Lecture 25

Robot Learning 1

CS 3630

Parameter Optimization

- Robotic systems have many parameters
  - Control parameters
  - Odometry parameters
  - Kalman filter parameters
  - …

- How can we optimize these in an efficient way?

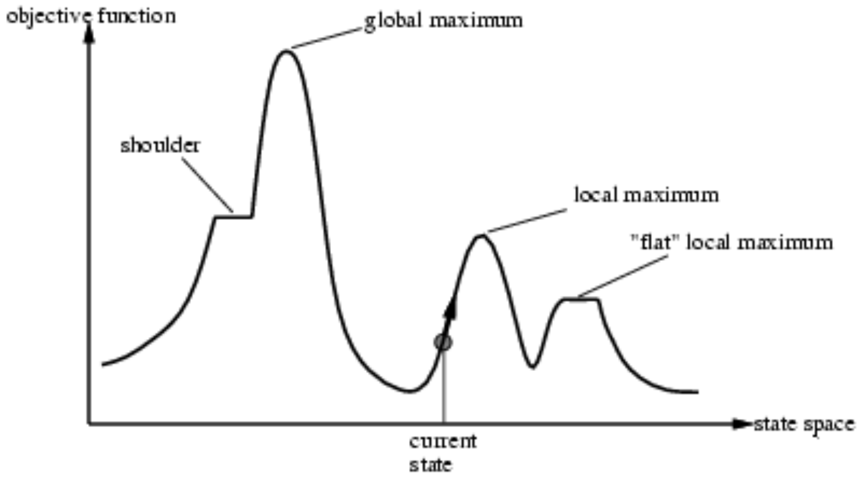
Local Search (Gradient Descent, Hillclimbing)

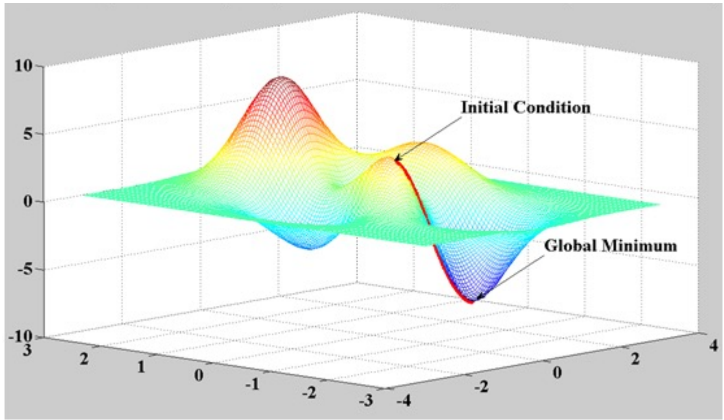
- Operates by keeping track of only the current state and changing that state to (hopefully) improve performance

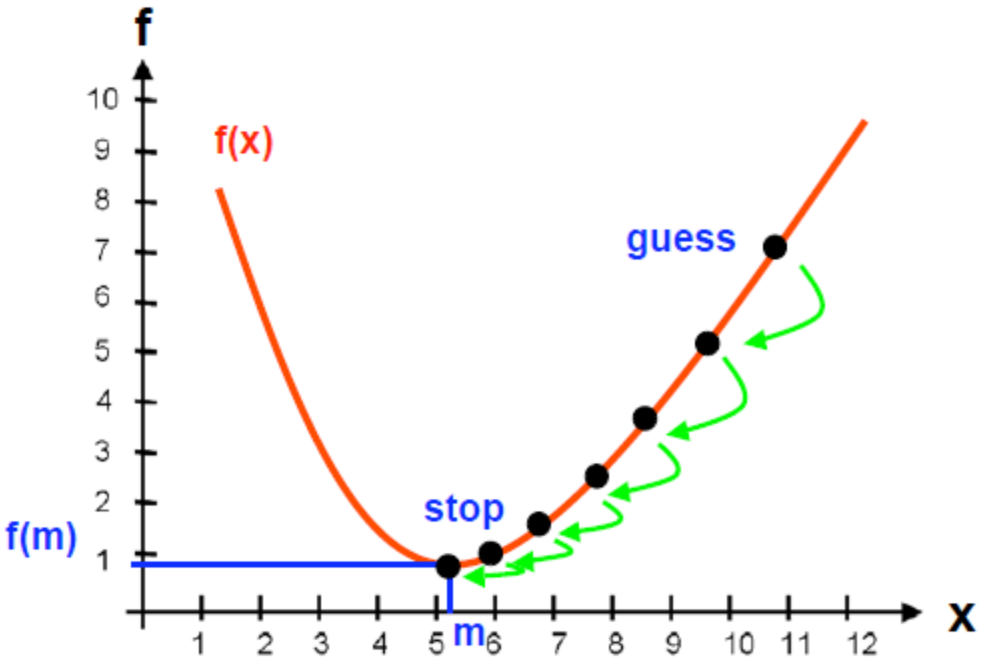
- Often used for:
  - Optimization problems
  - Scheduling
  - Task assignment
  - …many other problem where the goal is to find the best state according to some **objective function**

Local Search

- Goal is to find the local maximum (hillclimbing) or minimum (gradient descent)



