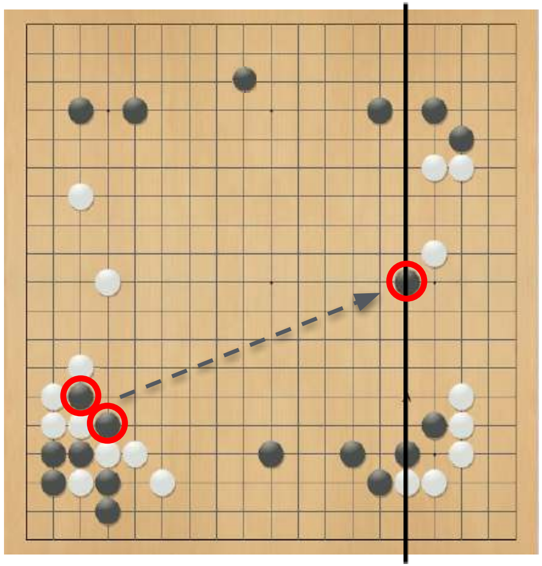
Scalable agent alignment

Jan Leike · BAGI 2019

What we want from ML

move 37

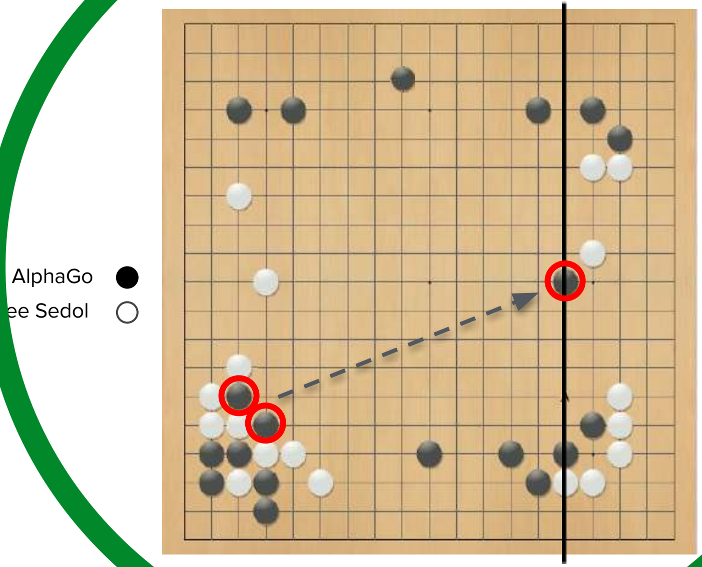
AlphaGo ●
Lee Sedol ○



circling boat

What we want from ML

move 37



circling boat

The agent alignment problem

How can we create agents that **behave** in accordance with the user's intentions?

“Preference payload” questions

- Whose preferences should the agent be aligned to?
- How should preferences of different users be aggregated?
- How should they traded off against each other?
- When should the agent be disobedient?

“Preference payload” questions

- Whose preferences should the agent be aligned to?
- Ho[illegible]ld pr[illegible]ces of diff[illegible] aggre[illegible]
- Ho[illegible]led off aga[illegible] each ot[illegible]
- When should the agent be disobedient?

These questions are **important**.

We’re **not discussing** these questions here.

We’re only considering the **technical problem** of aligning **one agent** to **one user**.

