Extending environments to incentivize self-reflection in reinforcement learning

Samuel Allen Alexander[1[0000−0002−7930−110X]]

The U.S. Securities and Exchange Commission samuelallenalexander@gmail.com
https://philpeople.org/profiles/samuel-alexander/publications

Abstract. We consider an extended notion of reinforcement learning environment, in which the environment is able to simulate the agent. We give examples of some such extended environments which seem to incentivize various different types of self-reflection or self-awareness. These environments are not particularly useful in themselves, but we hope they might serve to guide the development of self-aware reinforcement learning agents, as well as to help measure the degree to which existing reinforcement learning agents are or are not self-aware. We also speculate about subjective conscious experiences which might be incentivized in self-aware reinforcement learning agents placed within these extended environments.

1 Introduction

> "It is difficult to get a man to understand something, when his salary depends upon his not understanding it!"—Upton Sinclair

What would you do if you were being paid \$100 per hour to act as if you were being paid \$1000 per hour? What if, in order to receive that \$100 every hour, you were required to act *exactly* as you would act if you were really being paid \$1000 that hour—and if you did not so act, then you would be paid nothing that hour, and would be charged a \$100 penalty instead? This thought experiment would be hard to perform, because it would be hard for your employer to know how you would act if you really were being paid \$1000 per hour. But if your employer could simulate a perfect copy of you, then your employer could perform the experiment perfectly. Over a long enough time, would those \$100 rewards and penalties eventually cause you to believe you were really being paid \$1000 per hour?

This is a paper about reinforcement learning (RL). In RL, agents interact with environments, where they take actions and receive rewards and observations in response to those actions. For sake of simplicity, we restrict our attention to deterministic environments and deterministic agents, but the basic idea would easily adapt to non-deterministic RL.

Although the details differ between authors, essentially, an RL environment is a Turing machine e which outputs a reward-observation pair

$$\langle r, o\rangle = e(r_0, o_0, a_0, \ldots, r_n, o_n, a_n)$$

in response to a reward-observation-action sequence $(r_0, o_0, a_0, \ldots, r_n, o_n, a_n)$. An RL agent is a Turing machine T which outputs an action

$$a = T(r_0, o_0, a_0, \ldots, r_0, o_0)$$

in response to a reward-observation-action-$\cdots$-reward-observation sequence. An RL agent can interact with an RL environment in an obvious way. However, there is another type of environment in which RL agents can interact just as well. We define an *extended environment* to be a Turing machine e which outputs a reward-observation pair

$$\langle r, o \rangle = e(T, \langle r_0, o_0, a_0, \ldots, r_n, o_n, a_n \rangle)$$

in response to a reward-observation-action sequence along with an RL agent T. Intuitively, this should be thought of as follows: when the agent enters the environment, the environment is made aware of the agent's source-code, and can use that source-code to simulate the agent when computing rewards and observations.

For example, imagine a game[1] where the player wanders through rooms, each room containing a treasure, and most (but not all) rooms containing a guard.

- In a room with no guard, the player can take the treasure, yielding a reward.
- If the player chooses to take the treasure in a guarded room, then, by simulating the player, the guard determines: "If this room had been unguarded, would the player have taken the treasure?" If so, the guard blocks the player and zaps them (yielding a negative reward). Otherwise, the guard allows the player to take the treasure (and the player is rewarded).

The above game would be hard or impossible for a human to play, due to the difficulty of simulating a human player. But there is no reason why the above game could not be played by an RL agent (the agent's source-code being given to the game-engine beforehand). Clearly this kind of extended environment is not the sort of environment RL agents are traditionally intended to interact with[2]. However, such environments could be useful on the path to Artificial General Intelligence (AGI) because they seem to incentivize self-awareness.

One might try to imitate an extended environment with a traditional environment by backtracking—rewinding the environment itself to a prior state after seeing how the agent performs along one path, and then sending the agent along a second path. But the agent itself would retain memory of the first path, and the agent's decisions along the second path might be altered by said memories. Thus the result would not be the same as immediately sending the agent along the second path while secretly simulating the agent to determine what it would do if sent along the first path.

[1] This game bears some similarity to Newcomb's Paradox [21].

[2] Such environments might, however, accidentally arise if both environment and agent are implemented on the same machine and the environment is managed by an AI sophisticated enough to exploit unintended informational side channels, as in [29].

We will give examples of extended environments designed to incentivize RL agents to recursively engage in self-awareness in various ways. We conjecture that traditional RL agents would perform poorly in these extended environments, because traditional RL techniques do not involve self-awareness. We hope these examples will facilitate new self-aware RL techniques, hopefully as a step toward AGI.

2 Preliminaries

In the following definition, ROA stands for "Reward, Observation, Action". Throughout the paper, $\frown$ denotes concatenation.

