Chapter 5
Challenges for Reinforcement Learning

Theories destroy facts.

Peter Medawar [221]

Reinforcement learning was historically established as a descriptive model of learning in animals [234], [324], [32], [279] then recast as a framework for optimal control [331]. The definition of RL has been progressively expanded to be so broad and unconstrained [143] that virtually any form of learning can ultimately be described as RL. This has led some to consider RL as a plausible candidate for unifying cognition at large, hence subsuming general intelligence. In this chapter and the next, we identify fundamental issues that challenge this view. In summary:

- The notion of a *fixed a priori reward function* acts counter to open-endedness: assumptions of stationarity and the validity of 'once and for all' behavioral specifications are simply not adequate for open-ended behavior in the real world.
- For both efficiency and safety reasons, the notion of *reward sampling* prevents RL from being performed in the real world.
- The notion of *policy* conflates knowledge (world models) and motivation (goals)[1] via the direct mapping from states to actions: this opposes continual world modeling and plan composition.

To its credit, function approximation using 'deep' neural architectures has advanced the capacity of RL substantially beyond the capabilities of the original tabular setting. However, as described in Chap. 4, deep function approximation has serious limitations.

[1] Even though model-based RL leverages world models during learning, its end-product still remains a policy.

J. Swan et al., *The Road to General Intelligence*,
Studies in Computational Intelligence 1049,
https://doi.org/10.1007/978-3-031-08020-3_5

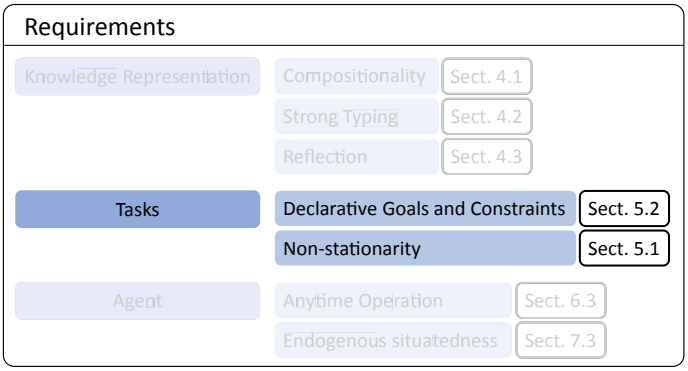

Fig. 5.1 In this chapter we argue that the centrality in RL of a priori fixed reward functions oppose key requirements for general intelligence

Claim 4

Reinforcement learning techniques which use deep neural networks for function approximation will inherit the issues stated in Chap. 4.

As per Sect. 2.1, we center our requirements around the continual adaptation of a world model to achieve a variety of arbitrary goals. We also stipulate that (1) learning without forgetting should be relatively cheap and (2) safety must be certifiable.

5.1 A Priori Reward Specification

Arguably one the most central concept in RL is the *value function* which usually express the value of a state or state-action pair in terms of the expected returns under some policy starting from that state—recall from Section 3.1 that a policy is simply a distribution of actions conditioned on states. The iteration of value functions toward a fixed point makes sense because the reward in RL is specified beforehand and is assumed to remain stationary. If we entirely commit to this strategy, then the only way to become more general is to widen stationarity over broader notions of tasks and environments. Learning is then extended to maximize reward over an entire distribution of MDPs [293], [305], [277] which share a state and action space whereas the reward function is selected from a distribution. These modifications, which are examples of *meta*-RL, purportedly allow for an agent to learn how to succeed over a variety of tasks, and would seem to be the most developed route in fully stationary RL towards general intelligence.

Notwithstanding those extensions, the value function is still an ideal place to scrutinize the foundations of RL. The assumptions of RL make it suitable to apply to problems where there is a naturally occurring quantitative objective to be maximized

and there are 'operationalized' actions [11], as exemplified by e.g. video games, where even random behavior has non-negligible correlation with success. Even then, the RL engineer may have to do significant work to make a dense, shaped reward from a *sparse* reward, in order to obtain a sufficiently rich reward signal. This is not all that different from having to invest effort into designing the architecture of neural networks to accommodate inductive biases and ease nonconvex optimization. In the context of general intelligence, we therefore present the following claim:

Claim 5

The notion of a *fixed a priori reward function* acts counter to open-endedness: assumptions of stationarity and sufficient human foresight cannot be guaranteed to hold in the domains where intelligent agents are expected to operate.

We argue that since open-ended learning is a necessary trait of general intelligence, there cannot be any true fixed-point optimality condition to achieve. It might be possible that certain primitives can be hard-coded as policies over state observations, such as for basic locomotion. However, it is at the very least counter-intuitive that the higher-order cognition for achieving a variety of goals might be expressible as an optimization over a very wide distribution. Instead, we must dismantle the assumption that rewards are stationary and that states (or state-action pairs) accordingly have a pre-defined value.

Indeed, the literature on prospects for continual learning in RL highlights this same point [171]. In this context, general intelligence may be approached as mastering a sequence of MDPs rather than a distribution. Within each one, there is still an assumption of a priori optimality, and this may perhaps make sense for individual tasks which are suitably framed as optimization problems. The more general scenario, however, involves tasks which the user himself may wish to modify/extend/retract as time goes on. Naturally, a framework for task specification should support the modification of a specification without requiring that it be treated as an entirely new one. This could be partially enabled by having reward functions which are compositional, which is explored to a degree in literature on general value functions and scalarized multi-objective RL [350].

