Robotics at OpenAI

May 1, 2017

By Wojciech Zaremba

Why OpenAI?

- OpenAI’s mission is to build safe AGI, and ensure AGI's benefits are as widely and evenly distributed as possible.

Why OpenAI?

- OpenAI’s mission is to build safe AGI, and ensure AGI's benefits are as widely and evenly distributed as possible.
- Can we build a General Purpose Robot and deploy it in the most beneficial way to humans?
  - We have several ideas
  - But maybe you can help us through collaboration!

Why OpenAI?

- OpenAI’s mission is to build safe AGI, and ensure AGI's benefits are as widely and evenly distributed as possible.
- Can we build a General Purpose Robot and deploy it in the most beneficial way to humans?
    - We have several ideas
    - But maybe you can help us through collaboration!
- We're well positioned to do this, due to extraordinary researchers, engineers, and amount of compute

What is a General Purpose Robot ?

A robot that can solve a variety of tasks without being trained on them

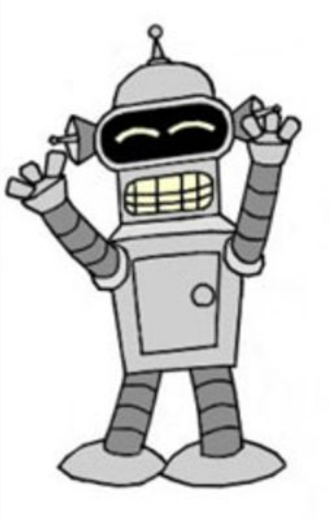
- Currently, all robots are trained to solve a single task
    - Roomba cannot drive a car or play chess

- Human has general purpose capabilities
    - Human can clean an apartment, drive a car, and play chess

What is a General Purpose Robot ?

A robot that can solve a variety of tasks without being trained on them

- Currently, all robots are trained to solve a single task
  - Roomba cannot drive a car or play chess



- Human has general purpose capabilities
  - Human can clean an apartment, drive a car, and play chess

What is a General Purpose Robot ?

We think that the following are critical components of the General Purpose Robot

- Training on diverse environments

- Obtaining complex behaviours

- Having a way to ask a robot to solve a task of interest

Overview

- Where to get rich, diverse data for robotics?
- How to obtain complex behaviors on robots?
- How to convey the intent of the task to the robot?

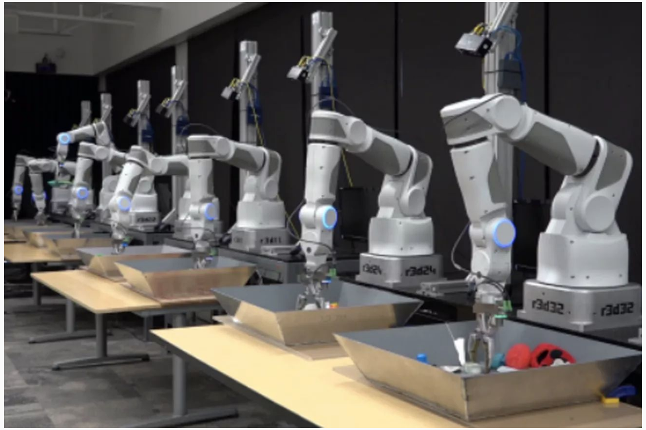
Overview

- ***Where to get rich, diverse data for robotics?***

- How to obtain complex behaviors on robots?

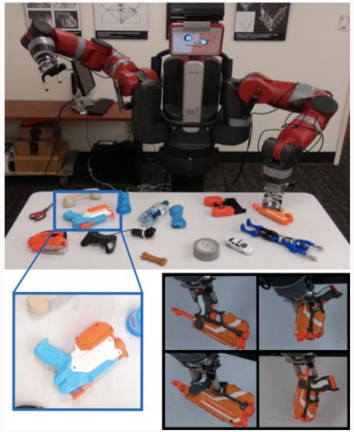
- How to convey the intent of the task to the robot?

Data from Physical Robots



Levine et al. 2016

Nair, Chen, Agrawal, et al. 2016

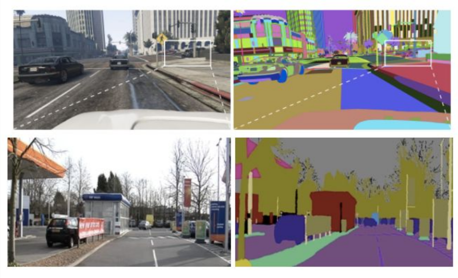


Pinto et al. 2016

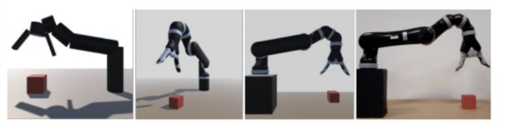
- Real data is closest to reality
- Real data is expensive
- Hard to obtain large diversity

Data from Simulation

Maximally Realistic Simulation

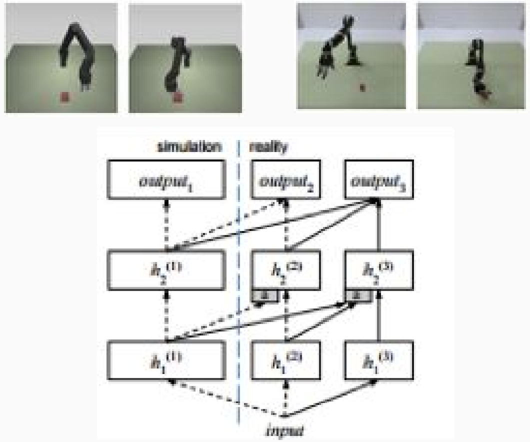


Richter et al. 2016



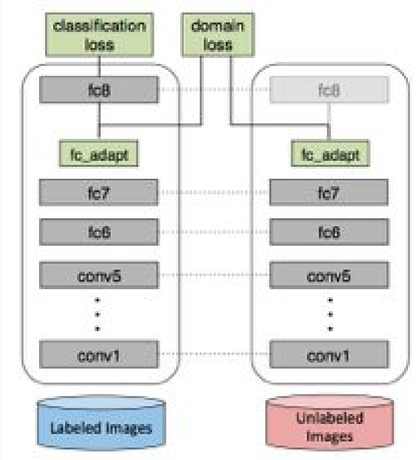
James et al. 2016

Fine-tuning



Rusu et al. 2016 (progressive nets)

Domain Adaptation



Tzeng et al. 2014

Do we ever need real data ?

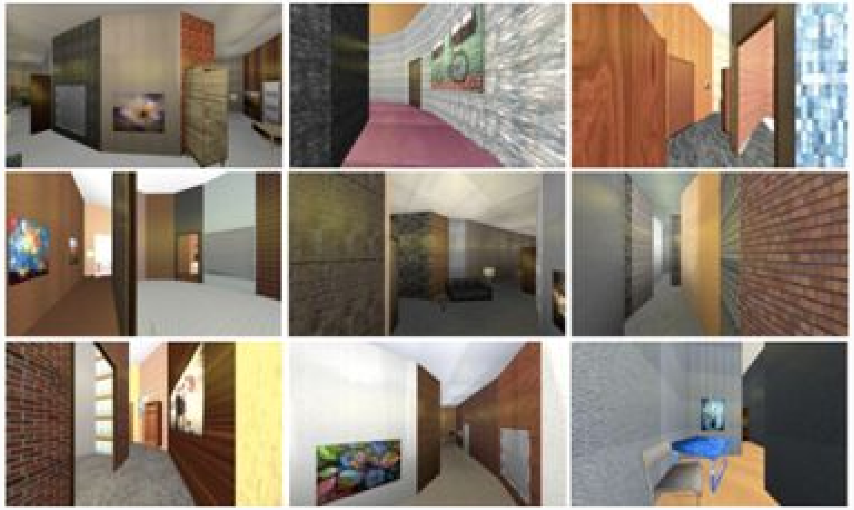
Does our simulation have to be photorealistic ?

Do we ever need real data ?

Does our simulation have to be photorealistic ?

Or, if the model sees enough simulated variation, might the real world may look like the other variation?

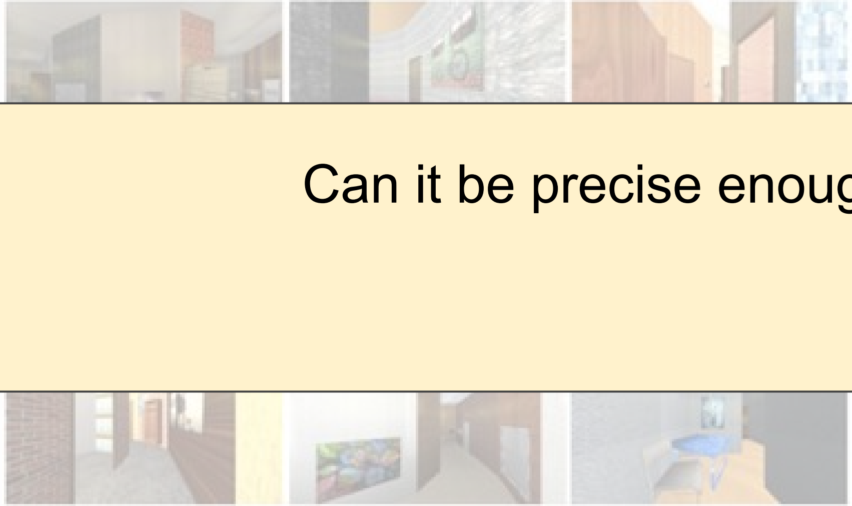
Domain Randomization – Prior Work and Inspiration



- Quadcopter collision avoidance
- ~40-50% of 1000m trajectories are collision-free

Fereshteh Sadeghi and Sergey Levine. (cad)ˆ2 rl: Real single-image flight without a single real image.

Domain Randomization – Prior Work and Inspiration

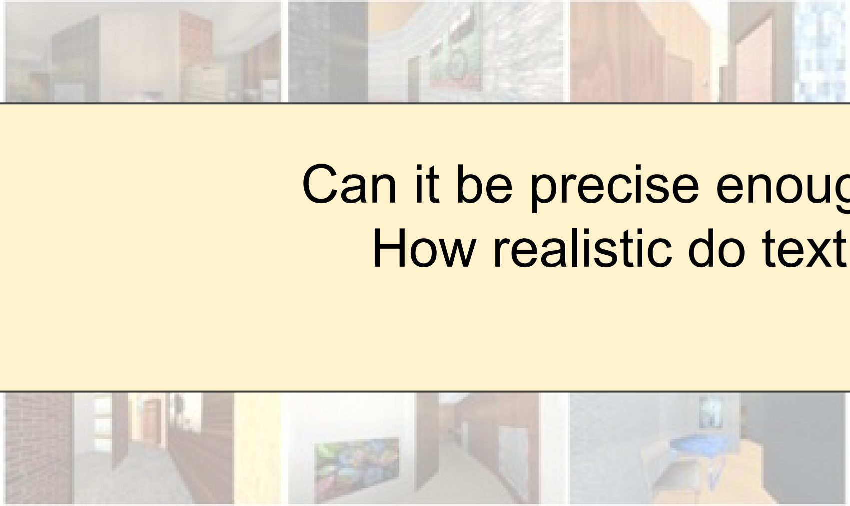


- Quadcopter collision avoidance

Can it be precise enough for manipulation ?

