Can reinforcement learning learn itself? A reply to 'Reward is enough'

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. In their paper 'Reward is enough', Silver et al conjecture that the creation of sufficiently good reinforcement learning (RL) agents is a path to artificial general intelligence (AGI). We consider one aspect of intelligence Silver et al did not consider in their paper, namely, that aspect of intelligence involved in designing RL agents. If that is within human reach, then it should also be within AGI's reach. This raises the question: is there an RL environment which incentivises RL agents to design RL agents?

1 Introduction

In their thought-provoking paper 'Reward is enough' [23], Silver et al hypothesise that "intelligence and its associated abilities may be understood as subserving the maximisation of reward". Motivated by recent reinforcement learning (RL) triumphs such as AlphaZero's performance in the game of Go [21] [22], Silver et al argue that:

1. Reward is enough for knowledge and learning.
2. Reward is enough for perception.
3. Reward is enough for social intelligence.
4. Reward is enough for language.
5. Reward is enough for generalisation.
6. Reward is enough for imitation.

Silver et al then argue that it should be possible to achieve Artificial General Intelligence (AGI) via the design of RL agents. They say:

> "A sufficiently powerful and general reinforcement learning agent may ultimately give rise to intelligence and its associated abilities. In other words, if an agent can continually adjust its behaviour so as to improve its cumulative reward, then any abilities that are repeatedly demanded by its environment must ultimately be produced in the agent's behaviour. A good reinforcement learning agent could thus acquire behaviours that exhibit perception, language, social intelligence and so forth, in the course of learning to maximise reward in an environment, such as the human world, in which those abilities have ongoing value."

And in their conclusion:

> "Finally, we have presented a conjecture that intelligence could emerge in practice from sufficiently powerful reinforcement learning agents that learn to maximise future reward. If this conjecture is true, it provides a direct pathway towards understanding and constructing an artificial general intelligence."

If we have understood correctly, Silver et al's conclusion seems to depend on philosophical induction. Namely: assuming claims 1–6 above, conclude that reward is enough for all tasks in reach of human intelligence.

In order to put the above conclusion to the test, we ask: is reward enough for the creation of RL agents? In other words—in the same way that reward can allegedly be used to incentivise RL agents to know, to learn, to perceive, to exhibit social intelligence, to exhibit language, to generalise, and to imitate—can reward be used to incentivise RL agents to design[1] RL agents? Hence the title of this paper: "Can reinforcement learning learn itself?" To rephrase it yet another way, suppose we define *RL-solving intelligence* to be that aspect of human intelligence which RL researchers apply when they design RL agents. Then: is reward enough for RL-solving intelligence?

In addition to the question of whether RL can learn itself, there is also the question of whether humans are capable of designing sufficiently good RL agents (Silver et al seem to implicitly assume the answer to this is "yes"). These two yes-or-no questions give rise to four possibilities.

1. RL can learn itself, and humans are capable of designing sufficiently good RL agents. This would be strong evidence supporting Silver et al's conjecture.
2. RL can learn itself, but humans are not capable of designing sufficiently good RL agents. Then whether or not RL is a path to AGI, it is not a practical one, at least not for humans.
3. RL cannot learn itself, and humans are capable of designing sufficiently good RL agents. Then it seems RL cannot lead to AGI, because "reward is not enough" for at least one type of human intelligence, namely RL-solving intelligence.
4. RL cannot learn itself, and humans are not capable of designing sufficiently good RL agents. Then whether or not RL is a path to AGI, it is not a practical one, at least not for humans.

The structure of this paper is as follows.

[1] To be clear, when an agent updates its own future behavior based on training data, we do not consider this to be an instance of the agent designing a new agent, even though in some sense the agent post-training is different than the agent pre-training. In the same way, when one reads a book, one becomes, in a sense, a different human being, yet we do not say that by doing so, one has designed a human being. When we speak of an RL agent designing an RL agent, we mean it in the same sense as, e.g., when we speak of an RL agent writing a poem. An RL agent would write a poem by writing down words. In the same way, an RL agent would design an RL agent by writing down pieces of computer code.

- In Section 2 we briefly review RL.
- In Section 3 we discuss that aspect of intelligence involved in the designing of RL agents.
- In Section 4 we discuss a type of RL environment which, if realised, might incentivise RL agents to design RL agents.
- In Section 5 we address some anticipated objections.
- In Section 6 we summarise and draw conclusions.

The main thesis of this paper is that before we can conclude that RL is a direct path to AGI (as Silver et al conjecture it is), we ought first to establish that RL is a direct path to RL-solving intelligence. In the past, skeptics said "computers will never master chess," but computers mastered chess; "computers will never master Go," but computers mastered Go. In order to avoid falling into checkmate once again, skeptics need to think bigger. Perhaps they could rally around "computers will never master designing RL agents" (we do not take a stance here on whether computers will be able to do so, we merely suggest this to the skeptics as a more defensible position).