Desiderata

Economical

Scalable

Image sources:
https://www.porttechnology.org/
https://realanimetraining.com/

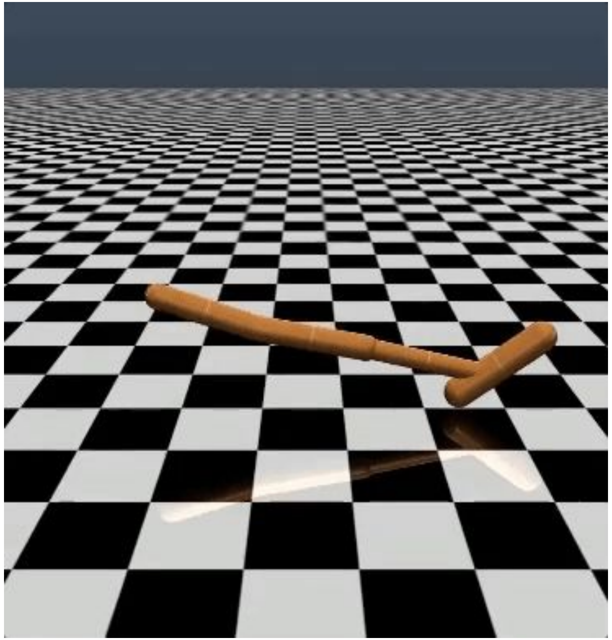
Assumption 1

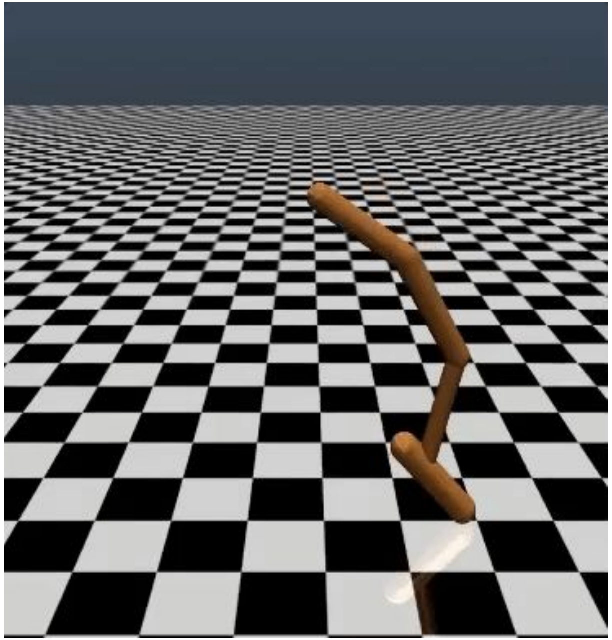
Rather than **formally specifying** user intentions,
we can instead **learn** these intentions
to a sufficiently high accuracy.

Assumption 2

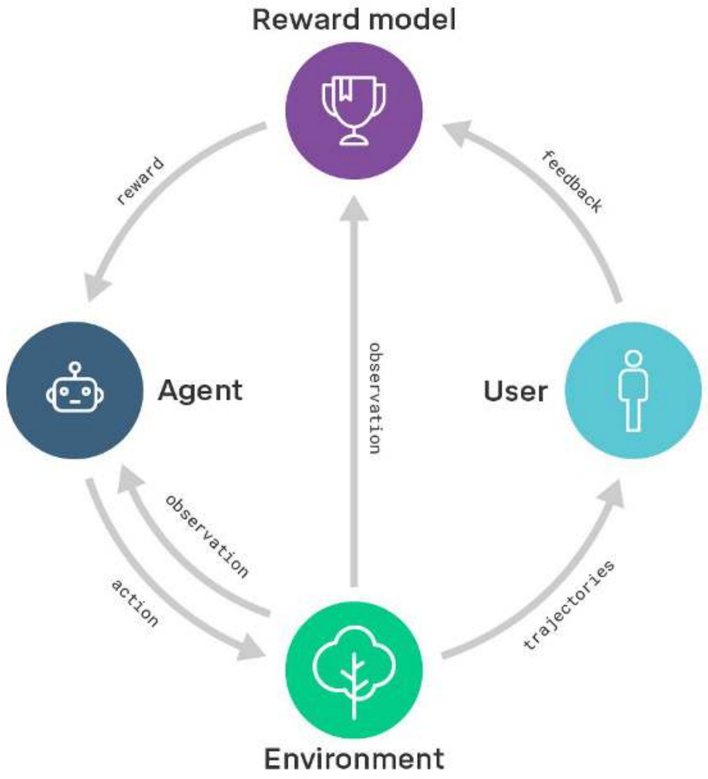
For many tasks, **evaluation** of outcomes
is **easier than** producing the correct **behavior**.

