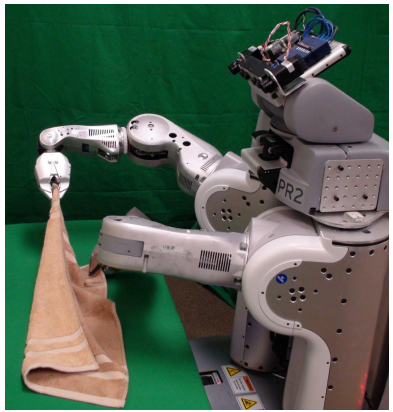
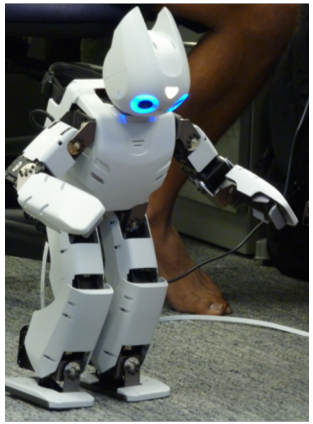
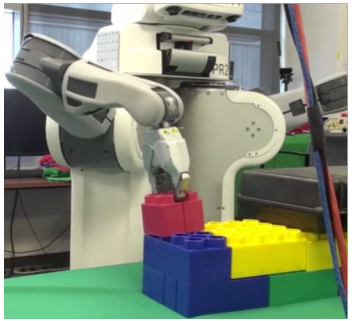