Fereshteh Sadeghi and Sergey Levine. (cad)ˆ2 rl: Real single-image flight without a single real image.

Domain Randomization – Prior Work and Inspiration

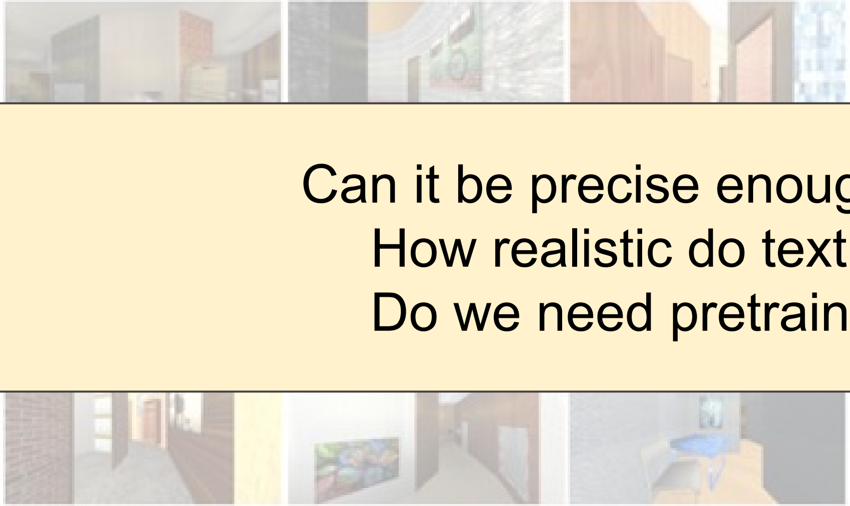


- Quadcopter collision avoidance

Can it be precise enough for manipulation ?
How realistic do textures need to be ?

Fereshteh Sadeghi and Sergey Levine. (cad)ˆ2 rl: Real single-image flight without a single real image.

Domain Randomization – Prior Work and Inspiration

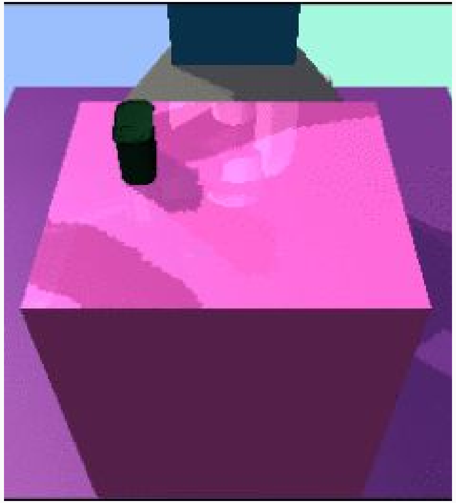


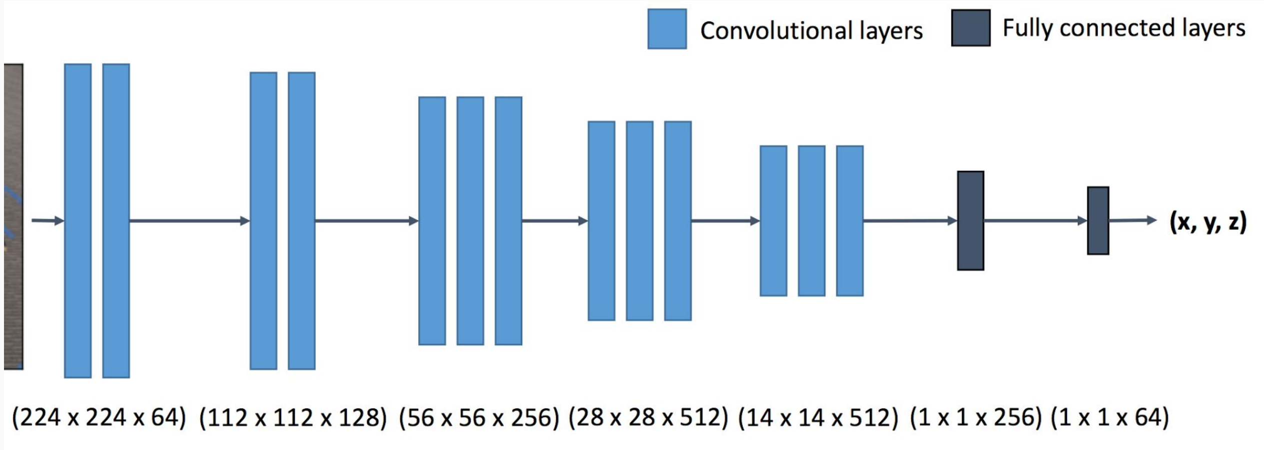
- Quadcopter collision avoidance

Can it be precise enough for manipulation ?
How realistic do textures need to be ?
Do we need pretraining on real data ?

Fereshteh Sadeghi and Sergey Levine. (cad)ˆ2 rl: Real single-image flight without a single real image.

Our Approach - Domain Randomization





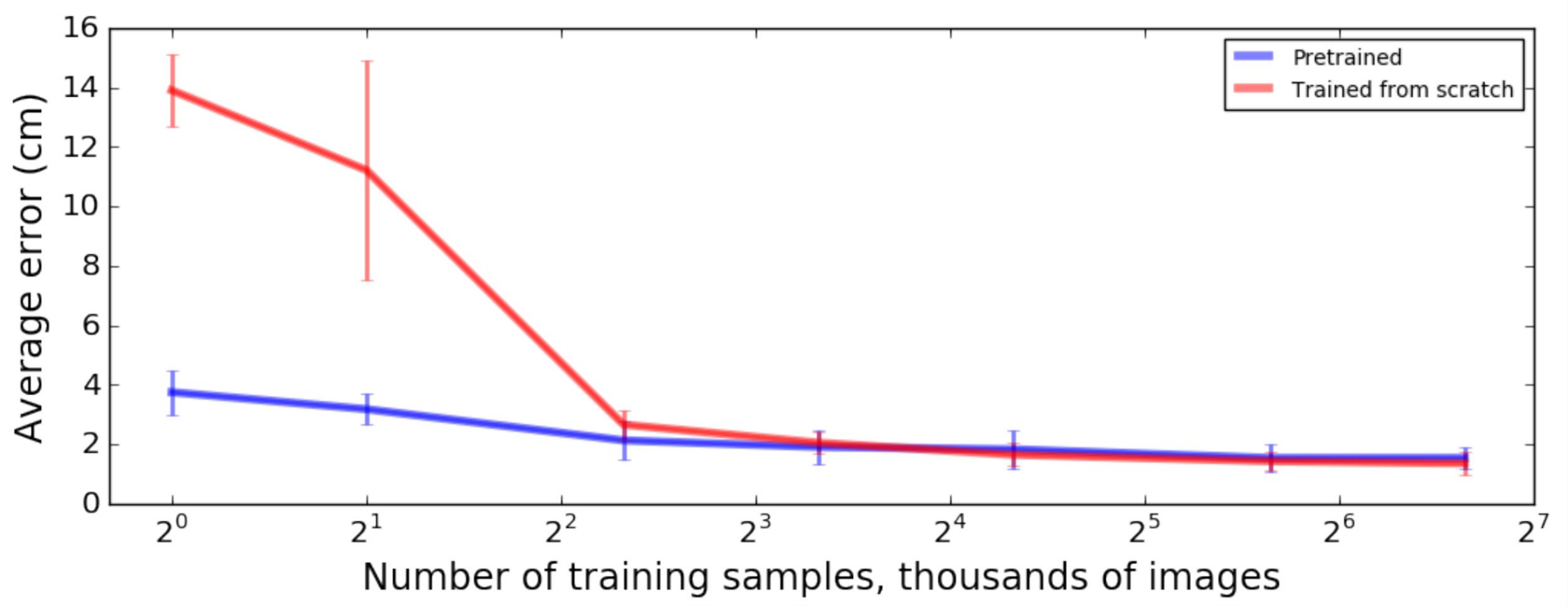
Randomized 100k scenes

- lighting
- textures (checkerboards and solid)
- camera position

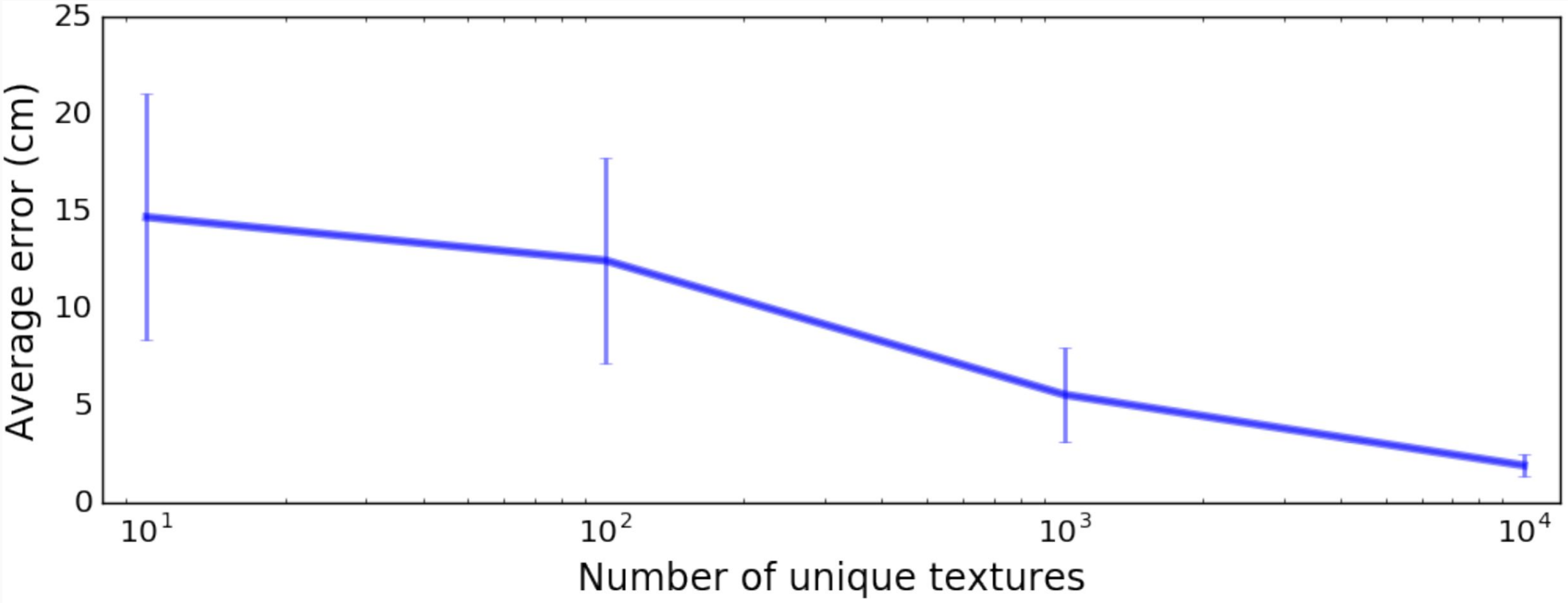
By Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, Pieter Abbeel