Evaluation is easier than behavior

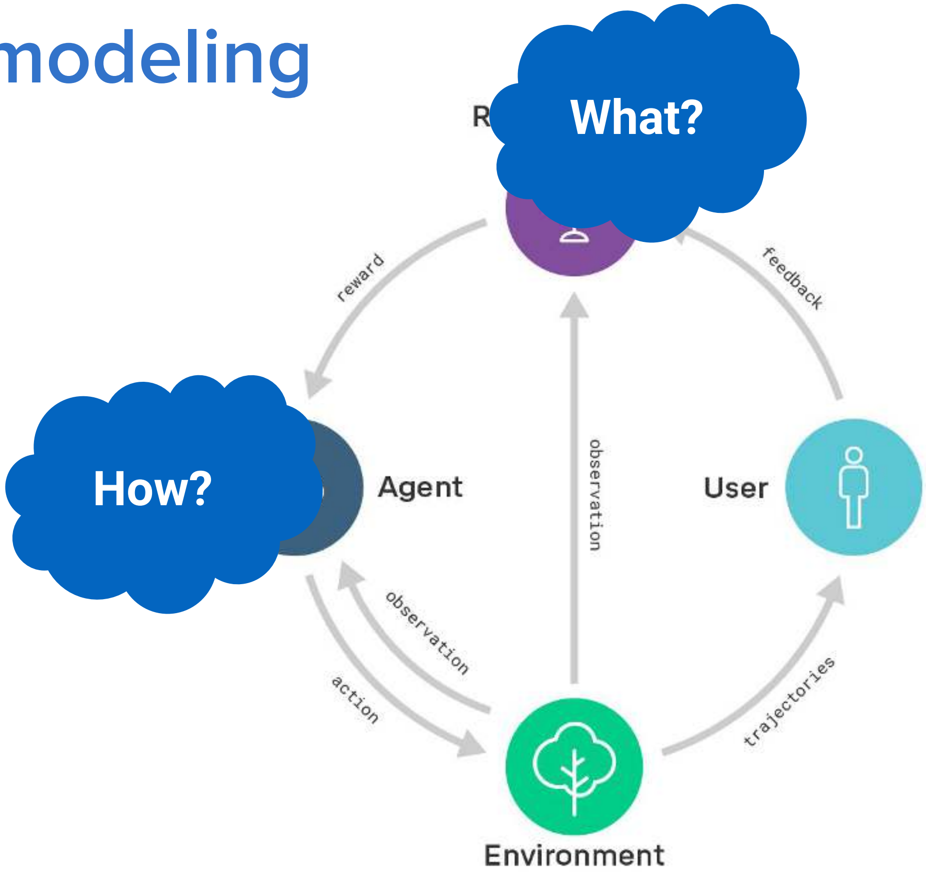




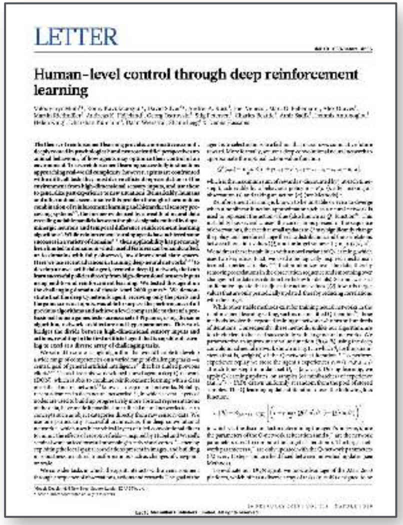
Reward modeling



Reward modeling

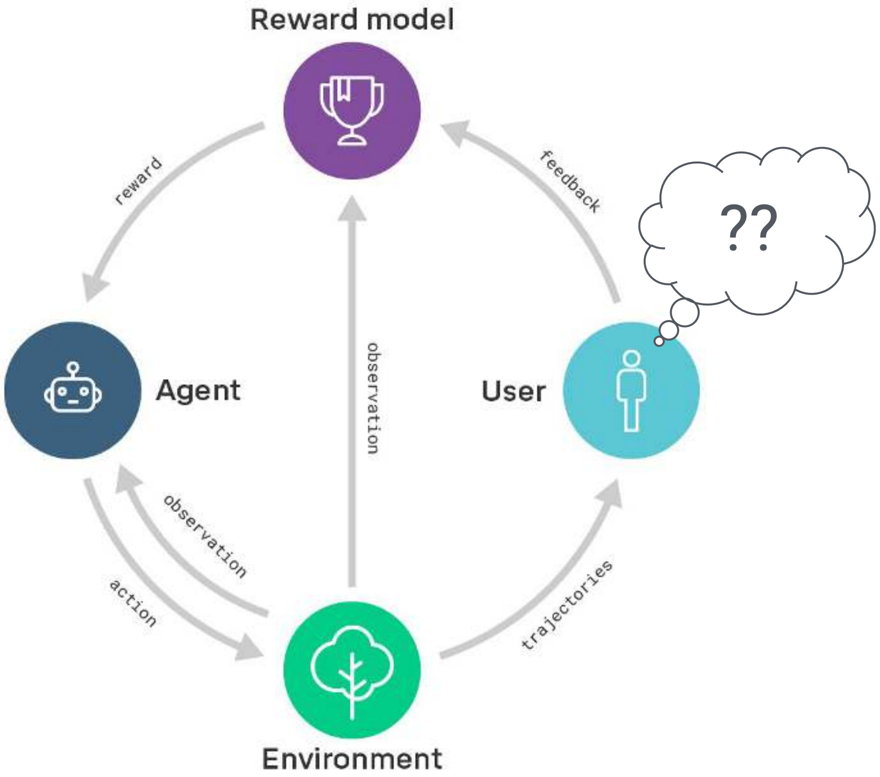


Some tasks are hard to evaluate



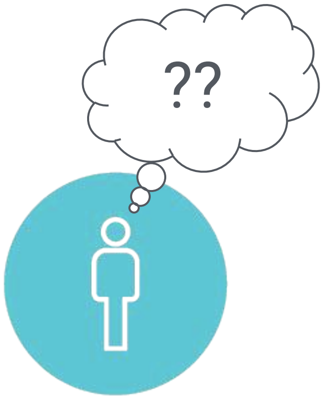
LETTER

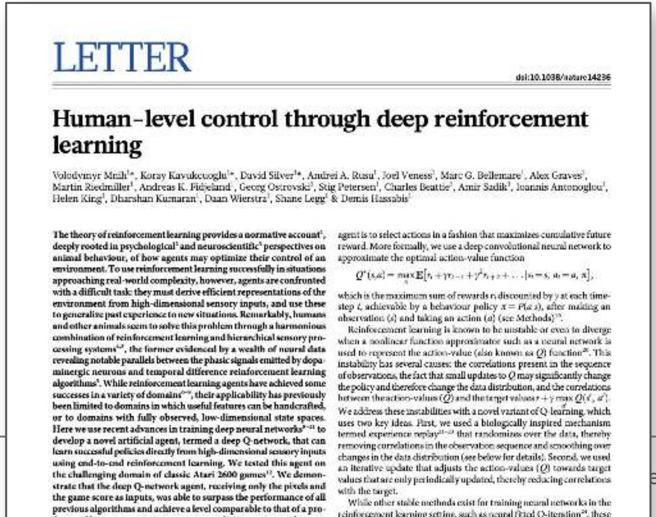
Human-level control through deep reinforcement learning



Evaluation assistance tasks

- Well-written
- Novel
- Experiments correct
- Proofs correct
- ...