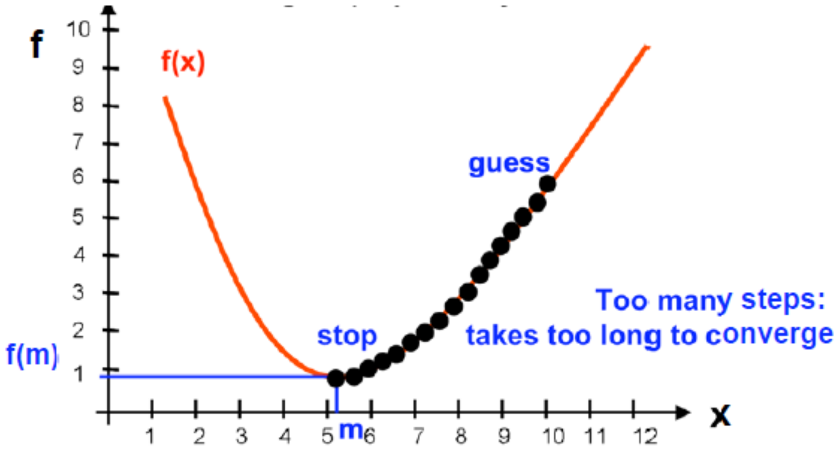


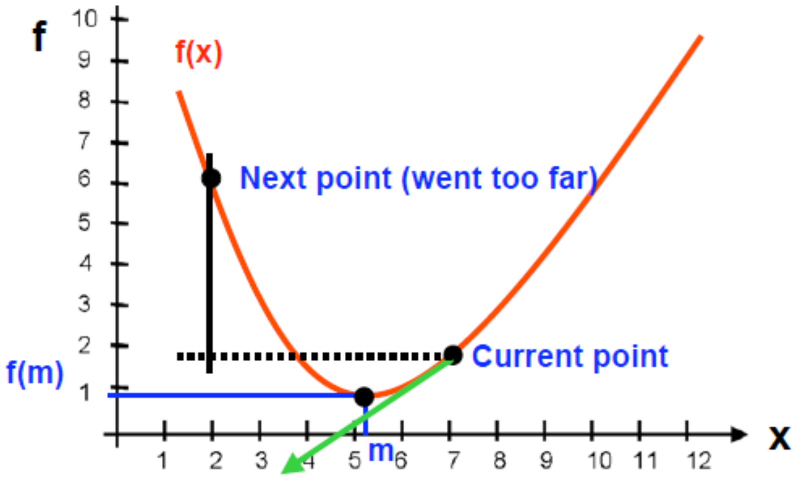


f
10
9
8
7
6
5
4
3
2
1
f(x)
guess
stop
f(m)
m
1
2
3
4
5
6
7
8
9
10
11
12
x

Parameters

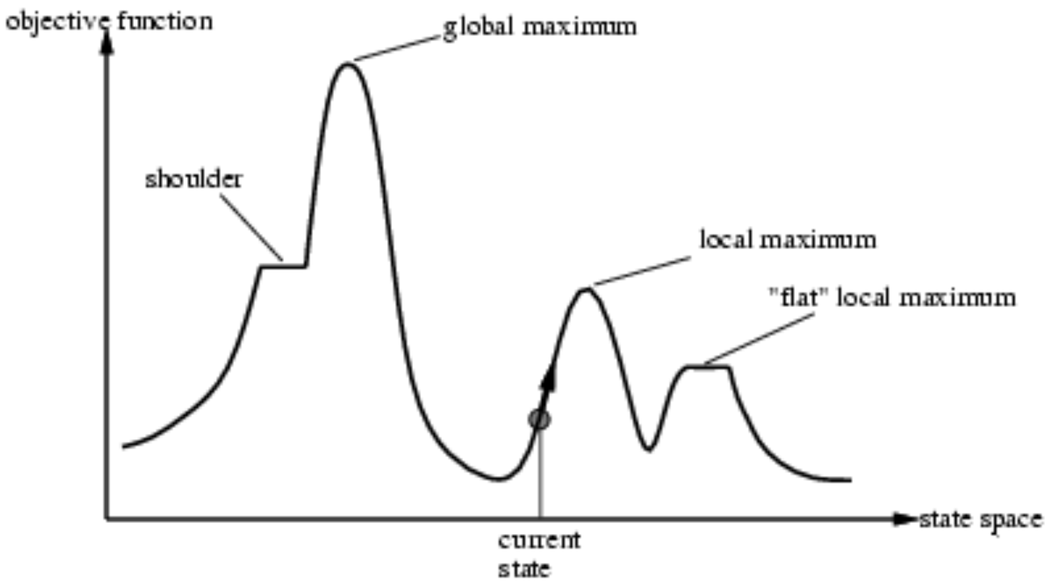
- Step size
  - Small steps: inefficient
  - Large steps: potentially bad results





Gradient Descent

- Benefits:
  - Efficient for multidimensional domains (if you know the gradient)
  - Well-suited for smooth objective and constraint functions
- Drawbacks
  - Can get stuck in local min/max



Approaches to overcoming local optima

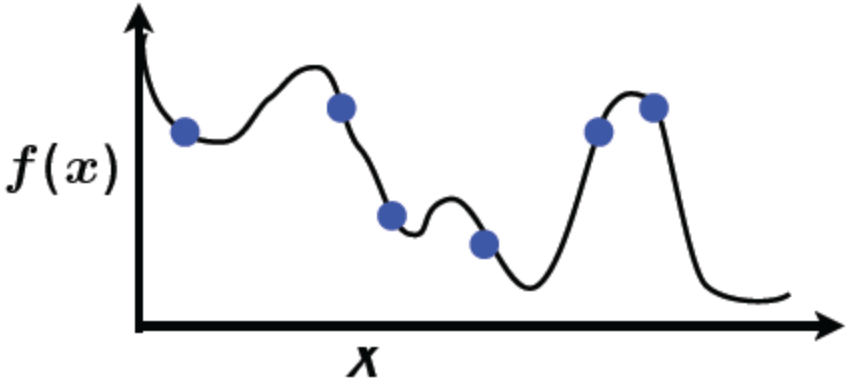
- Try again
  - Run algorithm some number of times and return the best solution. Initial start location is usually chosen randomly.

- Sideways moves
  - If stuck on a plateau, if we wait awhile and allow flat moves, will become unstuck (sometimes)

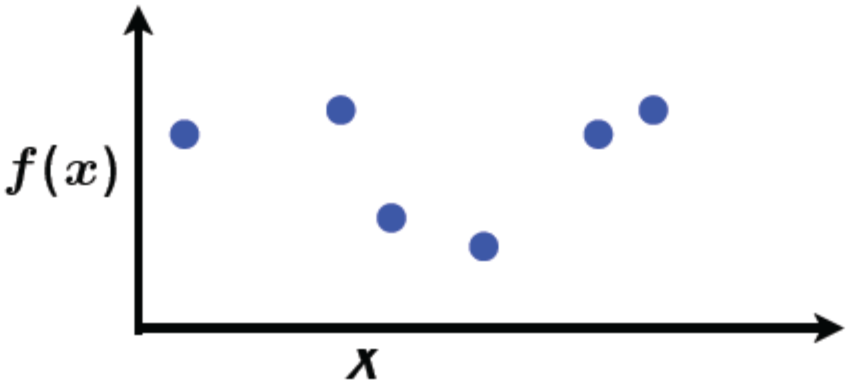
- Many other extensions: look ahead, simulated annealing, …

Local Search

- Challenge: we typically have no idea what function we are trying to minimize.
- We can only compute the function f at a finite number of points, and each evaluation is expensive



