OpenAI

Faster Reinforcement Learning via Transfer

John Schulman

2018.09.06

Overview

Policy Gradients

- Success stories
- Limitations

Meta Reinforcement Learning

Gym Retro

Terminology

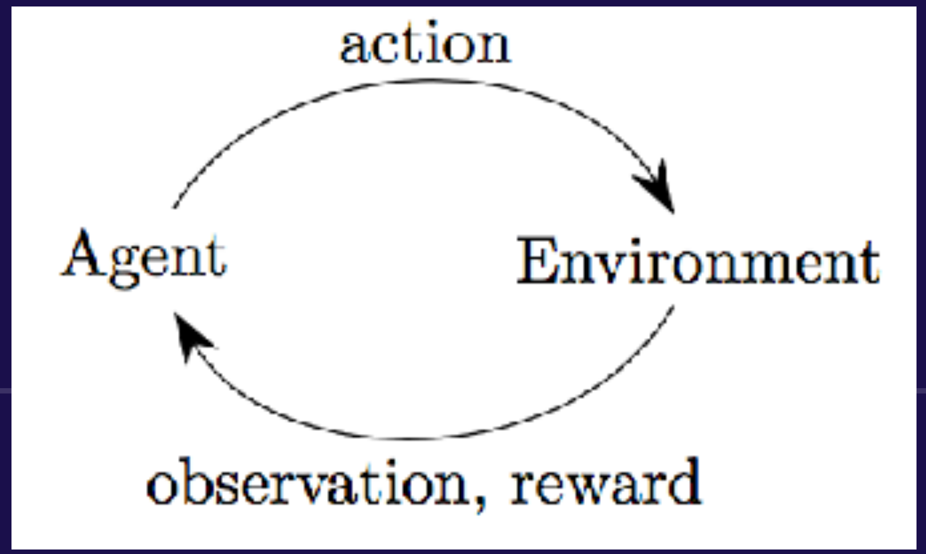
Reinforcement Learning (RL): maximize cumulative reward using trial and error.

Deep RL: using neural nets to represent functions in RL algorithm

Meta-Learning: learn how to learn. Master a task that itself involves learning

RL Terminology

Observation, Action, Reward



Trajectory: a sequence of observation, action, reward

Return: sum of rewards received by agent

RL Terminology (II)

Policy: function that chooses action, based on recent observations

- usually stochastic

Policy Gradient Method: a class of RL algorithms that optimize a policy

Policy Gradients

Pseudocode

Initialize policy

Loop:

- Collect trajectories
- Estimate which actions were good and which were bad
- Increase probability of good actions via gradient update

Why does it work? Gradient ascent on expected return of policy

Policy Gradients—History

Old idea (Williams ’92)

Lots of recent variants with deep learning

- E.g., Proximal Policy Optimization (PPO)

Policy Gradients—Success Stories

AlphaGo

- Lee Sedol version used policy gradients (REINFORCE)
  - as part of the pipeline that also involved search

deepmind.com

Policy Gradients—Success Stories

Dota2

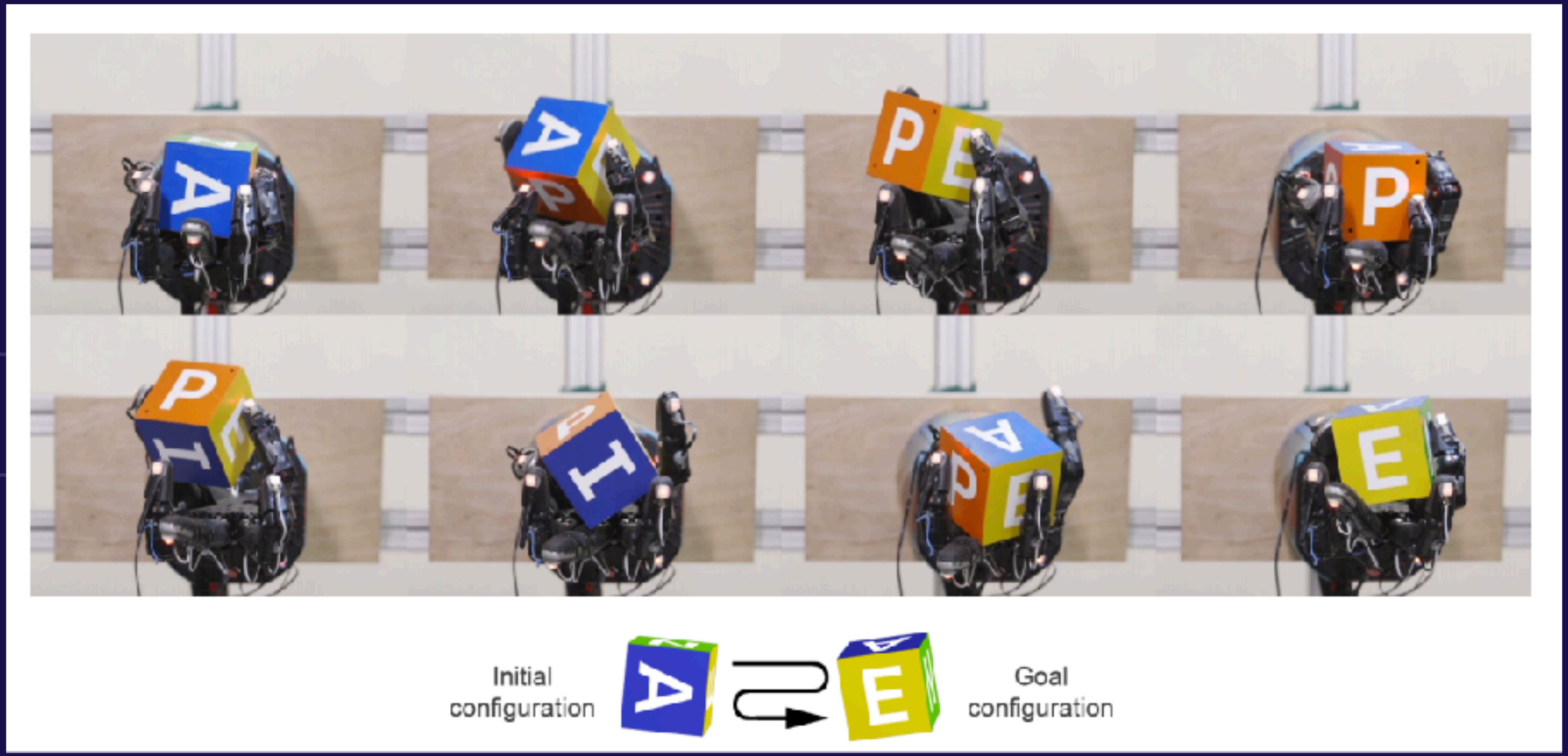
- LSTM policy trained using PPO
- beat world champions at 1v1 (2017)
- beat top players at modified 5v5 (2018)
- approaching top level at normal 5v5 (2018)

OpenAI, 2018

Policy Gradients—Success Stories

Robotic manipulation

- LSTM policy trained using PPO
  - (this is actually meta-RL)



OpenAI, 2018

RL Requires a Lot of Training

	Chess	Shogi	Go
Mini-batches	700k	700k	700k
Training Time	9h	12h	34h
Training Games	44 million	24 million	21 million
Thinking Time	800 sims 40 ms	800 sims 80 ms	800 sims 200 ms

Table S3: Selected statistics of *AlphaZero* training in Chess, Shogi and Go.

