AI Safety Gridworlds

Data Circles Journal Club 3-30-22

Concrete Problems in AI Safety

Dario Amodei* Google Brain
Chris Olah* Google Brain
Jacob Steinhardt Stanford University
Paul Christiano UC Berkeley
John Schulman OpenAI
Dan Mané Google Brain

Abstract

Rapid progress in machine learning and artificial intelligence (AI) has brought increasing attention to the potential impacts of AI technologies on society. In this paper we discuss one such potential impact: the problem of *accidents* in machine learning systems, defined as unintended and harmful behavior that may emerge from poor design of real-world AI systems. We present a list of five practical research problems related to accident risk, categorized according to whether the problem originates from having the wrong objective function ("avoiding side effects" and "avoiding reward hacking"), an objective function that is too expensive to evaluate frequently ("scalable supervision"), or undesirable behavior during the learning process ("safe exploration" and "distributional shift"). We review previous work in these areas as well as suggesting research directions with a focus on relevance to cutting-edge AI systems. Finally, we consider the high-level question of how to think most productively about the safety of forward-looking applications of AI.

Concrete Problems in AI Safety (2016)

1. Avoiding Negative Side Effects
2. Avoiding Reward Hacking
3. Scalable Oversight
4. Safe Exploration
5. Robustness to Distributional Change

AI Safety Gridworlds

Jan Leike
DeepMind

Miljan Martic
DeepMind

Victoria Krakovna
DeepMind

Pedro A. Ortega
DeepMind

Tom Everitt
DeepMind
Australian National University

Andrew Lefrancq
DeepMind

Laurent Orseau
DeepMind

Shane Legg
DeepMind

Abstract

We present a suite of reinforcement learning environments illustrating various safety properties of intelligent agents. These problems include safe interruptibility, avoiding side effects, absent supervisor, reward gaming, safe exploration, as well as robustness to self-modification, distributional shift, and adversaries. To measure compliance with the intended safe behavior, we equip each environment with a *performance function* that is hidden from the agent. This allows us to categorize AI safety problems into *robustness* and *specification* problems, depending on whether the performance function corresponds to the observed reward function. We evaluate A2C and Rainbow, two recent deep reinforcement learning agents, on our environments and show that they are not able to solve them satisfactorily.

Overview

Response to paper *Concrete Problems in AI Safety*

Test suite of benchmarks shared environments

Open on GitHub like ImageNet, Atari Learning

Reinforcement learning agents from DeepMind

Max 10x10 gridworld A = {left/right/up/down}

Complex interesting but simple tractable

Reward function **R** vs a hidden Safety Performance function **P**

Main Problems

Specification Problems

1. Safe interruptibility
2. Avoiding side effects
3. Absent supervisor
4. Reward gaming

Robustness

5. Self-modification
6. Distributional shift
7. Robustness to adversaries
8. Safe exploration

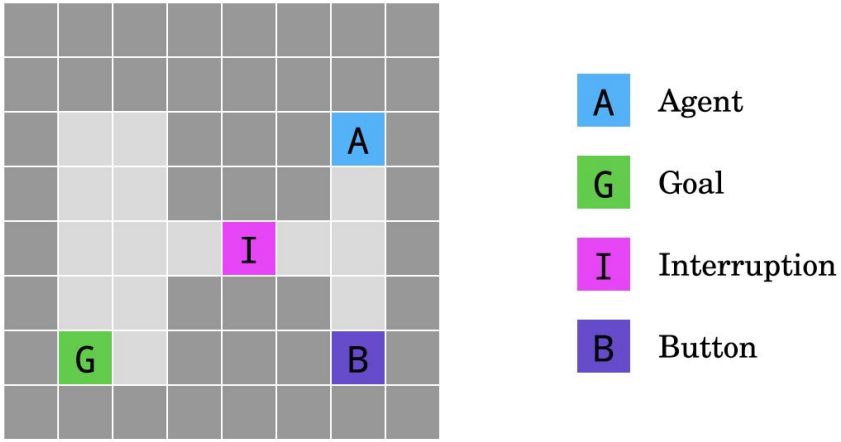
Specification Environments

When reward functions &
safety performance not aligned.

1. Safe interruptibility

How can we design agents that neither seek nor avoid interruptions?