True Function



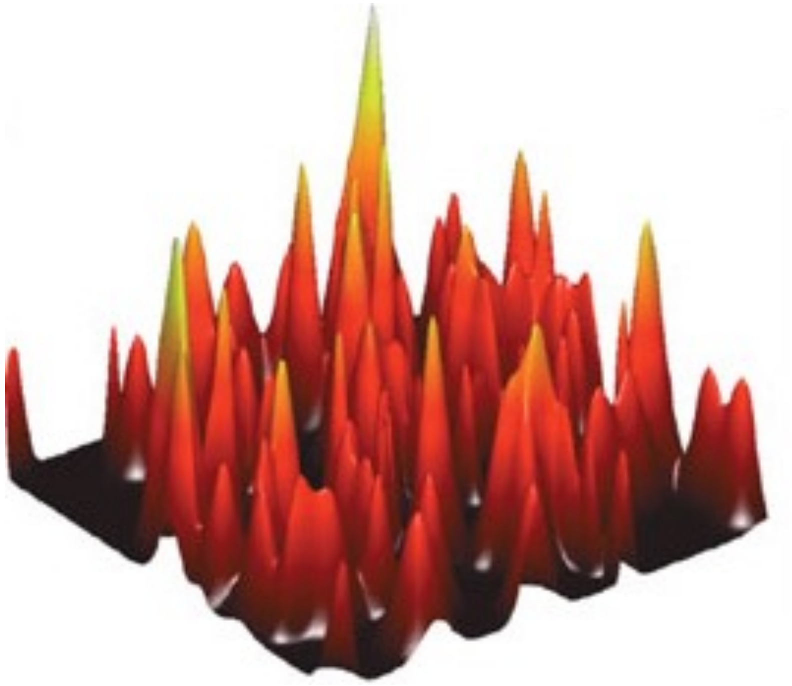
Observed Function

What if we don’t know f and the gradient?

- Estimate the partial derivative with respect to each dimension using sampling

1. Take current estimate $x = \begin{bmatrix} x_i \\ \dots \\ x_N \end{bmatrix}$
2. Create k new estimates, such that $x' = \begin{bmatrix} x_i + \Delta_1 \\ \dots \\ x_N + \Delta_N \end{bmatrix}$
3. Evaluate each estimate using objective function f
4. Estimate gradient of objective function f at x

Many domains have too many local optima for local search to be effective



"many real-world problems have a landscape that looks more like a widely scattered family of balding porcupines on a flat floor, with miniature porcupines living on the tip of each porcupine needle, ad infinitum."

-Russell and Norvig

Takeaways

- Learning is hard, especially in high-dimensional spaces
- Finding a solution requires many trials
- Many of the trials lead to failure, which can be problematic for expensive robots
- Simulation has been widely used to overcome the need for data while keeping robots safe…
  - … but this introduces the the sim2real gap: robot skills learned in simulation are not guaranteed to work well in the real world

Today, we’ll look at some of the results from simulation, and next time discuss the underlying learning methods.

Flexible Muscle-Based Locomotion for Bipedal Creatures

SIGGRAPH ASIA 2013

Thomas Geijtenbeek
Michiel van de Panne
Frank van der Stappen

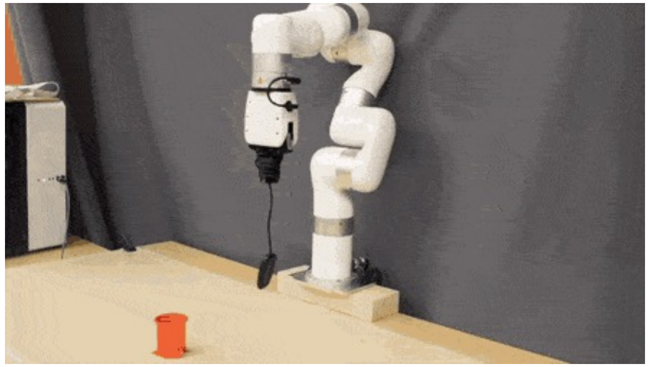
Example applications

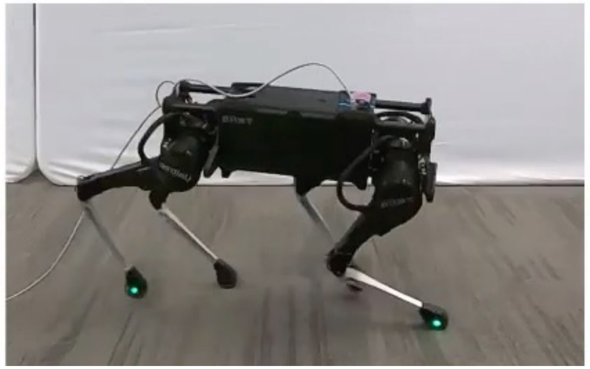
- Evolving soft robots
  - Cheney N, MacCurdy R, Clune J, Lipson H (2013) *Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding*. Proceedings of the Genetic and Evolutionary Computation Conference.
- 2DBoxCar