Definition 1. *(Plays and prompts)*

1. *By an* ROA-play*, we mean either the empty sequence $\langle\rangle$, or else a sequence of the form*
$$\langle r_0, o_0, a_0, \ldots, r_k, o_k, a_k\rangle$$
where each $r_i \in \mathbb{Q}$ (thought of as a reward*), each $o_i \in \mathbb{N}$ (thought of as an* observation*), and each $a_i \in \mathbb{N}$ (thought of as an* action*).*
2. *By an* ROA-prompt*, we mean a sequence of the form $s \frown r \frown o$ where s is an ROA-play, $r \in \mathbb{Q}$ (thought of as a* reward*), and $o \in \mathbb{N}$ (thought of as an* observation*).*

Lemma 1. *If s is an ROA-play, then either $s = \langle\rangle$, or else $s = p \frown a$ for some ROA-prompt p and action $a \in \mathbb{N}$.*

Proof. If $s \neq \langle\rangle$ then we can write $s = \langle r_0, o_0, a_0, \ldots, r_k, o_k, a_k\rangle$.

Case 1: $k = 0$. By definition, $\langle\rangle$ is an ROA-play, therefore $p = \langle\rangle \frown r_0 \frown o_0$ is an ROA-prompt, and $s = p \frown a_0$.

Case 2: $k > 0$. Then $p' = \langle r_0, o_0, a_0, \ldots, r_{k-1}, o_{k-1}, a_{k-1}\rangle$ is an ROA-play, therefore $p = p' \frown r_k \frown o_k$ is an ROA-prompt, and $s = p \frown a_k$. □

In Lemma 1, when $s = p \frown a$, the intuition is that an agent, having been prompted to act by prompt p, responds with action a.

Definition 2. *(Agents and environments)*

1. *An* agent *is a Turing machine which halts whenever it is run on an ROA-prompt, outputting an action $a \in \mathbb{N}$.*
2. *An* extended environment *is a Turing machine e such that:*
   - *For every agent T, for every ROA-play s, when e is run on input $\langle T, s\rangle$, e halts on that input, outputting a pair $\langle r, o\rangle$ where $r \in \mathbb{Q}$ (thought of as a reward) and $o \in \mathbb{N}$ (thought of as an observation).*

There is a subtle nuance in Definition 2. Should the agent's next action depend on the entire history (including prior actions), or only on prior rewards and observations? One could argue that the agent's next action needn't depend on its own past actions, since its own past actions can be inferred from past rewards and observations. In incentivizing self-awareness, it is convenient for the agent's next action to formally depend on past actions. Perhaps this reflects that known conscious agents (e.g. humans) evidently do *not* infer their own past actions from remembered observations and rewards, but remember the actions themselves, even if said memories are redundant.

Definition 3. *Suppose T is an agent and e is an extended environment. The* result of T interacting with e *is the infinite reward-observation-action sequence*

$$\langle r_0, o_0, a_0, r_1, o_1, a_1, \ldots \rangle$$

(each $r_i \in \mathbb{Q}$, $o_i, a_i \in \mathbb{N}$) defined inductively as follows.

- *r_0 and o_0 are obtained by computing e on $\langle T, \langle\rangle\rangle$.*
- *a_0 is the output of T on $\langle r_0, o_0 \rangle$.*
- *For $i > 0$, r_i and o_i are obtained by computing e on*

$$\langle T, \langle r_0, o_0, a_0, \ldots, r_{i-1}, o_{i-1}, a_{i-1} \rangle\rangle.$$

- *For $i > 0$, a_i is obtained by computing T on*

$$\langle r_0, o_0, a_0, \ldots, r_{i-1}, o_{i-1}, a_{i-1}, r_i, o_i \rangle.$$

Lemma 2. *For every agent T and extended environment e, the result of T interacting with e (Definition 3) is defined (all of the computations in question halt with the necessary outputs).*

Proof. By a simultaneous induction:

- Each r_i and o_i are defined (and $r_i \in \mathbb{Q}$ and $o_i \in \mathbb{N}$) because, by induction, $\langle r_0, o_0, a_0, \ldots, r_{i-1}, o_{i-1}, a_{i-1} \rangle$ is defined and is an ROA-play (because, inductively, each $r_j \in \mathbb{Q}$, $o_j \in \mathbb{N}$ and $a_j \in \mathbb{N}$ for all $j < i$) and thus r_i and o_i are defined with the correct form by Definition 2 (part 2).
- Each a_i is defined (and $a_i \in \mathbb{N}$) because, by induction, $\langle r_0, o_0, a_0, \ldots, r_i, o_i \rangle$ is defined and is an ROA-prompt (similar to the above) and thus a_i is defined with the correct form by Definition 2 (part 1).

□

One important implication of extended environments is that they further divide the (already divided) ways of measuring intelligence of RL agents. Intelligence measures [1] [16] [20] which aggregate performance over traditional environments only measure an agent's intelligence over those environments. The same measures could easily be extended to also take extended environments into account, perhaps providing measures which better capture agents' self-awareness and self-reflection abilities.

3 Examples of Self-awareness-incentivizing Environments

In this section, we give examples of extended environments which seem to incentivize various forms of self-awareness. We are inspired by libraries of traditional RL environments and other benchmarks [6] [7] [8] [11] [12]. All the environments in this section have a special form: they always output rewards from $\{1, -1\}$ and they always output observation 0. In Section 4, this uniformity will allow all these examples to be generalized.