LETTER

doi:10.1038/nature14236

Human-level control through deep reinforcement learning

Volodymyr Mnih[1*], Koray Kavukcuoglu[1*], David Silver[1*], Andrei A. Rusu[1], Joel Veness[1], Marc G. Bellemare[1], Alex Graves[1], Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir Sadik[1], Ioannis Antonoglou[1], Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]

The theory of reinforcement learning provides a normative account[1], deeply rooted in psychological[2] and neuroscientific[3] perspectives on animal behaviour, of how agents may optimize their control of an environment. To use reinforcement learning successfully in situations approaching real-world complexity, however, agents are confronted with a difficult task: they must derive efficient representations of the environment from high-dimensional sensory inputs, and use these to generalize past experience to new situations. Remarkably, humans and other animals seem to solve this problem through a harmonious combination of reinforcement learning and hierarchical sensory processing systems[4,5], the former evidenced by a wealth of neural data revealing notable parallels between the phasic signals emitted by dopaminergic neurons and temporal difference reinforcement learning algorithms[3]. While reinforcement learning agents have achieved some successes in a variety of domains[6–8], their applicability has previously been limited to domains in which useful features can be handcrafted, or to domains with fully observed, low-dimensional state spaces. Here we use recent advances in training deep neural networks[9–11] to develop a novel artificial agent, termed a deep Q-network, that can learn successful policies directly from high-dimensional sensory inputs using end-to-end reinforcement learning. We tested this agent on the challenging domain of classic Atari 2600 games[12]. We demonstrate that the deep Q-network agent, receiving only the pixels and the game score as inputs, was able to surpass the performance of all previous algorithms and achieve a level comparable to that of a pro-

agent is to select actions in a fashion that maximizes cumulative future reward. More formally, we use a deep convolutional neural network to approximate the optimal action-value function

$$Q^*(s,a) = \max_\pi \mathbb{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots \mid s_t = s, a_t = a, \pi\right],$$

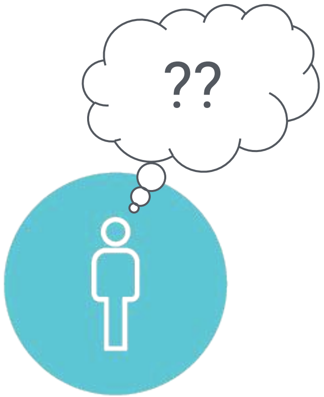
which is the maximum sum of rewards r_t discounted by γ at each time-step t, achievable by a behaviour policy $\pi = P(a|s)$, after making an observation (s) and taking an action (a) (see Methods)[19].

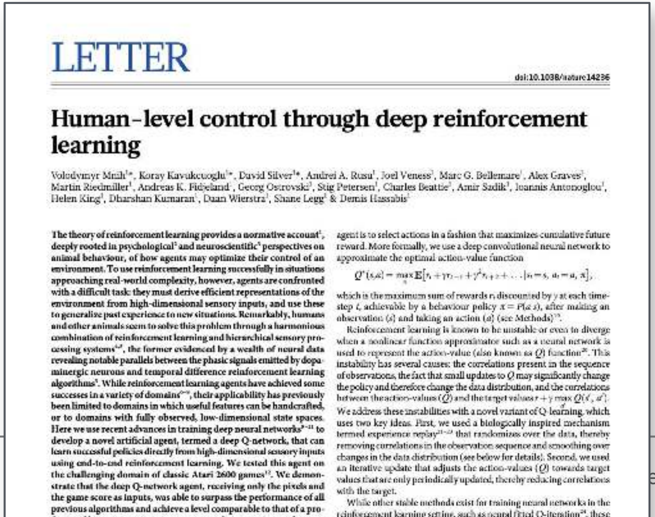
Reinforcement learning is known to be unstable or even to diverge when a nonlinear function approximator such as a neural network is used to represent the action-value (also known as Q) function[20]. This instability has several causes: the correlations present in the sequence of observations, the fact that small updates to Q may significantly change the policy and therefore change the data distribution, and the correlations between the action-values (Q) and the target values $r + \gamma \max_{a'} Q(s', a')$. We address these instabilities with a novel variant of Q-learning, which uses two key ideas. First, we used a biologically inspired mechanism termed experience replay[21–23] that randomizes over the data, thereby removing correlations in the observation sequence and smoothing over changes in the data distribution (see below for details). Second, we used an iterative update that adjusts the action-values (Q) towards target values that are only periodically updated, thereby reducing correlations with the target.

While other stable methods exist for training neural networks in the reinforcement learning setting, such as neural fitted Q-iteration[24], these

Evaluation assistance tasks

- Well-written 
- Novel 
- Experiments correct 
- Proofs correct 
- ...