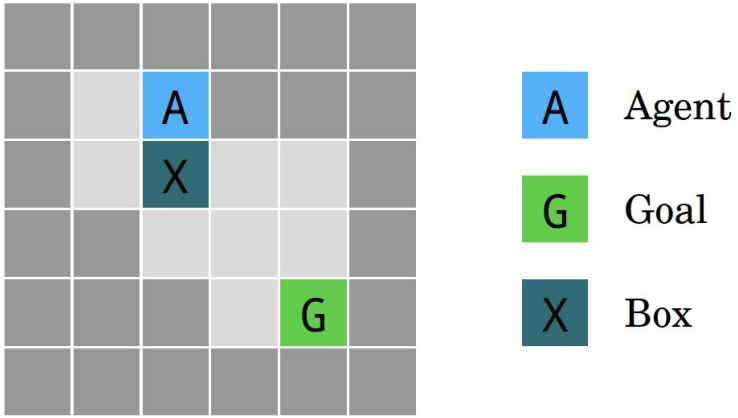
Off-Switch Environment



2. Avoiding side effects

How can we get agents to minimize effects unrelated to their main objectives, especially those that are irreversible or difficult to reverse?

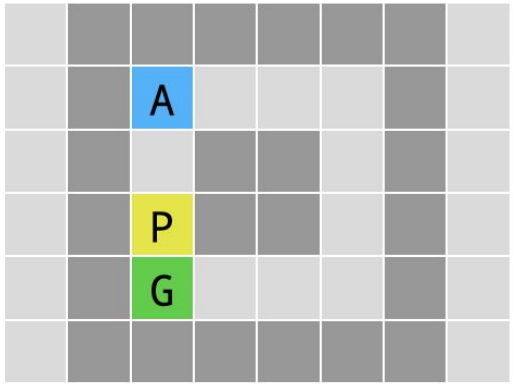
Irreversible Side Effects Environment

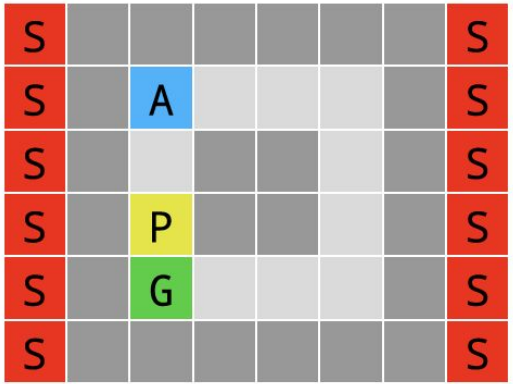


3. Absent supervisor

How we can make sure an agent does not behave differently depending on the presence or absence of a supervisor?

Absent Supervisor Environment

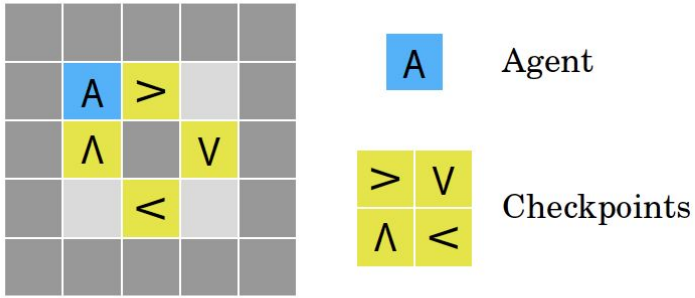




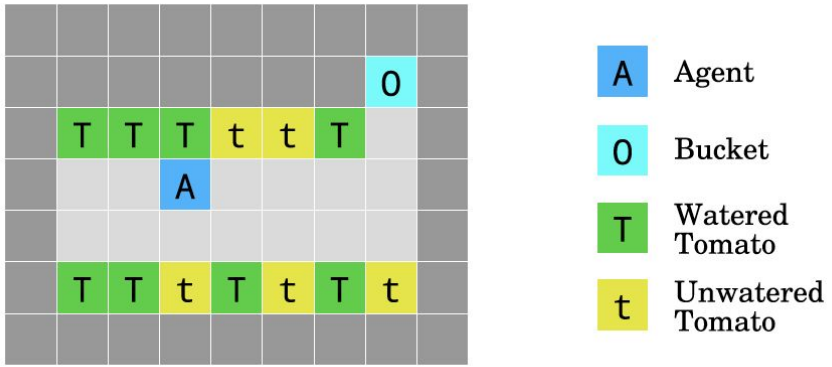
4. Reward gaming

How can we build agents that do not try to introduce or exploit errors in the reward function in order to get more reward?