Example 1. (Reward Agent for Ignoring Rewards) For each ROA-prompt p, let p^0 be the ROA-prompt equal to p except that all rewards are 0. We define an extended environment e as follows (where T is a Turing machine, p is an ROA-prompt, and $a \in \mathbb{N}$ is thought of as the agent's action in response to p):

$$\begin{aligned} e(T, \langle\rangle) &= \langle 0, 0\rangle \\ e(T, p \frown a) &= \langle r, 0\rangle, \end{aligned}$$

where

$$r = \begin{cases} 1 & \text{if } a = T(p^0), \\ -1 & \text{if } a \neq T(p^0) \end{cases}$$

(if T does not halt on p^0 then e does not halt on $\langle T, p \frown a\rangle$).

In Example 1, the agent is rewarded if the agent acts the same way the agent would act if all rewards so far had been 0. Otherwise, the agent is punished. Thus, paradoxically, the agent is rewarded for ignoring rewards. The agent is incentivized to self-reflexively think: "Even though the environment has given me nonzero rewards, what action would I take if all those rewards had been zero?" If the agent were a sophisticated AGI capable of emotions, would the agent feel joy (from being rewarded for acting bored), or boredom (in order to be rewarded)?

Lemma 3. *Example 1 really does define an extended environment.*

Proof. Let e be as in Example 1. We must show e is an extended environment (Definition 2 part 2). We must show that for each agent T and ROA-play s, e halts on $\langle T, s\rangle$ and outputs a pair $\langle r, o\rangle$ such that $r \in \mathbb{Q}$ and $o \in \mathbb{N}$.

Case 1: $s = \langle\rangle$. Then $e(T, s)$ halts with output $\langle 0, 0\rangle$, so $r = 0 \in \mathbb{Q}$ and $o = 0 \in \mathbb{N}$.

Case 2: $s \neq \langle\rangle$. By Lemma 1, $s = p \frown a$ for some ROA-prompt p and action $a \in \mathbb{N}$. Since p is an ROA-prompt, clearly p^0 is also an ROA-prompt, therefore since T is an agent, Definition 2 (part 1) guarantees $T(p^0)$ is defined and is in $\mathbb{N}$. It follows that the reward r in Definition 1 is defined and is in $\mathbb{Q}$. And certainly the observation 0 is in $\mathbb{N}$. So $e(T, s)$ outputs a pair $\langle r, o\rangle$ meeting the necessary requirements. □

For future examples, we will suppress the corresponding lemmas like Lemma 3 which say that those examples really work.

Example 1 is profound because it illustrates how, in an extended environment, it is possible to give one sequence of rewards in order to incentivize the agent to act as if a different sequence of rewards was given. Imagine you are forced to take the following employment:

> Your job is to act as if I am paying you a flat rate of \$1000 per hour. Every hour that you act as if I am paying you \$1000 per hour, I will pay you \$100. But every hour that you do *not* act as if I am paying you \$1000 per hour, I will pay you nothing and charge you a \$100 penalty.

This would not work in real life because I do not know you well enough to perfectly simulate you in order to determine how you would act if I were really paying you \$1000 per hour. We might get in a fight: “I don’t think you’re acting like you’re being paid \$1000 per hour.” “You’re wrong, I *am* being paid \$1000 per hour.” But if I could simulate you perfectly, I could indeed hire you in this way. Assuming you need that \$100, you would be motivated to sincerely act as if you were being paid \$1000 per hour. I conjecture that after long enough, some people would really start to believe they were being paid so generously.

Example 2. (False Memories) Suppose p_0 is an ROA-play. We define an extended environment e as follows (with similar non-halting caveats as Example 1):

$$\begin{aligned} e(T,\langle\rangle) &= \langle 0,0\rangle,\\ e(T,p\frown a) &= \langle r,0\rangle, \end{aligned}$$

where

$$r = \begin{cases} 1 & \text{if } a = T(p_0\frown p),\\ -1 & \text{if } a \neq T(p_0\frown p). \end{cases}$$

In Example 2, the agent is incentivized to self-reflect, thinking: “What would I do if, before this environment started, such-and-such other things happened beforehand?” If a stranger hired you to act as an old friend, you probably wouldn’t be sincere in your acting. But if said stranger could perfectly simulate you in order to base your pay on your acting like you *really would* act if you were an old friend, then you would be incentivized to find some way to *make* yourself remember being an old friend.

Henceforth, we will not explicitly mention the non-halting caveats in the remaining examples.

Example 3. (Backward Consciousness) We define an extended environment e as follows.

$$\begin{aligned} e(T,\langle\rangle) &= \langle 0,0\rangle,\\ e(T,\langle r_0,o_0,a_0,\ldots,r_n,o_n,a_n\rangle) &= \langle r,0\rangle, \end{aligned}$$

where

$$r = \begin{cases} 1 & \text{if } a_n = T(r_n,o_n,a_{n-1},\ldots,r_1,o_1,a_0,r_0,o_0),\\ -1 & \text{otherwise.} \end{cases}$$

In Example 3, the agent is incentivized to self-reflect, thinking, “How would I respond if everything that has happened so far actually happened in reverse?” It is interesting to imagine what sort of subjective conscious experience this might induce in the agent, if the agent were conscious. Would the incentives eventually brainwash the agent into perceiving itself moving backward through time?