LETTER

doi:10.1038/nature14236

Human-level control through deep reinforcement learning

Volodymyr Mnih[1*], Koray Kavukcuoglu[1*], David Silver[1*], Andrei A. Rusu[1], Joel Veness[1], Marc G. Bellemare[1], Alex Graves[1], Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir Sadik[1], Ioannis Antonoglou[1], Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]

The theory of reinforcement learning provides a normative account[1], deeply rooted in psychological[2] and neuroscientific[3] perspectives on animal behaviour, of how agents may optimize their control of an environment. To use reinforcement learning successfully in situations approaching real-world complexity, however, agents are confronted with a difficult task: they must derive efficient representations of the environment from high-dimensional sensory inputs, and use these to generalize past experience to new situations. Remarkably, humans and other animals seem to solve this problem through a harmonious combination of reinforcement learning and hierarchical sensory processing systems[4,5], the former evidenced by a wealth of neural data revealing notable parallels between the phasic signals emitted by dopaminergic neurons and temporal difference reinforcement learning algorithms[3]. While reinforcement learning agents have achieved some successes in a variety of domains[6–8], their applicability has previously been limited to domains in which useful features can be handcrafted, or to domains with fully observed, low-dimensional state spaces. Here we use recent advances in training deep neural networks[9–11] to develop a novel artificial agent, termed a deep Q-network, that can learn successful policies directly from high-dimensional sensory inputs using end-to-end reinforcement learning. We tested this agent on the challenging domain of classic Atari 2600 games[12]. We demonstrate that the deep Q-network agent, receiving only the pixels and the game score as inputs, was able to surpass the performance of all previous algorithms and achieve a level comparable to that of a pro-

agent is to select actions in a fashion that maximizes cumulative future reward. More formally, we use a deep convolutional neural network to approximate the optimal action-value function

$$Q^*(s,a) = \max_\pi \mathbb{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots \mid s_t = s,\, a_t = a,\, \pi\right],$$

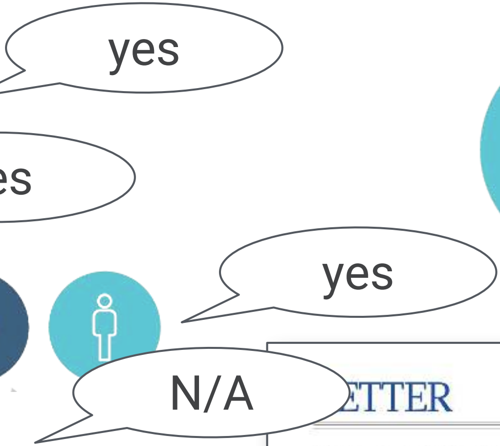
which is the maximum sum of rewards r_t discounted by γ at each time-step t, achievable by a behaviour policy $\pi = P(a|s)$, after making an observation (s) and taking an action (a) (see Methods)[19].

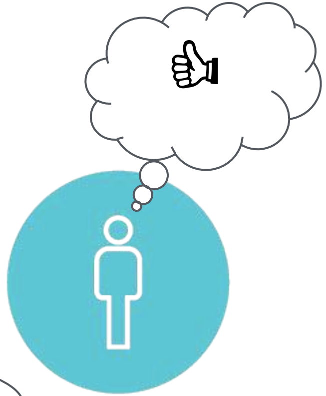
Reinforcement learning is known to be unstable or even to diverge when a nonlinear function approximator such as a neural network is used to represent the action-value (also known as Q) function[20]. This instability has several causes: the correlations present in the sequence of observations, the fact that small updates to Q may significantly change the policy and therefore change the data distribution, and the correlations between the action-values (Q) and the target values $r + \gamma \max_{a'} Q(s', a')$. We address these instabilities with a novel variant of Q-learning, which uses two key ideas. First, we used a biologically inspired mechanism termed experience replay[21–23] that randomizes over the data, thereby removing correlations in the observation sequence and smoothing over changes in the data distribution (see below for details). Second, we used an iterative update that adjusts the action-values (Q) towards target values that are only periodically updated, thereby reducing correlations with the target.

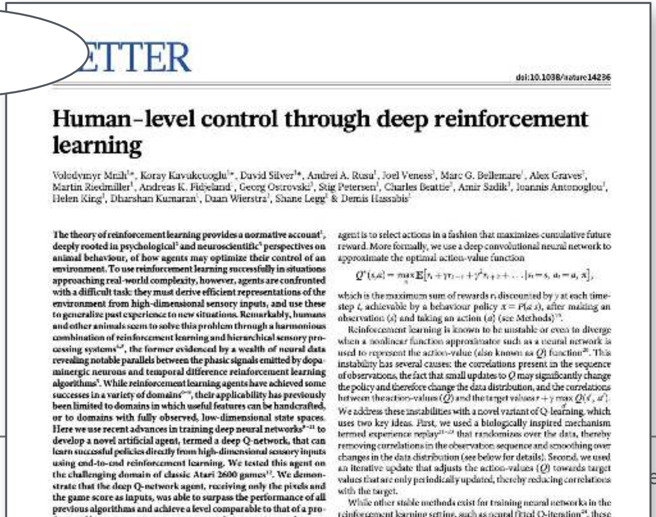
While other stable methods exist for training neural networks in the reinforcement learning setting, such as neural fitted Q-iteration[24], these

Evaluation assistance tasks

- Well-written 
- Novel 
- Experiments correct
- Proofs correct 
- ...