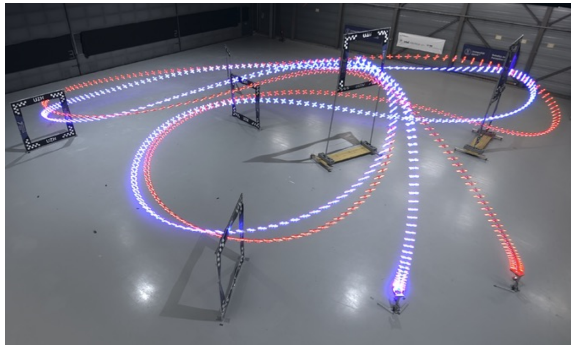
Learning-based approaches

Imitation Learning

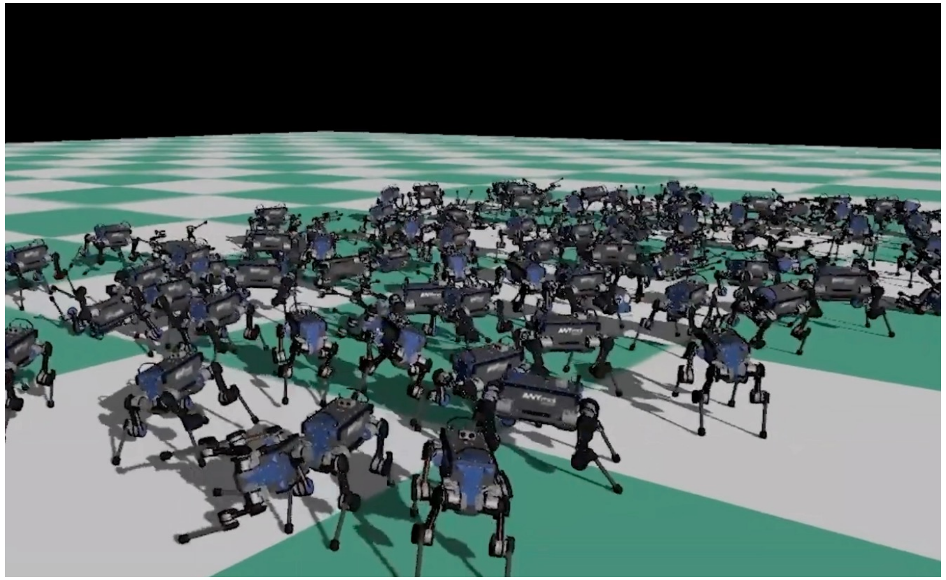
Reinforcement Learning



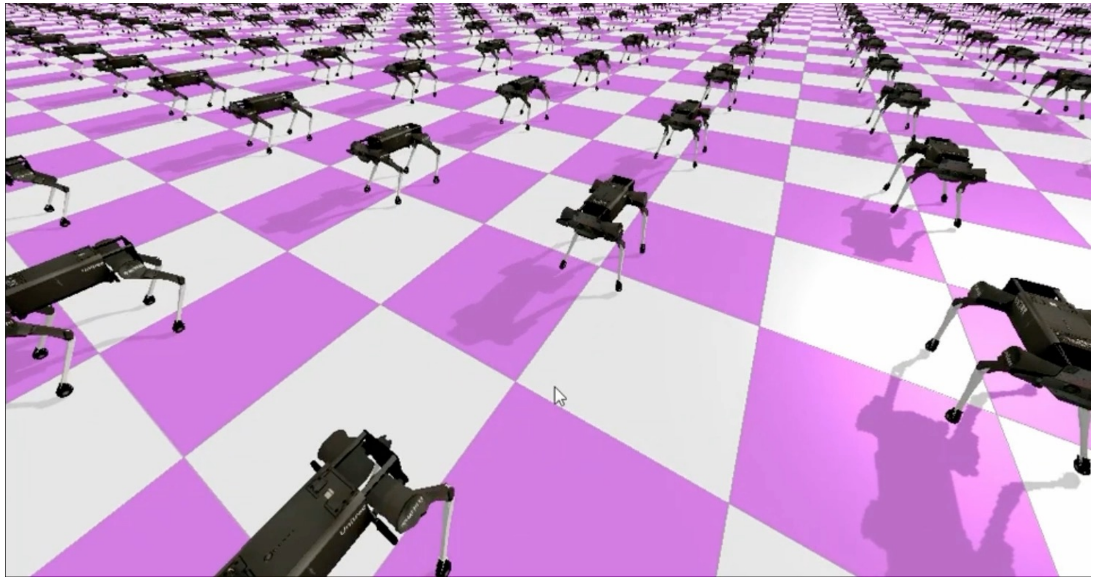




Simulators

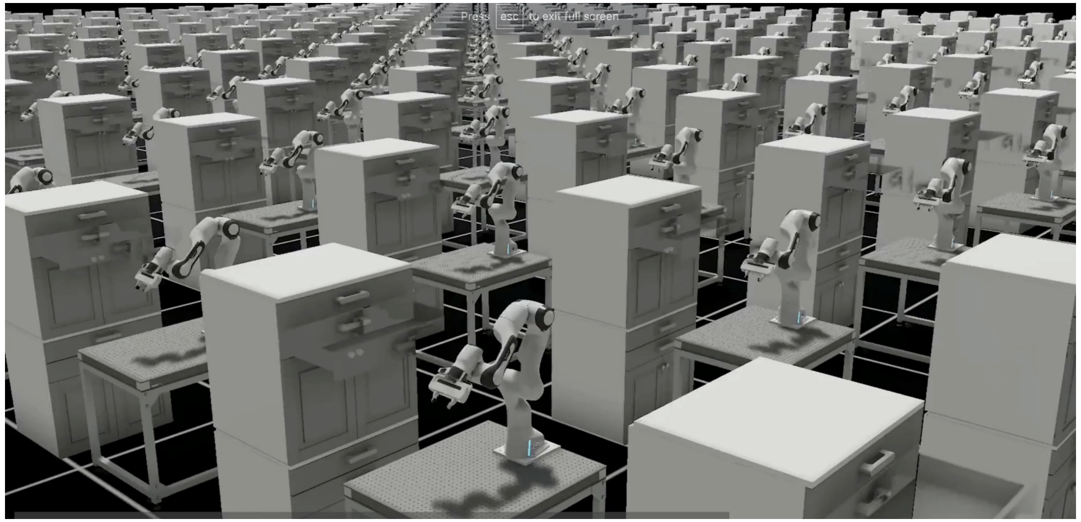


Hwangbo et al. 2019



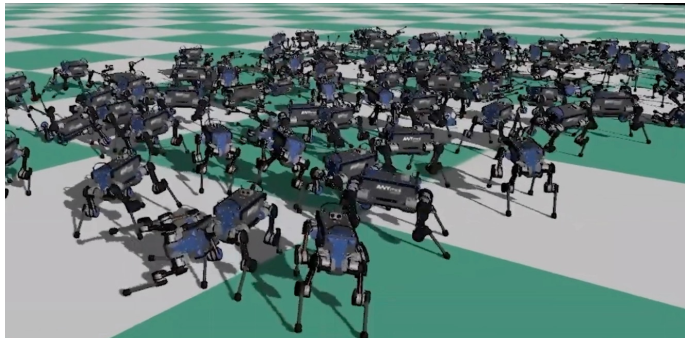