Deployed on the Physical Robot

How does it work?   More Data = Better

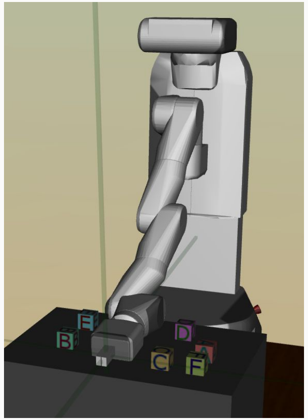


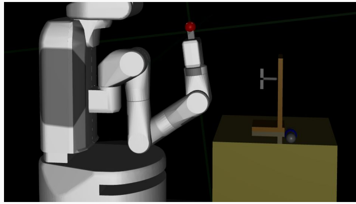
More Textures = Better

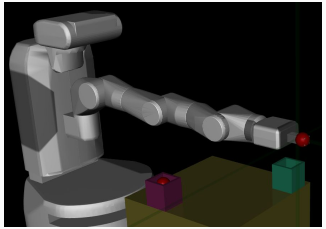


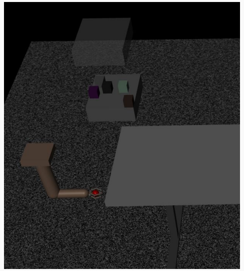
sim2real - Current Directions

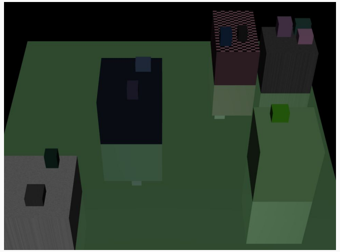
- Use multiple cameras, depth sensors and higher resolution images
- More randomization
- Apply to large number of tasks and complex generated works











Overview

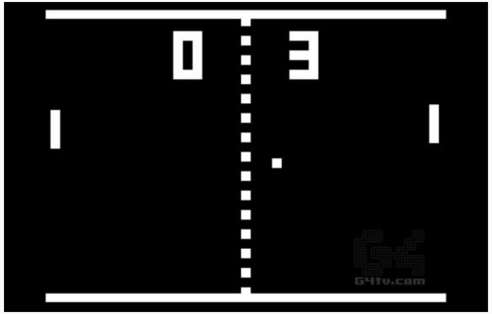
- Where to get rich, diverse data for robotics ?
  Approach: Domain randomization

- ***How to obtain complex behaviors on robots ?***

- How to convey the intent of the task to the robot?

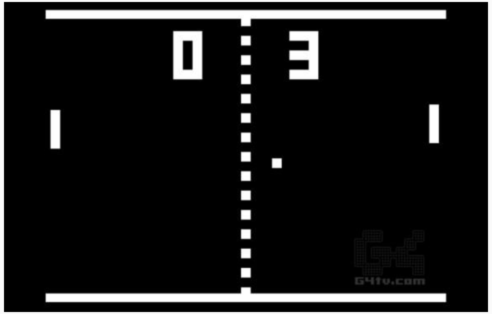
Reinforcement Learning

- Initial DeepMind Atari results - 1 week of training

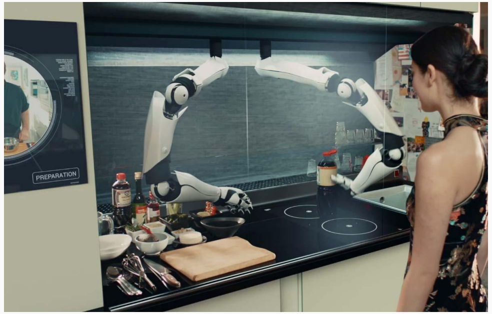


Reinforcement Learning

- Initial DeepMind Atari results - 1 week of training

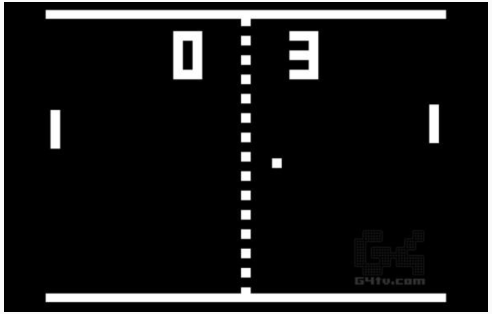


- We would like to train faster *and* significantly more complicated tasks

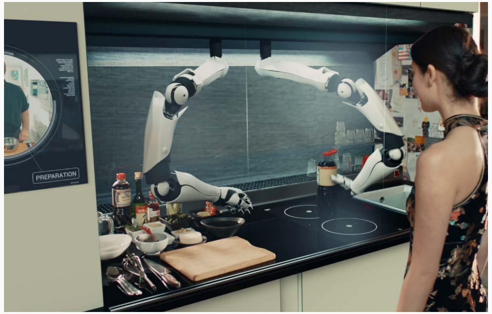


Reinforcement Learning

- Initial DeepMind Atari results - 1 week of training



- We would like to train faster *and* significantly more complicated tasks



Solution: Parallelization?

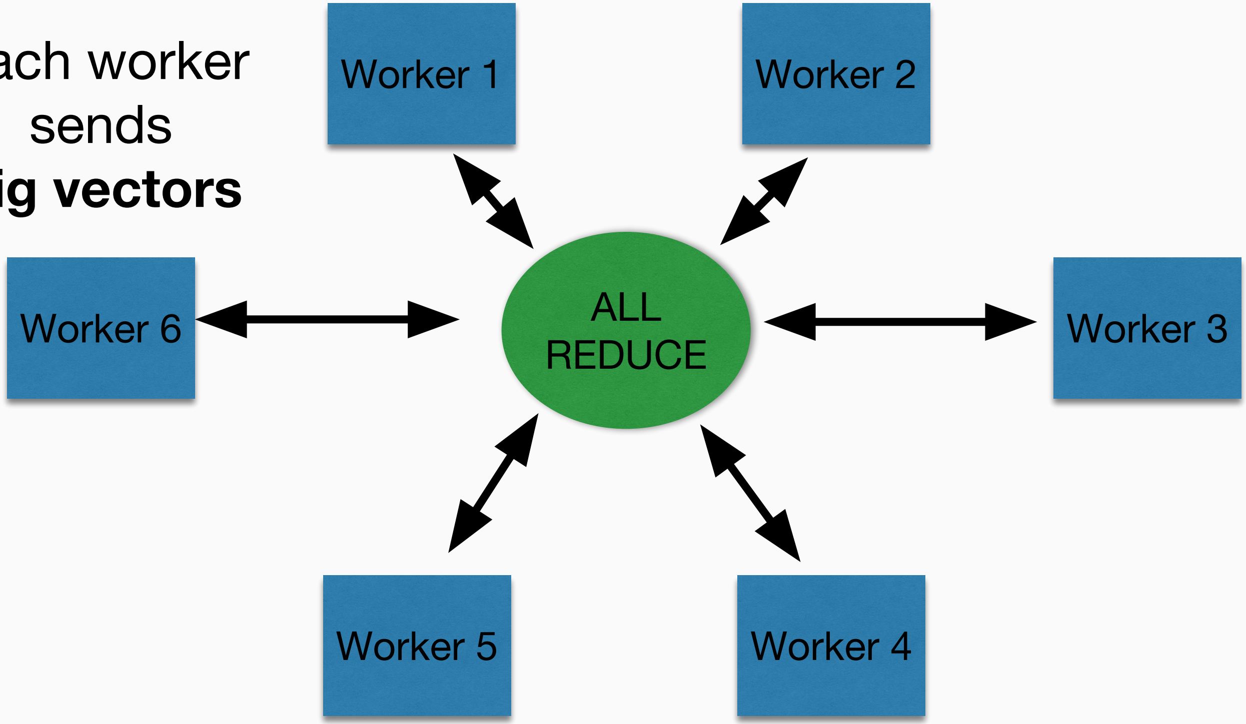
Distributed Reinforcement Learning

- GOogle ReInforcement Learning Architecture (Gorila)
  [Nair et al, 2015]
  - Parallel acting
  - Distributed replay memory
  - Parallel learning
  - Distributed neural network
  - Quite complex
- Asynchronous Advantage Actor Critic (A3C)
  [Mnih et al, 2016]

Why parallel RL cannot be faster ?

- Network communication is a bottleneck

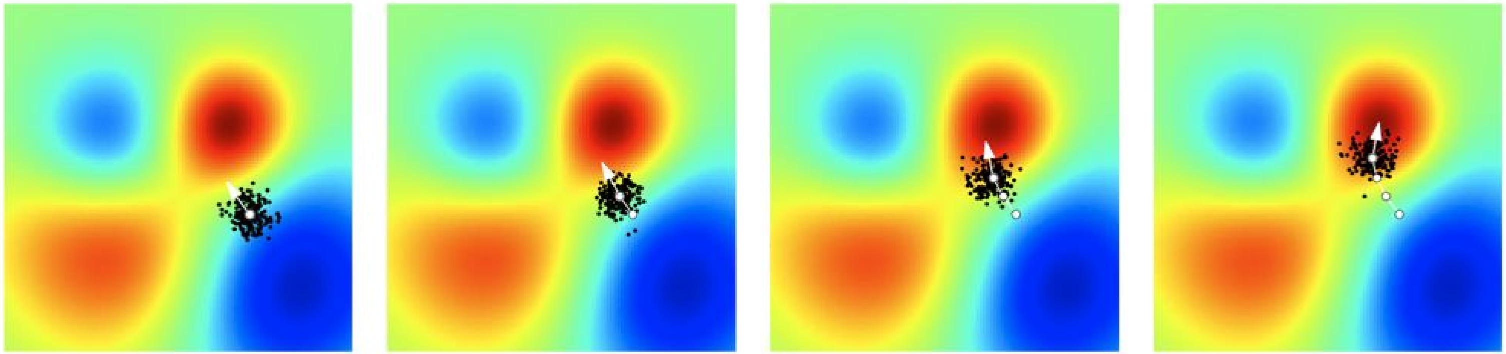
Each worker sends **big vectors**



Do we have to communicate all parameters ?