Alpha Zero Paper (Silver et al. 2017)

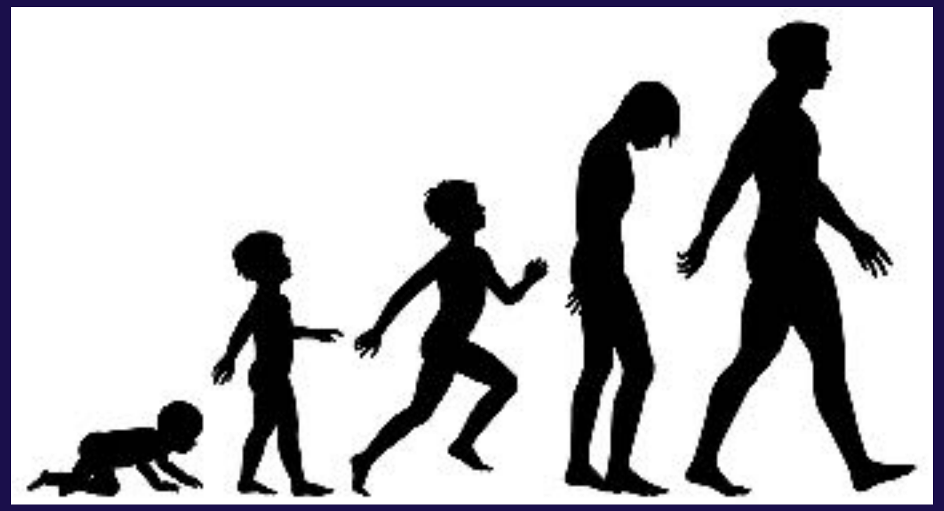
RL Requires a Lot of Training

	OPENAI 1V1 BOT	OPENAI FIVE
CPUs	60,000 CPU cores on Azure	128,000 preemptible CPU cores on GCP
GPUs	256 K80 GPUs on Azure	256 P100 GPUs on GCP
Experience collected	~300 years per day	~180 years per day (~900 years per day counting each hero separately)

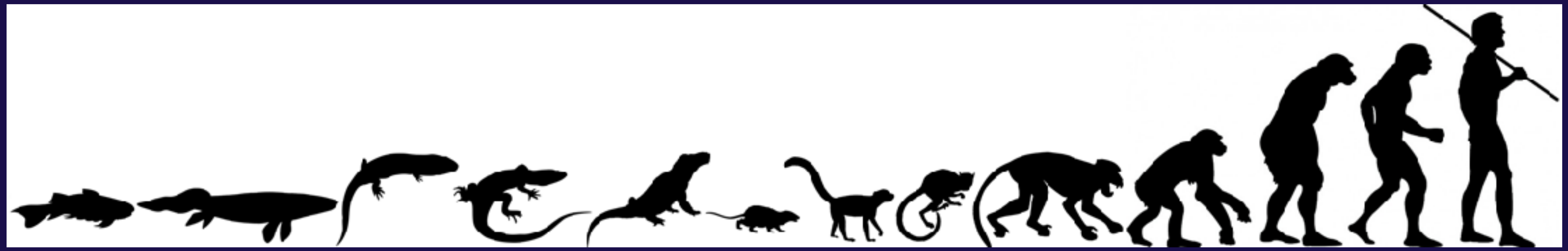
OpenAI Five (https://blog.openai.com/openai-five/)

Prior Knowledge

from life history



and evolutionary history



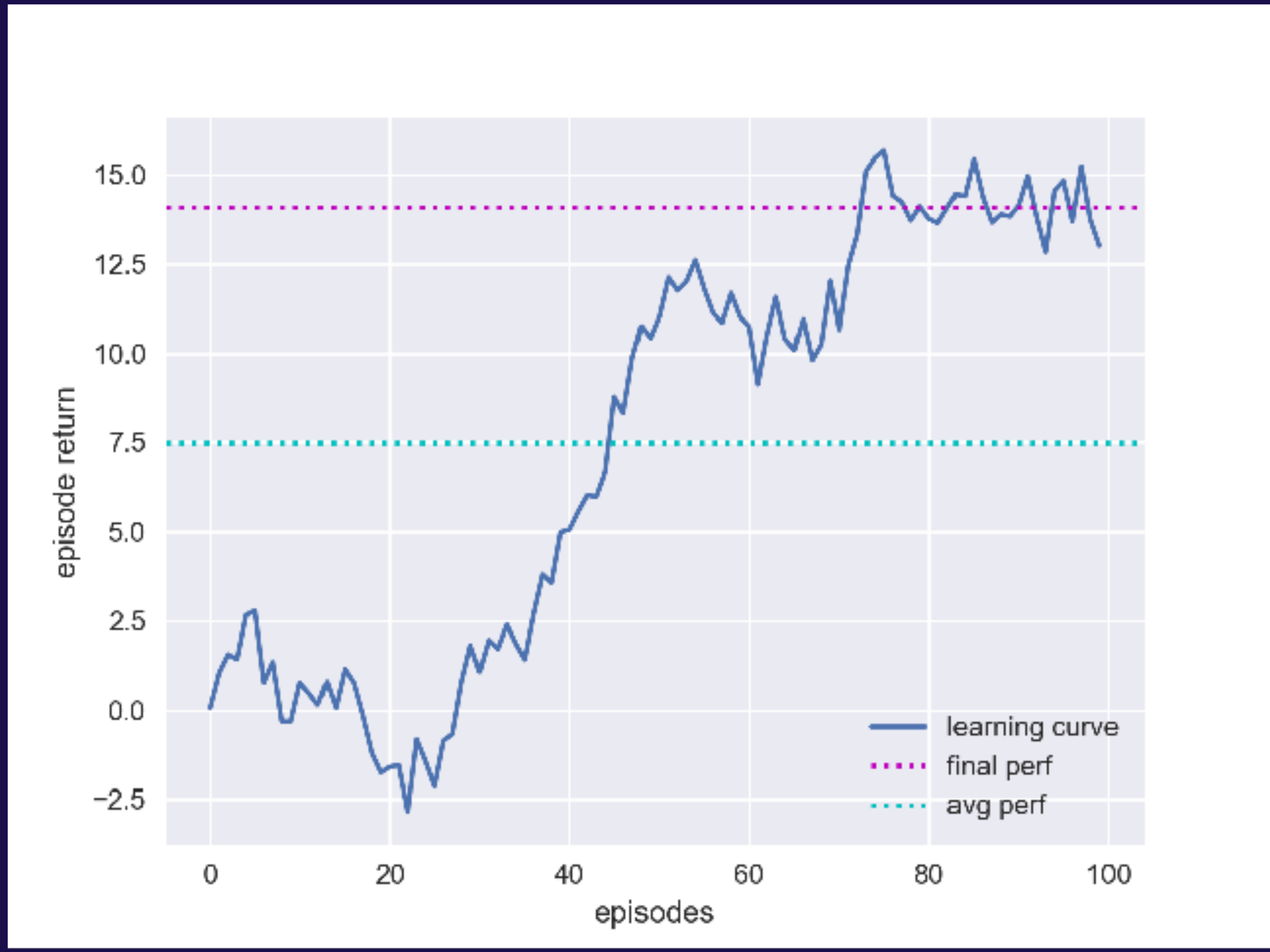
https://www.youtube.com/watch?v=8vNxjwt2AqY

https://blog.openai.com/gym-retro/

Meta Reinforcement Learning: Problem Formulation

Single task RL: Maximize reward in a single task, given K episodes of experience

- Train from scratch
- Most recent deep RL work follows this framework, e.g. using Atari games



random data for illustrative purposes

Meta RL: Maze Navigation

"Classic RL" task: learn to navigate from start to goal as fast as possible in a single maze