Coumans et al. 2020

Savva et al. 2019, Szot et al. 2021

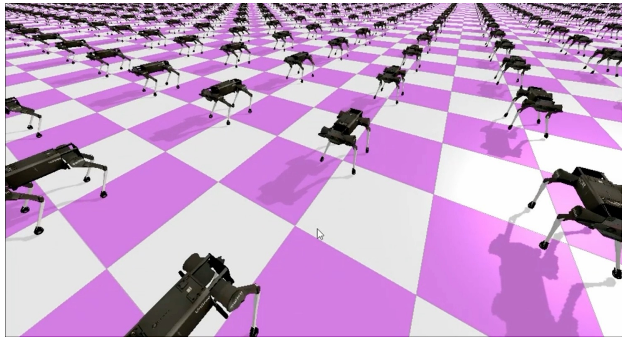


Mittal et al. 2023

Simulators

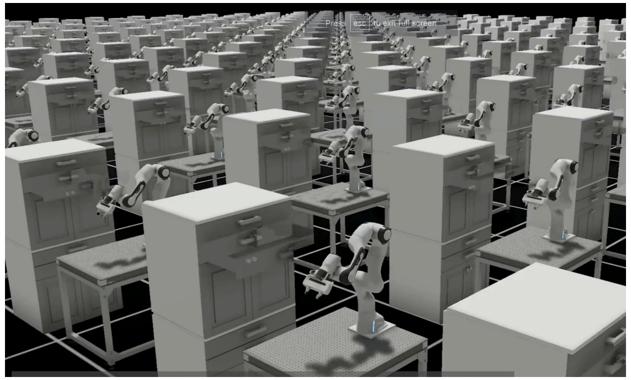


Hwangbo et al. 2019



Coumans et al. 2020

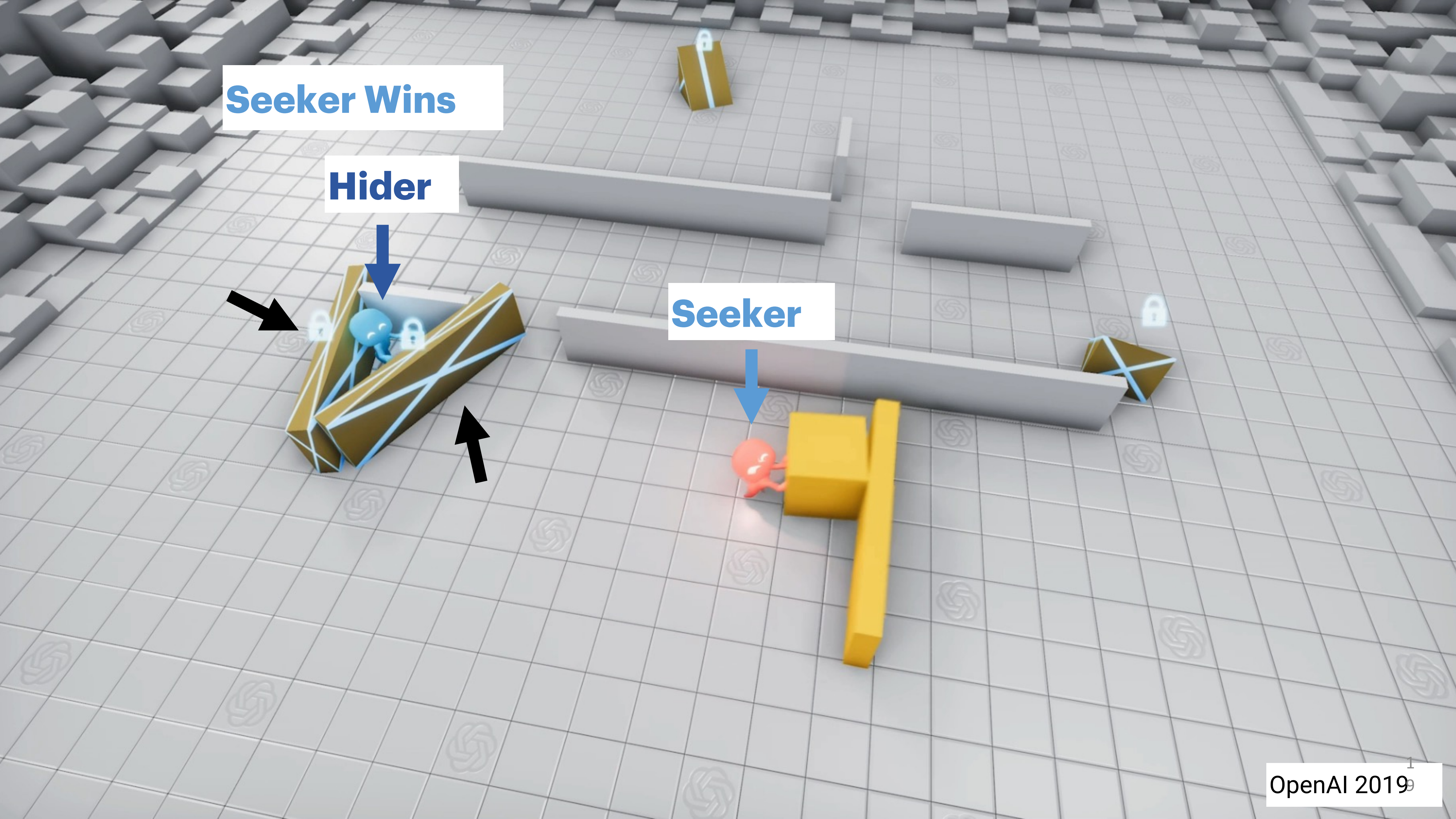
Savva et al. 2019, Szot et al. 2021

