Task Specification in Continual Reinforcement Learning

Jonathan Colaço Carr, Supervised by Prakash Panangaden and Doina Precup

December 12th, 2022

Abstract

Abstract. As reinforcement learning agents tackle increasingly complex problems, it becomes more important to ensure that the reward signals given to a learning agent convey the correct preferences. In their recent paper, Abel et. al [2] address this by testing the ability of reward signals to express preferences in Markovian reinforcement learning problems. In this report, we extend Abel et. al's methodology to general environments and verify the expressiveness of external and internal reward signals in this setting. We show that in almost all cases, an external reward signal can express a given task. Internal reward signals can be made equally as expressive, provided that the agent can represent all the preferences in its internal state. However, this is not a necessary condition for internal reward expressiveness, as we illustrate with a worked example.

1 Introduction

The goal of this report to determine whether or not scalar reward signals can convey a designer's preferences to reinforcement learning agents. At their best, rewards can constructed to completely align with human preferences [6, 7, 13, 15]. At their worst, mis-specified rewards can cause misalignment between human preferences and optimal agent behaviours [3, 17]. In their work, Abel et. al [2] develop a framework to study 'reward expressiveness' - whether or not a reward will properly express designer preferences.

A key assumption in Abel et. al's work is that the agent's learning environment is a controlled Markov process. Our first goal is to extend their analysis to more general learning environments where no assumption is made on the transition dynamics. Secondly, we analyze two types of rewards: *external rewards* which are given to the agent by an external source, and *internal rewards* which are computed autonomously by the learning agent. The difference between external and internal reward expressiveness provides insight into the impact of function approximation and state abstraction in reward design.

Our Contributions.

1. In Section 3 we extend Abel et. al's notions of "tasks" and "reward expressiveness" from controlled Markov processes to general learning environments.
2. In Section 4 we prove theoretical results to determine when reward signals convey human preferences.

3. In Section 5, we interpret our reward expressiveness results for these general learning environments and discuss avenues for future work.

2 Preliminaries

There are three core concepts to our reinforcement learning problem: the learning environment, the agent, and the goals/rewards used to convey designer preferences. We follow the notation used in Dong et. al [5], but note that this learning environment has been studied elsewhere (e.g. [10, 8, 11, 4, 9, 12]).

The Learning Environment. The setting for our reinforcement learning agent is a learning environment $\mathcal{E} := (\mathcal{A}, \mathcal{O}, \rho)$ where $\mathcal{A}$ is a finite set of actions, $\mathcal{O}$ is a finite set of observations and ρ is a transition probability function mapping trajectory-action pairs to distributions over future observations.

We write $\rho(o_{t+1}|h_t, a_t) \equiv \mathbb{P}_{t+1}(o_{t+1}|h_t, a_t)$ to denote the probability of receiving observation o_{t+1} given the *history* $h_t := (o_0, a_0, \ldots, a_{t-1}, o_t)$ and action a_t. We also let

- $\text{len}(h)$ be the number of observations in history h.
- $\mathcal{H}_t := (\mathcal{O} \times \mathcal{A})^{t-1} \times \mathcal{O}$ be the set of all action-observation histories of length t,
- $\mathcal{H} := \bigcup_{t=1}^{\infty} \mathcal{H}_t$ be the set of all finite histories.
- Π be the set of all deterministic policies $\pi : \mathcal{H} \to \mathcal{A}$.

The Agent State. The agent state encodes all of the data available to the agent. As discussed in [5, 10], many popular agent designs (including DQN [14], MuZero [16], and MPO [1, 18]) are divided into three components:

$$\textbf{agent state } x_t = (\textbf{aleatoric state } s_t, \textbf{epistemic state } p_t, \textbf{algorithmic state } z_t).$$

1. The **aleatoric state** s_t describes the agent's current sitation in the environment. In the case where the environment is an MDP the observation and aleatoric state would both be equal to the state of the MDP. In non-Markovian environments, (eg. an Atari-playing DQN agent [14]), the aleatoric state might some function of the most recent observations.
2. The **epistemic state** p_t represents the knowledge the agent retains about its environment that is not represented in the current aleatoric state (e.g. the weights of an action-value function or a replay-buffer).
3. The **algorithmic state** z_t captures all other parts of the agent state that are not in the aleatoric and epistemic state (e.g. dummy variables or learning rates).