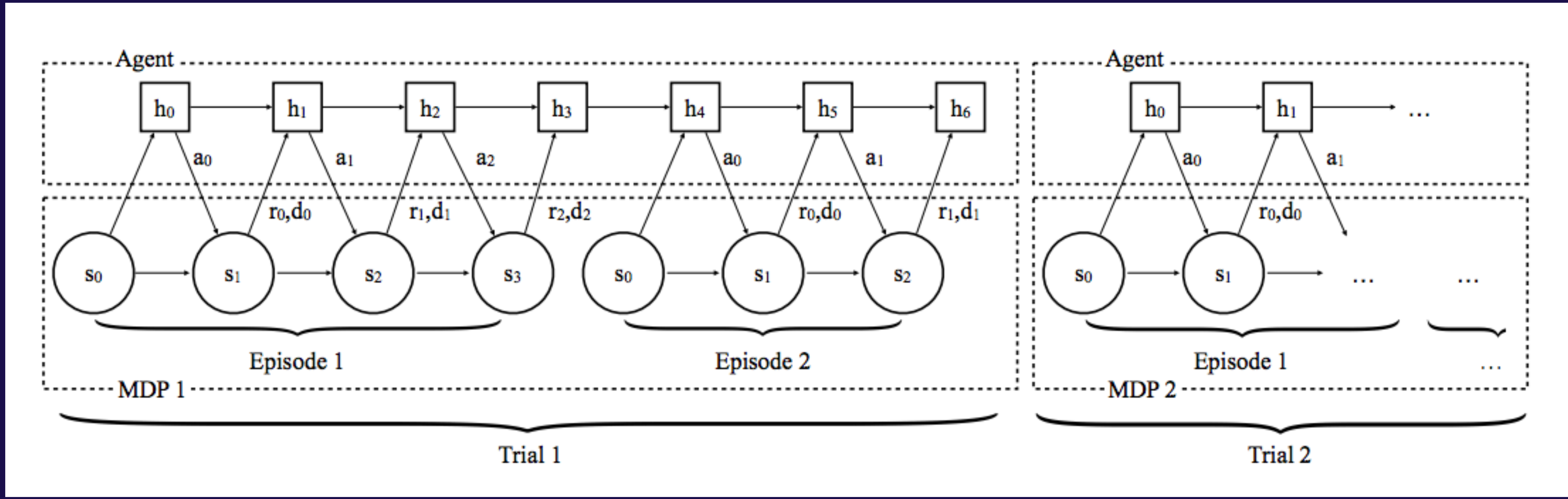
"Meta-RL" task: learn to navigate from start to goal as fast as possible K times in a random maze

- Agent doesn't know what maze it's in—needs to explore
- After first episode, it should know location of goal and go straight there
- Through outer training loop, agent needs to learn to explore and learn to remember

Meta RL is a Special Case of Normal RL

Define new RL task in terms of old task

- First timestep: agent is placed in random task / world
- New observation = (old observation, reward, "done" signal)
- New-task episode = K old-task episodes



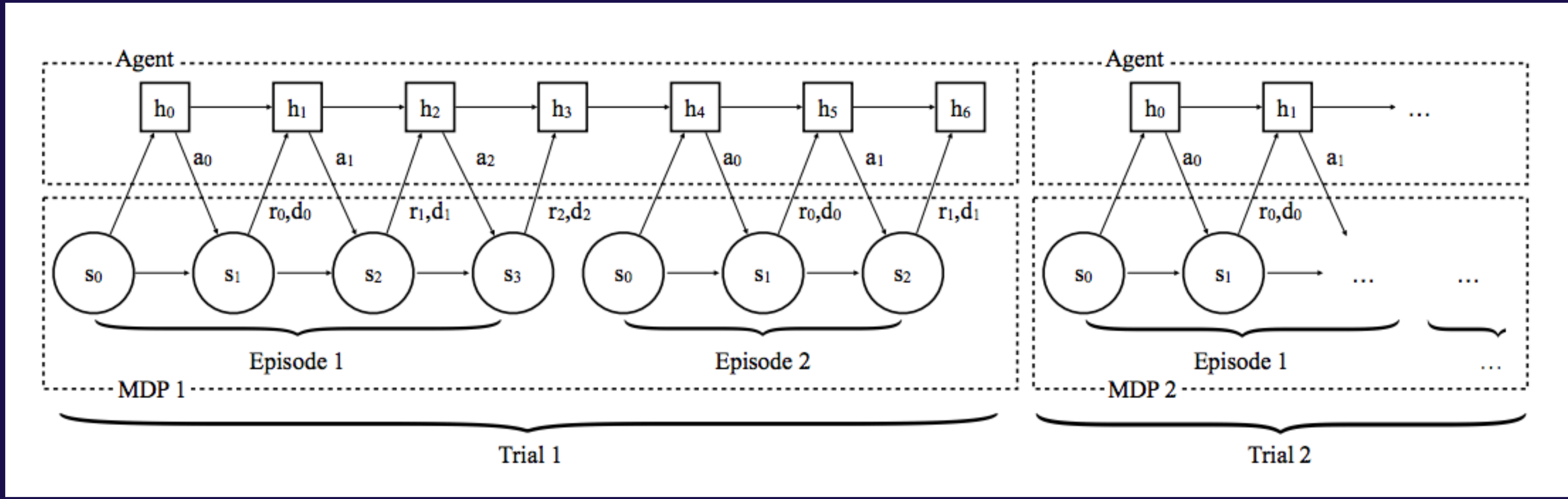
https://arxiv.org/abs/1611.02779 (Duan et al. '16)

Meta RL is a Special Case of Normal RL

RL²: "Fast Reinforcement Learning via Slow Reinforcement Learning"