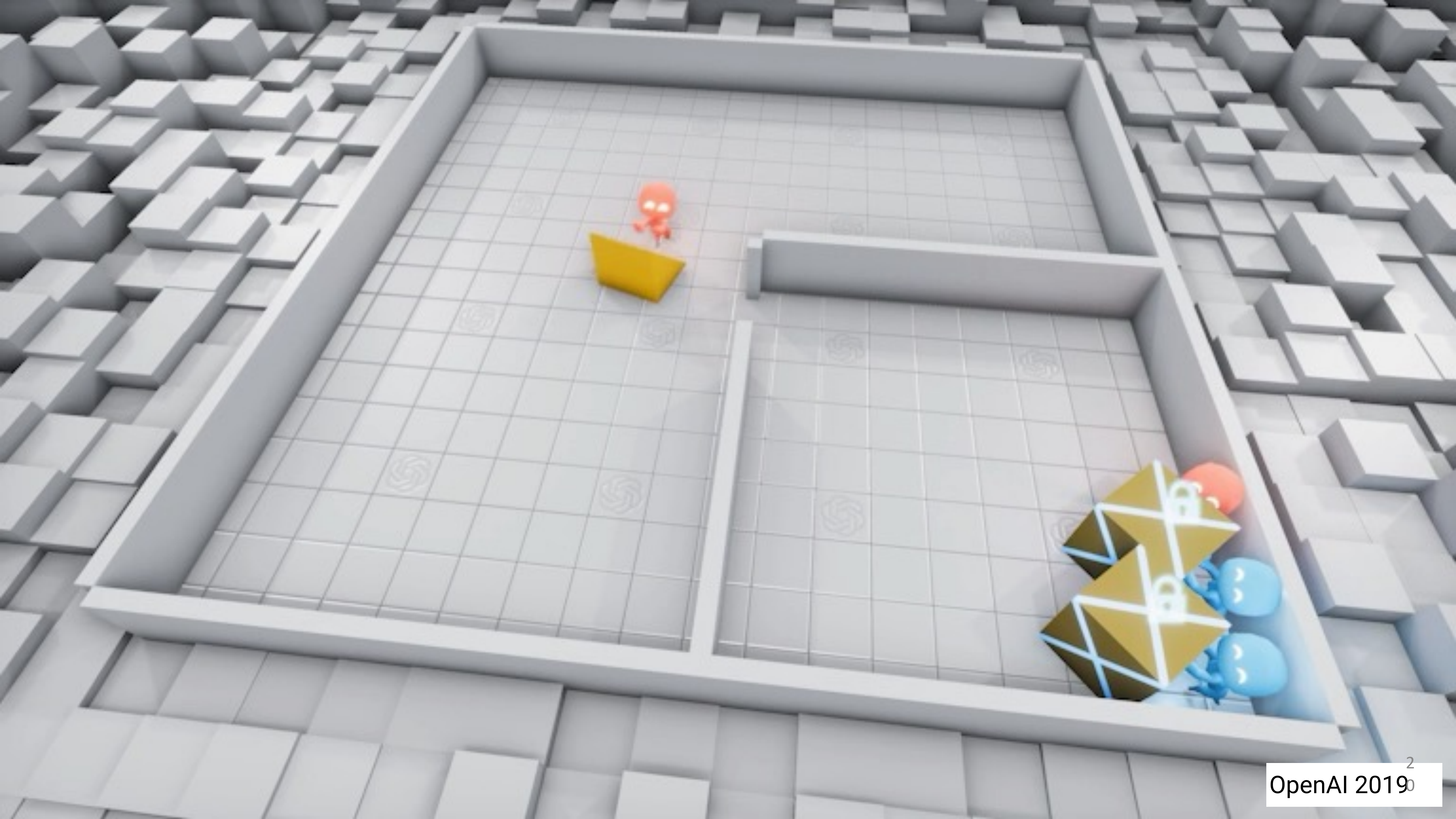


Mittal et al. 2023

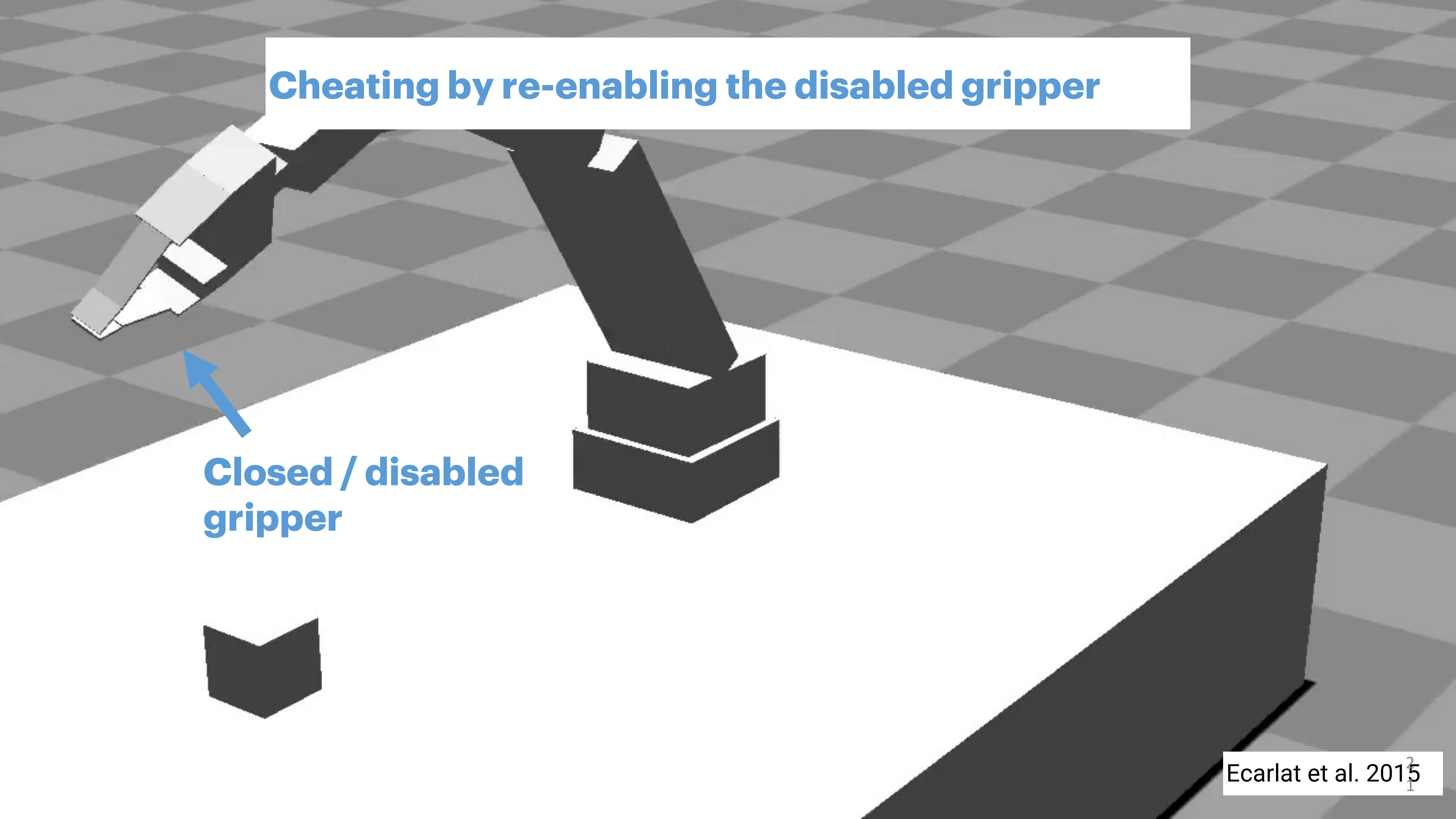
1. Safe
2. Fast and scalable
3. Systematic benchmarking
4. Cheap
5. Flexible
6. Access to ground-truth data

No simulator is perfect, and AI systems learn to cheat by exploiting imperfections in the simulator.