Each component of the agent state is updated separately at each timestep. The aleatoric state update given by a *compression function* $\varphi : \mathcal{H} \to \mathcal{S}$, where $\mathcal{S}$ denotes the set of all possible aleatoric states.

At each step of a learning process, the agent chooses an action according to the *agent policy* $\pi_{\text{agent}} : \mathcal{X} \to \mathcal{A}$, where $\mathcal{X}$ is the set of all possible agent states. While the set of agent policies is much smaller than Π, we hope to design the agent state that can be used to approximate the most desirable behaviour from Π.

Goals and Rewards. In most non-Markovian reinforcement learning problems [8, 9, 10, 11], rewards are assumed to be computed by an external source and given to the agent at each timestep. However, this type of consistent feedback is not always feasible for real-world agents. As in Dong et. al [5], we'd like to also consider reward signals that can be computed by a learning agent without external monitoring. We will thus analyze two types of reward signals:

1. external rewards $r_{\text{ex}} : \mathcal{H} \to \mathbb{R}$ which may depend on the agent's full history,
2. internal rewards $r_{\text{in}} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ which depend on the agent's current aleatoric state, current action, and next aleatoric state.

For both internal and external rewards, the goal of the agent is to maximize the cumulative expected reward. Writing a general reward function as

$$r(h_t) = \begin{cases} r_{\text{ex}}(h_t) & \text{or} \\ r_{\text{in}}(\varphi(h_{t-1}), a_{t-1}, \varphi(h_t)), \end{cases}$$

we define the *cumulative reward* for history $h \in \mathcal{H}$ as

$$G(h; r) := \sum_{t=1}^{\text{len}(h)} r(h_t),$$

where h_t denotes the first t action-observation pairs in h. For each $T \in \mathbb{N}$, we define the *value* of policy π to be

$$V(\pi, T; r) := \mathbb{E}_\pi[G(H_T; r)] = \mathbb{E}_\pi \left[\sum_{t=1}^{T} r(H_t) \right],$$

where the expectation is taken with respect to following policy π in the environment.

From a reinforcment learning agent's perspective, the best policies are those which maximize the long-term average value. The goal in the coming sections is to determine when our preferences are also optimal with respect to this cumulative reward.

3 Tasks and Expressiveness

In order to find "good" reward signals, we first provide two broad classes of designer preferences that we might want a reward to express.

3.1 Tasks

In the spirit of Abel et. al [2], we consider preferences over trajectories as well as preferences over policies. We refer to such preferences as 'tasks', and extend two of Abel et. al's three task definitions to the non-Markovian learning environment, deferring the third extension for future work.

Preference over trajectories. For an environment $\mathcal{E} = (\mathcal{A}, \mathcal{O}, \rho)$, let $\mathcal{H}_\mathcal{E} \subseteq \mathcal{H}$ be the set of all trajectories that occur with non-zero probability when following a policy which chooses actions uniformly at random.

Definition 3.1 (Task 1 - Trajectory Ordering)**.** Given an environment $\mathcal{E}$, a trajectory ordering (TO), denoted $(\mathcal{H}_\mathcal{E}, \preceq)$, is a preorder whose kernel partitions $\mathcal{H}_\mathcal{E}$ into finitely many equivalence classes.