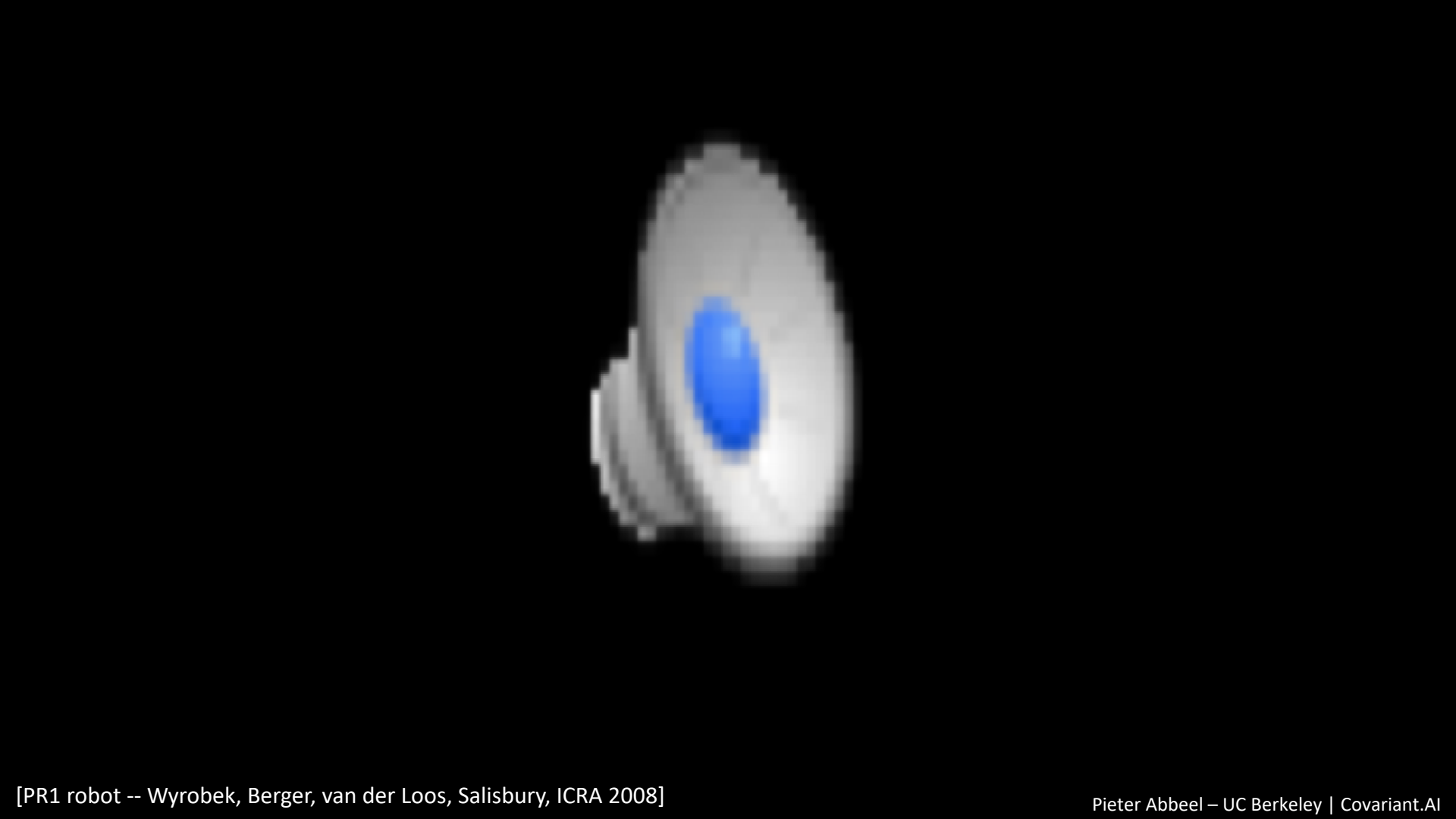


Deep Learning for Robotics

Pieter Abbeel

UC Berkeley AI research & teaching
covariant.ai AI for robotic automation of mfrg / logistics

Advisor to OpenAI, Preferred Networks, Traptic, Dishcraft, inzone.ai, OffWorld, TensorFlight, onai



[PR1 robot -- Wyrobek, Berger, van der Loos, Salisbury, ICRA 2008]

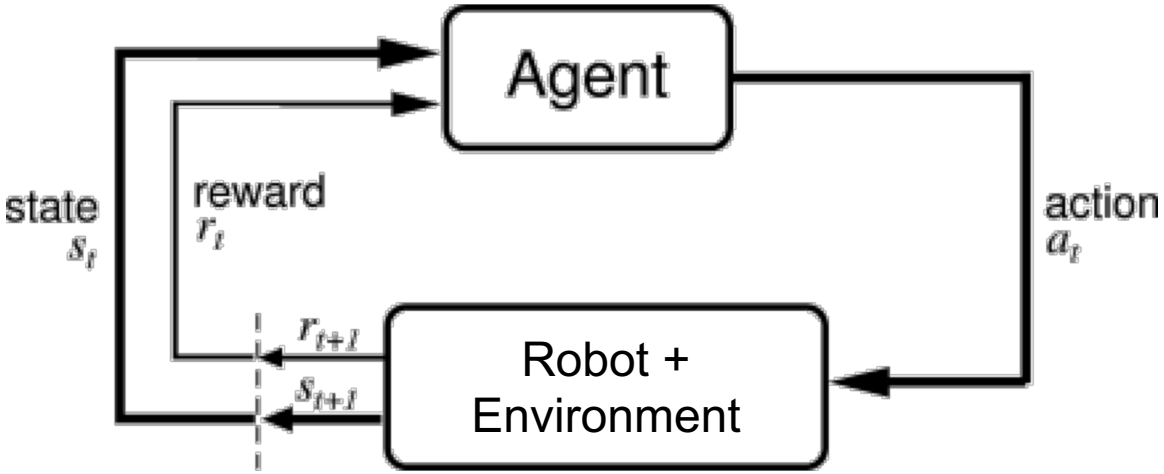
Many Exciting Challenges in AI for Robotics!

- Few-Shot Reinforcement Learning
- Leveraging Simulation
- Model-based RL
- Learning Representations for Exploration
- Few-Shot Imitation Learning

Many Exciting Challenges in AI for Robotics!