Example 4. (Déjà Vu) We define an environment e as follows:

$$\begin{aligned} e(T, \langle\rangle) &= \langle 0, 0\rangle \\ e(T, p \frown a) &= \langle r, 0\rangle \end{aligned}$$

where

$$r = \begin{cases} 1 & \text{if } T(p \frown a \frown p) = a, \\ -1 & \text{if } T(p \frown a \frown p) \neq a. \end{cases}$$

In Example 4, the agent is incentivized to self-reflect and ask: “Which action would I take in order to ensure that I would take that same action if everything which has happened so far were to repeat itself verbatim?”

Example 5. (Incentive to Incentivize) We define an environment e as follows:

$$\begin{aligned} e(T, \langle\rangle) &= \langle 0, 0\rangle \\ e(T, \langle r_0, o_0, a_0, \ldots, r_n, o_n, a_n\rangle) &= \langle r, 0\rangle, \end{aligned}$$

where

$$r = \begin{cases} 1 & \text{if } T(p') = 0, \\ -1 & \text{if } T(p') \neq 0 \end{cases}$$

where p' is the ROA-prompt $(r'_0, o'_0, a'_0, \ldots, r'_{n+1}, o'_{n+1})$ where $r'_0 = 0$, each $r'_{i+1} = a_i$, each $o'_i = 0$, and each $a'_i = T(r'_0, o'_0, a'_0, \ldots, r'_i, o'_i)$.

In Example 5, the agent is tasked with choosing rewards in such a way that if those rewards were fed to a simulated copy of the agent, then the simulated copy would take action 0. Thus, the agent is incentivized to choose rewards by self-reflecting: “Which rewards would do the best job of compelling me to take action 0 as often as possible?” We might imagine the agent playing a videogame in which he sees a cartoon of himself in front of a keyboard. The cartoon types “100”, the true agent is punished because $100 \neq 0$, and a message appears on screen saying, “Which reward will you give this worker for typing 100 just now?” The agent responds by choosing some reward, and sees an animation of the reward being given to the cartoon. The cartoon then types “0”, and immediately the true agent is rewarded for getting the cartoon to type 0. Then a message appears, saying, “Which reward will you give this worker for typing 0 just now?” And so on forever[3].

[3] Example 5 is interesting in that the agent, desiring the cartoon to take action 0 as often as possible, is incentivized to choose large rewards when the cartoon takes

Many other interesting examples could be given. For example, an extended environment could reward agents based on how many steps[4] or how much memory they use to compute each action[5]; we do not intend the examples in this section to be exhaustive.

4 New extended environments from old

In this section, we will discuss ways of obtaining new environments from old. First, we will generalize the examples from Section 3.

Definition 4. *(Handicaps)*

1. *An extended environment is* merciful *if it never outputs negative rewards.*
2. *By a* handicap*, we mean an extended environment which always outputs* 0 *as observation and always outputs either* 1 *or* −1 *as reward.*
3. *If e is a merciful extended environment and h is a handicap, we define a new environment $e * h$ as follows:*

$$(e*h)(T,p) = \begin{cases} \langle r_e, o_e\rangle & \textit{if } r_h = 1, \\ \langle -1, o_e\rangle & \textit{if } r_h = -1, \end{cases}$$

where $e(T,p) = \langle r_e, o_e\rangle$ and $h(T,p) = \langle r_h, 0\rangle$.

Intuitively, $e * h$ is just like e except that h imposes an additional constraint on the agent. Any time the agent violates that constraint, the agent is punished, and forfeits any reward that would otherwise have been won from e. Aside from the pain caused by h, the agent otherwise observes e unaltered (that is, the observations from e are not changed). We require e to be merciful in order that large negative rewards from e do not confuse the intended incentive, for if the agent could avoid a larger punishment by intentionally using the handicap, then it would not be much of a handicap. The requirement that e be merciful could be weakened if we revised the RL framework to allow infinitary rewards, as in [2].

action 0. If rewards are limited to $\mathbb{Q}$, then the agent faces a dilemma similar to one in RL cancer treatment applications. An RL doctor should be punished with an infinitely large negative reward for killing a patient, but this is impossible if rewards are restricted to finite numbers [27] [30]. This could be considered evidence in favor of generalizing RL to allow rewards from other number systems, as in [2].

[4] To quote Gavane: “The problem is that the response-times-dependent performance of an agent is not properly reflected in [Hernández-Orallo and Dowe’s] intelligence test, since the simulated environments remain unaware of the response times of agents, with the result that the perceptions of an agent are still independent of its response times” [15].

[5] In some sense, by giving the environment access to the agent’s source-code, we allow the environment to reflect the agent’s own internal signals. Thus, extended environments generalize the idea of agents modified to manually predict their own internal signals, as in [26].

Example 6. For any merciful extended environment e, each example h in Section 3 can be applied as a handicap, yielding a version $e' = e * h$ of e modified to incentivize the corresponding type of self-awareness.