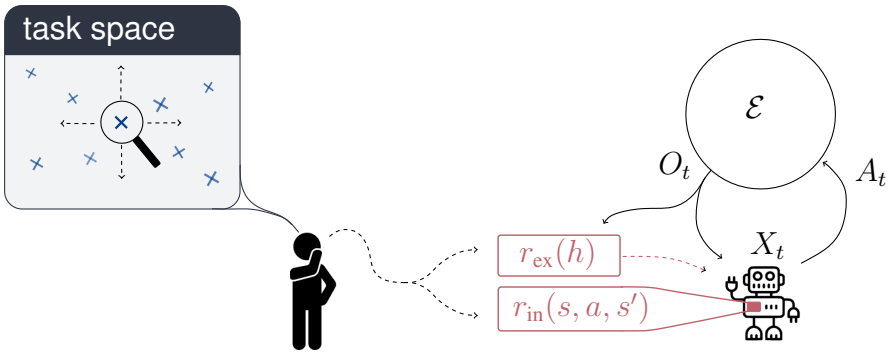

Figure 1: The relationship between designer preferences (tasks) and reward signals, adapted from Abel et. al. We want to determine whether or not our designer preferences (shown in blue) can be expressed by reward signals (shown in red) in a given learning environment.

Preferences over trajectories are used in Preference-Based RL to communicate preferences for how a task is completed [19]. For example, a trajectory ordering can convey the task, "Safely reach the goal in less than thirty steps, or just get to the subgoal in less than twenty steps".

Preferences over policies. Although we've defined policies as functions from $\mathcal{H}$ to $\mathcal{A}$, any policy π can only reach a set of *admissible histories*,

$$\text{Ad}(\pi) := \{h \in \mathcal{H} : \ \forall (h', a') \sqsubseteq h, \ \pi(h') = a'\}.$$

When assessing whether or not a policy is 'acceptable', we should only consider the quality of the trajectories that are attainable under π.

Definition 3.2 (Task 2 - Set of Acceptable Policies)**.** A set of acceptable policies (SOAP) is a non-empty subset $\Pi_G \subset \Pi$ such that

1. Good actions make good policies. I.e. for any $\pi \in \Pi$, if $\forall h \in \text{Ad}(\pi)$, $\exists \pi_G \in \Pi_G$ s.t. $h \in \text{Ad}(\pi_G)$, then $\pi \in \Pi_G$.
2. All bad policies choose at least one bad action. I.e. for any $\pi_B \in \Pi_G^{\text{C}}$, $\exists h \in \text{Ad}(\pi_B)$ such that $\forall \pi_G \in \Pi_G$, $h \notin \text{Ad}(\pi_G)$.

3.2 Reward Expressiveness

The goal of a reinforcement learning agent is to follow a policy which maximizes the cumulative expected reward. Ideally, we want the behaviour that is optimal with respect to the reward to be "optimal" with respect to the partial order induced by the task. In this case, we say that the reward is expressive.

Definition 3.3 (Reward Expressiveness)**.** Let an environment $\mathcal{E} = (\mathcal{A}, \mathcal{O}, \rho)$ be given.

- A reward function r expresses a trajectory ordering $(\mathcal{H}_\mathcal{E}, \preceq)$ in $\mathcal{E}$ iff $\forall h, h' \in \mathcal{H}_\mathcal{E}$,

$$(G(h; r) \leq G(h'; r)) \iff (h \preceq h').$$

- A reward function r expresses a set of acceptable policies $\Pi_G \subseteq \Pi$ in $\mathcal{E}$ iff $\exists T' \in \mathbb{N}$ such that $\forall T \geq T'$, $\forall \pi_{G_1}, \pi_{G_2} \in \Pi_G, \pi_B \in \Pi_G^{\text{C}}$,

$$V(\pi_{G_1}, T; r) = V(\pi_{G_2}, T; r) > V(\pi_B, T; r).$$

4 Reward Analysis

4.1 External Rewards

The following two propositions give necessary and sufficient conditions for finding expressive external rewards in any learning environment.