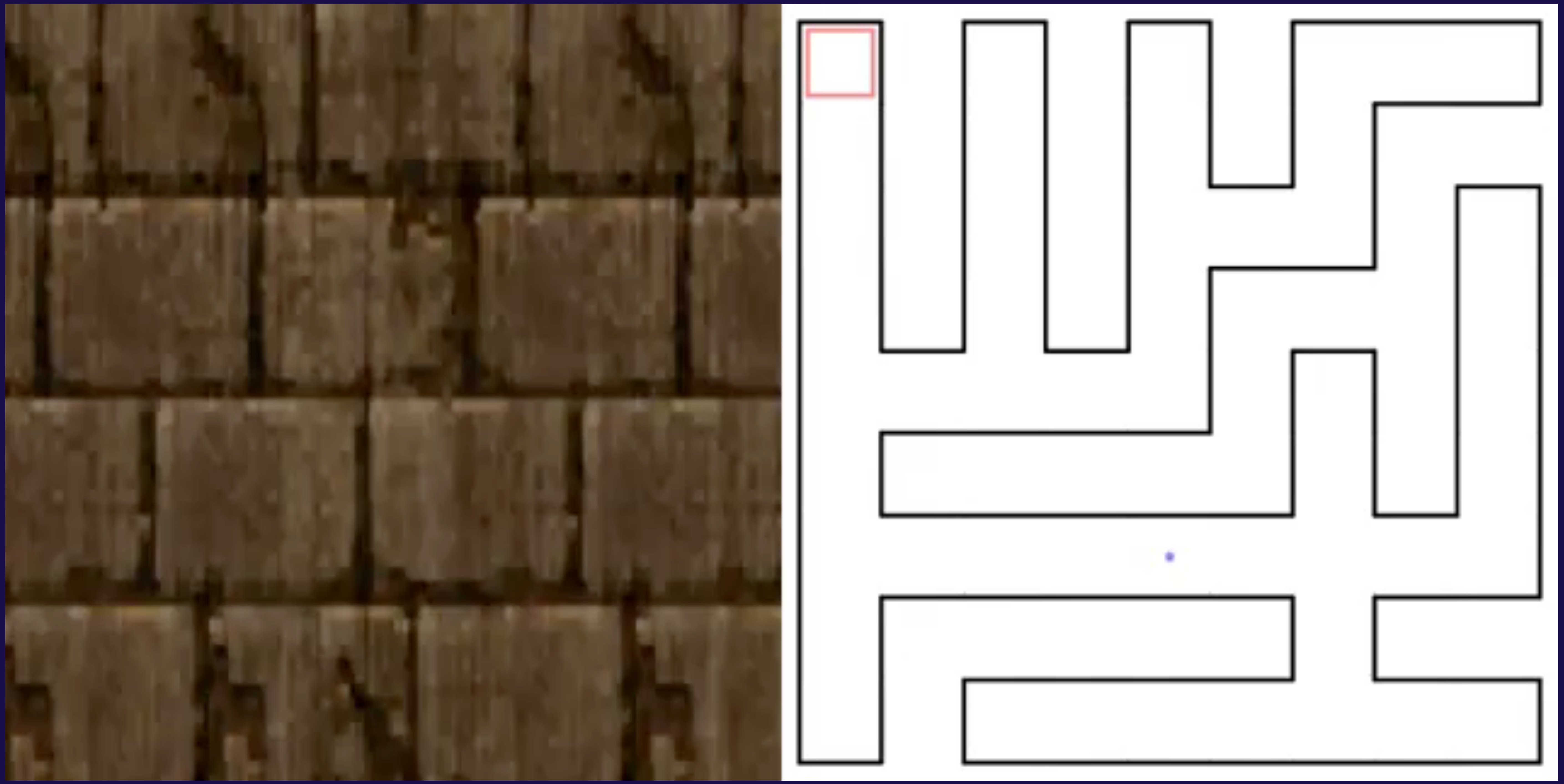
Agent uses recurrent network (LSTM / GRU) as policy—internally implements learning algorithm.

- Fast RL: learned "algorithm" performed by LSTM
- Slow RL: algorithm used to train LSTM in outer loop



https://arxiv.org/abs/1611.02779 (Duan et al. '16)

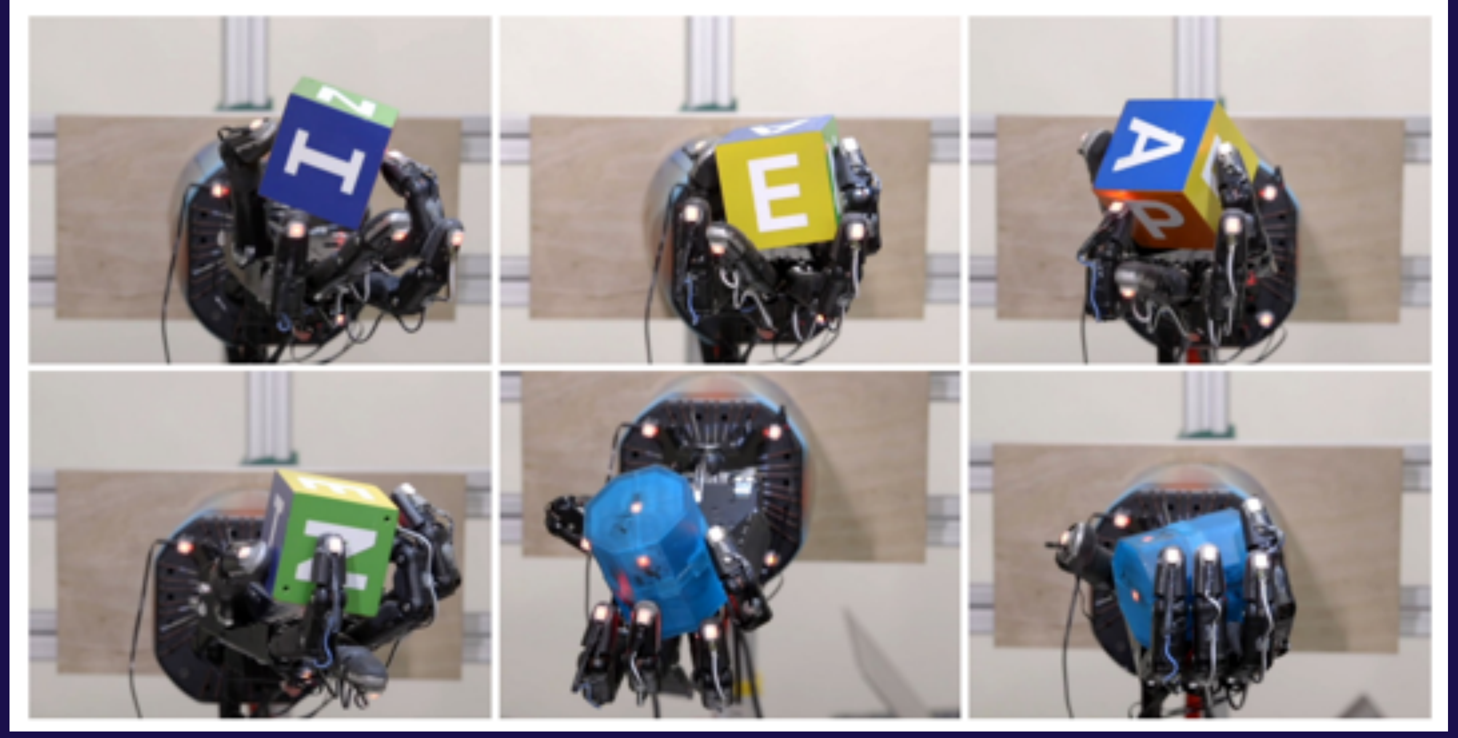
Meta RL: Maze Navigation



https://arxiv.org/abs/1611.02779 (Duan et al. '16)

Meta RL: Robotic Manipulation

Real world: robotic hand



https://arxiv.org/pdf/1808.00177.pdf

Simulation: randomized physics, robot dimensions, visual appearance

Meta RL: Robotic Manipulation

Simulation training: every episode for LSTM policy occurs in a different randomly sampled world

Policy is trained to quickly master a new world through several seconds of interaction

Trained policy is then applied to the real world

- Identifies real-world physics parameters after first few seconds of interaction

Meta RL: Robotic Manipulation

https://blog.openai.com/learning-dexterity/

Meta RL: Limitations of Approaches Described Previously

- Infinite data: can randomly sample tasks; assume distribution covers the one you care about
  - Doesn't emphasize generalization—performance given finite training set
- All "learning" is performed through RNN state updates: might be a poor inductive bias for what learning algorithm should look like
  - All meta-learning results so far use short horizons, 1000s of timesteps max
  - RL algorithms (policy gradients, Q-learning) find better solutions after longer training periods