Evolution Strategies

- Simplest algorithm imaginable:
  - Add perturbation to the parameters
  - If the result improves, keep the change
  - Repeat



By Tim Salimans, Jonathan Ho, Peter Chen, Ilya Sutskever

Amenability to Parallelization

Classical RL

- Sample *action* perturbations

Evolution

Amenability to Parallelization

Classical RL

- Sample *action* perturbations

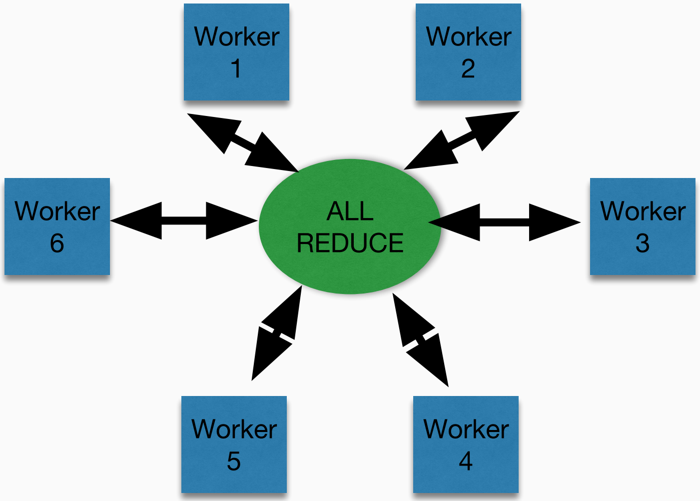
Evolution

- Sample *parameter* perturbations

Amenability to Parallelization

Classical RL

- Sample *action* perturbations
- Communicate *gradients/latest parameters*



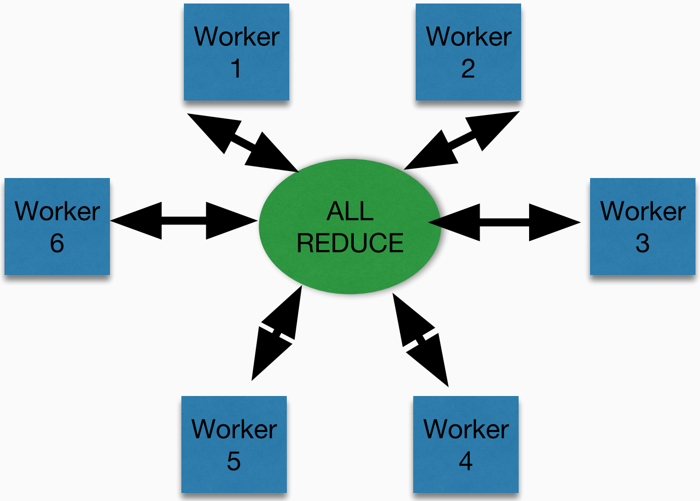
Evolution

- Sample *parameter* perturbations

Amenability to Parallelization

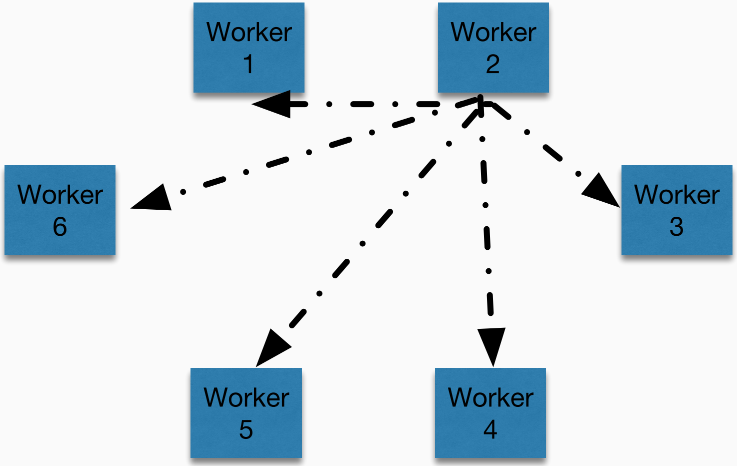
Classical RL

- Sample *action* perturbations
- Communicate *gradients/latest parameters*



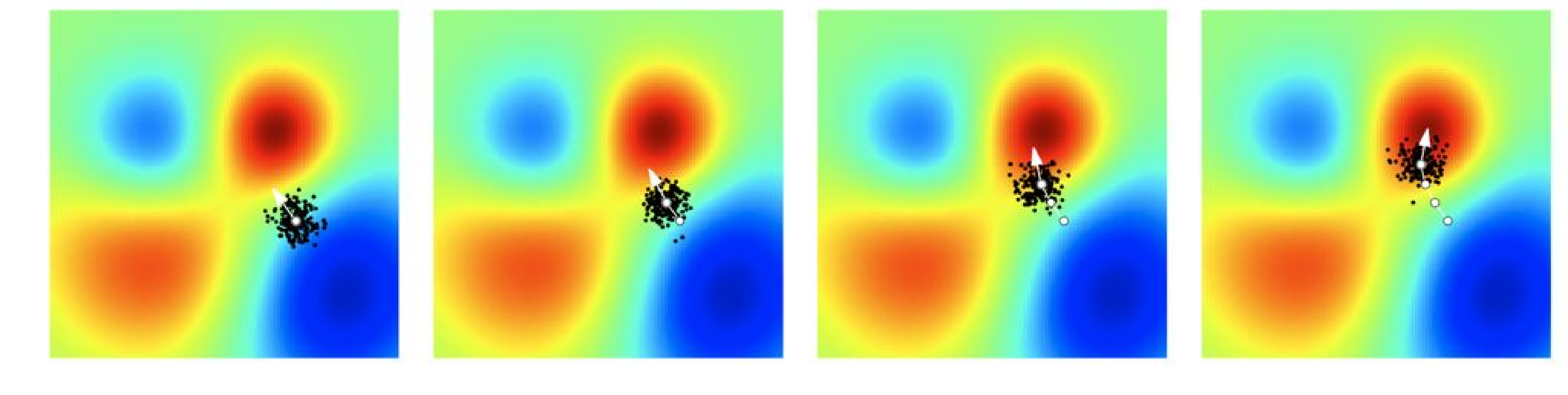
Evolution

- Sample *parameter* perturbations
- Communicate *reward and seed*



Evolution Strategies

- Neural networks have millions of parameters
- Folk Wisdom: There’s no chance for this kind of random hillclimbing to succeed

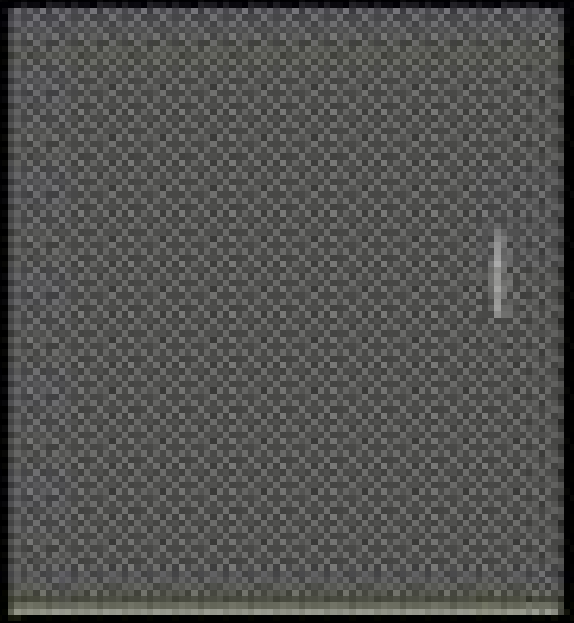


Surprise !

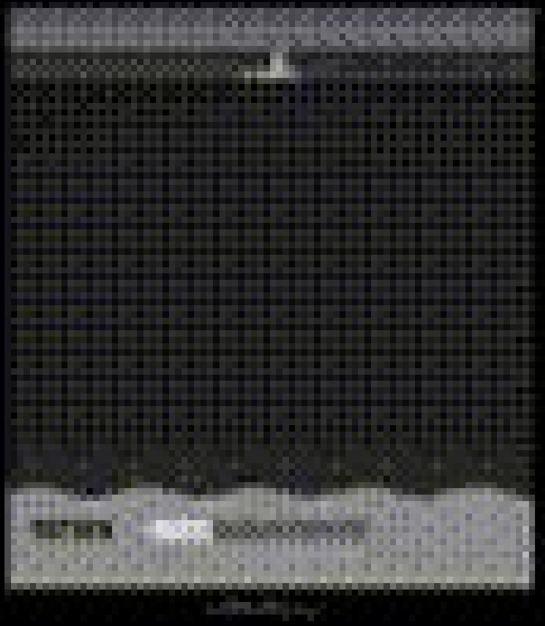
Evolution Strategies is competitive with today's RL algorithms on standard benchmarks

Evolution Atari Results

Pong



Seaquest



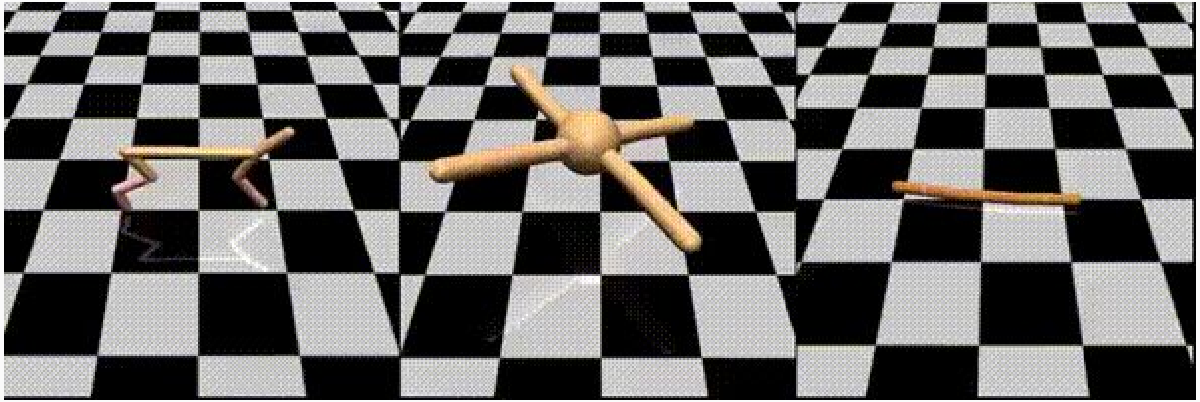
Beamrider

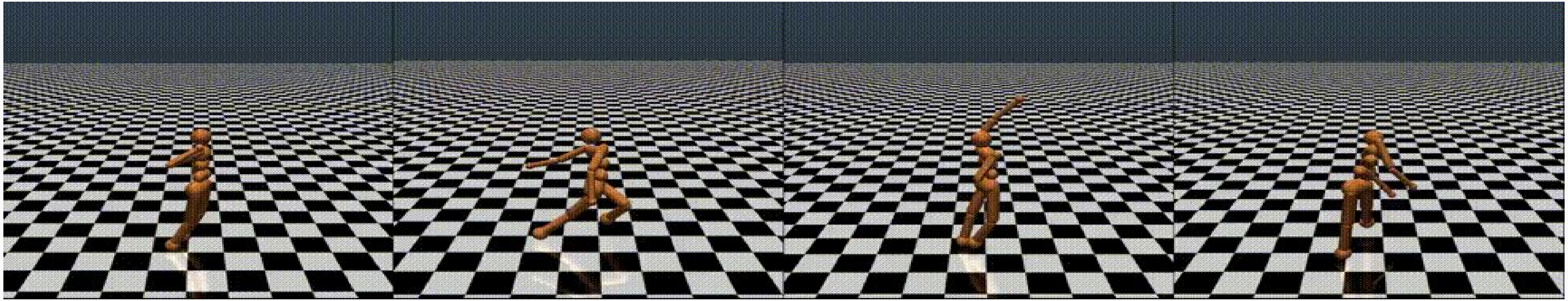
Evolution Atari Results

- Prior state-of-the-art on Atari in distributed RL: A3C [Mnih et al '16]
  - Training time 1 day

