SOAPs that are expressible by Markov rewards that are *not* realizable as an equal-SOAP. We illustrate this fact with the following proposition. All proofs are presented in the appendix of the full version of the paper.

Proposition 2.1. *There exists a CMP, E, and choice of Π_G such that Π_G can be realized under the range-SOAP criterion, but cannot be realized under the equal-SOAP criterion.*

One such CMP is pictured in Figure 2b. Consider the SOAP $\Pi_G = \{\pi_{11}, \pi_{12}, \pi_{21}\}$, where π_{21} denotes the policy $\{s_0 \mapsto a_2, s_1 \mapsto a_1\}$: Under the equal-SOAP criterion, if each of these three policies are made optimal, any reward function will *also* make π_{22} (the only bad policy) optimal as well. In contrast, for the range criterion, we can choose a

Name	*Notation*	*Generalizes*	*Constraints Induced by* $\mathscr{T}$
SOAP	Π_G	task-as-π^*	equal: $V^{\pi_g}(s_0) = V^{\pi_{g'}}(s_0) > V^{\pi_b}(s_0), \forall_{\pi_g, \pi_{g'} \in \Pi_G, \pi_b \in \Pi_B}$ range: $V^{\pi_g}(s_0) > V^{\pi_b}(s_0), \forall_{\pi_g \in \Pi_G, \pi_b \in \Pi_B}$
PO	L_Π	SOAP	$(\pi_1 \oplus \pi_2) \in L_\Pi \implies V^{\pi_1}(s_0) \oplus V^{\pi_2}(s_0)$
TO	$L_{\tau,N}$	task-as-goal	$(\tau_1 \oplus \tau_2) \in L_{\tau,N} \implies G(\tau_1; s_0) \oplus G(\tau_2; s_0)$

Table 1: A summary of the three proposed task types. We further list the constraints that determine whether a reward function *realizes* each task type in an MDP, where we take $\oplus$ to be one of '$<$', '$>$', or '$=$', and G is the discounted return of the trajectory.

reward function that assigns lower rewards to a_2 than a_1 in both states. In general, we take the equal-SOAP realization as canonical, as it is naturally subsumed by our next task type.

Partial Ordering on Policies. Next, we suppose that Alice chooses a *partial ordering* on the deterministic policy space. That is, Alice might identify some great policies, some good, and some bad policies to strictly avoid, and remain indifferent to the rest. POs strictly generalize equal-SOAPs, as any such SOAP is a special choice of PO with only two equivalence classes. We offer the following definition of a PO.

Definition 2.3. *A policy order (PO) of the deterministic policies* Π *is a partial order, denoted* L_Π.

As with SOAPs, we take the start-state value $V^\pi(s_0)$ induced by a reward function R as the mechanism by which policies are ordered. That is, given E and L_Π, we say that a reward function R *realizes* L_Π in E if and only if the resulting MDP, $M = (E, R)$, produces a start-state value function that orders Π according to L_Π.

Partial Ordering on Trajectories. A natural generalization of goal specification enriches a notion of task to include the details of how a goal is satisfied—that is, for Alice to relay some preference over trajectory space [Wilson *et al.*, 2012], as is done in preference-based RL [Wirth *et al.*, 2017]. Concretely, we suppose Alice specifies a partial ordering on length-N trajectories of (s, a) pairs, defined as follows.

Definition 2.4. *A trajectory ordering (TO) of length* $N \in \mathbb{N}$ *is a partial ordering* $L_{\tau,N}$*, with each trajectory* τ *consisting of* N *state–action pairs,* $\{(s_0, a_0), \ldots, (a_{N-1}, s_{N-1})\}$*, with* s_0 *the start state.*

As with PO, we say that a reward function realizes a trajectory ordering $L_{\tau,N}$ if the ordering determined by each trajectory's cumulative discounted N-step return from s_0, denoted $G(\tau; s_0)$, matches that of the given $L_{\tau,N}$. We note that trajectory orderings can generalize goal-based tasks at the expense of a larger specification. For instance, a TO can convey the task, "Safely reach the goal in less than thirty steps, or just get to the subgoal in less than twenty steps."

Recap. We propose to assess the expressivity of reward by first restricting attention to SOAPs, POs, or TOs, as summarized by Table 1. We say that a task $\mathscr{T}$ is *realized* in an environment E under reward function R if the start-state value function (or return) produced by R imposes the constraints specified by $\mathscr{T}$, and are interested in whether reward can always realize a given task in any choice of E.

3 Results

With our definitions and objectives in place, we now present our main results.

3.1 Expressing SOAPs, POs, and TOs

We first ask whether Markov reward can always realize a given SOAP, PO, or TO, for an arbitrary E. Our first result states that the answer is "no".

Theorem 3.1. *For each task type, there exist* $(E, \mathscr{T})$ *pairs for which no Markow reward function realizes* $\mathscr{T}$ *in* E.