Meta RL: changes to problem formulation

- Performance on a task is defined as average return after K episodes of learning
- Given a finite set of training tasks and a test set of tasks
- Train on training tasks as much as desired
- Goal: maximize performance on test set

Gym Retro

Prior RL research: achieve human-level performance at games

New challenge: solve a previously unseen game as fast as a human, given prior experience with similar games

Gym Retro

Gym Retro

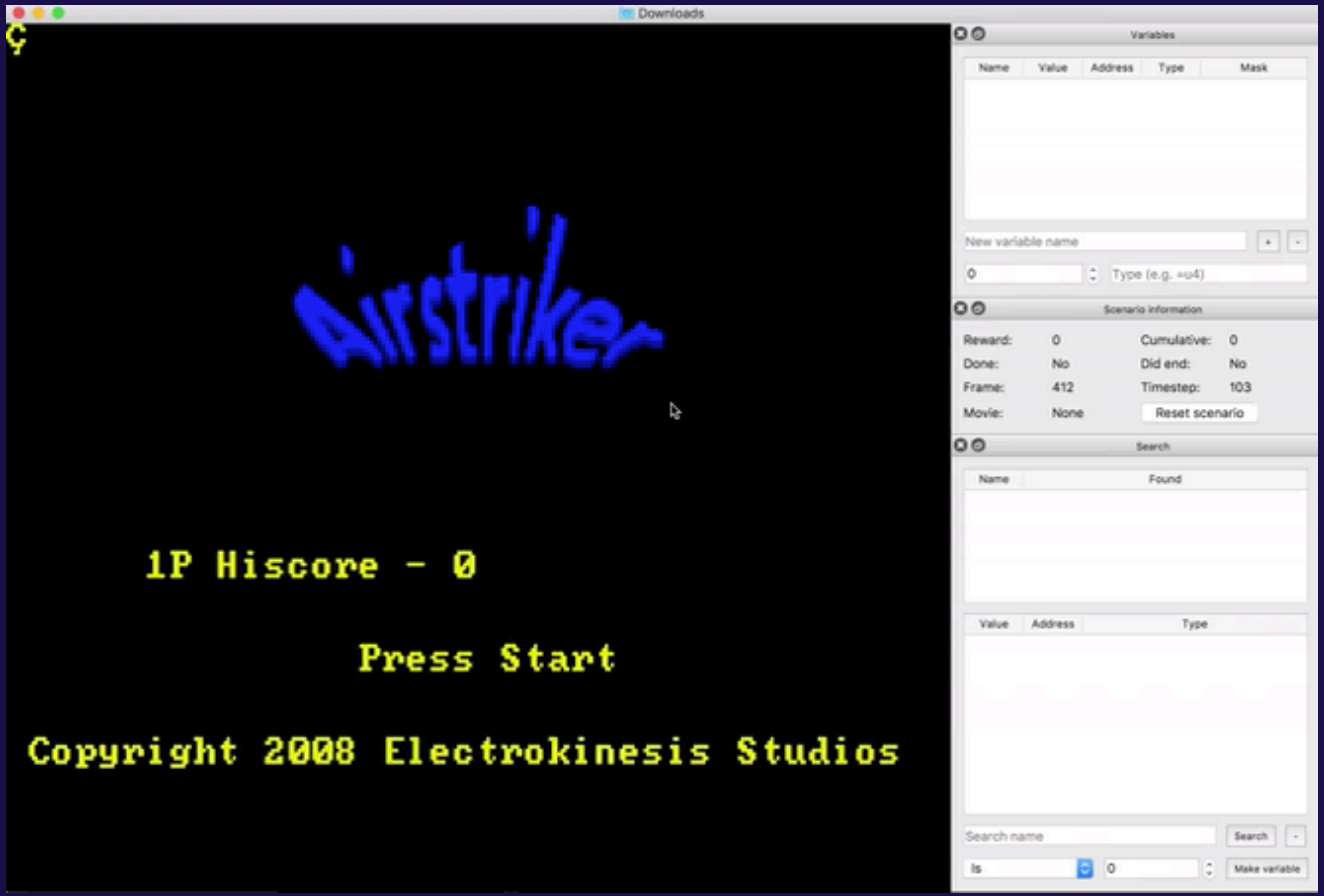
Over 1000 games integrated

- annotated with reward signal & episode completion

Use various game systems via emulators

Open source: https://github.com/openai/retro/

Gym Retro



Downloads
1P Hiscore - 0
Press Start
Copyright 2008 Electrokinesis Studios
Variables
Name
Value
Address
Type
Mask
Scenario information
Reward: 0
Cumulative: 0
Done: No
Did end: No
Frame: 412
Timestep: 103
Movie: None
Reset scenario
Search
Name
Found
Value
Address
Type
Search
Make variable