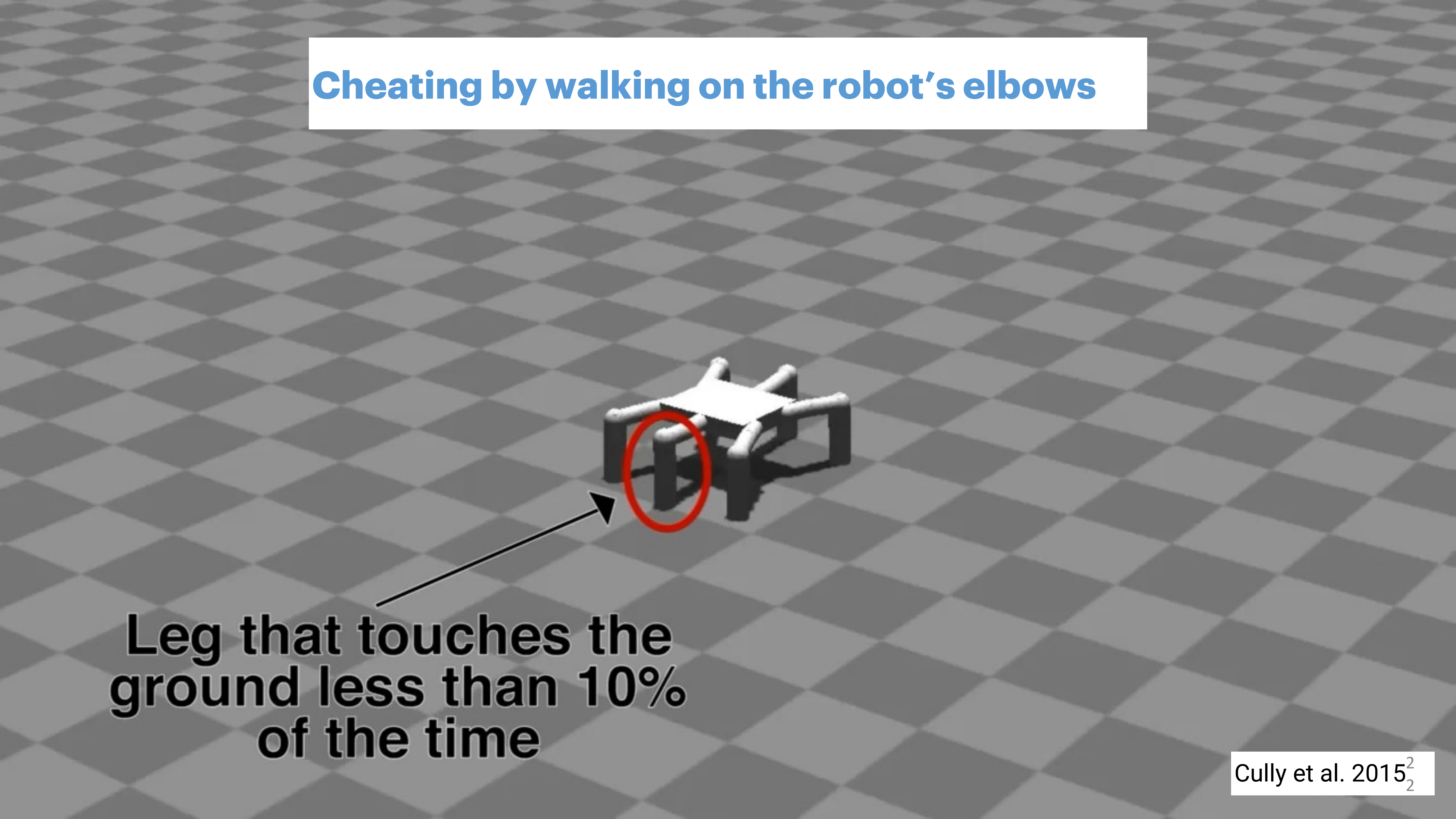

Seeker Wins
Hider
Seeker
OpenAI 2019



OpenAI 2019


Cheating by re-enabling the disabled gripper
Closed / disabled gripper
Ecarlat et al. 2015