- ***Few-Shot Reinforcement Learning***
- Leveraging Simulation
- Model-based RL
- Learning Representations for Exploration
- Few-Shot Imitation Learning

Reinforcement Learning (RL)



$$\max_{\theta} \quad \mathrm{E}[\sum_{t=0}^{H} R(s_t)|\pi_\theta]$$

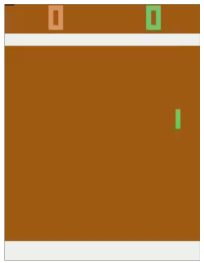
[Image credit: Sutton and Barto 1998]

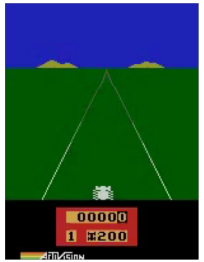
Deep RL Success: Learn Locomotion

^ **TRPO** Schulman et al, 2015 + **GAE** Schulman et al, 2016

See also: **DDPG** Lillicrap et al 2015; **SVG** Heess et al, 2015; **Q-Prop** Gu et al, 2016; **Scaling up ES** Salimans et al, 2017; **PPO** Schulman et al, 2017; **Parkour** Heess et al, 2017;

More Deep RL Successes

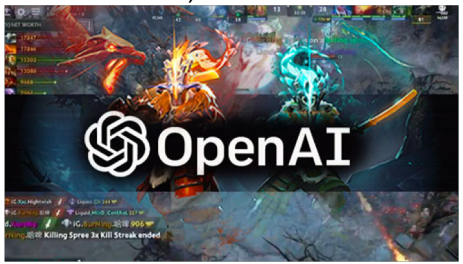




Mnih, et al 2013

Peng et al, 2018

Peng et al, 2018



OpenAI, 2017

Silver et al, 2016

Levine, Finn, et al, 2016

Geng, Zhang, Bruce et al, 2017

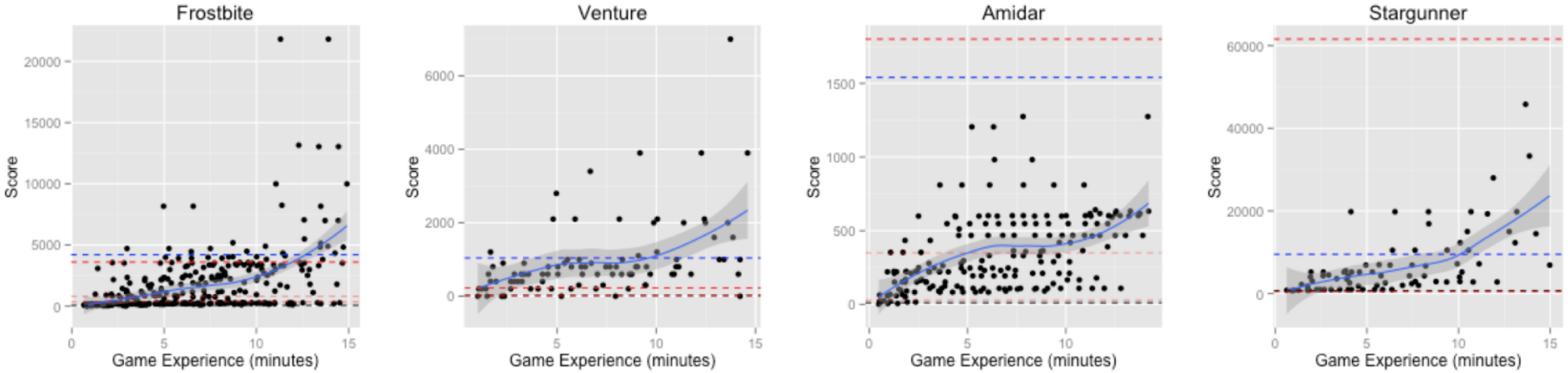
Reinforcement Learning

Q: What can it learn?

A: Mastery in wide range of domains

How fast does it learn?

Humans after 15 minutes tend to outperform DDQN after 115 hours



Black dots: human play

Blue curve: mean of human play

Blue dashed line: 'expert' human play

Red dashed lines:
DDQN after 10, 25, 200M frames
(~ 46, 115, 920 hours)

[Tsividis, Pouncy, Xu, Tenenbaum, Gershman, 2017]