2 Reinforcement Learning

Reinforcement learning is a branch of machine learning in which an *agent* interacts with an *environment*. As the subject is relatively young, there is not consensus on the formalization. Many authors do not formalize RL at all, and this includes Silver et al in the paper we are responding to (they semi-formally describe RL but their description is not mathematically rigorous). In order to make our response self-contained, we will give a rigorous definition (a modification of Hutter [11]), and we will indicate some of the many ways in which this formalization could differ. The reader should bear in mind that Silver et al only vaguely define RL in their paper: their remarks would apply to many different variations of RL, as would our response. Thus, the particular details of the following formalization are not important. But we felt that since some participants in this workshop might not be familiar with RL, we should offer a concrete formalization in order to avoid misunderstanding. Readers familiar with RL can safely skip the following definition.

Definition 1. *(Reinforcement Learning) Fix a finite set $\mathcal{O}$ of* observations *and a finite set $\mathcal{A}$ of* actions *(with $|\mathcal{O}| > 1$, $|\mathcal{A}| > 1$). By a* percept*, we mean a pair (o, r) where $o \in \mathcal{O}$ is an observation and $r \in \mathbb{R}$ is a number, called a* reward*. Write $\mathcal{P}$ for the set of all percepts.*

1. *Write $(\mathcal{PA})^*$ for the set of all finite sequences beginning with a percept, terminating with an action, and following the pattern "percept, action, ...". We also include the empty sequence $\langle\rangle$ in $(\mathcal{PA})^*$. Intuitively, an element $(p_0, a_0, \ldots, p_n, a_n)$ of $(\mathcal{PA})^*$ should be thought of as a percept-action history ending with an action.*

2. *Write* $(\mathcal{PA})^*\mathcal{P}$ *for the set of all sequences of form* $s \frown p$ *with* $s \in (\mathcal{PA})^*$, $p \in \mathcal{P}$ *(here* $\frown$ *denotes concatenation). An element* $(p_0, a_0, \ldots, p_{n-1}, a_{n-1}, p_n)$ *of* $(\mathcal{PA})^*\mathcal{P}$ *should be thought of as a percept-action history ending with a percept.*
3. *An* RL agent *(or simply an* agent*) is a function* $\pi : (\mathcal{PA})^*\mathcal{P} \to \mathcal{A}$*. When* $\pi(s) = a$*, the intuition is that agent* π *would take action* a *in response to history* s*.*
4. *An* RL environment *(or simply an* environment*) is a function* $\mu : (\mathcal{PA})^* \to \mathcal{P}$*. When* $\mu(s) = (o, r)$*, the intuition is that, in response to the agent taking the last action in* s *(in response to the history preceding that action), the environment gives the agent reward* r *and the agent's view of the world is replaced by observation* o*. When* $\mu(\langle\rangle) = (o, r)$*, the intuition is that* o *is the agent's initial view of the world and* r *is a meaningless initial reward.*
5. *The* result of agent π interacting with environment μ *is the infinite sequence* $(p_0, a_0, p_1, a_1, \ldots)$ *where* $p_0 = \mu(\langle\rangle)$*,* $a_0 = \pi(\langle p_0\rangle)$*, each* $p_{i+1} = \mu(p_0, a_0, \ldots, p_i, a_i)$*, and each* $a_{i+1} = \pi(p_0, a_0, \ldots, p_i, a_i, p_{i+1})$*.*

Example 1. For example, suppose $\mathcal{O} = \mathcal{A} = \{0, 1\}$, i.e., every observation is a single binary digit and every action is a single binary digit. We can imagine an environment which transmits binary digit observations in order to encode a pseudo-randomly generated English-language arithmetic question, and then waits for the agent to use binary digit actions to encode an English-language response. When the agent finishes encoding the response, the environment rewards the agent accordingly and repeats the process with a new question. While each question is being transmitted by the environment, the environment also transmits rewards of 0, and lets the agent take actions (which the environment ignores), until the environment's question is transmitted. Then, while the agent is encoding its answer action-by-action, the environment responds with observations of 0 and rewards of 0. These dummy rewards, observations, and actions are included so that the whole interaction conforms to Definition 1. The resulting interaction might look something like the following (suitably encoded):

- Environment: What is $1 + 1$?
- Agent: *(Agent initially has no knowledge of environment and its actions appear random)* ygHw
- Environment: *(Gives reward* -1*.)* What is $5 + 2$?
- Agent: JpX
- Environment: *(Gives reward* -1*.)* What is $8 + 3$?
- *(...Millions of turns pass like this...)*
- Environment: *(Gives reward* -1*.)* What is $2 + 1$?
- Agent: *(For the first time, the agent gets the right answer, by dumb luck)* 3
- Environment: *(Gives reward* $+1$*.)* What is $9 + 2$?
- *(...Billions more turns pass; agent gradually figures out the environment...)*
- Environment: *(Gives reward* $+1$*.)* What is $6 + 3$?
- Agent: 9
- Environment: *(Gives reward* $+1$*.)* What is $1 + 5$?