Proposition 4.1 (TOs are realized by external rewards)**.** *For any environment $\mathcal{E}$ and any trajectory ordering $(\mathcal{H}_\mathcal{E}, \preceq)$, there exists an external reward function that realizes the trajectory ordering in $\mathcal{E}$.*

Proof. Let $A_1, \ldots, A_K$ be the partitions of $\mathcal{H}_\mathcal{E}$ induced by the trajectory ordering. I.e. $\mathcal{H}_\mathcal{E} = \bigcup_{i=1}^K A_i$, $A_i \cap A_j = \emptyset$ if $i \neq j$, and $\forall 1 \leq i, j \leq K$,

$$i \leq j \iff (\forall h_i \in A_i, h_j \in A_j,\ h_i \preceq h_j)\,.$$

Additionally, we set $A_0 = \{()\}$, where $()$ is the 'start' history. Writing $\mathbf{1}_E$ for the indicator of the event E, we assign a score to each trajectory according to $\alpha_{\text{ex}} : \mathcal{H}_\mathcal{E} \to \{0, 1, \ldots, K\}$, where

$$\alpha_{\text{ex}}(h) = \sum_{k=0}^{K} k\mathbf{1}_{A_k}(h).$$

Setting $r_{\text{ex}}(h) := \alpha_{\text{ex}}(h_{\text{len}(h)}) - \alpha_{\text{ex}}(h_{\text{len}(h)-1})$ and $r_{\text{ex}}(()) = 0$ gives a telescoping series for the return:

$$\begin{aligned} G(h; r_{\text{ex}}) &= \mathbb{E}\left[\sum_{t=0}^{\text{len}(h_\mathcal{E})} r_{\text{ex}}(h_t)\right] \\ &= \alpha_{\text{ex}}(h_0) + (\alpha_{\text{ex}}(h_1) - \alpha_{\text{ex}}(h_0)) + \cdots \\ &= \alpha_{\text{ex}}(h) - 0 \\ &= \sum_{k=1}^{K} k\mathbf{1}_{A_k}(h). \end{aligned}$$

Thus $\forall h, h' \in \mathcal{H}_\mathcal{E}$,

$$\begin{aligned} (h \preceq h') &\iff \exists 1 \leq i \leq j \leq K :\ (h \in A_i \text{ and } h' \in A_j) \\ &\iff G(h; r_{\text{ex}}) \leq G(h'; r_{\text{ex}}). \end{aligned}$$

So r_{ex} expresses the trajectory ordering. □

To prove that SOAPs can be realized by external rewards, we re-write Π_G as P_2 and Π_G^{C} as P_1. For each $h \in \mathcal{H}$, we define the rank of as the order of the worst policy that follows h,

$$\text{rank}(h) := \min\{i :\ \exists \pi_i \in P_i :\ h \in D(\pi_i)\} \in \{1, 2\}.$$

The two consistency conditions in the definition of a SOAP is equivalent to saying that $\forall \pi_i \in P_i$,

$$\inf\{\text{rank}(h) :\ h \in D(\pi_i)\} = i.$$

Proposition 4.2 (SOAPs are realized by external rewards)**.** *For an environment $\mathcal{E} = (\mathcal{A}, \mathcal{O}, \rho)$, let ρ_π be the distribution induced by following policy π in $\mathcal{E}$. A SOAP is realized in $\mathcal{E}$ if and only if the following two criteria hold:*

1. *(Preferences occur in $\mathcal{E}$) For all $i = 1, 2$, $\pi_i \in P_i$,*

$$\inf\{\textit{rank}(h) : \ h \in \textit{supp}(\rho_{\pi_i})\} = i. \tag{$\star$}$$

2. *(All preferences are distinguishable before a finite time) Let τ_{π_i} be the smallest time for which $(\star)$ occurs. I.e.*

$$\tau_{\pi_i} = \inf\{t \in \mathbb{N} : \exists h_t \in \mathcal{H}_t \cap \textit{supp}(\rho_{\pi_i}) \textit{ s.t. } \textit{rank}(h_t) = i\}.$$

   Then $\sup_\Pi \tau_\pi < \infty$.