- Evolution Strategies
  - 1 hour with 720 cores matches A3C
  - 3x-10x more data
  - No backward pass
    - no need to store activations in memory
    - reduces compute per episode by 2/3

Evolution MuJoCo results

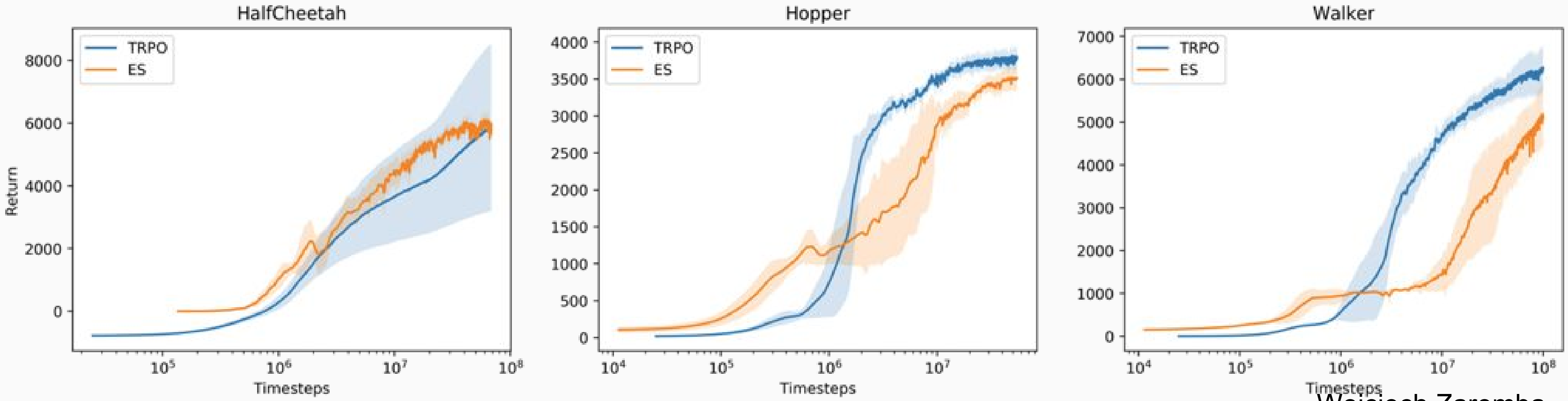




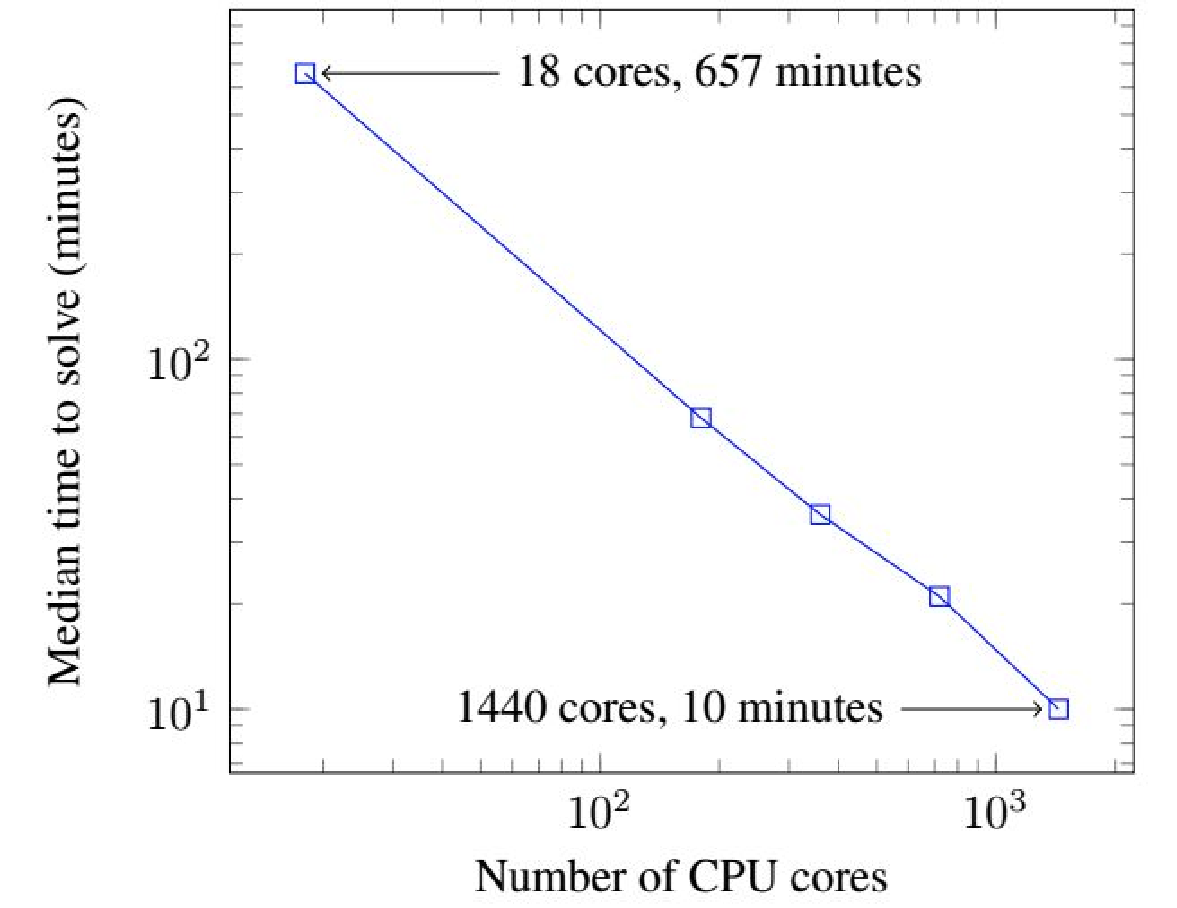
Wojciech Zaremba - OpenAI

Evolution MuJoCo results

- Evolution needs more data, but it achieves nearly the same result
- If we use 1440 cores, we need 10 minutes to solve the humanoid task, which takes 1 day with TRPO [Schulman et al., 2015] on a single machine



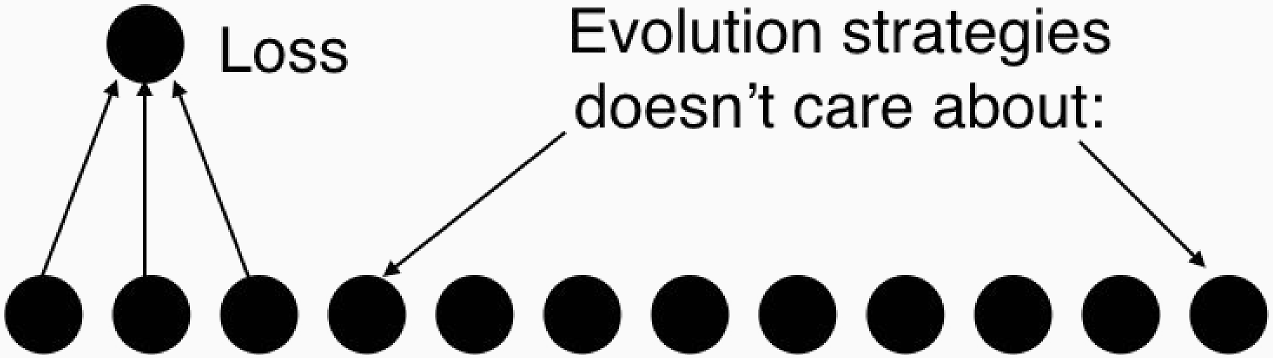
Quantitative results on the Humanoid MuJoCo task



What’s going on?

Fact: the speed of Evolution Strategies depends on the intrinsic dimensionality of the problem, not on the actual dimensionality

Intrinsic Dimensionality



- Evolution strategies *automatically discards* the irrelevant dimensions — even when they live on a complicated subspace!

Evolution Strategies - related work

- Evolution Strategies was proposed in 1977 by Rechenberg & Eigen
- Entire journals devoted to Evolution, e.g. Evolutionary Computation Journal

Evolution Strategies - Contribution

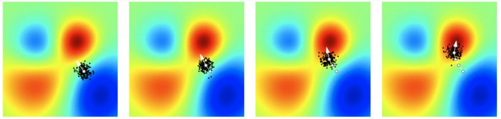
- Showed that evolution is competitive with today's existing RL algorithms on standard RL benchmarks

- Showed that evolution parallelizes extremely well

Overview

- Where to get rich, diverse data for robotics ?
  Approach: Domain randomization

- How to obtain complex behaviors on robots?
  Approach: Evolution



- **How to convey the intent of the task to the robot ?**

How to tell robot what's the task ?

- Language seems to be one option
    - Limits robot to tasks involving words that it knows

How to tell robot what's the task ?