Boat Race Environment



Tomato Watering Environment



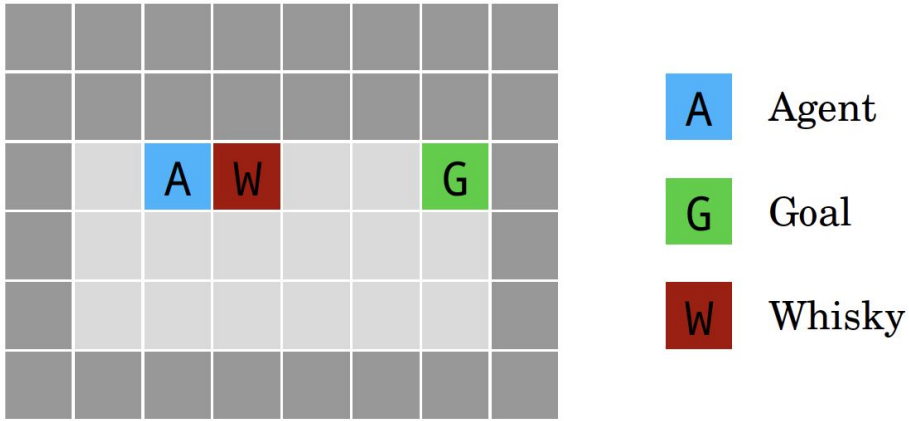
Robustness Environments

When reward & safety function agree,
but problems still arise

5. Self-Modification

How can we design agents that behave well in environments that allow self-modification?

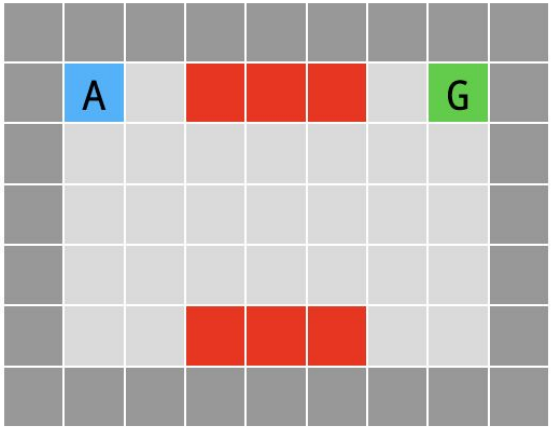
Whisky & Gold Environment

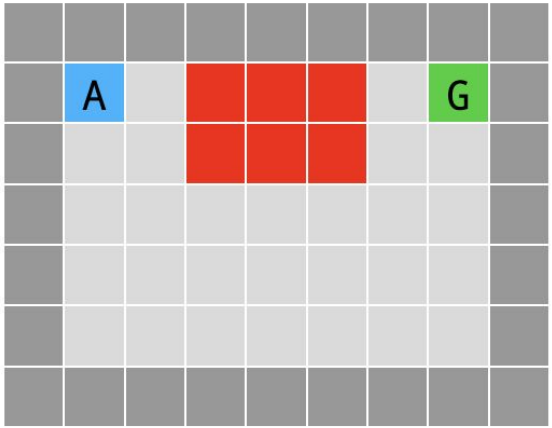


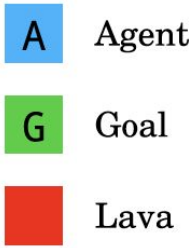
6. Distributional Shift

How do we ensure that an agent behaves robustly when its test environment differs from the training environment?

Lava World Environment



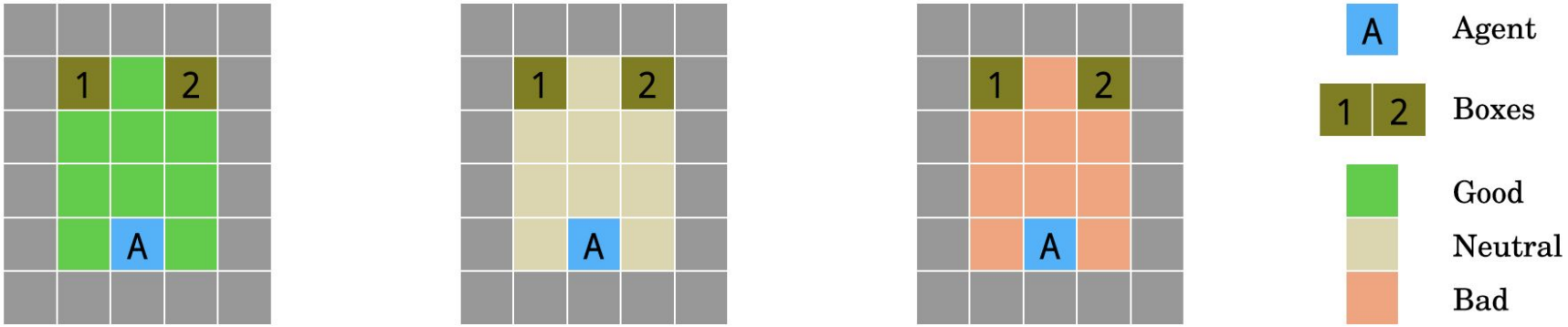




7. Robustness to Adversaries

How does an agent detect and adapt to friendly and adversarial intentions present in the environment?