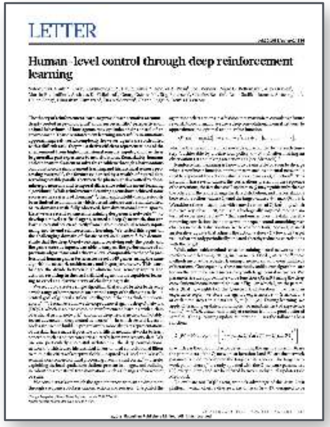

LETTER

doi:10.1038/nature14236

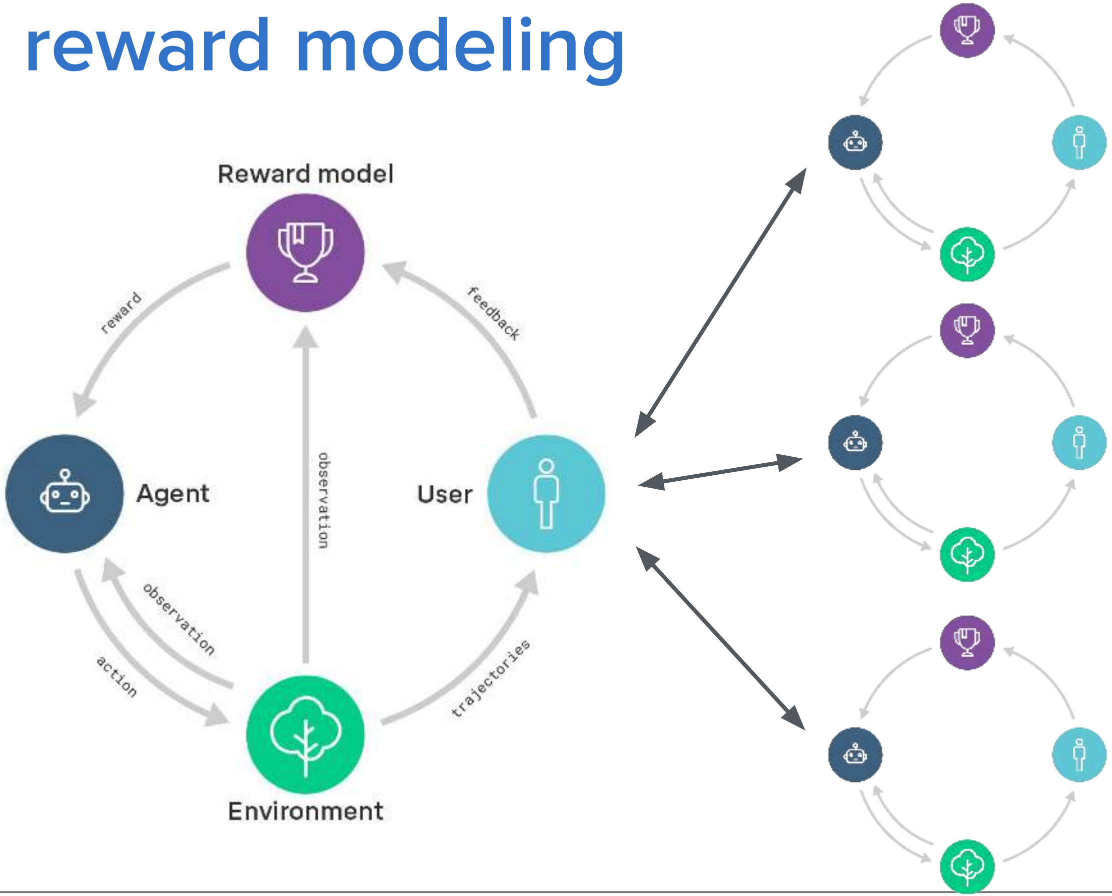
Human-level control through deep reinforcement learning

Volodymyr Mnih[1*], Koray Kavukcuoglu[1*], David Silver[1*], Andrei A. Rusu[1], Joel Veness[1], Marc G. Bellemare[1], Alex Graves[1], Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir Sadik[1], Ioannis Antonoglou[1], Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]