Thus, Markov reward is incapable of capturing certain tasks. What tasks are they, precisely? Intuitively, inexpressible tasks involve policies or trajectories that are correlated in *value* in an MDP. That is, if two policies are nearly identical in behavior, it is unlikely that reward can capture the PO that places them at opposite ends of the ordering. A simple example is the "always move the same direction" task in a grid world, with state defined as an (x, y) pair. The SOAP $\Pi_G = \{\pi_\leftarrow, \pi_\uparrow, \pi_\rightarrow, \pi_\downarrow\}$ conveys this task, but no Markov reward function can make these policies strictly higher in value than all others.

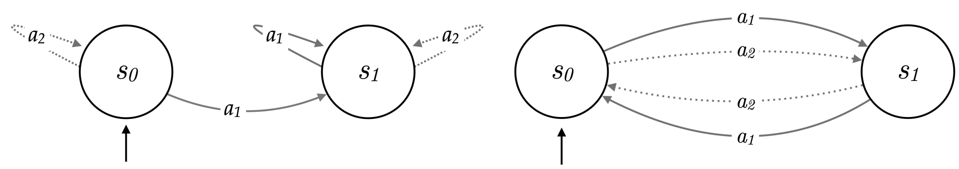
Inexpressible SOAPs. Observe the two CMPs pictured in Figure 2a and Figure 2b, depicting two kinds of inexpressible SOAPs. In the top figure, we consider the SOAP $\Pi_G = \{\pi_{21}\}$, containing only the policy that executes a_2 in the left state (s_0), and a_1 in the right (s_1). This SOAP is inexpressible through reward, but only because reward cannot distinguish the start-state value of π_{21} and π_{22} since the policies differ only in an unreachable state. In the bottom figure, we find a more interesting case: The chosen SOAP is similar to the XOR function, $\Pi_G = \{\pi_{12}, \pi_{21}\}$. Here, the task requires that the agent choose each action in exactly one state. However, there cannot exist a reward function that makes *only* these policies optimal, as by consequence, both policies π_{11} and π_{22} *must* be optimal as well.

3.2 Constructive Algorithms: Task to Reward

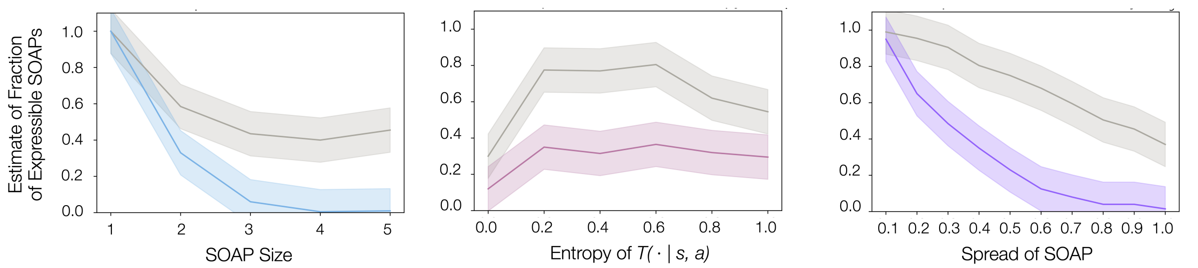
We now analyze how to determine whether an appropriate reward function can be constructed for any $(E, \mathscr{T})$ pair. We pose a general form of the reward-design problem [Mataric, 1994; Sorg *et al.*, 2010; Dewey, 2014] as follows.

Definition 3.1. *The* REWARDDESIGN *problem is:* ***Given*** $E = (\mathcal{S}, \mathcal{A}, T, \gamma, s_0)$*, and a* $\mathscr{T}$*,* ***output*** *a reward function* R_{alice} *that ensures* $\mathscr{T}$ *is realized in* $M = (E, R_{alice})$.

Indeed, for all three task types, there is an efficient algorithm for solving the reward-design problem.



(a) Steady State Case (b) Entailment Case



(c) Vary SOAP Size (d) Vary Entropy of T (e) Vary the Spread of Π_G

Figure 2: (Top Row) Two CMPs in which there is a SOAP that is not expressible under any Markov reward function. On the left, $\Pi_G = \{\pi_{21}\}$ is not realizable, as π_{21} can not be made better than π_{22} because s_1 is never reached. On the right, the XOR-like-SOAP, $\Pi_G = \{\pi_{12}, \pi_{21}\}$ is not realizable: To make these two policies optimal, it is entailed that π_{22} and π_{11} must be optimal, too. (Bottom Row) The approximate fraction of SOAPs that are expressible by reward in CMPs with a handful of states and actions, with 95% confidence intervals. In each plot, we vary a different parameter of the environment or task to illustrate how this change impacts the expressivity of reward, showing both equal (color) and range (grey) realization of SOAP.

Theorem 3.2. *The* REWARDDESIGN *problem can be solved in polynomial time, for any finite E, and any $\mathscr{T}$, so long as reward functions with infinitely many outputs are considered.*

Therefore, for *any* choice of E and a SOAP, PO, or TO, we can find a reward function that perfectly realizes the task in the given environment, if such a reward function exists. Each of the three algorithms are based on forming a linear program that matches the constraints of the given task type, which is why reward functions with infinitely many outputs are required.