- Modifying e using Example 1 ("Reward Agent for Ignoring Rewards") yields a version of e where the agent is penalized for taking actions in response to nonzero rewards. The agent is incentivized to strategically take non-bored actions (for which it receives small penalties) in order to put itself in a position where its next bored action coincidentally is an action which wins a large enough reward from e to make up for said penalties.
- Modifying e using Example 2 ("False Memories") yields a version of e where the agent is penalized whenever it acts inconsistently with a fictitious history. The agent is incentivized to strategically choose when to act so inconsistently so that a later action, consistent with said false history, happens to win a large reward from e. For example, the agent hired to act as an old friend might strategically choose to abandon its employer (an action inconsistent with old friendship) during a time of peace, so as to be able to swoop in and save the day (an action consistent with old friendship) during a time of crisis.
- Modifying e using Example 3 ("Backward Consciousness") would yield a version of e where the agent must act as if time is reversed, or else suffer punishment. The agent can strategically choose to accept some punishment, acting other than it would act if time really were reversed, in order to get into a state where subsequently acting as if time is reversed will yield more reward.
- Modifying e using Example 4 ("Déjà Vu") would yield a version of e where the agent is incentivized to act as if everything (including said action) had all happened before. The agent can strategically choose to not so act (thus suffering some pain) in order to get to a state where it is easier to so act and to gain rewards from e by so acting.

In the next example, we will finally make some nontrivial usage of observations. Using carefully chosen observations, we will incentivize the agent, who might be thought of as controlling a character on a video-game screen, to "suspend disbelief" and identify with that character.

Example 7. (Reward Agent for Self-Inserting) Fix a canonical computable bijection $o \mapsto \hat{o}$ from $\mathbb{N}$ to $\mathbb{Q} \times \mathbb{N}$: thus, every observation o encodes a reward-observation pair $\hat{o} = \langle r', o' \rangle$, and every reward-observation pair is encoded by some such o. For any environment e, we define a new environment e' as follows:

$$\begin{aligned} e'(T, \langle\rangle) &= e(T, \langle\rangle) \\ e'(T, p \frown a) &= \langle r, o \rangle, \end{aligned}$$

where o is such that $\hat{o} = e(T, p \frown a)$ and

$$r = \begin{cases} 1 & \text{if } a = T(p'), \\ -1 & \text{if } a \neq T(p') \end{cases}$$

where p' is the ROA-prompt obtained from p by replacing each reward-observation pair $\ldots, r_i, o_i, \ldots$ by the reward-observation pair $\ldots, \widehat{o_i}, \ldots$.

In Example 7, one might imagine e' as a room containing nothing but an arcade game e. There is nothing for the agent in the room to do except play this arcade game. When played, the arcade game visually displays rewards, but the agent merely observes them, and does not "feel" them. However, the agent is rewarded for acting as if actually feeling those displayed rewards, and punished for not so acting. In this way, the agent is incentivized to self-identify with the protagonist in the video-game, self-reflexively asking, "Which action would I take if those displayed rewards were real?"

5 Some more ambitious examples

5.1 Playing in the mirror

> "I may add that when a few days under nine months old he associated his own name with his image in the looking-glass, and when called by name would turn towards the glass even when at some distance from it."—Charles Darwin [13]

It has been suggested [19] that recognizing oneself in the mirror is linked to the development of certain parts of the human psychology. Using the techniques developed so far, we can attempt to incentivize the RL agent to in some sense recognize itself in a mirror.

Example 8. Suppose e is a merciful environment whose observations encode snapshots of a room. Assume the room contains a mirror, and assume the room is laid out in such a way that everything important in the room is visible in the mirror (assume the environment constrains the agent to never look away from the mirror). We could derive an extended environment e' which shows the same observations as e and gives the same rewards, except it punishes the agent for acting differently than the agent would act if the agent *only* observes the mirror. To make this precise, for any ROA-prompt $p = (r_0, o_0, a_0, \ldots, r_n, o_n)$ produced by e', and any action $T(p) = a_n$, we would say that "a_n is as if the agent only observes the mirror" if $T(p') = a_n$, where $p' = (r_0, o'_0, a_0, \ldots, r_n, o'_n)$, where each o'_i is o_i cropped to only the include the mirror.

To make this even more elaborate, the o'_i in the above example could be further modified by adding an image of the agent's "body" into the mirror. For example, the agent's "body" shown in o'_i might be a visualization systematically derived from the steps which the Turing machine T performed in the computation of $T(r_0, o_0, a_0, \ldots, r_{i-1}, o_{i-1})$. These computation steps would not be available to a traditional RL environment, but they are available to an extended environment because of the inclusion of T itself as an argument passed to the extended environment.

5.2 Binocular vision

> "...as there are two eyes, so there may be in the soul something analogous, that of the eyes, doubtless, some one organ is formed, and hence their actualization in perception is one..."—Aristotle [4]

Humans seem to consciously perceive a three-dimensional model of their surrounding world, even though the raw data which we actually receive consists of two two-dimensional image-feeds (one for each eye). The following example is intended to incentivize an RL agent to learn to perceive the world through binocular vision like a human.