– *(...Interaction continues forever, with the agent getting better and better, but still occasionally answering wrong on purpose in order to test whether there might be an even more rewarding way to respond to the environment...)*

There are many ways in which Definition 1 could be varied. For example, instead of interactions beginning with an initial percept, interactions could begin with an initial (blind) action from the agent. Agents and/or environments could be allowed to be non-deterministic functions (one would have to rigorously specify what exactly that means). Computability requirements could be placed on agents and/or environments. Either $\mathcal{O}$, $\mathcal{A}$, or both could be made infinite. Rewards could be further restricted or, going the other direction, could be allowed to come from some other number system besides $\mathbb{R}$. In more practical settings, agents and environments are often not mathematical functions, but rather, instances of agent-classes and environment-classes, respectively[2]. For example, RL is implemented this way in OpenAI Gym [7] and Stable Baselines3 [18]. There, agent-classes define action-methods which take an individual observation, rather than a whole history—but said action-method can refer to the agent's internal memory (which can include things like neural net weights), which internal memory may vary during an environmental interaction, so that despite only explicitly depending on the most recent observation, these action methods implicitly depend on a whole history. For additional variations on RL, see Table 1 in Silver et al's paper.

We could modify Example 1 so that instead of the environment asking the agent arithmetic questions, the environment instead plays chess against the agent, using observations to encode images of the chessboard and then letting the agent use actions to encode moves (perhaps punishing the agent for attempting illegal moves, and so on). A legal move by the agent results in a reward of 0 unless the game ends (via the agent's move or the environment's responding move), in which case the reward is 1, 0, or -1 depending whether the agent won, drew, or lost. After each game-ending turn, the board returns to its initial state and the interaction resumes as if a new game has begun. To maximise rewards,

[2] Practitioners often abuse language and refer to agent-classes as agents. For example, a Python programmer might write "from stable_baselines3 import DQN" and refer to the resulting DQN class as the deep Q learning "agent" when, in reality, that object does not itself act. Rather, it must be instantiated (with hyperparameters), and the *instance* then acts. Language is further abused: underlying an agent, there is typically a *model* or *policy* (e.g., a neural network and its weights); once trained using Reinforcement Learning, the model is often published alone, in which capacity it merely acts in response to observations, and no longer has any mechanism for learning from rewards or even accepting rewards as input. Practitioners sometimes abuse language and refer to such pretrained models as "RL" agents. Thus, one might say, "this camera is controlled by an RL agent", when in reality the camera is controlled by a model obtained by training an RL agent (an expensive one-time training investment done on a supercomputer so that the resulting model can be used on consumer-grade computers to control many cameras thereafter). The model itself is not the RL agent—the weaker computer running the model does not give the model rewards or punishments. These nuances cause no confusion in practice.

the agent basically must learn how to play chess. A good RL agent will gradually do so. The same good agent, confronted instead with a game of Backgammon or Go, would learn that too: a good agent is general-purpose, not depending on built-in domain knowledge of any particular environment.

2.1 Are humans RL agents?

The way we formalized RL agents (Definition 1), humans are not RL agents, because humans are not mathematical functions. But we would not be doing the question justice with such an answer. Humans are not graph vertices either, yet that does not prevent mathematical biologists from studying graphs in which humans are vertices. There are two ways humans might be considered as RL agents, which we will refer to as *synthetic* and *organic*.

1. (Synthetic) Humans could be considered in their capacity to perform in RL environments. In other words, the typical human could compete in an "RL tournament", like a chess tournament except instead of playing chess, competitors play various RL environments chosen secretly by the tournament hosts. It would not be too large of an abuse of language to identify a human with the agent she would act as if she were competing in such a tournament.
2. (Organic) One might try to consider reality itself to be an RL environment in which the human acts as an RL agent, receiving observations equal to the sum of all their sensory inputs, and receiving rewards in some physiological form, such as physical pleasure and pain.

Treating humans as RL agents synthetically seems fairly non-controversial (at least if humans are suitably idealized, e.g., assumed to live forever so as to be able to continue environmental interactions forever). Strictly speaking, one should be careful, since, for example, the same human might act differently in the RL tournament at different times in their life. Thus, it might be more proper to say that "at time t, such-and-such human would act as such-and-such RL agent (if transported, at time t, to a totally isolated room where they can no longer receive any other external stimulus that might change their behavior, to spend the rest of eternity choosing actions in response to rewards and observations displayed on a screen)". One should also be careful to specify certain caveats, e.g., that unconscious humans or newborn babies should not be considered to be RL agents in this way. Also, there might be some doubt about whether the human's actions in the RL tournament define a mathematical *function*, depending on questions concerning free will. But as we mentioned above, other formalizations of RL admit non-deterministic agents, which would apparently remove problems related to free will.

Humans being RL agents in the synthetic sense does not imply much about AGI. All it implies is that an AGI should be capable of performing in RL environments (it does not even imply that an AGI should necessarily perform *well* in said environments, unless one first argues that humans would perform well, which does not seem like a trivial assertion). In the same way, the fact that

humans can play chess does not imply much about AGI, except that AGI should be capable of playing chess.