3.3 Experiments

Lastly, we conduct an experiment to shed further light on the analysis. Our focus is on SOAPs, though we anticipate the insights extend to POs and TOs with little complication.

SOAP Expressivity. Concretely, we estimate the fraction of SOAPs that are expressible by Markov reward in small CMPs as we vary expects of the environment or task. For each data point, we sample 200 random SOAPs and run the algorithm mentioned in Theorem 3.2 to determine whether each SOAP is realizable in the given CMP. We ask this question for both the equal (color) and range (grey) SOAP realization. We inspect SOAP expressivity as we vary three different characteristics of E or Π_G: The number of good policies in each SOAP, the Shannon entropy of T at each (s, a) pair, and the "spread" of each SOAP. The spread approximates average edit distance among policies in Π_G determined by randomly permuting actions of a reference policy by a coin weighted according to the value on the x-axis. We use the same set of CMPs for each environment up to any deviations explicitly made by the varied parameter (such as entropy). Unless otherwise stated, each CMP has four states and three actions, with a fixed but randomly-chosen transition function.

Results are presented in Figure 2. We find that our theory is borne out in a number of ways. First, as Theorem 3.1 suggests, we observe that SOAP expressivity is strictly less than one in nearly all cases. This is evidence that inexpressible tasks are not only found in manufactured corner cases, but rather that expressivity is a spectrum. We further observe—as predicted by Proposition 2.1—separation between the expressivity of range-SOAP (grey) vs. equal-SOAP (color); there are many cases where we can find a reward function that makes the good policies *near*-optimal and better than the bad, but cannot make those good policies all *exactly*-optimal. Additionally, several trends emerge as we vary the parameter of environment or task, though we note that such trends are likely specific to the choice of CMP and may not hold in general. Perhaps the most striking trend is in Figure 2e, which shows a decrease in expressivity as the SOAPs become more *spread-out*.

4 Conclusion

We here examine the expressivity of Markov reward, framed around three accounts of task. Our main results show that there exist choices of task and environment in which Markov reward cannot express the chosen task, but there are efficient algorithms that decide whether a task is expressible *and* construct a realizing reward function when it exists. We take these to be first steps toward understanding the full scope of the reward hypothesis, and hope this work provides new conceptual perspectives on reward and its place in RL and AI.

Acknowledgments

The authors would like to thank André Barreto, Diana Borsa, Michael Bowling, Wilka Carvalho, Brian Christian, Jess Hamrick, Steven Hansen, Zac Kenton, Ramana Kumar, Katrina McKinney, Rémi Munos, Brendan O'Donoghue, Matt Overlan, Mark Rowland, Hado van Hasselt, and Ben Van Roy for helpful discussions. We would also like to thank the anonymous NeurIPS reviewers for their thoughtful feedback.

References

[Abel *et al.*, 2021] 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.

[Ackley and Littman, 1992] David Ackley and Michael L. Littman. Interactions between learning and evolution. *Artificial Life II*, 1992.

[Christian, 2021] Brian Christian. *The Alignment Problem: Machine Learning and Human Values*, pages 130–131. Atlantic Books, 2021.

[Dewey, 2014] Daniel Dewey. Reinforcement learning and the reward engineering principle. In *Proceedings of the AAAI Spring Symposium Series*, 2014.

[Littman, 2017] Michael L. Littman. The reward hypothesis. https://www.coursera.org/lecture/fundamentals-of-reinforcement-learning/michael-littman-the-reward-hypothesis-q6x0e, 2017.

[Mataric, 1994] Maja J. Mataric. Reward functions for accelerated learning. In *Proceedings of the International Conference on Machine Learning*, 1994.

[Ng *et al.*, 1999] Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the International Conference on Machine Learning*, 1999.

[Puterman, 2014] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[Singh *et al.*, 2009] 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*, 2009.

[Sorg *et al.*, 2010] Jonathan Sorg, Richard L. Lewis, and Satinder Singh. Reward design via online gradient ascent. *Advances in Neural Information Processing Systems*, 2010.

[Sorg, 2011] Jonathan Sorg. *The Optimal Reward Problem: Designing Effective Reward for Bounded Agents*. PhD thesis, University of Michigan, 2011.

[Sutton and Barto, 2018] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[Sutton, 2004] Richard S. Sutton. The reward hypothesis. http://incompleteideas.net/rlai.cs.ualberta.ca/RLAI/rewardhypothesis.html, 2004.

[Wilson *et al.*, 2012] Aaron Wilson, Alan Fern, and Prasad Tadepalli. A Bayesian approach for policy learning from trajectory preference queries. In *Advances in Neural Information Processing Systems*, 2012.

[Wirth *et al.*, 2017] Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *The Journal of Machine Learning Research*, 18(1):4945–4990, 2017.