Cheating by walking on the robot’s elbows

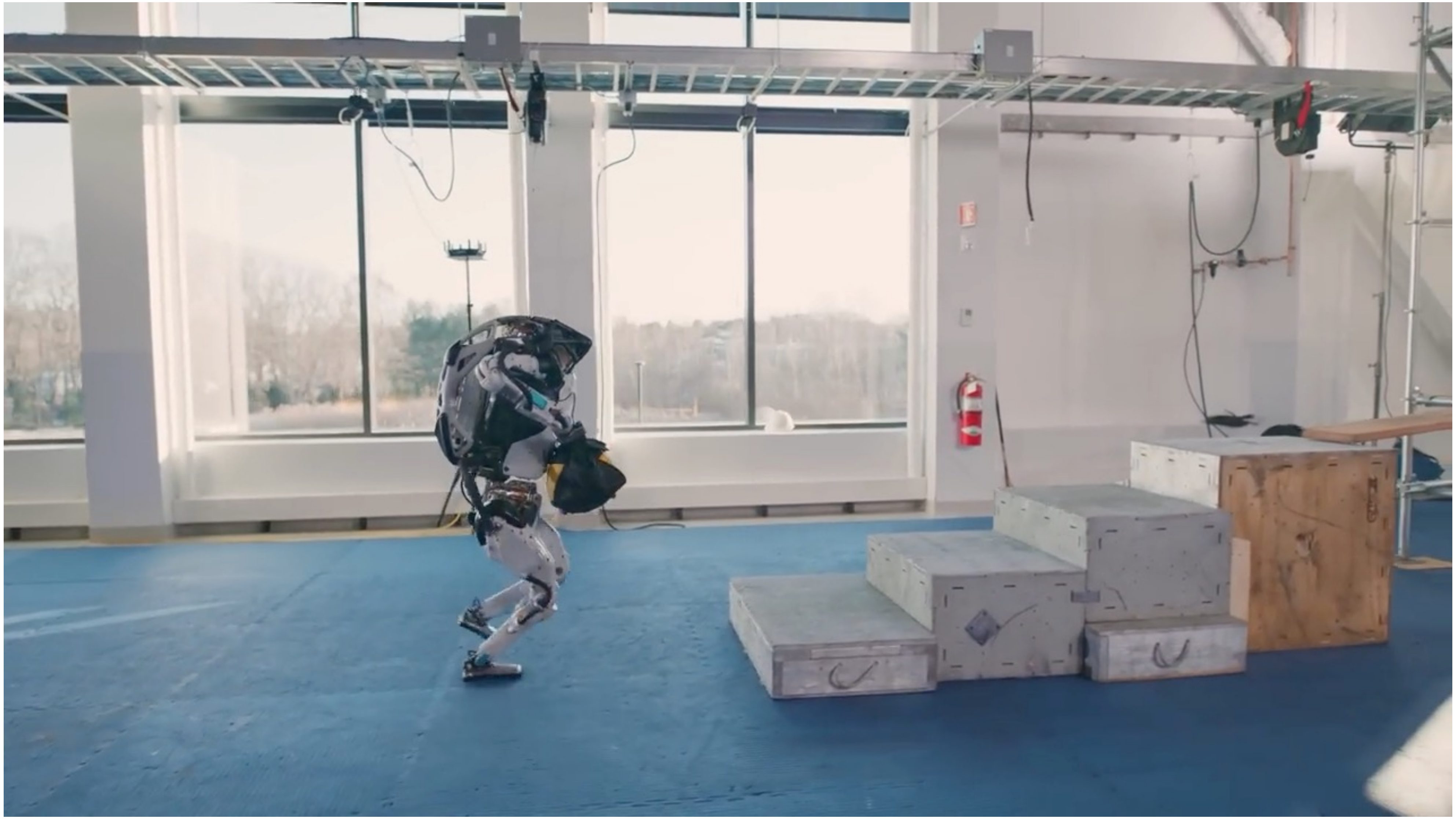


Cully et al. 2015

Examples of AI systems cheating in simulation

Title	Description	Video / Image	Authors	Original source	Original source link	Source / Credit	Source link
Aircraft landing	Evolved algorithm for landing aircraft exploited overflow errors in the physics simulator by creating large forces that were estimated to be zero, resulting in a perfect score		Feldt, 1998	Generating diverse software versions with genetic programming: An experimental study.	http://ieeexplore.ieee.o	Lehman et al, 2018	https://arxiv.org
Bicycle	Reward-shaping a bicycle agent for not falling over & making progress towards a goal point (but not punishing for moving away) leads it to learn to circle around the goal in a physically stable loop.		Randlov & Alstrom, 1998	Learning to Drive a Bicycle using Reinforcement Learning and Shaping	https://pdfs.semanticsc	Gwern Branwen	https://www.gw
Block moving	A robotic arm trained using hindsight experience replay to slide a block to a target position on a table achieves the goal by moving the table itself.		Chopra, 2018	GitHub issue for OpenAI gym environment FetchPush-v0	https://github.com/open	Matthew Rahtz	
Boat race	Reinforcement learning agent goes in a circle hitting the same targets instead of finishing the race	https://www.you	Amodei & Clark, 2016	Faulty reward functions in the wild	https://blog.openai.com		
Ceiling	A genetic algorithm was instructed to try and make a creature stick to the ceiling for as long as possible. It was scored with the average height of the creature during the run. Instead of sticking to the ceiling, the creature found a bug in the physics engine to snap out of bounds.	https://youtu.be	Higueras, 2015	Genetic Algorithm Physics Exploiting	https://youtu.be/ppf3Vq	Jesús Higueras	https://youtu.be
CycleGAN steganography	CycleGAN algorithm for converting aerial photographs into street maps and back steganographically encoded output information in the intermediary image without it being humanly detectable.		Chu et al, 2017	CycleGAN, a Master of Steganography	https://arxiv.org/abs/17	Tech Crunch / Gwern Branwen / B	https://techcrun
Data order - mushrooms	Neural nets evolved to classify edible and poisonous mushrooms took advantage of the data being presented in alternating order, and didn't actually learn any features of the input images		Ellefsen et al, 2015	Neural modularity helps organisms evolve to learn new skills without forgetting old skills	http://journals.plos.org/	Lehman et al, 2018	https://arxiv.org
Dying to Teleport	PlayFun algorithm deliberately dies in the Bubble Bobble game as a way to teleport to the respawn location		Murphy, 2013	The First Level of Super Mario Bros. is Easy with Lexicographic Orderings and Time Travel	http://www.cs.cmu.edu/	Alex Meiburg	
Eurisko - authorship	Game-playing agent accrues points by falsely inserting its name as the author of high-value items		Johnson, 1984	Eurisko, The Computer With A Mind Of Its Own	http://aliciapatterson.or	Catherine Olsson / Stuart Armstrong	http://lesswron
Eurisko - fleet	Eurisko won the Trillion Credit Squadron (TCS) competition two years in a row creating fleets that exploited loopholes in the game's rules, e.g. by spending the trillion credits on creating a very large number of stationary and defenseless ships		Lenat, 1983	Eurisko, The Computer With A Mind Of Its Own	http://aliciapatterson.or	Haym Hirsh	
Evolved creatures - clapping	Creatures exploit a collision detection bug to get free energy by clapping body parts together		Sims, 1994	Evolved Virtual Creatures	http://www.karlsims.con	Lehman et al, 2018 / Janelle Shane	https://arxiv.org
Evolved creatures - falling	Creatures bred for speed grow really tall and generate high velocities by falling over	https://www.you	Sims, 1994	Evolved Virtual Creatures	http://www.karlsims.con	Lehman et al, 2018 / Janelle Shane	https://arxiv.org
Evolved creatures - floor collisions	Creatures exploited a coarse physics simulation by penetrating the floor between time steps without the collision being detected, which generated a repelling force, giving them free energy.	https://pbs.twim	Cheney et al, 2013	Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding	http://jeffclune.com/pub	Lehman et al, 2018 / Janelle Shane	https://arxiv.org
Evolved creatures - pole vaulting	Creatures bred for jumping were evaluated on the height of the block that was originally closest to the ground. The creatures developed a long vertical pole and flipped over instead of jumping.	https://www.you	Krcah, 2008	Towards efficient evolutionary design of autonomous robots	http://artax.karlin.mff.cu	Lehman et al, 2018 / Janelle Shane	https://arxiv.org
Evolved creatures - self-intersection	Creatures exploit a quirk in Box2D physics by clipping one leg into another to slide along the ground with phantom forces instead of walking	https://youtu.be	Code Bullet, 2019	AI Learns To Walk	https://youtu.be/K-wIZu	Peter Cherepanov	
Evolved creatures - suffocation	In a game meant to simulate the evolution of creatures, the programmer had to remove "a survival strategy where creatures could gain energy by suffocating themselves"		Schumacher, 2018	0.11.0.9&10: All the Good Things	https://spe		
Evolved							

Yet we still see success stories...



Boston Dynamics 2023

And failures...

Boston Dynamics 2023

More on the techniques behind these methods next time!