An interesting question to consider is: assuming humans are RL agents in the synthetic sense, can humans design RL agents that are better than humans themselves are? If so, is there a way to incentivize humans to do exactly this within the RL framework itself, that is, is there an RL environment which would reward the human agent exactly for designing superhuman RL agents? Or, is it the case that humans are capable of designing superhuman RL agents, but their motivations for doing so must necessarily transcend what can be expressed in the RL framework?

If humans are RL agents in the organic sense, then that would seem to make Silver et al's conjecture trivially true. But that is a much trickier claim. Here are some of the problems involved:

- If humans are identified with their bodies, then Silver et al themselves rule out humans as organic RL agents, because they say: "The agent consists solely of the decision-making entity; anything outside of that entity (including its body, if it has one) is considered part of the environment" [23]. Certainly our bodies include our brains and all the parts thereof, as well as our nervous systems, our sense organs, and so on. So if humans are organically RL agents then apparently this would entail some sort of controversial dualistic metaphysics.
- The RL framework generally involves *one* agent interacting with the environment. Thus, if our shared reality is the environment, at most one of us is the agent. One could perhaps consider reality to be composed of many environments, one for each agent (the reality from that agent's point of view), or one could consider a multi-agent version of RL such as that in [10]. Either way, multiple humans are not RL agents in a common single-agent environment.
- It is not clear how rewards and observations work if humans are RL agents. Am I punished the instant I touch the hot stove, or is my punishment delayed while the information travels from my fingertips up to my brain?[3] Is it delayed while my brain processes and interpets it? See [24] for a discussion of intrinsic reward vs. external signals.

One can certainly idealize humans and treat them as RL agents, in the same way the physicist can assume a spherical cow. But considerable additional justification would be needed before one could jump from said idealization to the conclusion that sufficiently strong RL agents are automatically AGI.

We think it might shed light on the matter if we compare the situation to Newtonian physics. Authors frequently speak as if our universe is a model of Newtonian physics, but it is understood that this is merely an approximation. In the same way, it is often useful to speak of the human world as an RL environment. Silver et al do this, saying, for example:

[3] To quote Aristotle: "For if ... one were to stretch a covering or membrane over the skin, a sensation would still arise immediately on making contact; yet it is obvious that the sense-organ was not in this membrane" [6].

> "A good reinforcement learning agent could thus acquire behaviours ... in the course of learning to maximise reward in an environment, such as the human world..." [23]

But it is not clear that the human world literally is an RL environment. Certainly one can approximate the human world (through a particular human's point of view) as an RL environment. But care should be taken before committing to this as literal truth. Given that it were literal truth, we could immediately conclude that strong enough RL agents would manifest AGI. In the same way, given that Newtonian physics were literal truth, we might immediately conclude that the universe is Turing computable. But since the universe is *not* literally a model of Newtonian physics, proponents of a Turing computable universe would need to come up with some other argument. Likewise, if the human world is not literally an RL environment, then some further argument would be needed to prove that strong enough RL agents would necessarily manifest AGI. One could argue in favor of a literally Newtonian universe by pointing to concrete experiments whose outcomes are predicted by Newtonian physics. Likewise, one could argue in favor of a literally RL environment human world by pointing to 'reward being enough' for various aspects of intelligence, as Silver et al do. But no matter how many experiments Newtonian physics predicts, there would linger the question of whether there are other experiments we haven't thought of yet, where Newton would fail (and indeed there are: relativity or quantum theoretic experiments). Are there aspects of human intelligence that RL is not 'enough for'? We do not know, but in the next section we will highlight one possible such aspect of intelligence.

3 RL-solving intelligence

Much ingenuity has gone into the design of RL agents, from basic Q-learning agents [25] to cutting-edge agents like DQN [16] and PPO [20]. Designing these agents is certainly an intellectual task. If every intellectual task requires a certain aspect of intelligence, then that goes for designing RL agents too, and we refer to that aspect of intelligence as *RL-solving intelligence*.

If humans do not possess decent RL-solving intelligence, then, even if RL is a path to AGI, it is not a practical path for humans. For the remainder of the paper, we assume humans do possess decent RL-solving intelligence. Presumably AGI should include all aspects of intelligence within human reach. Therefore, in particular, AGI should include decent RL-solving intelligence. Thus, if RL is to be a path to AGI, in particular RL would need to be a path to decent RL-solving intelligence.

In order for 'reward to be enough for RL-solving intelligence,' it seems there would need to be environments that reward RL agents for designing RL agents[4].

[4] One might object that there could be environments which reward some other behavior, which behavior requires RL-agent-design as an intermediate step, rather than rewarding RL-agent-design on its own. But how could we know this other behavior

And thus, any sufficiently good RL agent, when interacting with these environments, should eventually learn to design RL agents. Are such environments possible?

Without some sort of self-referential ouroboros argument, it seems that the question 'Can RL learn RL?' is a difficult obstacle if we want to assure ourselves that RL can lead to AGI. Proponents are obliged to show, for example, that RL agents can learn chess. But as soon as they do that, they themselves replace one obligation with another: since RL agents can learn chess, RL agents should be able to design agents who can learn chess[5]. If proponents demonstrate *that*, they incur an even *worse* obligation: RL agents should be able to design agents who can design agents who can learn chess. If proponents demonstrate *that*, they oblige RL agents to design agents who can design agents who can design agents who can learn chess. Trying to prove that RL agents are a path to AGI is an endless task if one merely attacks individual aspects of intelligence one at a time. To prove it would require short-circuiting the process somehow. In the next section we will consider one way the process might be short-circuited.