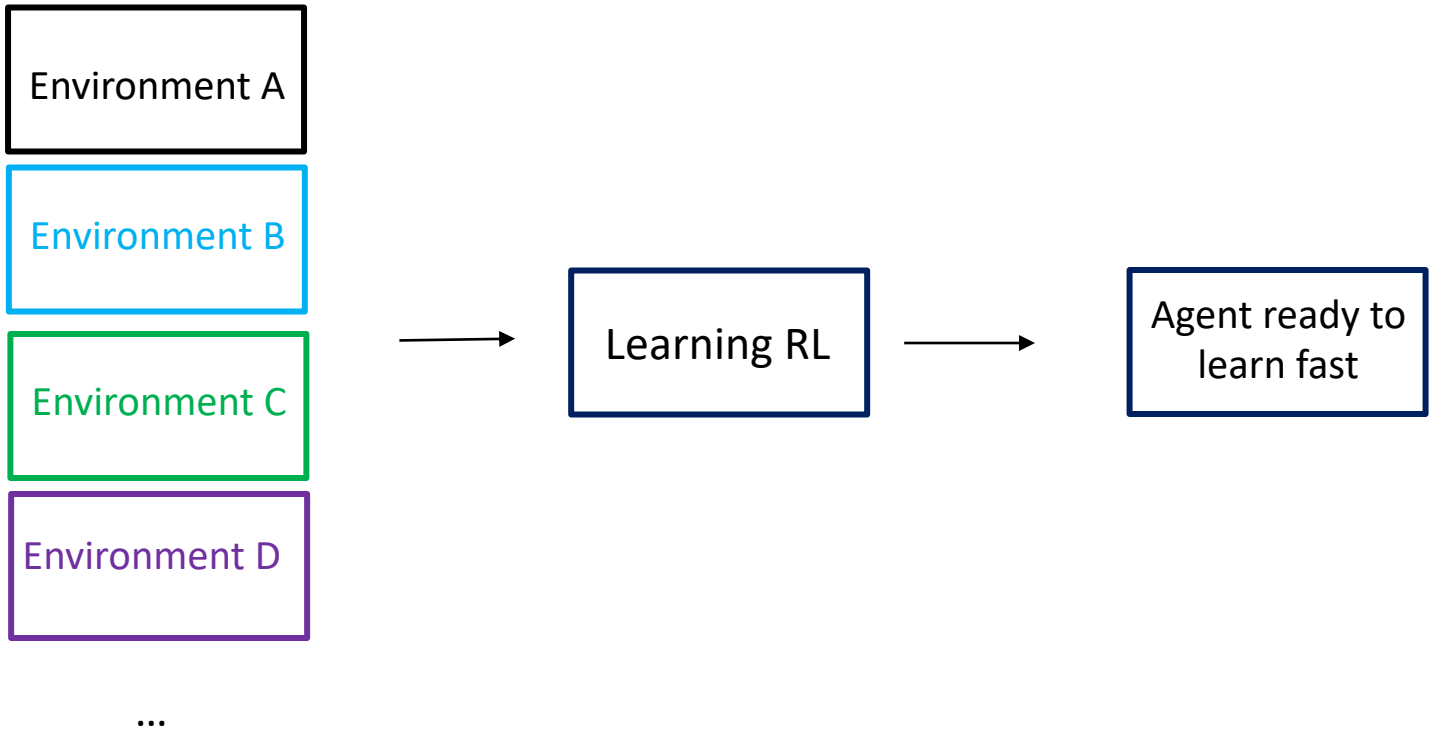
How to bridge this gap?

Humans might play Atari for the first time

BUT have done many other things before in their lives
AND implicitly through evolution