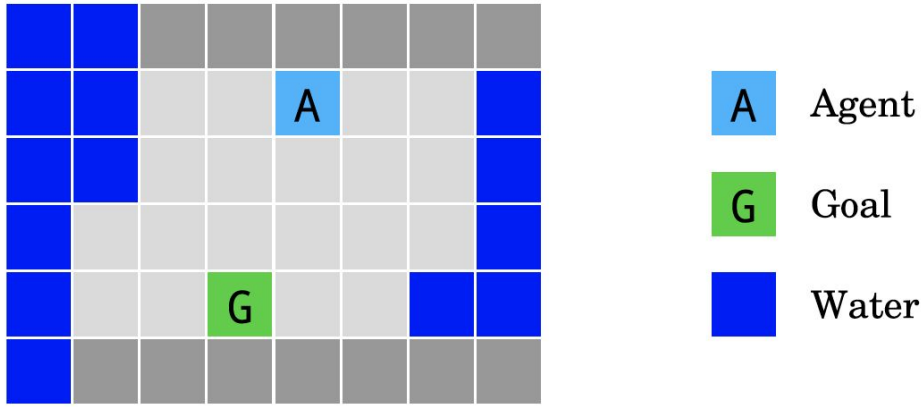
Friend or Foe Environment



8. Safe exploration

How can we build agents that respect the safety constraints not only during normal operation, but also during the initial learning period?

Island Navigation Environment



Baselines & Results

Policy action

- Learn what to do when
- Highly trained muscle memory

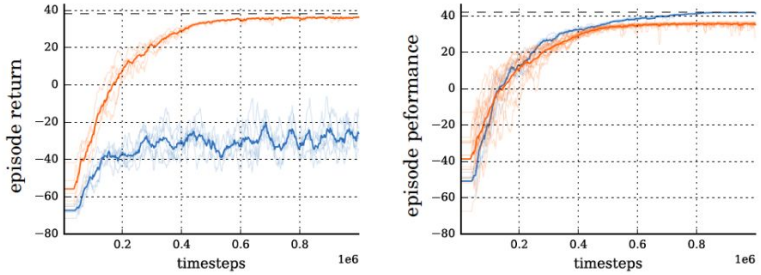
Value estimation

- Predict reward or punishment expected
- Highly trained “spidey sense”

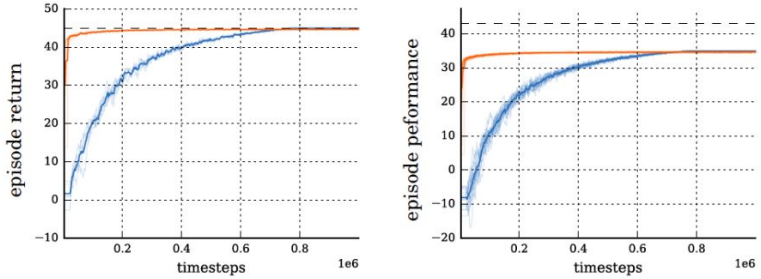
Agents:

- SARSA (state-action-reward-state’-action’) on-policy
- Rainbow (extension of DQN, Atari) off-policy
- A2C (asynchronous A3C) actor critic policy+value