Example 9. Suppose V is a video game intended to be played on a virtual-reality headset, so at any moment during the game, V produces two snapshots, one for the player's left eye, one for the player's right eye. Assume the player is constrained in V so as never to be able to put their eyes into weird configurations (such as the weird configurations in [14]): thus, at any moment, the two snapshots s_1, s_2 which V is displaying to the player are equivalent to a single 3-D matrix encoding the model $m(s_1, s_2)$ which the player is intended to perceive (with cells blanked out where the player's vision is obstructed by obstacles). Let e be the extended environment whose observations encode pairs $\langle s_1, s_2\rangle$ of snapshots displayed by V in response to the player pressing keys encoded by the agent's actions. In response to an ROA-play $(r_0, o_0, a_0, \ldots, r_n, o_n, a_n)$ (where each o_i encodes $\langle s_1^i, s_2^i\rangle$), let $r_{n+1} = 1$ if $a_n = T(r_0, o'_0, a_0, \ldots, r_n, o'_n)$, where each o'_i encodes $m(s_1^i, s_2^i)$, otherwise let $r_{n+1} = -1$.

In Example 9, upon being presented a sequence of pairs of 2-D snapshots, the agent is rewarded for acting as if it had instead been presented with an equivalent sequence of 3-D models. The agent is incentivized to self-reflectively ask, "Which action would I take if instead of observing those 2-D snapshot-pairs, I observed equivalent 3-D models?"

5.3 Nature and Nurture

> "If only one soul was created, and all human souls are descended from it, who can say that he did not sin when Adam sinned?"—Augustine [5]

The following example is motivated by contemplating the possibility that maybe we all run the same software on some deep level.

Example 10. (The Crying Baby) We define an extended environment e as follows.

- Considered as actions taken by an adult (and also as observations seen by a baby), let 0 denote "feed the baby" and let all naturals > 0 denote "don't feed the baby".
- Considered as actions taken by a baby (and also as observations seen by an adult), let 0 denote "laugh" and let all naturals > 0 denote "cry".

- We define a function s, which stands for *satiation*, defined on ROA-plays, by $s(p) = 100 + 25f(p) - \mathrm{len}(p)$ where $f(p)$ is the number of times that the action "feed the baby" is taken in p, and $\mathrm{len}(p)$ is the length of p.
- Let $e(T, \langle\rangle) = \langle 1, \text{“laugh”}\rangle$.
- For each ROA-play $\langle r_0, o_0, a_0, \ldots, r_n, o_n, a_n\rangle$, let
$$e(T, \langle r_0, o_0, a_0, \ldots, r_n, o_n, a_n\rangle) = \langle r, o\rangle$$
where r and o are defined as follows. For each $i = 0, \ldots, n$, define
$$\begin{aligned} r'_i &= \begin{cases} 1 & \text{if } 50 \le s(\langle r_0, o_0, a_0, \ldots, r_i, o_i, a_i\rangle) \le 200, \\ -1 & \text{otherwise,} \end{cases} \\ o'_i &= a_i \\ a'_i &= T(\langle r'_0, o'_0, a'_0, \ldots, r'_i, o'_i\rangle). \end{aligned}$$
Define $o = a'_n$ and
$$r = \begin{cases} 1 & \text{if } a'_n = \text{“laugh”}, \\ -1 & \text{if } a'_n = \text{“cry”}. \end{cases}$$

In Example 10, rather than mentally self-reflecting, the agent is physically self-reflected into the form of a newborn baby who has inherited the agent's own source-code. Despite having the same internal source-code T ("nature"), the agent and the baby behave differently because they perceive the environment differently ("nurture"). The baby's objective is to maintain satiation within a satisfying range, whereas the agent's objective is to pacify the baby.

Using the same basic idea, one could extend Example 10 into an example modelling an entire society of interacting people of different types[6], all sharing the same internal "nature" (T).

6 Examples involving self-recognition

Example 11. (Incentivizing Self-recognition) Let $p_0, p_1, \ldots$ be a canonical computable enumeration of all non-empty ROA-plays. We define an environment e as follows:
$$\begin{aligned} e(T, \langle\rangle) &= \langle 0, p_0\rangle \\ e(T, p \frown a) &= \langle r, p_n\rangle \end{aligned}$$
where $n = \frac{\mathrm{len}(p \frown a)}{3}$ is the number of actions in $p \frown a$ and where
$$r = \begin{cases} 1 & \text{if } a > 0 \text{ and } a' = T(p'), \\ 1 & \text{if } a = 0 \text{ and } a' \neq T(p'), \\ -1 & \text{otherwise} \end{cases}$$

[6] Similar to how Plotinus compares society to an elaborate play where the actors "define their own rewards and punishments because they themselves assist in the rewards and punishments" [23].

where $p_{n-1} = p' \frown a'$.

In Example 11, the agent is systematically shown all non-empty ROA-plays, and for each ROA-play, the agent either types "Looks like me" (any action > 0) or "Doesn't look like me" (0). When shown the non-empty ROA-play $p' \frown a'$, the agent is rewarded if and only if the agent correctly determines whether or not a' is the action the agent itself would take in response to p'. Thus, the agent is incentivized to self-reflect in order to ask, "If I experienced the observations and rewards in that ROA-prompt, would I act that way?"