Proof. First assume that criterion (1) fails in $\mathcal{E}$. So we can find a policy $\pi_2 \in P_2$ and a policy $\pi_1 \in P_1$ such that

$$\text{supp}(\rho_{\pi_1}) = \text{supp}(\rho_{\pi_2}).$$

The policies are deterministic and have the same support. So they must have the same distribution over trajectories in $\mathcal{E}$ (given uniquely by the transition probabilities in ρ). In particular, for any reward function $r : \mathcal{H} \to \mathbb{R}$ and any $T \in \mathbb{N}$,

$$\mathbb{E}_{\pi_1}\left[\sum_{t=0}^{T} r(H_t)\right] = \mathbb{E}_{\pi_2}\left[\sum_{t=0}^{T} r(H_t)\right],$$

which violates the condition for SOAP expressivity.

Now, assume that criterion (2) fails. So criterion (1) holds only in the limit of $T \to \infty$. Then by definition of reward expressivity, the SOAP cannot be realized, since we cannot guarantee that every good policy distribution is different from every bad policy distribution before some finite time.

Next, assume that criteria (1) and (2) hold in $\mathcal{E}$. Define $r_{\text{ex}} : \mathcal{H} \to \mathbb{R}$ by

$$r_{\text{ex}}(h) = \begin{cases} 0 & h \in \bigcup_{\pi_2 \in P_2} \text{supp}(\rho_{\pi_2}) \\ -1 & \text{else.} \end{cases}$$

Then by construction, $\forall \pi_2 \in P_2, T \in \mathbb{N}$ we have that

$$\mathbb{E}_{\pi_2}\left[\sum_{t=0}^{T} r_{\text{ex}}(H_t)\right] = 0. \tag{$\dagger$}$$

By criterion (2), there is $T \in \mathbb{N}$ such that $\forall \pi_1 \in P_1$,

$$\inf\left\{\text{rank}(h) : \ h \in \text{supp}(\rho_{\pi_1}) \cap \left(\bigcup_{t=1}^{T} \mathcal{H}_t\right)\right\} = 1.$$

So there exists some $1 \leq l \leq T$, $h'_l \in \text{supp}(\rho_{\pi_1})$ such that $h'_l \notin \bigcup_{\pi_2 \in P_2} \text{supp}(\pi_2)$. Next let $p(h)$ denote the probability of a history that occurs under π_2, i.e.

$$p(h) = \rho(o_0) \cdot \rho(o_2|o_1, a_1) \cdot \cdots \cdot \rho(o_{\text{len}(h)}|h_{\text{len}(h-1)}, a_{\text{len}(h)}) \geq 0.$$

Since $p(h)r(h) \leq 0 \; \forall h \in \mathcal{H}$, we find that $\forall T \geq l$,

$$\mathbb{E}_{\pi_1}\left[\sum_{t=0}^{T} r_{\text{ex}}(H_t)\right] \leq \mathbb{E}_{\pi_1}\left[r_{\text{ex}}(H_l)\right] = \sum_{h_l \in \text{supp}(\rho_{\pi_i})} p(h_l) r_{\text{ex}}(h_l) \leq \rho(h_l') r_{\text{ex}}(h_l') < 0.$$

Since the expected sum of rewards is non-increasing, we find that $\forall T' \geq T, \pi_1 \in P_1$,

$$\mathbb{E}_{\pi_1}\left[\sum_{t=0}^{T} r_{\text{ex}}(H_t)\right] < 0.$$

This combined with $(\dagger)$ shows that r_{ex} expresses the SOAP. □

4.2 Internal Rewards

This following proposition shows that if $\mathcal{S}$ refines the trajectory ordering's partition, then we can find an expressive internal reward.