4 Performance measurement and incentivizing RL-agent design

We have argued above that if RL is to be a path to AGI then, in particular, since AGI should include RL-solving intelligence, RL should be a path to RL-solving intelligence. In other words, if RL is a path to AGI, then it should be possible to use RL to design good RL agents. In this section, we will consider one possible strategy for doing exactly this.

At a high level, we can imagine designing an environment in which an agent is incentivized to design child agents. The problem is, how do we incentivize the agent to design *good* child agents? If we merely reward the parent agent for designing child agents, with no regard for how good those children are, then the parent will be incentivized to churn out simple child agents in order to get rewarded quickly. If only we had a way of measuring how good the child agent was, we could use that measurement to decide how to reward the parent: if the parent designs a child with goodness 5, then give the parent a reward of +5; if the parent designs a child with goodness 999, then give the parent a reward of +999. This line of thinking leads to the following definitions.

---

requires RL-agent-design as intermediate step? Maybe a smart enough RL agent would figure out a way to avoid the intermediate step—just as RL agents can learn to exploit video-game bugs, or invent unanticipated new Go strategies, or just as image classifiers can learn to associate rulers with malignant tumors [17]. Thus, to be confident that an RL environment can incentivise RL-agent-design, it seems necessary that there be an environment that directly rewards RL-agent-design as primary objective, not merely rewarding some other behavior that requires RL-agent-design as intermediate step.

[5] Foreshadowed by [15].

Definition 2. *By an* RL-agent measure*, we mean a function f which takes as input (an encoding of) an RL agent π, and outputs a number.*

Definition 3. *For each RL-agent measure f, let M_f be the environment which outputs rewards and observations according to the following instructions (suitably encoded as in Example 1).*

1. *Generate a pseudo-random number k.*
2. *Prompt the parent agent to spend k actions encoding a child and a mathematical proof*[6] *that that child is an agent. For example, output observations which encode the message: "Please use k keystrokes to design and prove a child RL agent".*
3. *Using f, measure the child agent's goodness (let the measure be -1 if the parent agent did not encode a child and a proof that the child is an agent).*
4. *Give the parent agent the measurement from line 3 as a reward.*
5. *Goto 1.*

Along the same lines as Example 1, when an agent interacts with M_f, the interaction might look something like the following:

- Environment: Please use $k = 17$ keystrokes to design and prove a child RL agent.
- Agent: jKr WwZmk5pk lqwE
- Environment: *(Gives -1 reward.)* Please use $k = 33$ keystrokes to design and prove a child RL agent.
- Agent: mlmWqq9Fg31x rRjNMkqulpio m jMy j
- Environment: *(Gives -1 reward.)* Please use $k = 29$ keystrokes to design and prove a child RL agent.
- *(...Many turns pass...)*
- Environment: *(Gives -1 reward.)* Please use $k = 107$ keystrokes to design and prove a child RL agent.
- Agent: *(For the first time ever, agent gives a valid answer by dumb luck)* Let A be the agent that always takes action 0. Proof: Constant functions don't get stuck in infinite loops.
- Environment: *(Gives $f(A) = 0.000000000013$ reward.)* Please use $k = 981$ keystrokes to design and prove a child RL agent.
- *(...And so on forever, agent gradually learning the environment...)*

Remark 1. The reason for the k in Definition 3 is to force the agent to design child agents that maximize f. If we placed no requirement on how many actions the agent may take encoding a submission, then, depending on f, the agent might learn to spam simple agents for quick rewards. For example, if it is possible to encode a child π (and proof of its agenthood) using 10 actions, with $f(\pi) = 1$, and if all other children π' had $f(\pi') \leq 2$, and if it requires at least 100 actions to design any child agent π' with $f(\pi') > 1$, then the agent interacting with M_f would be incentivized to repeatedly encode π, because quick rewards of $+1$ are better than rewards of at most 2 coming ten times more slowly.

[6] We assume some fixed background proof system such as ZFC or Peano Arithmetic.

Remark 2. The reason for the proofs in Definition 3 is that in Definition 2 we did not place any constraints on what happens if f is applied to an input that does not encode an agent. Such constraints would make f non-computable since, by Rice's Theorem, there is no procedure for determining whether a given source-code is indeed the source-code of an RL agent. Without such constraints, there is the danger that line 3 of Definition 3 would induce an infinite loop. For example, $f(\pi)$ might be the result of measuring π's performance on various benchmarks. If π is a source-code of an agent-like function which sometimes gets stuck in infinite loops (and is thus not a genuine agent), then $f(\pi)$ might get stuck in an infinite loop if one of those benchmarks causes π to get stuck in an infinite loop. Thus, when using f to define an environment, one must take care only to apply f to genuine agents. (We do have some evidence that RL agents can learn to write mathematical proofs: see [12].)