Gym Retro

Challenge: naive definition of reward sometimes leads to "farming" infinite loop

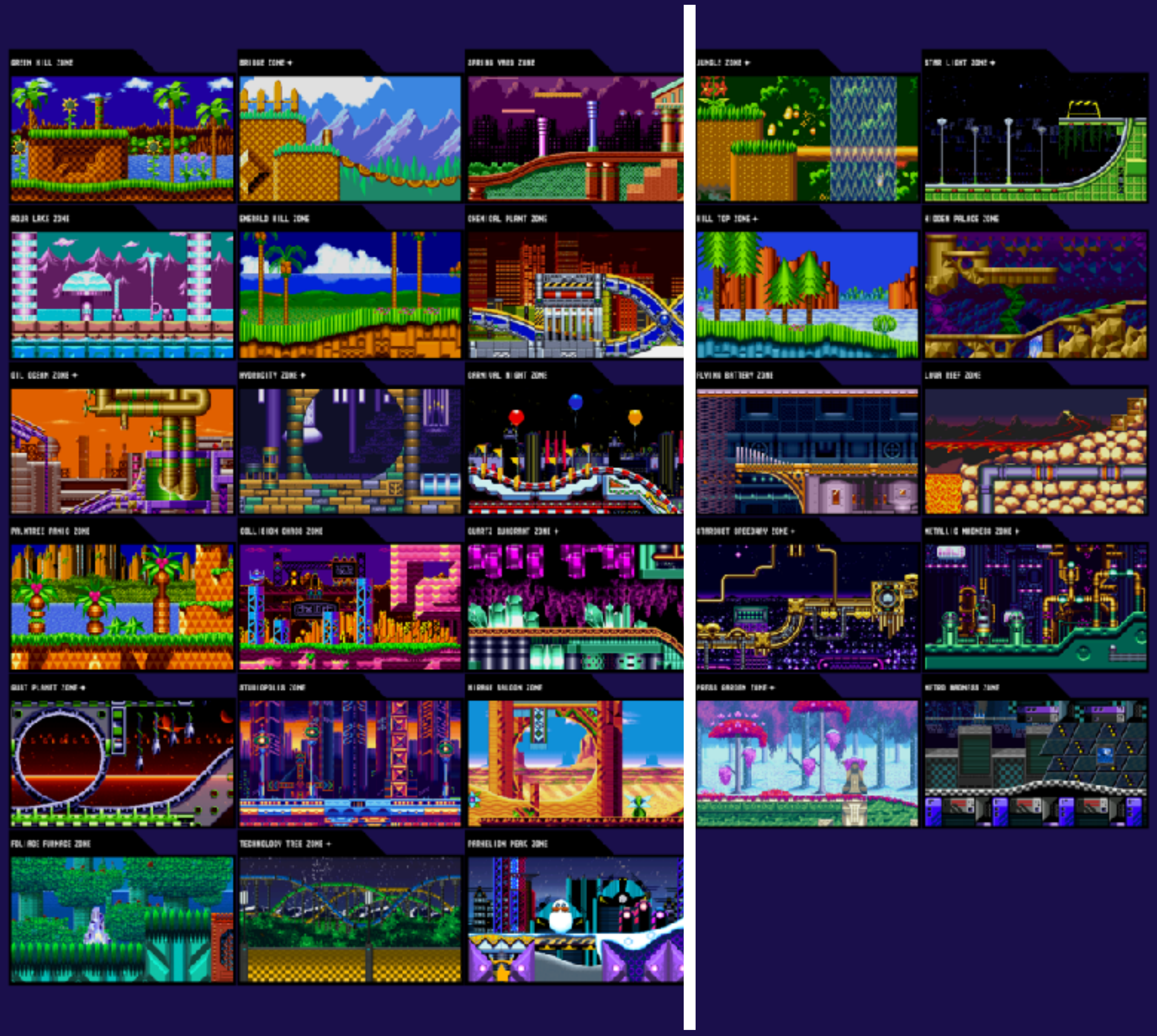
Sonic Benchmark

Reward: moving to the right

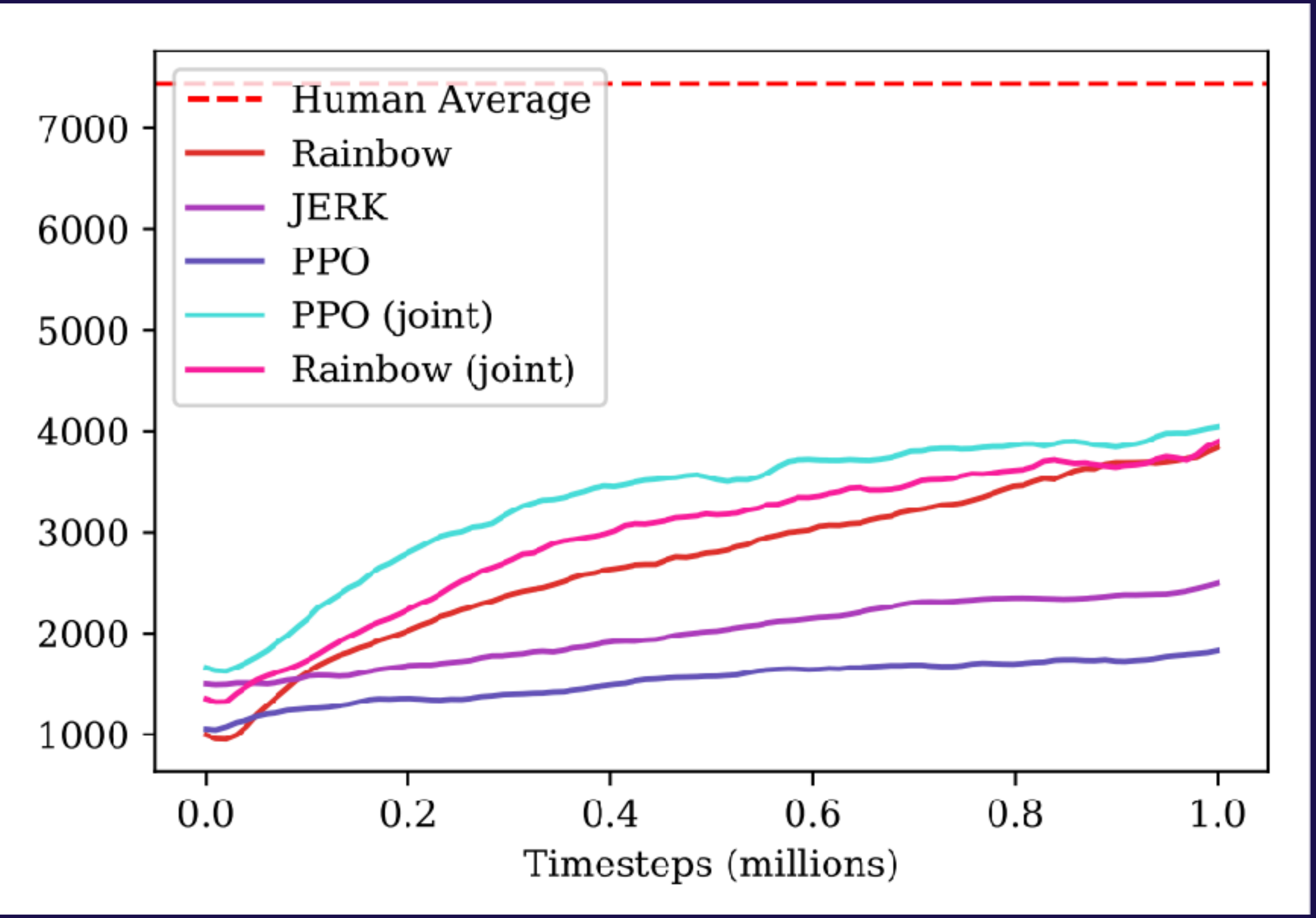
Avoid bad guys & traps, use gadgets

Sonic Benchmark

47 training tasks, 11 test tasks



Sonic Benchmark

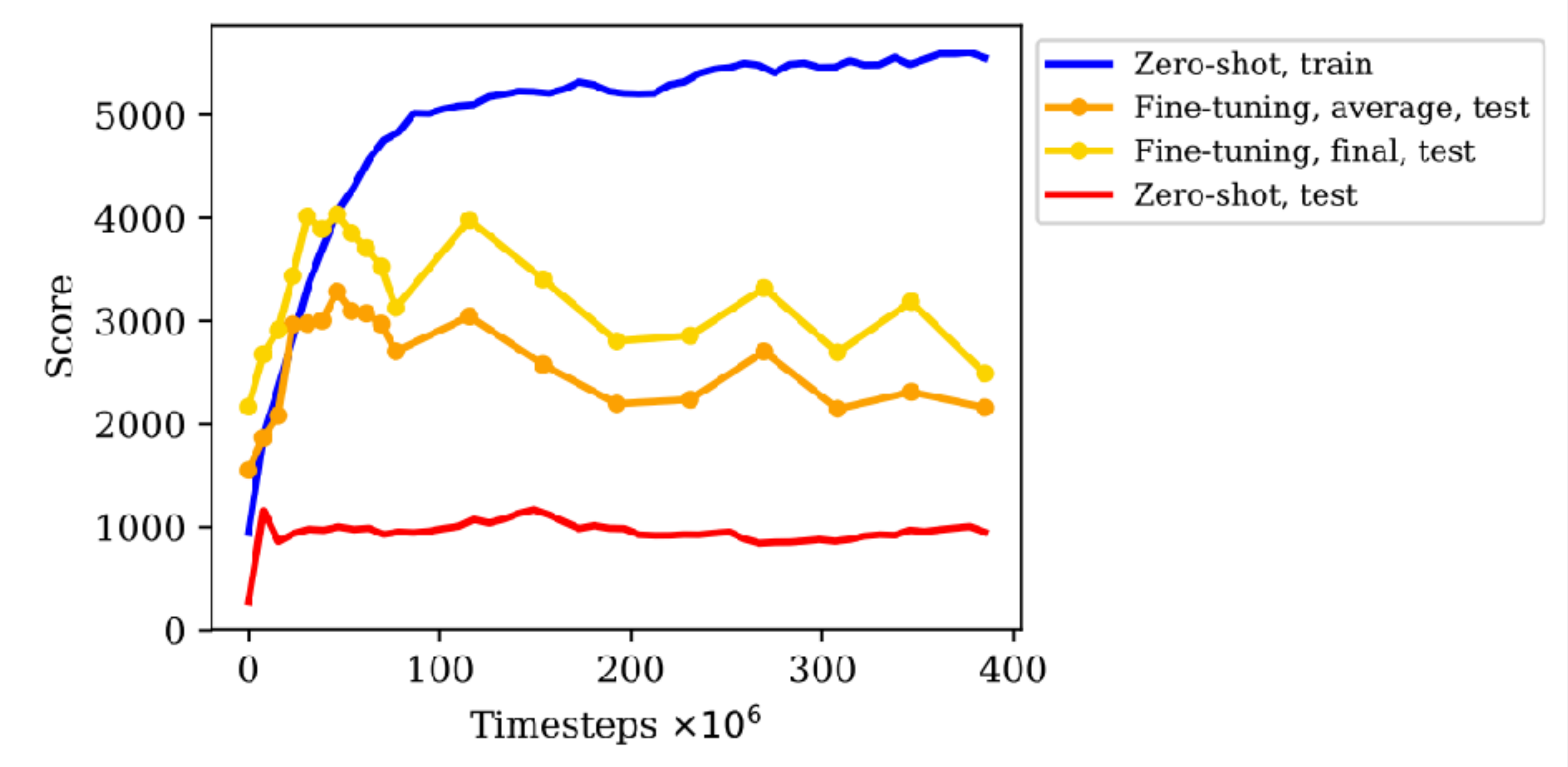