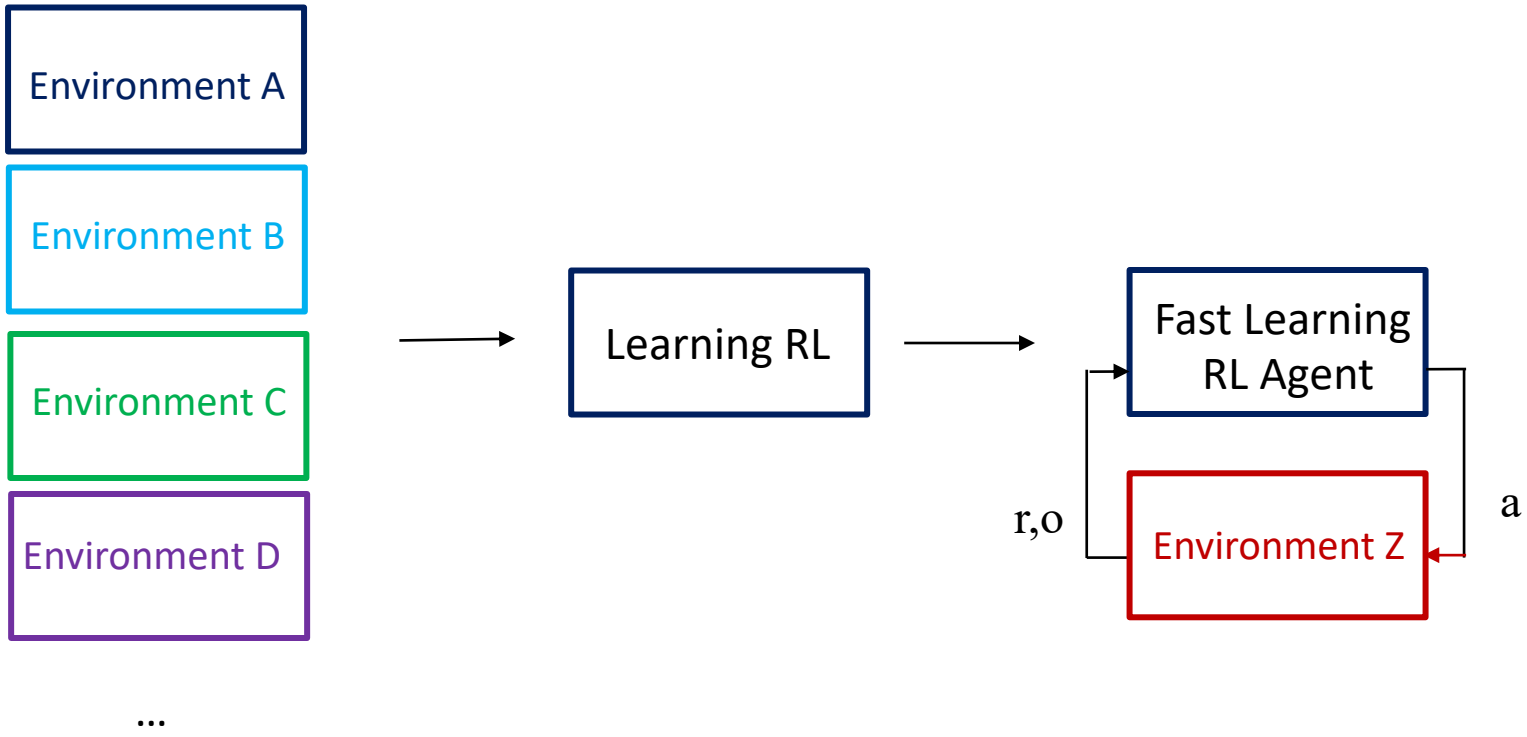
Can AI systems similarly exploit past
experiences to be ready to learn more quickly?

Learning to Reinforcement Learn



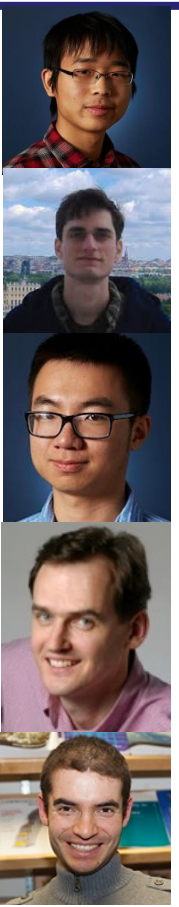
[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016] also: [Wang et al, 2016]

Learning to Reinforcement Learn



[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016] also: [Wang et al, 2016]