This challenge of modifying a priori reward specifications takes on even greater importance in AI safety literature through the core issues of *misspecification* and *alignment* [4], [82], [128], [292]. Even if we assume that humans have a fixed reward function (or distribution or sequence thereof), the challenge of building a powerful general intelligence aligned with that becomes extremely daunting. Inverse reinforcement learning [1] attempts to learn a (parametric) reward function for situations where an analytic description is a poor fit for human concerns. As with the usual a priori case, this reward function is taken to be essentially stationary and so the same concerns arise [129], [130], [195]. Ultimately, this may prove futile, since humans exhibit preferences which do not conform to the Von Neumann-Morgenstern axioms [353] of rationality, and hence cannot be said to possess a stationary utility function [80]. In any case, it is clearly desirable to avoid a formulation of general intelligence

which explicitly requires a *proxy* for desired states and would consequently be vulnerable to Goodhart's Law: "When a measure becomes a target, it ceases to be a good measure." [122].

We believe that RL is given credit for much of the work that is actually attributable to engineers through their intelligent design of reward functions and model architectures. More fundamentally, we argue that a framework for general intelligence is better founded if it does not assume that rewards (or more generally, objectives) can ever be fully known, even by the stakeholders. While some things can be optimized to a fixed point, a generally intelligent agent intended to replace human labor would not benefit from such essentially stationary notions. For example, domestic robots that are expected to work well around growing children must be able to accommodate ever-changing constraints and preferences from the family as its youngest members grow up. In fact, it becomes increasingly evident that, in the context of general intelligence, the very concept of 'reward function' ends up creating more problems than it solves. This notion is explored further in Sect. 6.1.

5.2 Sampling: Safety and Efficiency

In Sect. 4.3, we emphasized that *reflection* is a key requirement for knowledge representation. As with our preceding discussion of deep supervised learning, we have identified that feedback mechanisms are a key bottleneck in RL. In the former, we focused on the limitations of iteratively updating neural networks through gradient descent. Here, we discuss an analogous issue which is fundamental to RL, irrespective of reliance on function approximation using deep architectures. In essence, in the MDP formalism the agent receives a reward at each timestep as a result of its state and chosen action, and hence, goals (taken to be regions of the state space) are only expressed via sampling. Note that RL based on *model-predictive control* (MPC) often takes a reward function in closed form, visible to the agent, though this still leaves open the problem of finding its optima.

For arguably the majority of human activities we wish to automate, we take advantage of the fact that we can interpret our desired goals prior to the need to sample any information from the environment. Granted, this requires sufficient world knowledge to make sense of a goal description, but in such a case, this approach confers multiple benefits. We explore ways to make use of this perspective in Sect. 6.1. Here, we highlight the undesirable side effects of a sample-based approach to goal descriptions, of which RL reward feedback is an example. The key point is given in the following claim:

Claim 6

The reliance on sampled rewards means that it is inherently unsafe to train RL systems in the real world. Even if we approached this with very high fidelity simulations, RL would need to exhibit good sample efficiency, which it currently does not.

For humans, a reward function can in principle be made *intensional*: a 'white-box' function of state and action. In contrast, the RL procedure incrementally constructs an *extensional* description of the reward, and starts out entirely blind in its sampling. This raises serious concerns for the safety of systems developed in this way. Recent work has emerged which tries to specifically address safe exploration in deep RL [282], [24], [83], [181], but the fundamental issues of using deep neural networks for estimation and having to sample during training remain. Of course, one natural response would be that we can use resettable simulations to safely train agents until we certify them ready to act in the real world. The use of simulation may appear to be a panacea, but this strategy simply delays confronting the issue, analogous to our observations on domain randomization in Sect. 4.2. The primary challenge is the fidelity of the simulation. As a minimum, there is the issue of verisimilitude in the behavior of physical objects, with well-known precision issues [326]. More daunting is the credible simulation of other agents.

Moreover, the *sample efficiency* of deep RL agents is a pressing concern. Sample efficiency is a performance measure which is inversely proportional to the number of samples that the system needs to obtain from the environment in order to achieve success at some objective, e.g., loss, accuracy. Weak sample efficiency is countered in the deep RL community by an increasing reliance on extremely large computational resources to train on vast amounts of data. The associated engineering remedies are hence superficial and reduce visibility of the underlying obstacles.

Deep RL's most high-profile achievements are some of the best examples to showcase the growing challenge of sample efficiency. OpenAI Five [22] was trained with roughly 40,000 years of compressed real-time experience over the course of ten months. AlphaStar [351] made use of more sophisticated inductive biases and replaced self-play [17] with population-based training [156] but still required roughly 200 years of real-time gameplay per agent, which was parallelized over massive resources in order to achieve a training time of 14 days. In a task of dextrous object manipulation, OpenAI's Rubik's Cube agent [247] required 13,000 years of compressed real-time simulation to learn a single task. Note how the single task required a similar amount of experience to that required for a highly complex multi-agent video game for OpenAI Five, clearly showing that the former is nontrivially more challenging for RL to handle.

In the following chapter, we proceed to address the issue of a priori rewards by proposing a framework for *Work on Command*.