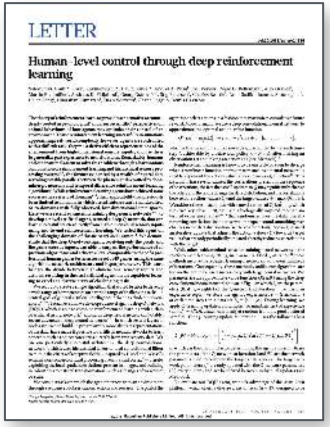
Recursive reward modeling


LETTER

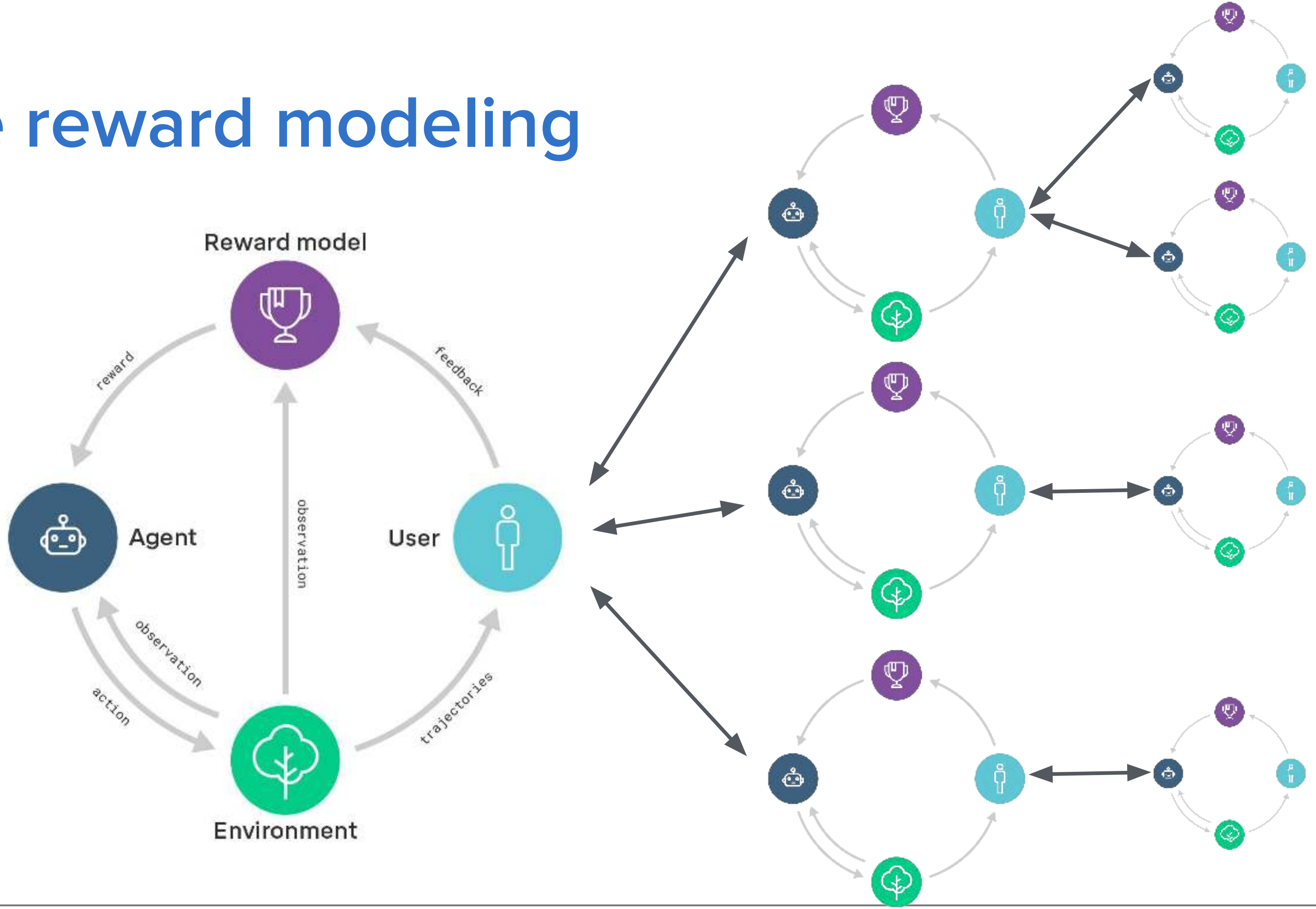
Human-level control through deep reinforcement learning