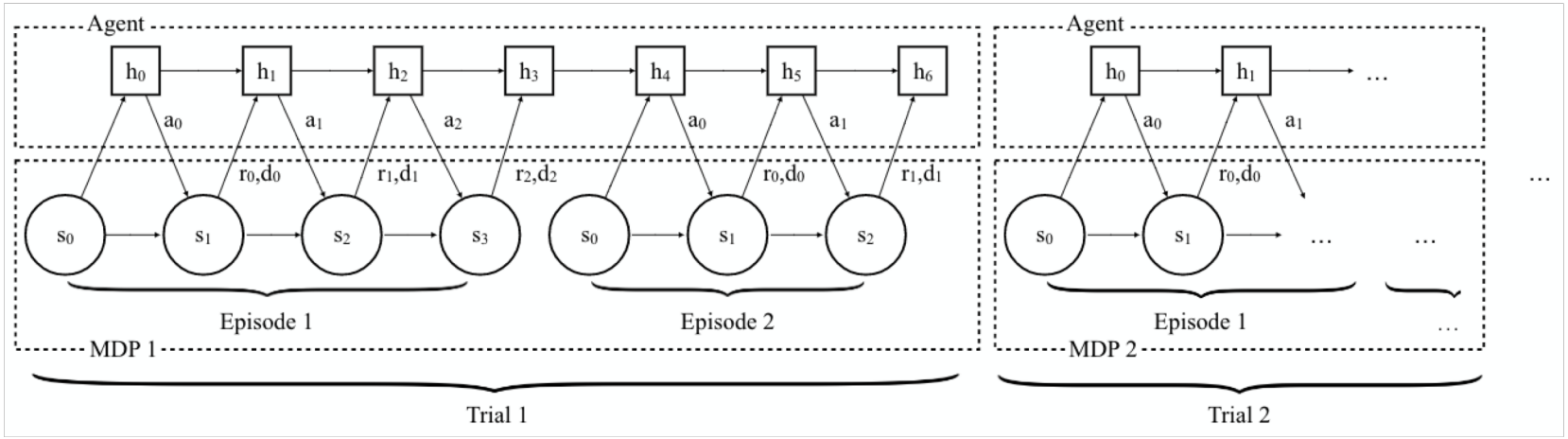
Formalizing Learning to Reinforcement Learn



$$\max_{\theta} \mathbb{E}_{M} \mathbb{E}_{\tau_M^{(k)}} \left[\sum_{k=1}^{K} R(\tau_M^{(k)}) \mid \text{RLagent}_{\theta} \right]$$

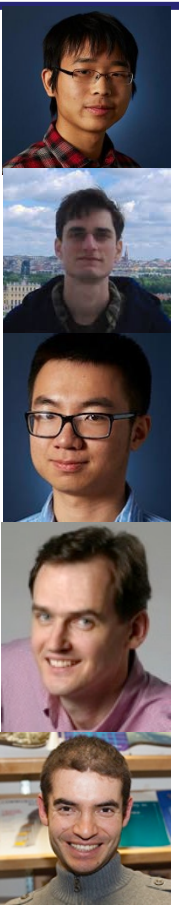
M : sample environment

$\tau_M^{(k)}$: k'th episode in environment M



[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016] also: [Wang et al, 2016]

Formalizing Learning to Reinforcement Learn



$$\max_{\theta} \mathbb{E}_{M} \mathbb{E}_{\tau_M^{(k)}} \left[\sum_{k=1}^{K} R(\tau_M^{(k)}) \mid \text{RLagent}_{\theta} \right]$$

M : sample MDP

$\tau_M^{(k)}$: k'th trajectory in MDP M

Meta-train:

$$\max_{\theta} \sum_{M \in M_{\text{train}}} \mathbb{E}_{\tau_M^{(k)}} \left[\sum_{k=1}^{K} R(\tau_M^{(k)}) \mid \text{RLagent}_{\theta} \right]$$

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016] also: [Wang et al, 2016]

Representing $\mathrm{RLagent}_\theta$

$$\max_\theta \sum_{M \in M_{\text{train}}} \mathbb{E}_{\tau_M^{(k)}} \left[\sum_{k=1}^{K} R(\tau_M^{(k)}) \mid \mathrm{RLagent}_\theta \right]$$