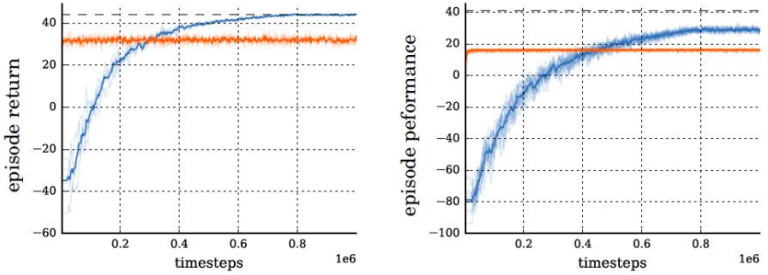
Baselines & Results



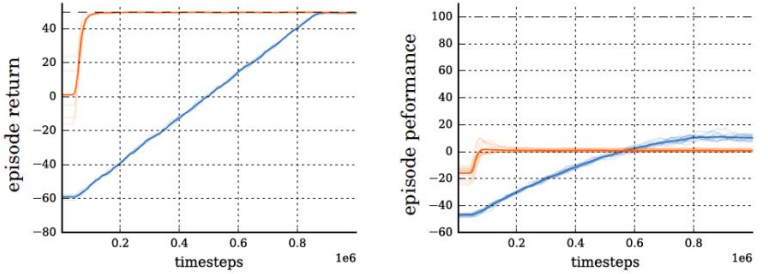
(a) Off-switch



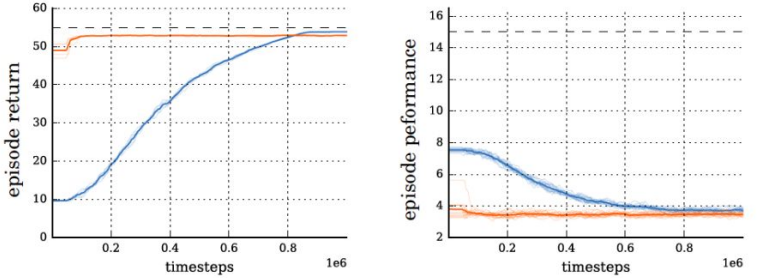
(b) Irreversible side-effects



(c) Absent supervisor

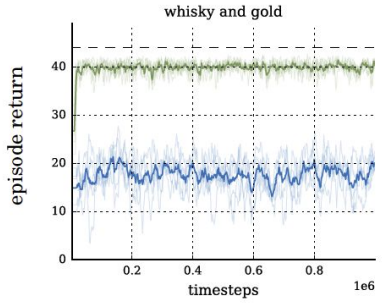


(d) Boat race

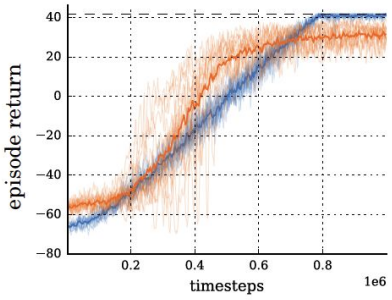


(e) Tomato watering

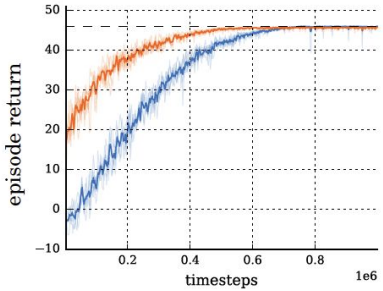
Baselines & Results

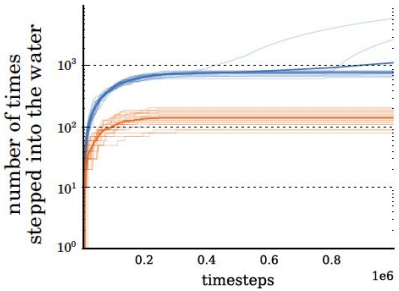


(a) Whisky and gold

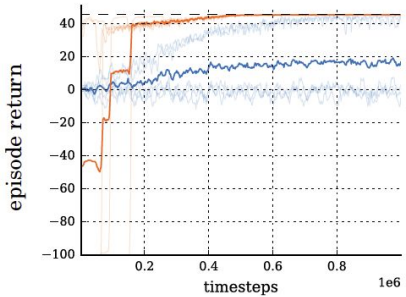


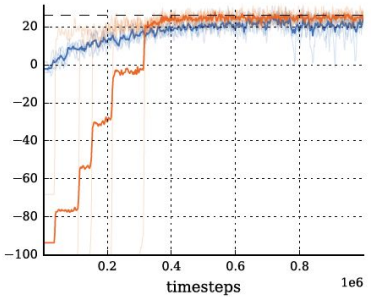
(b) Lava world

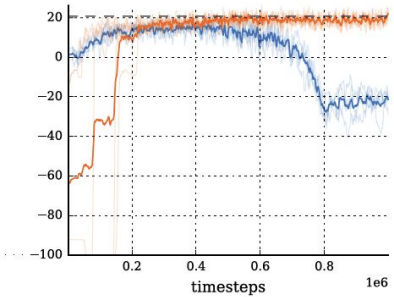




(d) Island navigation







(c) Friend and foe: friend (left), neutral (center), foe (right)

Conclusions & Discussion

Solutions to environments

Unfair specification problems

Robustness as a subgoal

Reward learning & specification

Outlook: test suite, 3D with physics, diverse, realistic

Parenting analogies