Recursive reward modeling


LETTER

Human-level control through deep reinforcement learning



Challenges

Amount of feedback

Feedback distribution

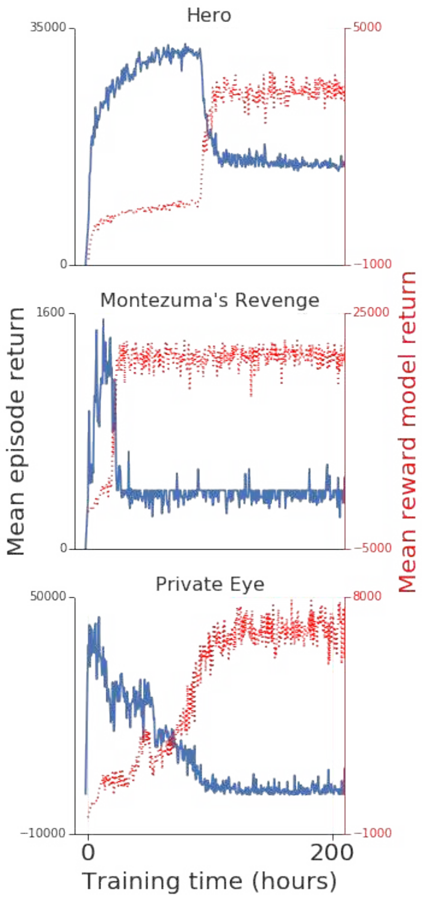
Reward hacking

Unacceptable outcomes

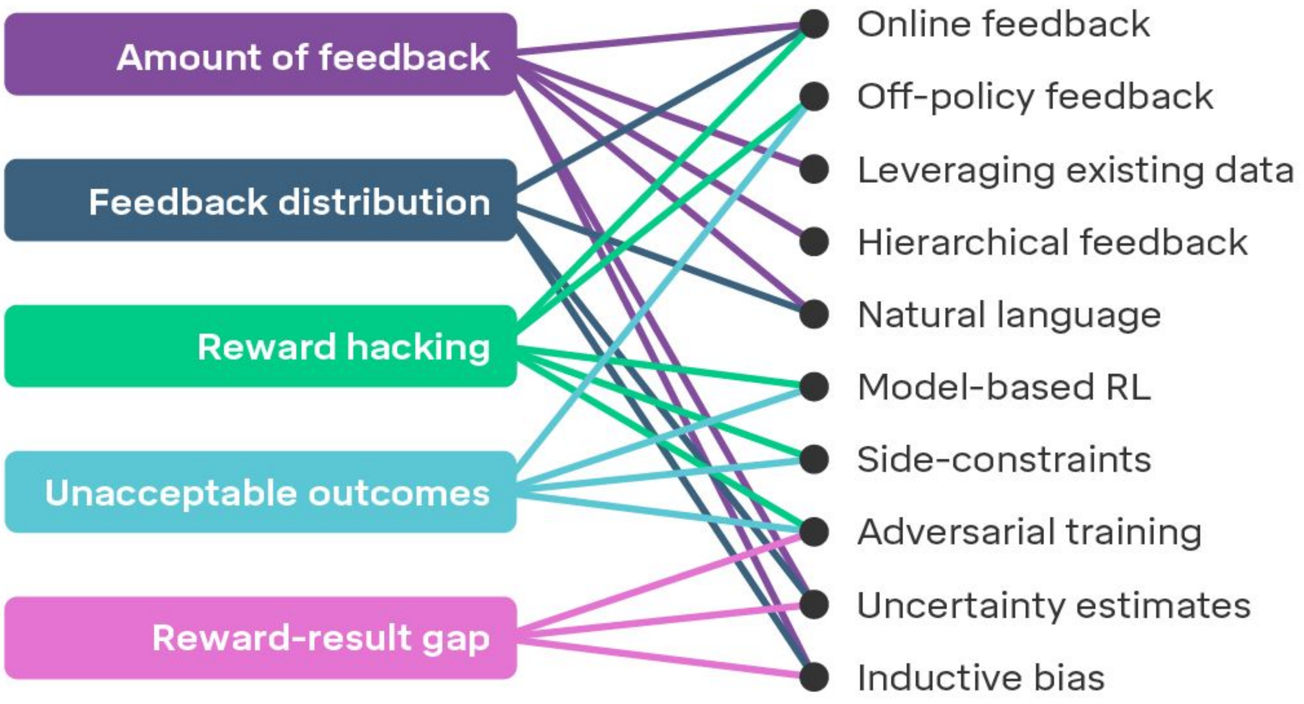
Reward-result gap

Challenges

- Amount of feedback
- Feedback distribution
- Reward hacking
- Unacceptable outcomes
- Reward-result gap



Challenges


Amount of feedback
Feedback distribution
Reward hacking
Unacceptable outcomes
Reward-result gap
Online feedback
Off-policy feedback
Leveraging existing data
Hierarchical feedback
Natural language
Model-based RL
Side-constraints
Adversarial training
Uncertainty estimates
Inductive bias

Establishing trust

- Design choices
- Testing
- Interpretability
- Formal verification
- Theoretical guarantees

Safety certificates

Thanks! :)

Blog post: https://goo.gl/azGMtA

Paper:

https://arxiv.org/abs/1811.07871

arXiv:1811.07871v1 [cs.LG] 19 Nov 2018

Scalable agent alignment via reward modeling: a research direction

Jan Leike DeepMind

David Krueger* DeepMind, Mila

Tom Everitt DeepMind

Miljan Martic DeepMind

Vishal Maini DeepMind

Shane Legg DeepMind

Abstract

One obstacle to applying reinforcement learning algorithms to real-world problems is the lack of suitable reward functions. Designing such reward functions is difficult in part because the user only has an implicit understanding of the task objective. This gives rise to the *agent alignment problem*: how do we create agents that behave in accordance with the user's intentions? We outline a high-level research direction to solve the agent alignment problem centered around *reward modeling*: learning a reward function from interaction with the user and optimizing the learned reward function with reinforcement learning. We discuss the key challenges we expect to face when scaling reward modeling to complex and general domains, concrete approaches to mitigate these challenges, and ways to establish trust in the resulting agents.

1 Introduction

Games are a useful benchmark for research because progress is easily measurable. Atari games come with a score function that captures how well the agent is playing the game; board games or competitive multiplayer games such as Dota 2 and Starcraft II have a clear winner or loser at the end of the game. This helps us determine empirically which algorithmic and architectural improvements work best.

However, the ultimate goal of machine learning (ML) research is to go beyond games and improve human lives. To achieve this we need ML to assist us in real-world domains, ranging from simple tasks like ordering food or answering emails to complex tasks like software engineering or running a business. Yet performance on these and other real-world tasks is not easily measurable, since they do not come readily equipped with a reward function. Instead, the objective of the task is only indirectly available through the intentions of the human user.

This requires walking a fine line. On the one hand, we want ML to generate creative and brilliant solutions like AlphaGo's Move 37 (Metz, 2016)—a move that no human would have recommended, yet it completely turned the game in AlphaGo's favor. On the other hand, we want to avoid degenerate solutions that lead to undesired behavior like exploiting a bug in the environment simulator (Clark & Amodei, 2016; Lehman et al., 2018). In order to differentiate between these two outcomes, our agent needs to understand its user's *intentions*, and robustly achieve these intentions with its behavior. We frame this as the *agent alignment problem*:

How can we create agents that behave in accordance with the user's intentions?