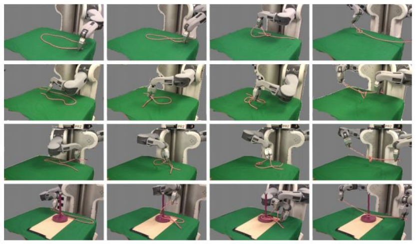
- Language seems to be one option
  - Limits robot to tasks involving words that it knows

- Alternative is to show the task

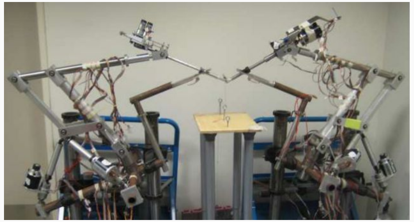
Learning from Demonstrations - Prior Work

- Abbeel, Coates, Ng: “Autonomous Helicopter Aerobatics through Apprenticeship Learning”

- Schulman et al. “Learning from Demonstrations Through the Use of Non-Rigid Registration”



- van den Berg et al. “Superhuman Performance of Surgical Tasks by Robots using Iterative Learning from Human-Guided Demonstrations”

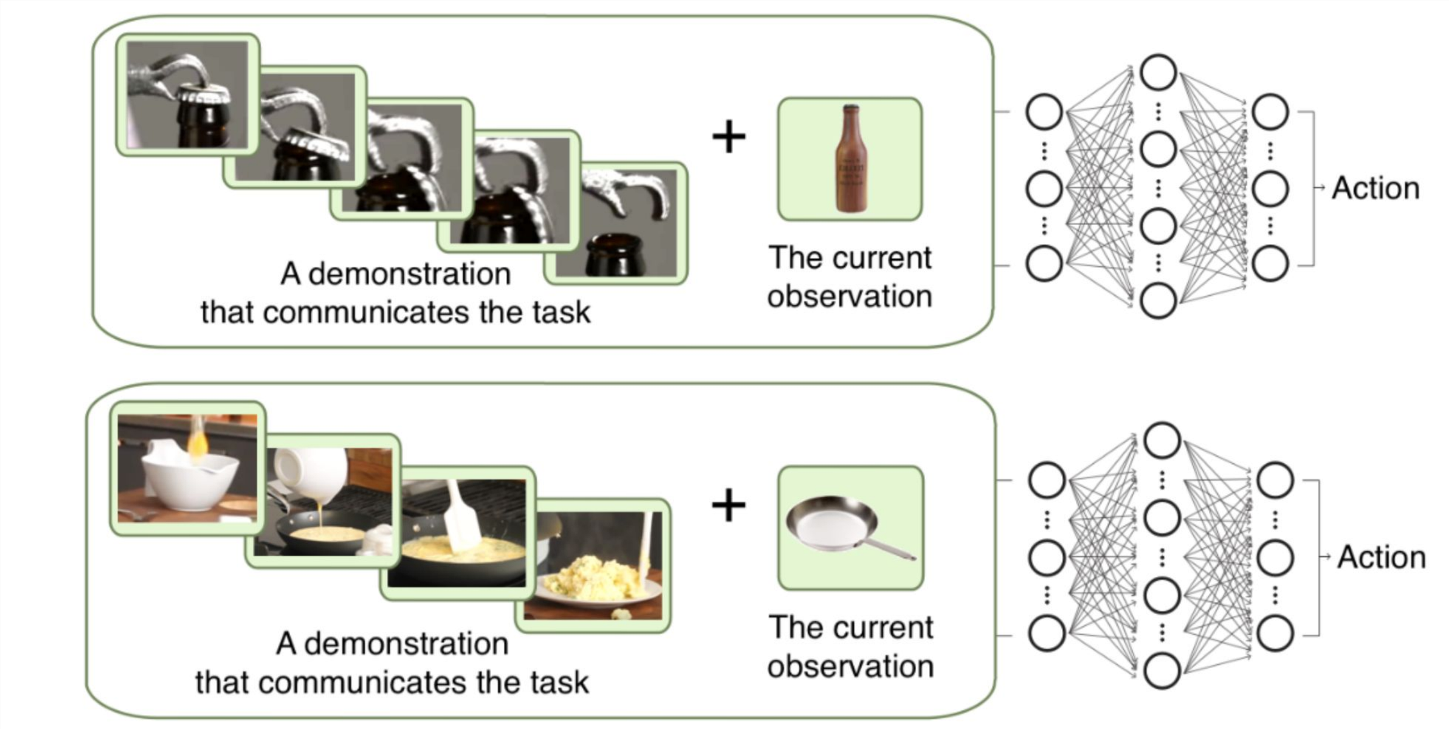


Prior work, proposes various imitation algorithms to learn from multiple demonstrations

Prior work, proposes various imitation algorithms to learn from multiple demonstrations

Instead, we *learn an imitation algorithm* that imitates based on a single demonstration

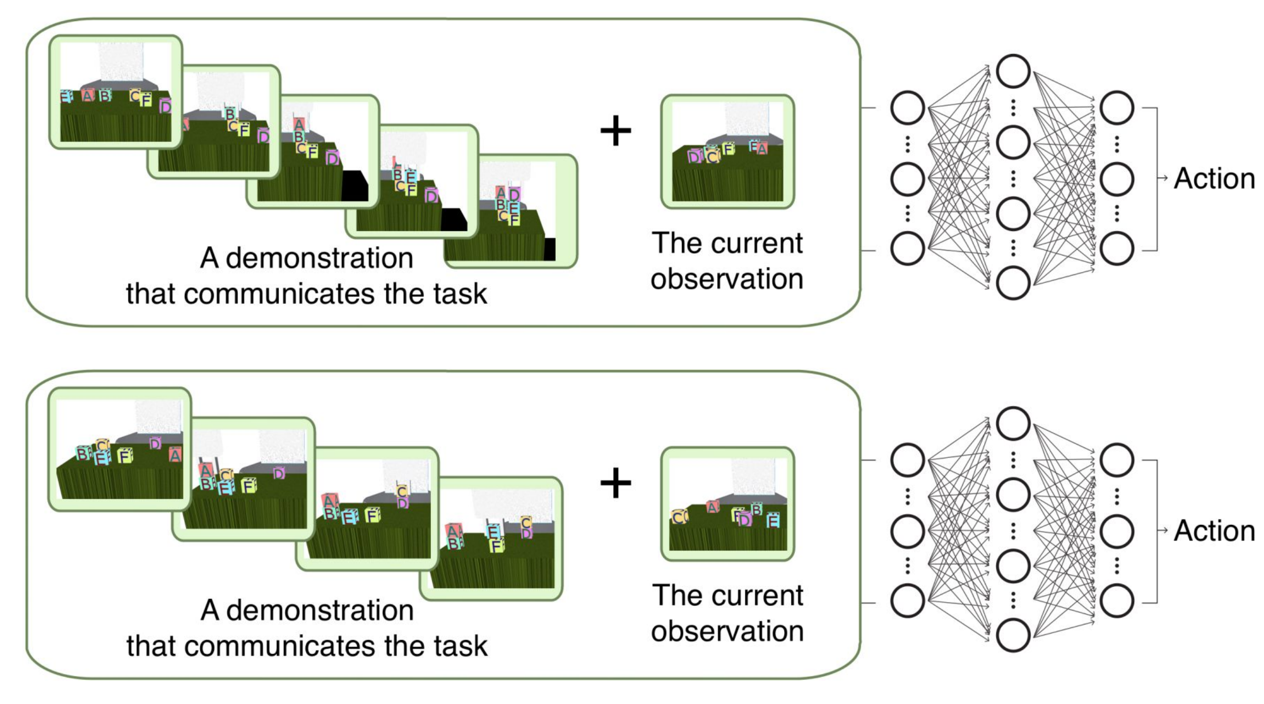
Learning the Imitation Algorithm - Idea



Learning the Imitation Algorithm: Algorithm

- Sample a task
- Sample an input demonstration from the task
- Sample a target demonstration from the task (in different initial condition)
- Train network given input demonstration to predict the target demonstration

Learning the Imitation Algorithm - Our Setup



By Rocky Duan, Marcin Andrychowicz, Bradly Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, Wojciech Zaremba

Simulated Block Stacking — Proof-of-Concept

- Each task is specified by the desired final layout

- Example: *abcd*
    - “Place *c* on top of *d*,
      place b on top of *c*,
      place a on top of *b*”

Simulated Block Stacking — Proof-of-Concept

- Each task is specified by the desired final layout

- Example: *abc def gh*
    - “Place *b* on top of *c*; a on top of *b*;”
    - “Place e on top of *f*; d on top of *e*;”
    - “Place *g* on top of *h*.”

Simulated Block Stacking — Proof-of-Concept

Size of dataset

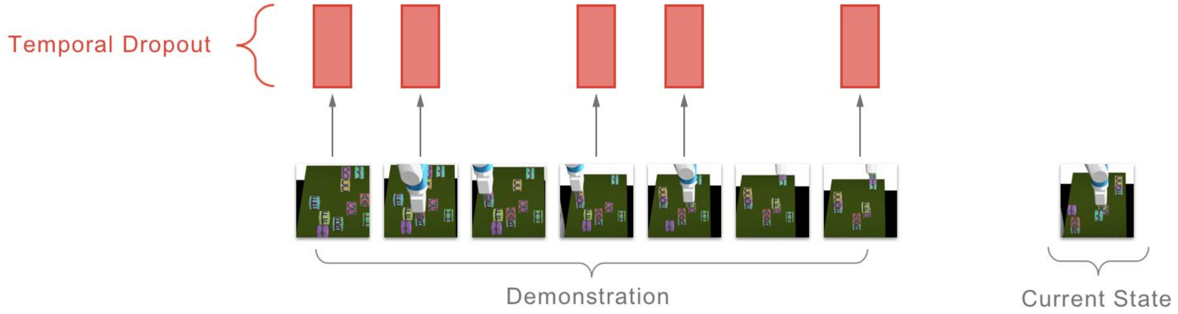
- Number of blocks vary from 2 to 10
- 183 distinct tasks, not counting equivalent permutations
- 140 tasks for training, and 43 tasks for testing

Simulated Block Stacking — Proof-of-Concept

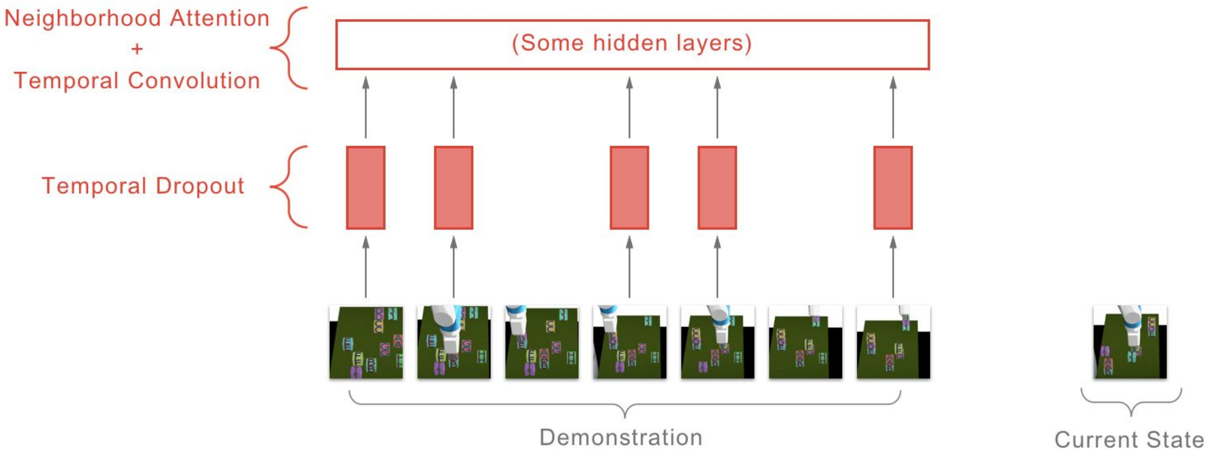
- Works with demonstrations of different size
- Works with very long demonstrations
- Works with variable number of blocks