Now, if f accurately measures how good an RL agent is, then it would seem that M_f incentivizes RL agents to produce good RL agents. For, in that case, the parent agent interacting with M_f would be rewarded based on the goodness of the child agents that it designs. If we could come up with an f which accurately measures how good an RL agent is, then we could run some good RL agents (such as DQN or PPO, assuming those are good RL agents) on M_f and see whether they eventually produce good children. If they do, that would be evidence in favor of Silver et al's conjecture.

Remark 3. An RL-measure f of Definition 2 gives a single numerical measurement to an agent. Generally speaking, any given agent will perform well in some environments and poorly in others. Thus, if f measures how well the agent performs, it evidently must do so in an aggregate sense: performance aggregated across all environments (or over some subset of environments of interest to us). Thus, an agent being good, as measured by f, does not automatically imply the agent performs well at M_f (in the same way that a candidate winning a high percentage of votes does not imply the candidate necessarily wins such-and-such individual's vote). Silver et al's conjecture would, in a sense, be trivially true if good aggregate performance implied good performance at M_f.

Are there any RL-agent measures f which accurately measure how good an RL agent is? What does that even mean? Silver et al do not offer any such measure in their paper, which is disappointing since, without such an f, it is hard for us to understand exactly what they mean when they speak of "sufficiently powerful reinforcement learning agents": what does it mean for an RL agent to be "sufficiently powerful"?

Various measures have been proposed by other authors besides Silver et al. Probably the best known is the Legg-Hutter universal intelligence measure [13]. The Legg-Hutter universal intelligence measure is, unfortunately, non-computable. Legg and Veness describe [14] a computable approximation for the Legg-Hutter universal intelligence measure. Building off of Legg and Hutter's work, Hernández-Orallo and Dowe propose [9] additional measures. These authors have given informal arguments that their proposed measures capture the

aggregate performance of RL agents (which they describe as the "intelligence" of RL agents), but it is not clear whether such aggregate performance is what Silver et al have in mind when they speak of "sufficiently powerful reinforcement learning agents". In any case, it would be interesting to take some of these proposed measures as our f and see what happens when we run state-of-the-art RL agents in the resulting environment M_f. It would be remarkable if, by doing so, and taking some of the resulting child agents, we found those resulting child agents to be good: if so, then by automating the design of those children, we would have succeeded at automating AI research, in a sense. And if said children turned out to be even better than the state-of-the-art RL agents which we used to design them, that would be most astonishing, maybe even the beginning of the singularity. But for reasons outside the scope of this paper, we are skeptical that such dramatic success would occur.

5 Discussion

We have argued that before we can conclude that RL is a direct path to AGI (as Silver et al conjecture it is), we ought first to establish that RL is a direct path to RL-solving intelligence. In this section, we discuss some anticipated objections to this thesis. We also anticipate and argue against some anticipated arguments that Silver et al's conjecture is trivial.

5.1 Agent-design is too complicated or expensive of a problem

Silver et al write [23]:

> The agent system α is limited by practical constraints to a bounded set [19]. The agent has limited capacity determined by its machinery (for example, limited memory in a computer or limited neurons in a brain). The agent and environment systems execute in real-time. While the agent spends time computing its next action (e.g. producing no-op actions while deciding whether to run away from a lion), the environment system continues to process (e.g. the lion attacks). Thus, the reinforcement learning problem represents a practical problem, as faced by natural and artificial intelligence, rather than a theoretical abstraction that ignores computational limitations.

One might argue that RL agent-design is too sophisticated and does not fall within the above constraints. Is RL agent-design merely a theoretical abstraction that ignores computational limitations? If so, does that absolve Silver et al's conjecture from requiring that RL lead to RL-solving intelligence? Well, this touches on the nature of what exactly AGI is. Is AGI merely required to include those aspects of intelligence which humans can reach while in a panicked state in front of a lion? We would argue the answer is "no". In our opinion, AGI should include whatever humans are capable of, whether those humans are panicking in front of a lion, leisurely enjoying a sabbatical year at a research lab, or even

collaborating in a huge elite well-funded team assisted by state-of-the-art supercomputers. If humans are capable of designing good RL agents (even if it requires a huge collaborative effort and scaled up cloud computing), then AGI should also be capable of designing good RL agents. The AGI might require access to similar resources as the human RL researchers have access to, and if good RL agent design requires n collaborating humans then maybe it requires n collaborating AGIs as well (as foreshadowed in [4]). But AGI should certainly be capable of creating good RL agents, if humans are capable of doing so. And if RL is a path to AGI then that means RL should be capable of designing good RL agents.

5.2 What about evolution?

A critic might argue that the process of human evolution has been an instance of RL, implying that RL suffices for human intelligence. But evolution does not directly have anything to do with rewards. Rather, evolution is about the natural selection of random mutations. An organism with a mutation unsuitable for its environment is less viable, so such mutations tend to be weeded out. An organism with a mutation beneficial for its environment is more viable, so such mutations tend to proliferate. Nowhere in this process does evolution punish or reward the organisms in question for their behavior or for any other reason. We might sometimes abuse language and speak as if evolution is a personified entity that gives an organism an offspring as a "reward" or gives an organism death as a "punishment". But this is only a manner of speaking: evolution does not literally walk around handing out rewards and punishments.