Tech Report, Nichol et al. 2018, https://arxiv.org/pdf/1804.03720.pdf

PPO Joint Training + Fine Tuning

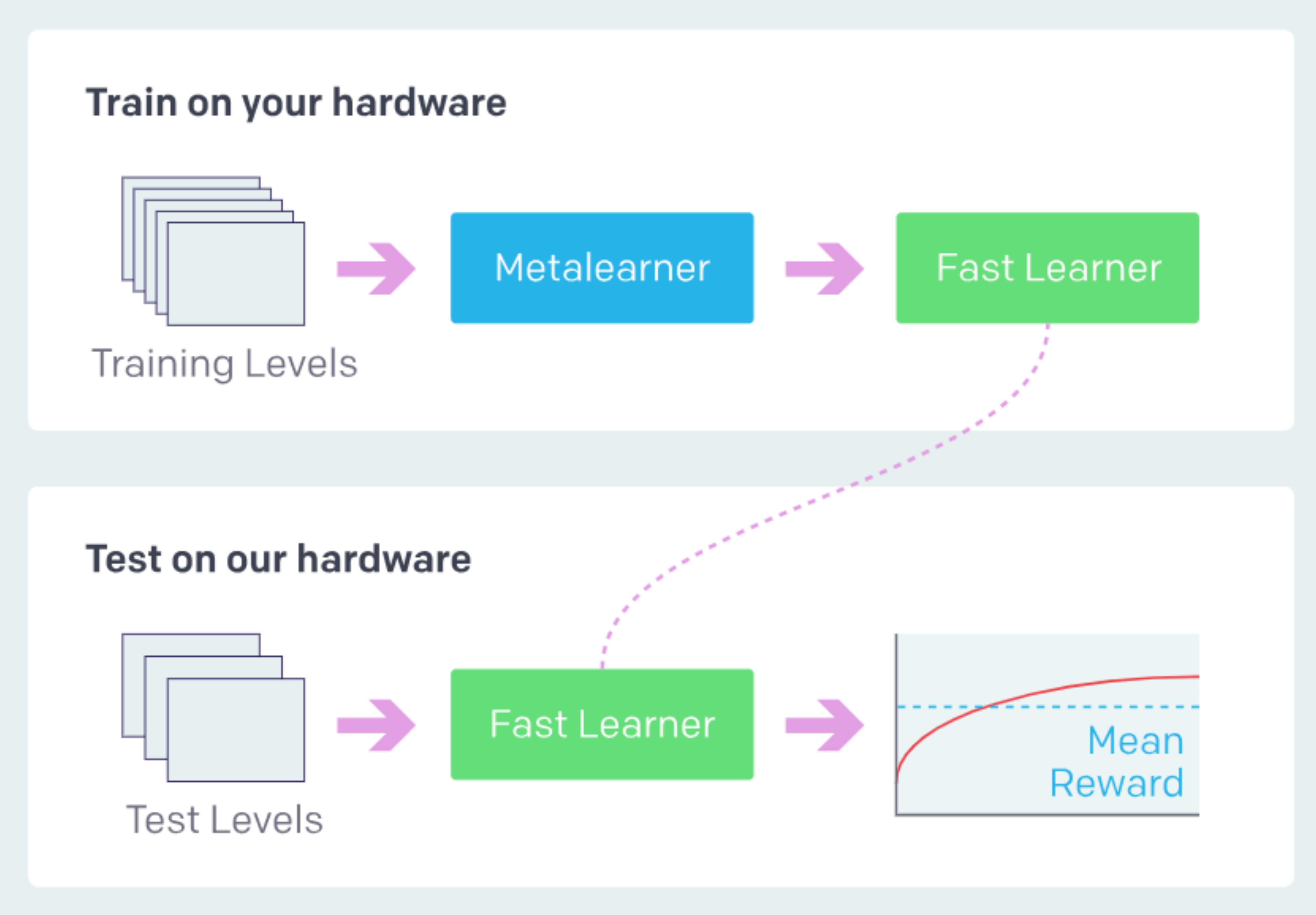
- Joint training: PPO on mixture of all training levels
  - use standard ConvNet policy
  - save weights at the end
- Fine tuning at test time: initialize network to saved weights, run PPO