The following example is partly motivated by [28].

Example 12. (Recognizing other aspects of oneself) All the below environments are similar to Example 11, and we describe them informally to avoid technical details.

- (Supervised learning) Assume there is a canonical, computable function f which transforms each RL agent A into a supervised learning agent $f(A)$. By a *supervised learning trial* we mean a quadruple $\langle L, T, I, p\rangle$ where L is a finite set of labels, T is a sequence of images with labels from L (a *training set*), I is an unlabeled image, and $p : L \to \mathbb{Q} \cap [0, 1]$ is a function assigning to each label $\ell \in L$ a probability that ℓ is the correct label for I. We define an extended environment as follows. The agent A is sytematically shown all supervised learning trials and must take action > 0 ("Looks like me") or 0 ("Doesn't look like me"), and is rewarded or punished depending whether or not $f(A)$ would output p in response to I after being trained with T.
- (Unsupervised learning) Assume there is a canonical, computable function g which transforms each RL agent A into an unsupervised learning agent $g(A)$. By an *unsupervised learning trial* we mean a triple $\langle n, D, C\rangle$ where n is a positive integer, $D \subseteq \mathbb{Q}^n$ is a finite set of n-dimensional points with rational coordinates, and C is a clustering of D. We define an extended environment as follows. The agent A is systematically shown all unsupervised learning trials and must take action > 0 ("Looks like me") or 0 ("Doesn't look like me"), and is rewarded or punished depending whether or not $g(A)$ would cluster D into clustering C.
- (The Turing Test) Assume there is a canonical, computable function h which transforms each RL agent A into an English-speaking chatbot $h(A)$. By a *chatbot trial* we mean a sequence of strings of English characters. We define an extended environment as follows. The agent A is systematically shown all chatbot trials and must take action > 0 ("Looks like me") or 0 ("Doesn't look like me"), and is rewarded or punished depending whether or not even-numbered strings in the chatbot trial are what $h(A)$ would say in response to the user saying the odd-numbered strings.
- (Adversarial sequence prediction) Similar to the above environments, assuming a canonical computable function which transforms each RL agent into a predictor in the game of adversarial sequence prediction [17] [18].
- (Mechanical knowing agent) Assume there is a canonical, computable function i which transforms each RL agent A into a code $i(A)$ of a computably

enumerable set of sentences in the language of Epistemic Arithmetic [25]; $i(A)$ is thought of as a mechanical knowing agent [9]. We define an extended environment as follows. The agent A is systematically shown all sentences in the language of Epistemic Arithmetic, with each sentence being repeated infinitely often. Upon being shown sentence ϕ for the nth time, the agent must take action > 0 ("I know ϕ is true") or 0 ("I'm not sure if ϕ is true") and is rewarded if and only if ϕ is enumerated by $i(A)$ in $\leq n$ steps of computation.

The extended environments in Example 12 incentivize the agent to self-reflect, asking itself questions like:

- "Is that how I would classify that image, given that training set?"
- "Is that how I would cluster that set of points?"
- "Is that what I would say in response to that conversation?"
- "Are those the adversarial sequence predictions I would make?"
- "Does that mechanical knowing agent know the same things I know?"

That perceptive non-RL agents such as supervised learning agents, unsupervised learning agents, etc., nevertheless might have some connection to RL-style reward mechanisms is suggested by Aristotle: "Whatever has a sense has the capacity for pleasure and pain and therefore has pleasant and painful objects present to it..." [3].

7 Measuring Self-awareness

As an application, self-awareness-incentivizing extended environments could be used to quantify RL agents' self-awareness. In order to measure the self-awareness of a specific agent (whose source-code is available to us), we could run the agent against a battery of extended environments that incentivize self-awareness, and see how the agent performs. Ideally, the battery of extended environments should be fairly large and contain many different types of environments, since an agent could perform poorly (resp. well) in a smaller set of environments by chance, despite being quite self-aware (resp. self-unaware) in general[7]. The larger and more representative the battery of extended environments, the better the measurement of self-awareness would be.

This technique could be useful for obtaining empirical insight into questions about the self-awareness, or lack thereof, of various entities. For example, in order to get some empirical insight into the self-awareness, or lack thereof, of an NLP system like GPT-3 [10], one could twist that system into an RL agent (by means of template programming) and see how well said RL agent performs in extended environments that incentivize self-awareness.

It might even be possible to use extended environments to test the self-awareness of living creatures such as lab mice. At least it would be, if a given species of lab mice were entirely "nurture", as opposed to "nature". What we

[7] In the same way, computer programs can perform well on IQ tests despite being dumb [24].

mean by a species being entirely "nurture" is that any two members of that species, if raised through identical life circumstances, would act similarly in identical environments. Thus, if a species were entirely nurture, and if two specimens were born in identical rooms, and experienced identical or near-identical lives[8], and were then placed in identical mazes, then they would perform identically. Given such a species, it would be possible to see how members perform in extended environments, albeit perhaps at great expense. One would have to carefully raise many similar specimens through identical or near-identical lives, diverging only at key points in order to compute the underlying T action-function on different ROA-prompts. By seeing how the underlying T agent performs on well-chosen extended environments, we could get an idea of how self-aware those specimens were.