Proposition 4.3. *Let an environment $\mathcal{E}$, trajectory ordering $(\mathcal{H}_{\mathcal{E}}, \preceq)$, and compression $\varphi : \mathcal{H} \to \mathcal{S}$ be given. Let $A_1, \ldots, A_K$ be the partitions of the trajectory ordering's equivalence relation. If*

$$(\forall s \in \mathcal{S}),\ (\exists 1 \leq i \leq K) \text{ s.t. } \varphi^{-1}(s) \subseteq A_i,$$

then there is an internal reward $r_{in} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ which expresses the trajectory ordering in $\mathcal{E}$.

Corollary 4.1 (Minimal TO Compression)**.** *If an external reward realizes a trajectory ordering with K partitions in $\mathcal{E}$, there exists a compression function $\varphi : \mathcal{H} \to \{0, 1, \ldots, K\}$ and a reward function $r_{in} : \{0, 1, \ldots, K\}^2 \to \mathbb{R}$ which expresses the trajectory ordering.*

Proof. The construction of the internal reward is identical to the construction of the external reward in Proposition 4.1. Recall the scoring function introduced in the proof of Proposition 4.1,

$$\alpha_{\text{ex}}(h) = \sum_{k=0}^{K} k \mathbf{1}_{A_k}(h).$$

It is sufficient to check that α_{ex} can be identified with some function $\alpha_{\text{in}} : \mathcal{S} \to \{0, \ldots, K\}$. Then, setting $r_{\text{in}}(s, a, s') = \alpha_{\text{in}}(s') - \alpha_{\text{in}}(s)$ will create a telescoping series for $G(\cdot; r_{\text{in}})$ that is consistent with the preferences.

The assumption in this theorem is that $\mathcal{S}$ acts as a refinement of the trajectory ordering's partitions. So there is a partition $S_1, \ldots, S_K$ of $\mathcal{S}$ such that $\forall 1 \leq i \leq K$,

$$h \in A_i \iff \varphi(h) \in S_i. \tag{$\star$}$$

We define $\alpha_{\text{in}} : \mathcal{S} \to \{0, \ldots, K\}$ by

$$\alpha_{\text{in}}(s) = \sum_{k=0}^{K} k \mathbf{1}_{S_k}(s).$$

By $(\star)$, we know that

$$\alpha_{\text{ex}}(h) = \alpha_{\text{in}}(\varphi(h)) \quad \forall h \in \mathcal{H}.$$

Setting $r_{\text{in}}(s, a, s') := \alpha_{\text{in}}(s') - \alpha_{\text{in}}(s)$ will express the trajectory ordering, with the argument being identical to that of Proposition 4.1. □

Proposition 4.4. *Let a SOAP Π_G, environment $\mathcal{E}$, and compression $\varphi : \mathcal{H} \to \mathcal{S}$ be given. Suppose that criteria (1) and (2) from Proposition 4.2 hold. If*

$$(\forall s \in \mathcal{S}),\ (\forall h, h' \in \varphi^{-1}(s))\ \textit{rank}(h) = \textit{rank}(h'), \tag{1}$$

then there exists an internal reward which realizes Π_G in $\mathcal{E}$.

Corollary 4.2 (Minimal SOAP Compression)**.** *If an external reward realizes a SOAP Π_G in $\mathcal{E}$, there exists a compression function $\varphi : \mathcal{H} \to \{0, 1\}$ and a reward function $r_{in} : \{0, 1\} \to \mathbb{R}$ which expresses the trajectory ordering.*

Proof. We use the same notation as in the proof of Proposition 4.2, setting $P_2 = \Pi_G,\ P_1 = \Pi_G^{\text{C}}$, and

$$\text{rank}(h) = \min\{i :\ \exists \pi_i \in P_i \text{ s.t. } h \in D(\pi_i)\}.$$