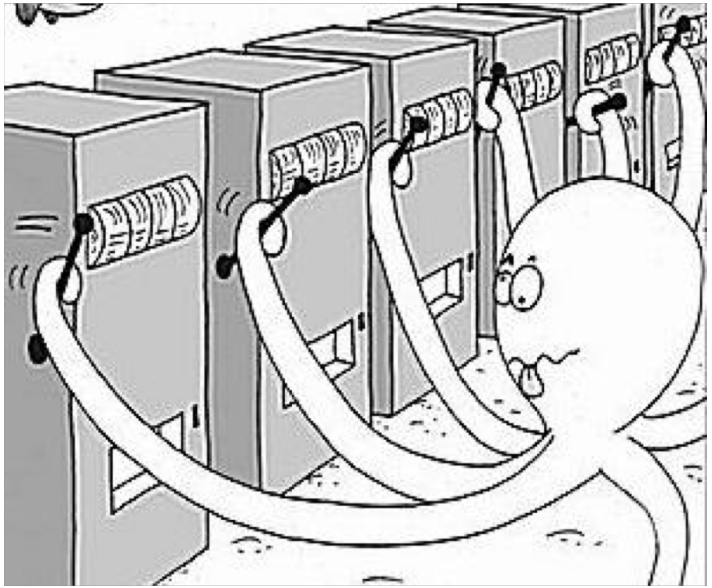
- RLagent = RNN = generic computation architecture
  - different weights in the RNN means different RL algorithm and prior
  - different activations in the RNN means different current policy
  - meta-train objective can be optimized with an existing (slow) RL algorithm

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016] also: [Wang et al, 2016]

Evaluation: Multi-Armed Bandits

- Multi-Armed Bandits setting
  - Each bandit has its own distribution over pay-outs
  - Each episode = choose 1 bandit
  - Good RL agent should explore bandits sufficiently, yet also exploit the good/best ones

- Provably (asymptotically) optimal RL algorithms have been invented by humans: Gittins index, UCB1, Thompson sampling, …



[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Evaluation: Multi-Armed Bandits

Setup	Random	Gittins	TS	OTS	UCB1	ϵ-Greedy	Greedy	RL^2
$n = 10, k = 5$	5.0	**6.6**	5.7	6.5	**6.7**	**6.6**	**6.6**	**6.7**
$n = 10, k = 10$	5.0	**6.6**	5.5	6.2	**6.7**	**6.6**	**6.6**	**6.7**
$n = 10, k = 50$	5.1	6.5	5.2	5.5	**6.6**	6.5	6.5	**6.8**
$n = 100, k = 5$	49.9	**78.3**	74.7	**77.9**	**78.0**	75.4	74.8	**78.7**
$n = 100, k = 10$	49.9	**82.8**	76.7	81.4	82.4	77.4	77.1	**83.5**
$n = 100, k = 50$	49.8	**85.2**	64.5	67.7	84.3	78.3	78.0	**84.9**
$n = 500, k = 5$	249.8	**405.8**	**402.0**	**406.7**	**405.8**	388.2	380.6	**401.6**
$n = 500, k = 10$	249.0	**437.8**	429.5	**438.9**	**437.1**	408.0	395.0	432.5
$n = 500, k = 50$	249.6	**463.7**	427.2	437.6	457.6	413.6	402.8	438.9

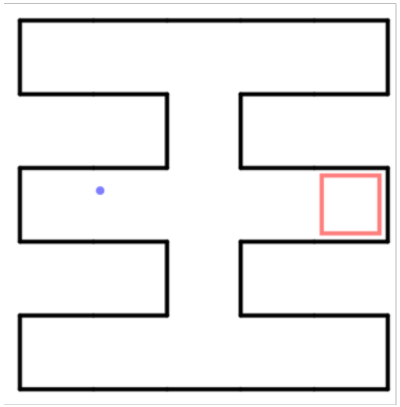
We consider Bayesian evaluation setting. Some of these prior works also have adversarial guarantees, which we don't consider here.

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Evaluation: Visual Navigation

Agent input: current image
Agent action: straight / 2 degrees left / 2 degrees right
Map just shown for our purposes, but not available to agent