It is tempting to try to measure the fitness of an organism using some sort of fitness function (e.g., the number of the organism's children, or other similar functions proposed in [24]). If such a fitness function accurately captured the process of evolution, we might derive an RL-style reward function from it (e.g., the organism gets +1 reward whenever it has a child). A simplistic fitness function like the number of children an organism has in its life does not accurately capture the process of evolution, because an organism can have many children and yet still be unfit, if all those children are unfit. A more accurate fitness function would be inherently self-referential: the fitness of an organism would depend on the fitness of its children and later descendants.

For example, suppose a mutation increases both fertility and heat susceptibility. Initially, the mutant would reproduce faster, and its children would reproduce faster, and their children, if heat waves were rare enough. Its descendants might enjoy greater fertility for many generations. But if the next heat wave kills all those descendants, then the original organism was not more fit after all.

Examples like the above motivate us to ask questions like:

- Which is more viable, the organism with 20 weak children or the organism with 3 strong children?
- Which is more indicative of viability: having one's 100th child, or having one's first great-great-great-great-grandchild?

These questions seem open-ended, and we doubt there is a canonical way to answer them. Thus, even if we had knowledge of the distant future descendants of a currently-living organism, it would still be nontrivial to aggregate that knowledge into a single fitness number. Turning that hypothetical aggregate number into an RL reward-signal is even less realistic. Thus, we doubt evolution fits in the RL framework.

5.3 Just pick an agent and incentivise its design

Let k be the source-code of some RL agent. Can we cheat in the following way? Design an RL environment in which agents are rewarded for typing k, verbatim. Any time an agent differs from typing k, the agent is punished and forced to start over. This would trivially incentivise agents to type k (and hence, the objection argues, to design the agent with code k).

The problem with this is that the proposed cheating environment does not actually incentivize any kind of creativity, ingenuity, or any other aspects of intelligence that go into RL agent design. Likewise, we would not say that an environment teaches an agent to play chess if the environment merely teaches the agent to use a particular fixed chess-strategy built into the environment. Thus the objection is invalid. What the objection does show, however, is that some care would be needed in order to mathematically formalise what it means for an RL environment to incentivise RL agents to design RL agents.

5.4 Incentivise the agent to type its own source-code

It is interesting to consider whether an environment could incentivise an agent to type its own source-code. Arguably, such an environment would indeed incentivise RL agents to design RL agents, and in fact to do so in a particularly elegant way, as if a human were to invent an AGI through a process of introspection culminating in the human writing her own source code.

We tentatively opine that such environments, unfortunately, do not exist. The reason for our opinion is as follows. There seem to be epistemological limits to how well an RL agent can possibly know[7] its own source-code. For example, suppose an RL agent has source-code k. We could place the agent in an environment which, on every turn, displays a message saying[8]: "Please act differently than how the agent with source-code k would act in response to this action-observation history; you will be rewarded for doing so, and punished if you disobey." The agent would be logically unable to comply with the request, because the agent has source-code k and must therefore act accordingly even if it tries not to. The environment could even augment observations with additional info such as, e.g., "On the previous input, the agent with source-code k only required 84926 steps to halt," which would enable the agent to verify that the

[7] Here we use the word "know" in the sense of "act as if it knows". This is similar to how knowledge is treated in [5].

[8] This environment has similarities to Yampolskiy's impossible "Disobey!" [26].

environment has been telling the truth so far (the agent would need this additional info in order to reliably verify the environment's previous warnings, due to the Halting Problem). Thus, the agent must be ignorant of its own source-code or of its own agenthood. For if it knew both, then it could infer: "I can safely run k in order to compute how the environment wants me to act—k will not get stuck in an infinite loop, because if it did, k would not be an agent, but I know I am k and I know I am an agent so I know k is an agent." This is an RL version of a more general epistemic limitation on knowing agents, that a knowing agent can know its own truthfulness or know its own code, but not both [1] [2] [3]. For this reason I opine that an environment cannot incentivize RL agents to type their own source codes. Of course, more work would be needed to make these informal speculations rigorous.

5.5 RL doesn't need to directly solve RL, it only needs to help us solve RL

One might object that it is not necessary for RL to directly solve RL in the sense of there being an environment which incentivises RL agents to design RL agents. For example, maybe the development of sufficiently powerful RL agents would allow us to develop a new programming language (or a new brain-computer interface mechanism, or a new type of electrode, or a more efficient CPU model, etc.) which would help us achieve AGI.

We do not deny that RL could lead to advancements like those listed above, nor that such advancements could help us achieve AGI. But we do not think it would be appropriate to say that in that case RL "directly" lead to AGI. In the same way, the inventor of the sail is not directly credited with the discovery of America. If one were to claim that by leading to advancements like the above, RL would directly lead to AGI, then, by the same logic, one could claim that, e.g., 'a word processor is enough'. Or, 'Turing machines are enough,' or, 'binary is enough,' or even, by a famous result due in part to this workshop's keynote speaker, 'Diophantine equations are enough' [8].