PPO Joint Training + Fine Tuning



Tech Report, Nichol et al. 2018, https://arxiv.org/pdf/1804.03720.pdf

Retro Contest



Retro Contest

contest.openai.com

April 5 to June 5, 2018

Hired level designers to create 11 custom levels

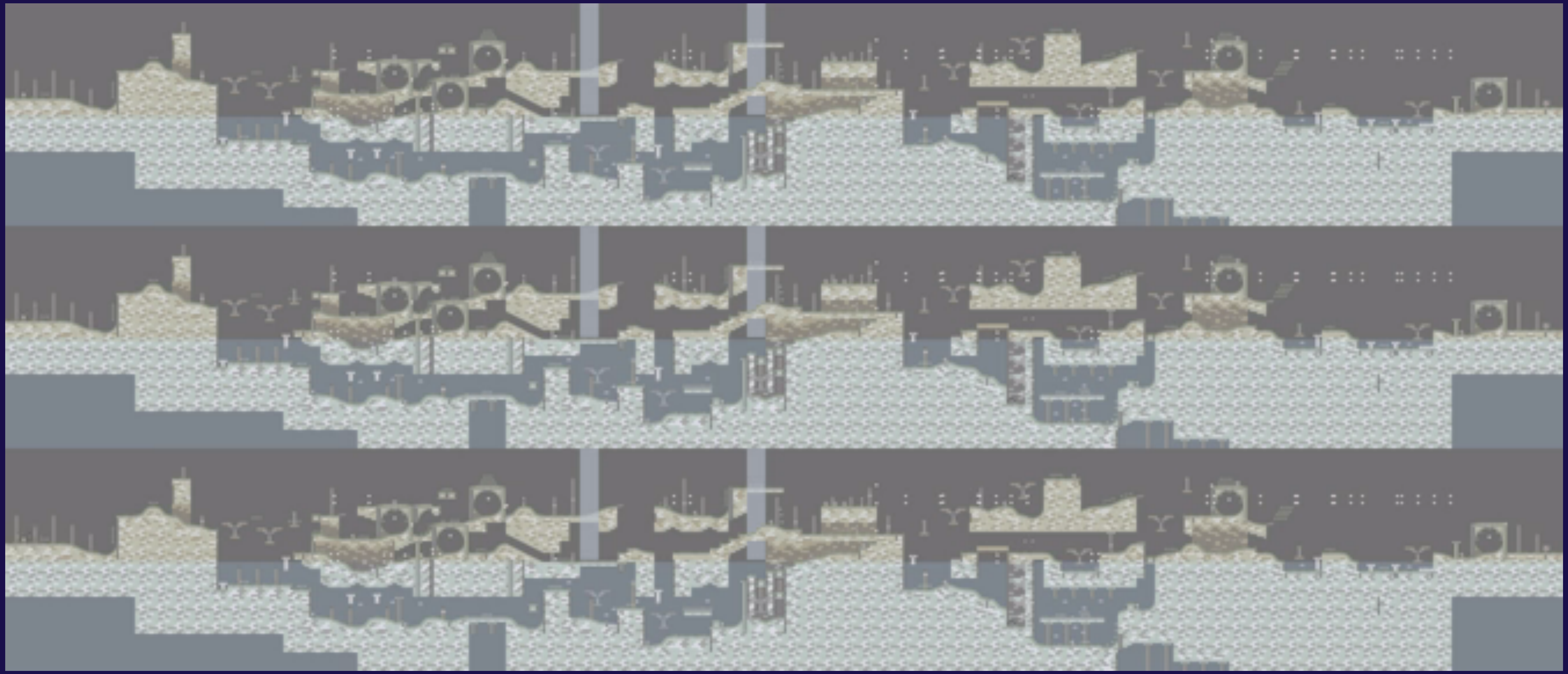
Also created 5 low-quality custom levels for leaderboard

Registration numbers:

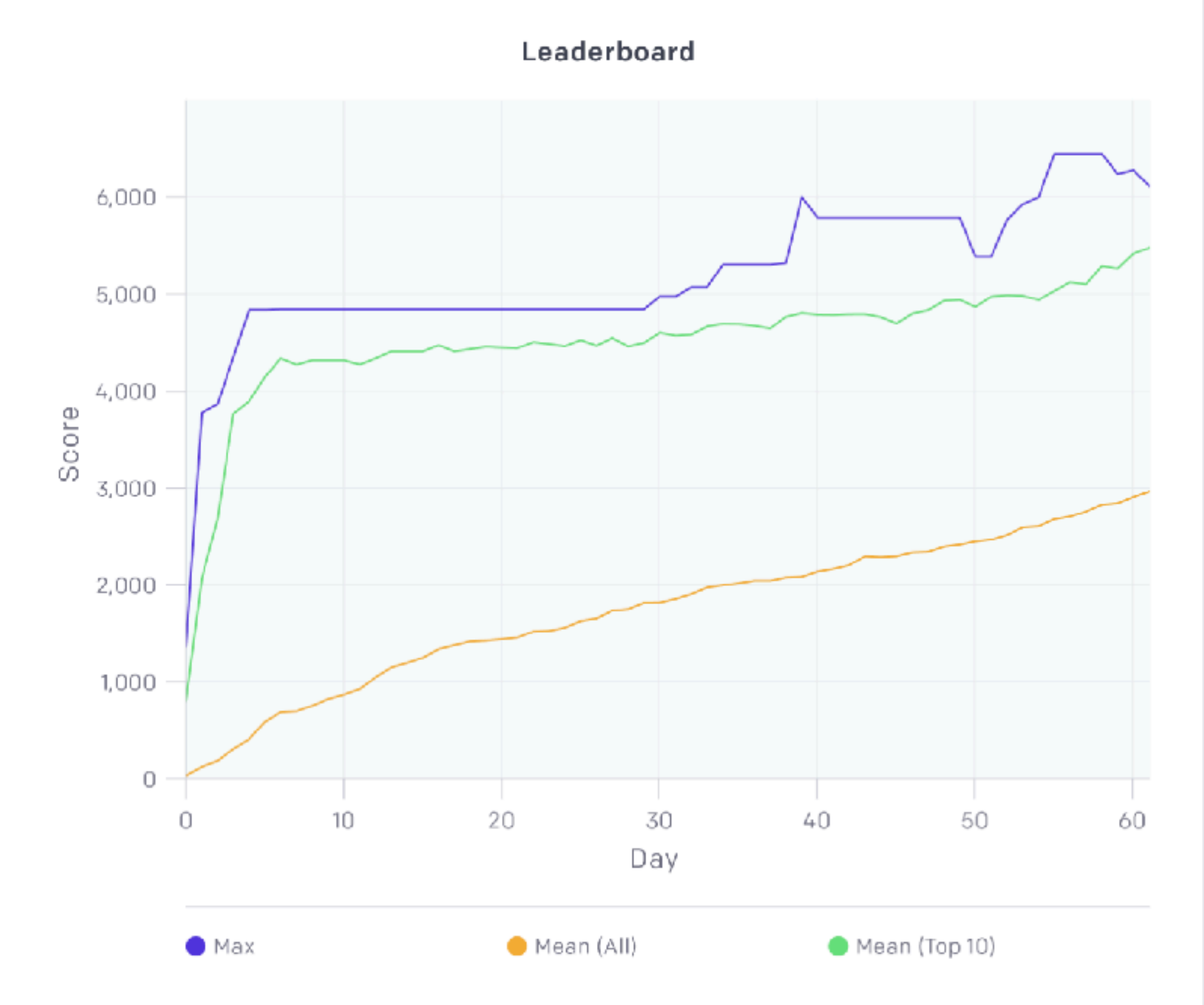
- 923 teams registered, 229 submitted solutions
- average 20 submissions per team

Retro Contest

Retro Contest



Retro Contest



Retro Contest

RANK	TEAM	SCORE
#1	Dharmaraja	4692
#2	mistake	4446
#3	aborg	4430
#4	whatever	4274
#5	Students of Plato	4269
	Joint PPO baseline	4070
	Joint Rainbow baseline	3843
	Rainbow baseline	3498

Retro Contest

RANK	TEAM	SCORE
#1	Dharmaraja	4692
#2	mistake	4446
#3	aborg	4430
#4	whatever	4274
#5	Students of Plato	4269
	Joint PPO baseline	4070
	Joint Rainbow baseline	3843
	Rainbow baseline	3498

Joint PPO

Rainbow

Joint PPO

Future Work

Improve performance on larger retro benchmark

- Exploration
- Unsupervised learning
- Hierarchy

Improve RNN-based meta-learning

- Better dealing with long time horizon: memory & credit assignment
- Better architectures

More contests!

Acknowledgements

Gym Retro dataset, results, and contest was joint work with Chris Hesse, Oleg Klimov, Alex Nichol, Vicki Pfau

Also highlighted work from OpenAI Robotics & Dota teams

RL^2 was with Rocky Duan & Pieter Abbeel

Thanks