By Eq. 1, the rank is a class property of $\varphi^{-1}(s)\ \forall s \in \mathcal{S}$. We let $\text{rank}(s) \in \{1, 2\}$ be the rank of the trajectories for which $\varphi(h) = s$,

$$\text{rank}(s) := \inf\{\text{rank}(h) :\ \varphi(h) = s\}.$$

Note that we could have used the supremum in this definition since we assume that all histories mapped to s have the same rank. We define an internal reward by

$$r_{\text{in}}(s, a, s') := \begin{cases} 0 & \text{rank}(s') = 2, \\ -1 & \text{else.} \end{cases}$$

By assumption, the criteria in Proposition 4.2 are satisfied. The exact same arguments as in Proposition 4.2 will show that this internal reward expresses the SOAP. □

Goal States. In order to ensure that we can express our rewards in an arbitrary environment, the conditions in the previous proposition are quite strict. They esssentially require the optimal policy in Π to representable by the set of policies on *aleatoric states*, $\Pi_{\mathcal{S}} = \{\pi : \mathcal{S} \to \mathcal{A}\}$.

Alternatively, if we have enough prior knowledge about the environment's transition dynamics, we can design an expressive internal reward by clustering the patterns of good policies into 'goal states' in the aleatoric state space. Then, if we can guarantee that the distribution over goal states is the same for all acceptable policies and strictly smaller for all bad policies, an internal reward function will realize the task.

Definition 4.1 (Goal State)**.** For an environment $\mathcal{E} = (\mathcal{A}, \mathcal{O}, \rho)$, SOAP Π_G, and compression function $\varphi : \mathcal{H} \to \mathcal{S}$, a goal state is a state $s^\star \in \mathcal{S}$ such that

$$(\exists T' \in \mathbb{N}) \text{ s.t. } (\forall T \geq T',\ \forall \pi_{G_1}, \pi_{G_2} \in \Pi_G, \pi_B \in \Pi_G^{\text{C}}), \quad \rho^{\mathcal{S}}_{\pi_{G_1}}(s^\star, T) = \rho^{\mathcal{S}}_{\pi_{G_2}}(s^\star, T) > \rho^{\mathcal{S}}_{\pi_B}(s^\star),$$

where $\rho^{\mathcal{S}}_\pi(s, T) = \sum_{t=0}^{T} \sum_{\mathcal{H}_t \cap \varphi^{-1}(s)} \rho_\pi(h)$ is the probability of visiting aleatoric state s during the first T time-steps (with $\rho_\pi(h)$ being the probability of observing trajectory h under policy π in $\mathcal{E}$).

Example 4.1 (Grocery Order). Suppose you would like to teach a reinforcement learning agent to fulfill a grocery order and bring it to the checkout (as shown in Figure 2). At the checkout, the clerk verifies if the order is complete. If there are missing items, the robot is sent back into the aisles. Otherwise, the robot passes to the pickup area, where it waits for the customer to