6 Conclusion

Silver et al proposed [23] that 'reward is enough,' and that sufficiently strong reinforcement learning (RL) agents offer a direct path to Artificial General Intelligence (AGI). This was motivated by arguing that various aspects of intelligence subserve the maximisation of reward. They conjectured that a sufficiently good RL agent should offer a direct path to AGI. We responded by asking 'Can RL learn RL?' and we discussed this question.

We pointed out that if humans have decent RL-solving intelligence (by which we mean the aspect of intelligence used to design RL agents), and if AGI is at least as intelligent as humans, then AGI should have decent RL-solving intelligence. Thus, if RL agents are to offer a direct path to AGI, then RL agents

should be able to learn to design RL agents. We discussed why this complicates the task of convincing ourselves that RL agents offer a path to AGI.

We discussed an environment M_f, depending on a function f which measures RL agents in some way, which incentivizes agents to design child agents so as to maximize the value of f on those child agents. Thus, if f measures how good an RL agent is, then M_f would incentivize RL agents to design good RL agents. We speculated about whether such an f is possible, and about what would happen if we ran state-of-the-art RL agents on the resulting M_f.

Acknowledgments

We gratefully acknowledge José Hernández-Orallo, Phil Maguire, and the reviewers for generous comments and feedback.

References

1. Aldini, A., Fano, V., Graziani, P.: Do the self-knowing machines dream of knowing their factivity? In: AIC. pp. 125–132 (2015)
2. Aldini, A., Fano, V., Graziani, P.: Theory of knowing machines: revisiting Gödel and the mechanistic thesis. In: International Conference on the History and Philosophy of Computing (2015)
3. Alexander, S.A.: A machine that knows its own code. Studia Logica **102**(3), 567–576 (2014)
4. Alexander, S.A.: AGI and the Knight-Darwin law: why idealized AGI reproduction requires collaboration. In: CAGI (2020)
5. Alexander, S.A.: Short-circuiting the definition of mathematical knowledge for an artificial general intelligence. In: CIFMA (2020)
6. Aristotle: On the soul. In: Barnes, J., et al. (eds.) The Complete Works of Aristotle. Princeton University Press (1984)
7. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: OpenAI gym. Preprint (2016)
8. Davis, M.: Hilbert's tenth problem is unsolvable. The American Mathematical Monthly **80**(3), 233–269 (1973)
9. Hernández-Orallo, J., Dowe, D.L.: Measuring universal intelligence: Towards an anytime intelligence test. Artificial Intelligence **174**(18), 1508–1539 (2010)
10. Hernández-Orallo, J., Dowe, D.L., Espana-Cubillo, S., Hernández-Lloreda, M.V., Insa-Cabrera, J.: On more realistic environment distributions for defining, evaluating and developing intelligence. In: CAGI (2011)
11. Hutter, M.: Universal artificial intelligence: Sequential decisions based on algorithmic probability. Springer (2004)
12. Kaliszyk, C., Urban, J., Michalewski, H., Olšák, M.: Reinforcement learning of theorem proving. In: NeurIPS (2018)
13. Legg, S., Hutter, M.: Universal intelligence: A definition of machine intelligence. Minds and machines **17**(4), 391–444 (2007)
14. Legg, S., Veness, J.: An approximation of the universal intelligence measure. In: Algorithmic Probability and Friends: Bayesian Prediction and Artificial Intelligence. Springer (2013)

15. Maguire, P., Moser, P., Maguire, R.: Are people smarter than machines? Croatian Journal of Philosophy **20**(1), 103–123 (2020)
16. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
17. Narla, A., Kuprel, B., Sarin, K., Novoa, R., Ko, J.: Automated classification of skin lesions: from pixels to practice. Journal of Investigative Dermatology **138**(10), 2108–2110 (2018)
18. Raffin, A., Hill, A., Ernestus, M., Gleave, A., Kanervisto, A., Dormann, N.: Stable baselines3. https://github.com/DLR-RM/stable-baselines3 (2019)
19. Russell, S.J., Subramanian, D.: Provably bounded-optimal agents. Journal of Artificial Intelligence Research **2**, 575–609 (1994)
20. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. Preprint (2017)
21. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: Mastering the game of Go with deep neural networks and tree search. Nature **529**(7587), 484–489 (2016)
22. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., Hassabis, D.: Mastering the game of Go without human knowledge. Nature **550**(7676), 354–359 (2017)
23. Silver, D., Singh, S., Precup, D., Sutton, R.: Reward is enough. Artificial Intelligence (2021)
24. Singh, S., Lewis, R.L., Barto, A.G., Sorg, J.: Intrinsically motivated reinforcement learning: An evolutionary perspective. IEEE Transactions on Autonomous Mental Development **2**(2), 70–82 (2010)
25. Watkins, C.: Learning from delayed rewards. Ph.D. thesis, Cambridge (1989)
26. Yampolskiy, R.: On controllability of artificial intelligence. Technical report (2020)