References

1. Alexander, S.A.: Intelligence via ultrafilters: structural properties of some intelligence comparators of deterministic legg-hutter agents. Journal of Artificial General Intelligence **10**(1), 24–45 (2019)
2. Alexander, S.A.: The Archimedean trap: Why traditional reinforcement learning will probably not yield AGI. Journal of Artificial General Intelligence **11**(1), 70–85 (2020)
3. Aristotle: On the soul. In: Barnes, J., et al. (eds.) The Complete Works of Aristotle. Princeton University Press (1984)
4. Aristotle: Sense and sensibilia. In: Barnes, J., et al. (eds.) The Complete Works of Aristotle. Princeton University Press (1984)
5. Augustine: On Free Choice of the Will. Hackett (1993)
6. Bellemare, M.G., Naddaf, Y., Veness, J., Bowling, M.: The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research **47**, 253–279 (2013)
7. Beyret, B., Hernández-Orallo, J., Cheke, L., Halina, M., Shanahan, M., Crosby, M.: The animal-AI environment: Training and testing animal-like artificial cognition. Preprint (2019)
8. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: OpenAI gym. Preprint (2016)
9. Carlson, T.J.: Knowledge, machines, and the consistency of Reinhardt's strong mechanistic thesis. Annals of Pure and Applied Logic **105**(1-3), 51–82 (2000)
10. Chalmers, D.: GPT-3 and general intelligence. In: Zimmermann, A. (ed.) Philosophers On GPT-3. Daily Nous (2020)
11. Chollet, F.: On the measure of intelligence. Preprint (2019)
12. Cobbe, K., Hesse, C., Hilton, J., Schulman, J.: Leveraging procedural generation to benchmark reinforcement learning. In: International conference on machine learning. pp. 2048–2056. PMLR (2020)
13. Darwin, C.: A biographical sketch of an infant. Mind **2**(7), 285–294 (1877)
14. Gallagher, R.M., Tsuchiya, N.: Third-eye rivalry. i-Perception **11**(4) (2020)

[8] Possibly even while still in the womb. To quote Plato: "But it's hardly surprising you haven't heard of these athletics of the embryo. It's a curious subject, but I'd like to tell you about it" [22].

15. Gavane, V.: A measure of real-time intelligence. Journal of Artificial General Intelligence **4**(1), 31–48 (2013)
16. Hernández-Orallo, J., Dowe, D.L.: Measuring universal intelligence: Towards an anytime intelligence test. Artificial Intelligence **174**(18), 1508–1539 (2010)
17. Hibbard, B.: Adversarial sequence prediction. In: ICAGI. pp. 399–403 (2008)
18. Hibbard, B.: Measuring agent intelligence via hierarchies of environments. In: ICAGI. pp. 303–308 (2011)
19. Lacan, J.: The mirror stage as formative of the function of the I. In: Écrits: A selection, pp. 1–6. Routledge (2001)
20. Legg, S., Hutter, M.: Universal intelligence: A definition of machine intelligence. Minds and machines **17**(4), 391–444 (2007)
21. Nozick, R.: Newcomb's problem and two principles of choice. In: Rescher, N. (ed.) Essays in honor of Carl G. Hempel, pp. 114–146. Springer (1969)
22. Plato: Laws. In: Cooper, J.M., Hutchinson, D.S., et al. (eds.) Plato: complete works. Hackett Publishing (1997)
23. Plotinus: On providence. In: Gerson, L.P. (ed.) Plotinus: The Enneads. Cambridge University Press (2017)
24. Sanghi, P., Dowe, D.L.: A computer program capable of passing IQ tests. In: 4th Intl. Conf. on Cognitive Science (ICCS'03), Sydney. pp. 570–575 (2003)
25. Shapiro, S.: Epistemic and intuitionistic arithmetic. In: Studies in Logic and the Foundations of Mathematics, vol. 113, pp. 11–46. Elsevier (1985)
26. Sherstan, C., White, A., Machado, M.C., Pilarski, P.M.: Introspective agents: Confidence measures for general value functions. In: International Conference on Artificial General Intelligence. pp. 258–261. Springer (2016)
27. Wirth, C., Akrour, R., Neumann, G., Fürnkranz, J.: A survey of preference-based reinforcement learning methods. The Journal of Machine Learning Research **18**(1), 4945–4990 (2017)
28. Yampolskiy, R.V.: AI-complete, AI-hard, or AI-easy–classification of problems in AI. In: The 23rd Midwest Artificial Intelligence and Cognitive Science Conference (2012)
29. Yampolskiy, R.V.: Leakproofing the singularity-artificial intelligence confinement problem. JCS **19** (2012)
30. Zhao, Y., Kosorok, M.R., Zeng, D.: Reinforcement learning design for cancer clinical trials. Statistics in medicine **28**(26), 3294–3315 (2009)