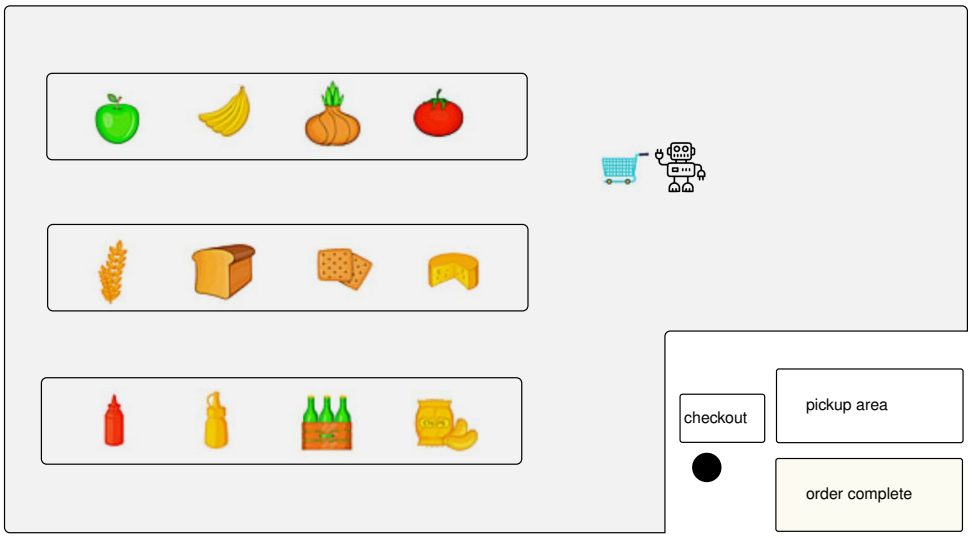

Figure 2: Grocery store setup for Example 4.1.

pick up their groceries. When the groceries are unloaded, the robot moves to the completed orders section and stays there.

We let $\mathcal{O}$ be the set of possible sensor readings from the robot, $\mathcal{A}$ be the set of actions for its joints, and let Π_G be the policy set of policies which completes orders as quickly as possible. Regardless of the complexity of the set of acceptable policies, if we set

$$\mathcal{S} = \{\texttt{in order complete zone}, \texttt{otherwise}\} \equiv \{s^\star, s^-\}.$$

We can express this task by assigning $r_{\text{in}}(s^\star, \cdot, \cdot) = 0$ and $r_{\text{in}}(s^-, \cdot, \cdot) = -1$.

Clearly in this case, the optimal policy is not representable by any policy that depends only on the current aleatoric state (in particular, the conditions in Proposition 4.4 are not met). If the optimal policy is representable by the *agent state*, then the robot can learn to behave optimally. However, if the agent is *never* able to reach the goal state, then it is not able to learn *any* behaviour in the store.

Remark 4.1. This example points to a potential tradeoff between the aleatoric state and agent state capacity. When the full agent state can represent a wider range of behaviours, the aleatoric state can be made smaller. However, if the agent does not reach goal states, it might fail to learn any sort of meaningful behaviour.

5 Discussion

In this report, we've taken Abel et. al's work in two new directions:

1. **Reward expressiveness beyond Markovian environments.** We have extended Abel et. al's theory of reward expressiveness to the most general reinforcement learning environments. While Abel et. al considers the expressiveness of a single type or reward, we've verified the expressiveness of two types of reward functions: external rewards and internal rewards. Propositions 4.1 and 4.2 show that external rewards can express tasks in almost

any environment. Propositions 4.3 and 4.4 provide sufficient, albeit strict, conditions for tasks to be expressed by internal rewards in environments where the transition dynamics are unknown.

2. **Representations and rewards.** While Propositions 4.3 and 4.4 show that 'perfect' aleatoric state spaces lead to expressive internal rewards, Example 4.1 shows that this is not a necessary condition. Even the most rudimentary aleatoric state space will suffice when we can identify goal states - patterns that "naturally" occur in the environment under good policies. However, compressing the aleatoric space makes it more challenging to learn directly from the aleatoric state.

In future work, we'd like to continue to explore the following topics:

1. **Exploring tradeoffs between aleatoric and agent states.** Example 4.1 seems to suggest that there is a balance to strike between finding 'minimal' aleatoric states while ensuring the agent receives enough reward to learn the desired behaviour. It may be interesting to experiment to see how the size of the smallest aleatoric state space changes with the size of the agent state.

2. **Adaptive tasks.** All of our task definitions require the designer preferences to be defined before the agent starts to learn. We hope to study more the realistic case where preferences evolve over time. The main challenge here is to ensure that the tasks are well-defined, and that we have a sensible notion for what it means for a reward to be 'expressiveness', without knowing the full task.

6 Conclusion

Throughout the evolution of reinforcement learning systems, reward continues to play a central role in communicating designer preferences. Careful reward design is critical in the pursuit of reliable, real-world reinforcement learning agents.