Architecture

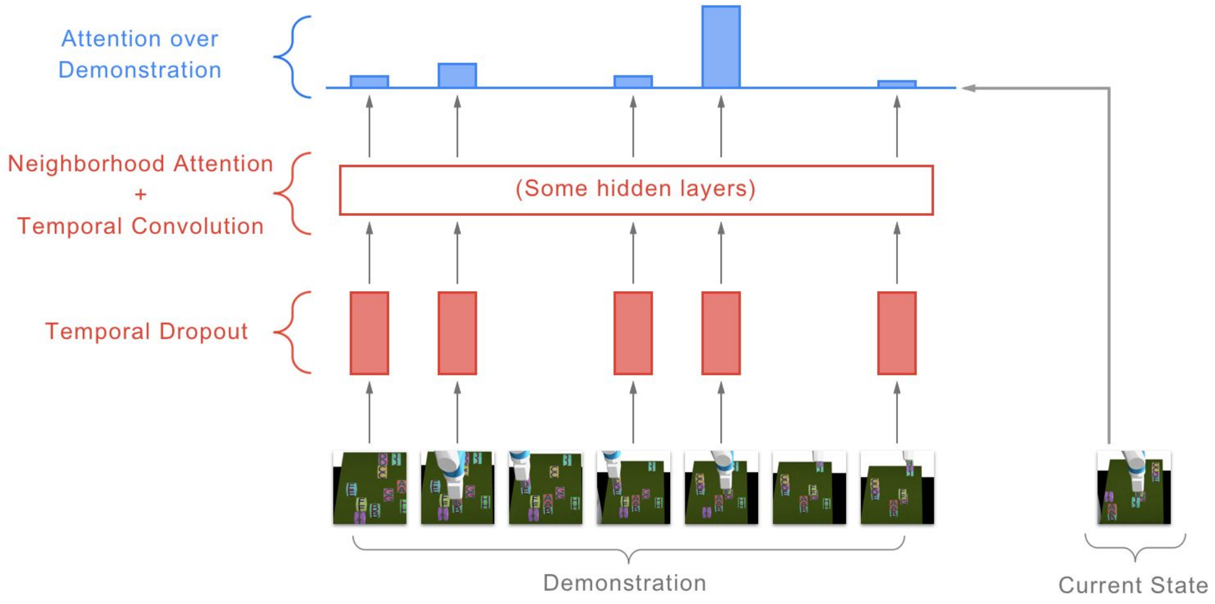
Architecture



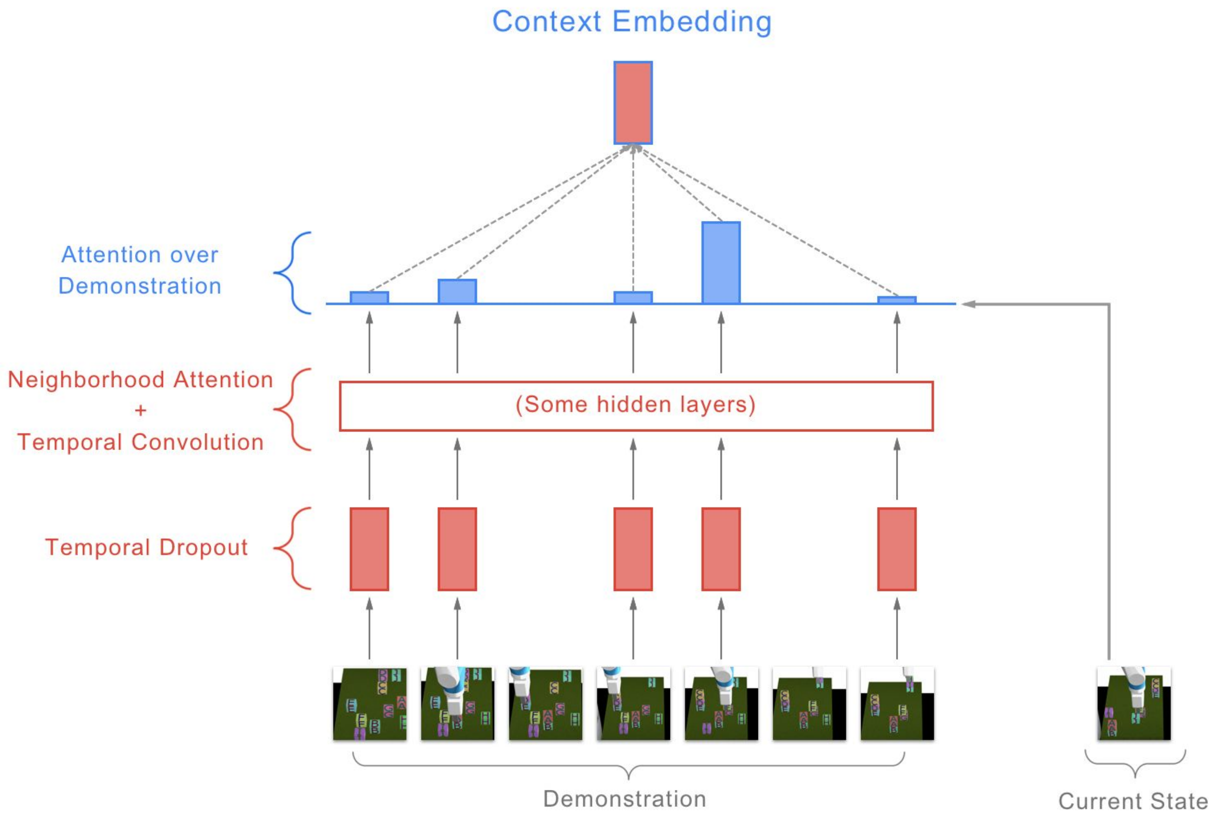
Architecture



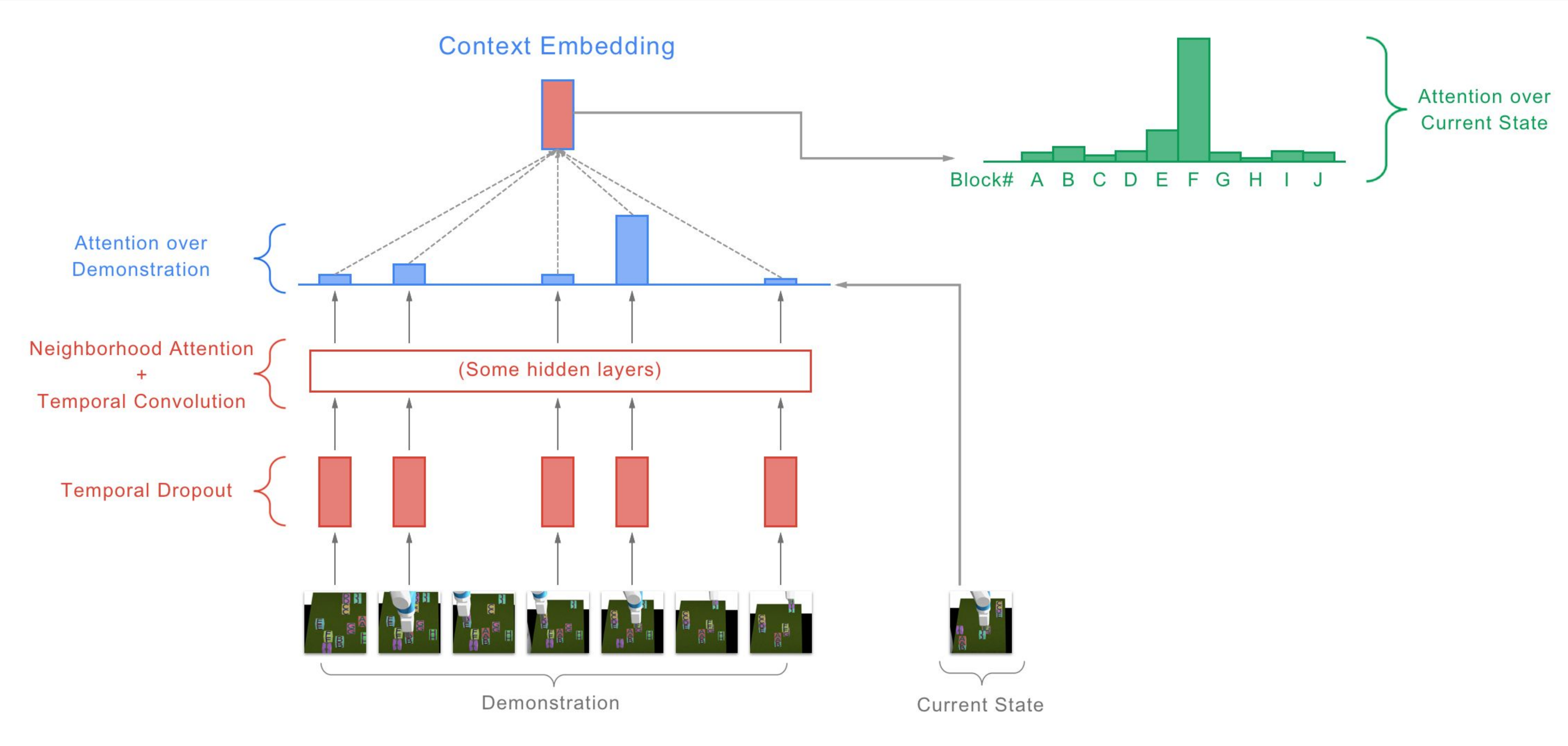
Architecture



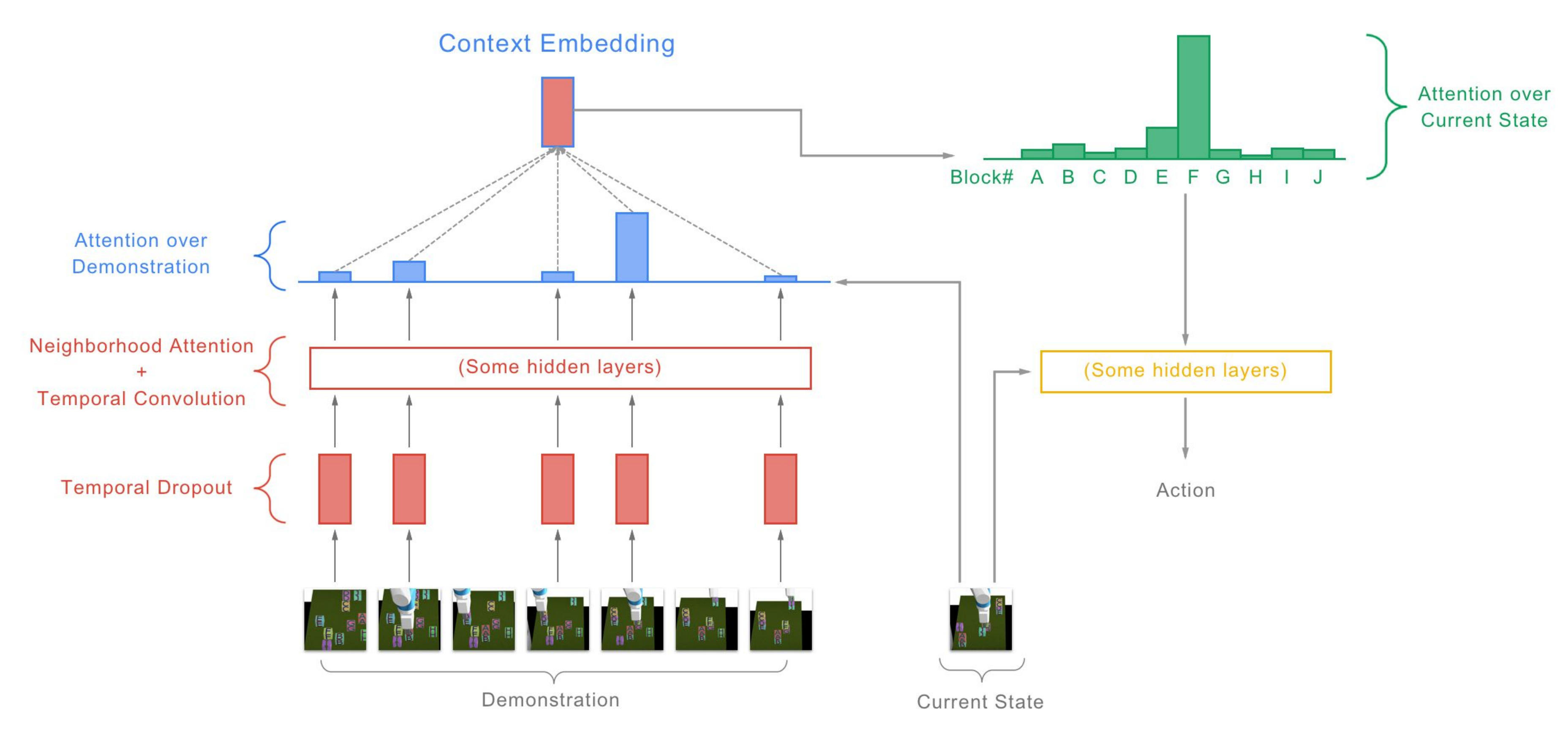
Architecture



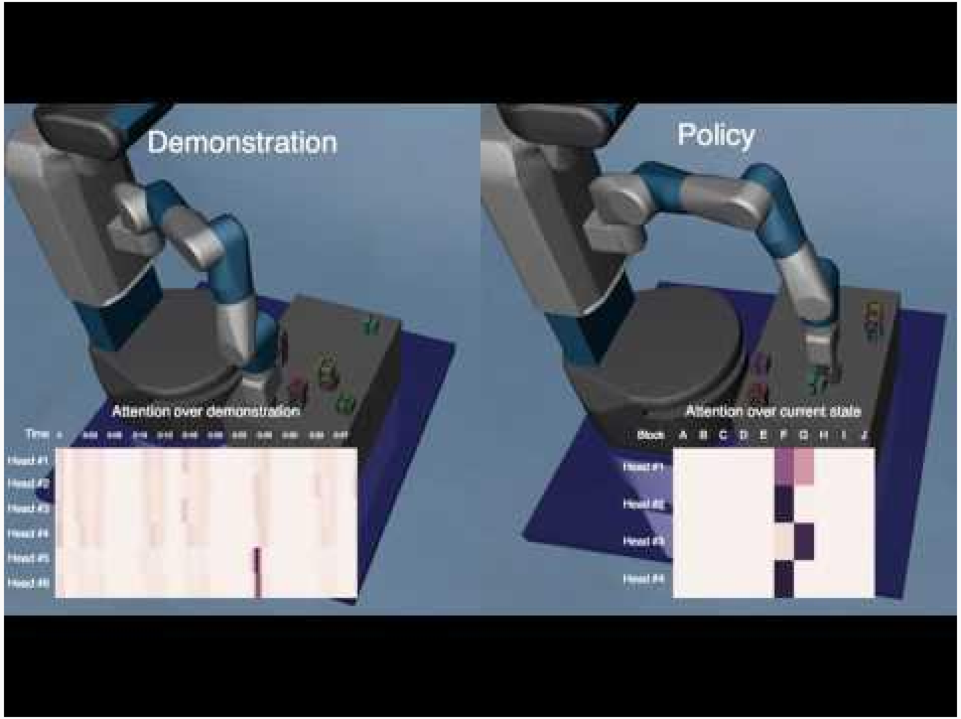
Architecture



Architecture

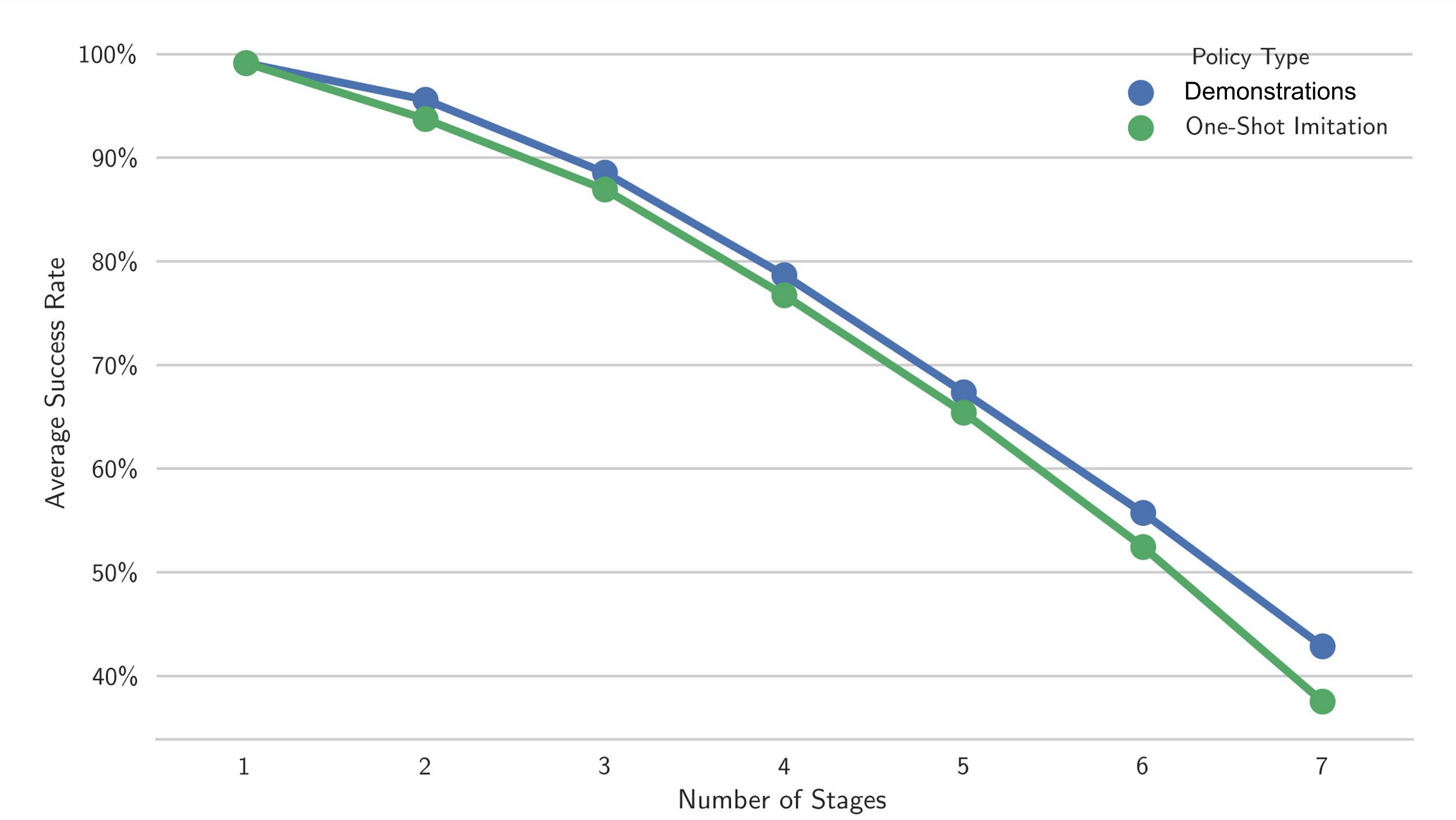


One-Shot Imitation — Proof-of-Concept



Wojciech Zaremba - OpenAI

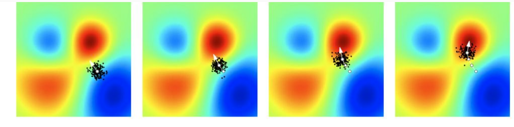
One-Shot Imitation — Numerical Results



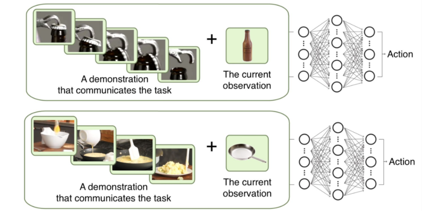
Summary

- Where to get rich, diverse data for robotics ?
  Our approach: Domain randomization

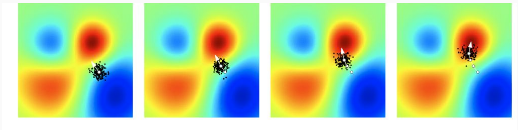
- How to obtain complex behaviors on robots?
  An approach: Evolution

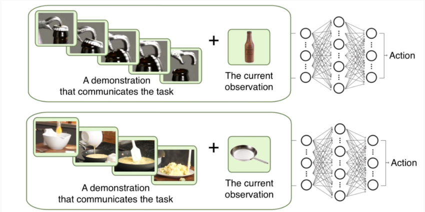


- How to convey the intent of the task to the robot ? An approach: One-shot imitation



Summary





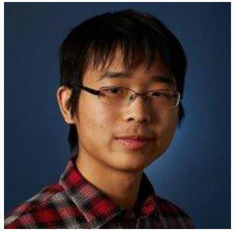
diverse data + scalable training + one-shot imitation

Still many limitations:

- Methods have not been evaluated on complex tasks such as cooking or cleaning
- Methods might break when simulated data oversimplifies real world
- Parallel gripper is relatively simple to control even without neural networks
- So far, all experiments are just a proof of concept

The robotics team


Marcin Andrychowicz


Rocky Duan


Bradly Stadie


Filip Wolski


Alex Ray


Jonas Schneider


Rachel Fong


Peter Welinder


Ankur Handa


Josh Tobin


Lukas Biewald


Pieter Abbeel


Erika Reinhardt


Bob McGrew


Vikash Kumar

Thank you

Ablation for one-shot