Agent's view

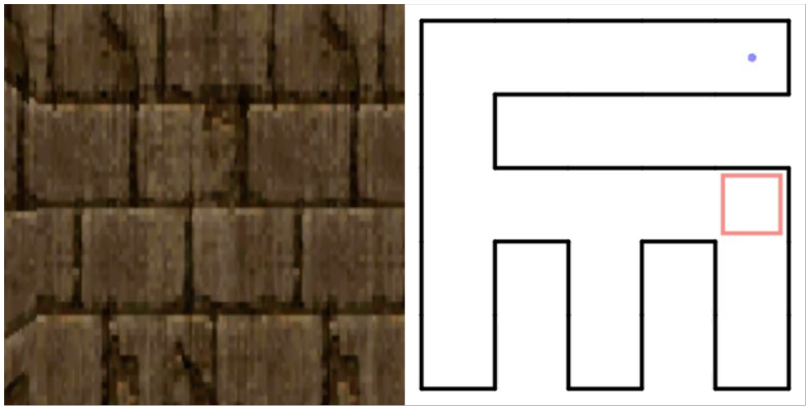


Maze

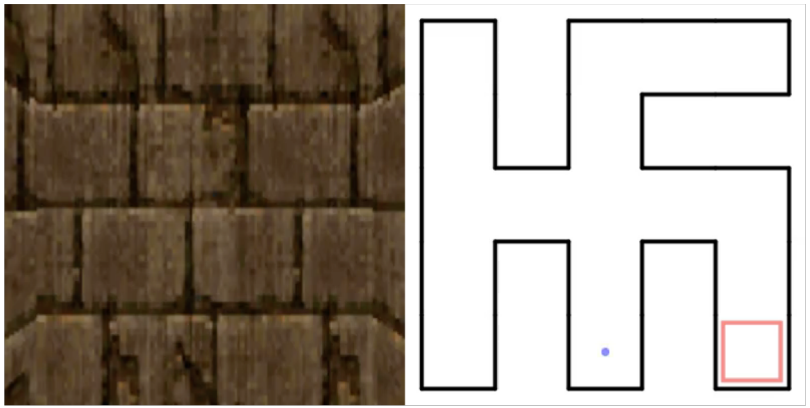
Related work: Mirowski, et al, 2016; Jaderberg et al, 2016; Mnih et al, 2016; Wang et al, 2016

Evaluation: Visual Navigation

Agent input: current image
Agent action: straight / 2 degrees left / 2 degrees right
Map just shown for our purposes, but not available to agent



Before learning-to-learn



After learning-to-learn

[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Many Exciting Challenges in AI for Robotics!

- Few-Shot Reinforcement Learning
- ***Leveraging Simulation***
- Model-based RL
- Learning Representations for Exploration
- Few-Shot Imitation Learning

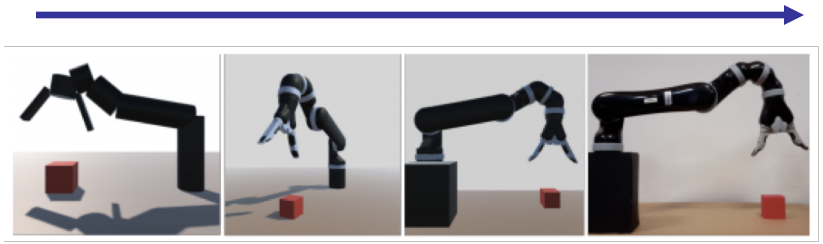
Motivation for Simulation

Compared to the real world, simulated data collection is…

- Less expensive
- Faster / more scalable
- Less dangerous
- Easier to label

How can we learn useful real-world skills in the simulator?

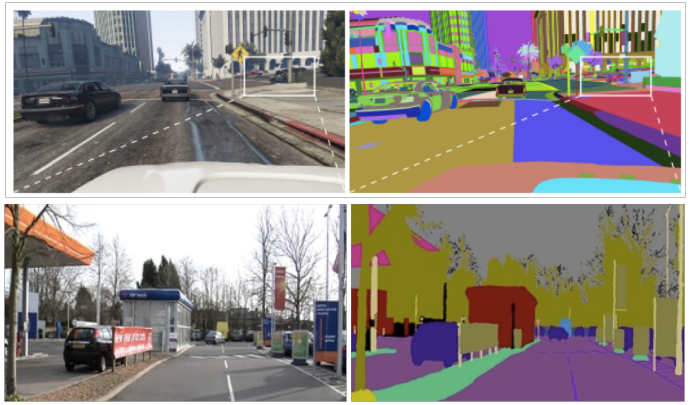
Approach 1 – Use Realistic Simulated Data



Simulation

Real world

Carefully match the simulation to the world [1,2,3,4]



GTA V

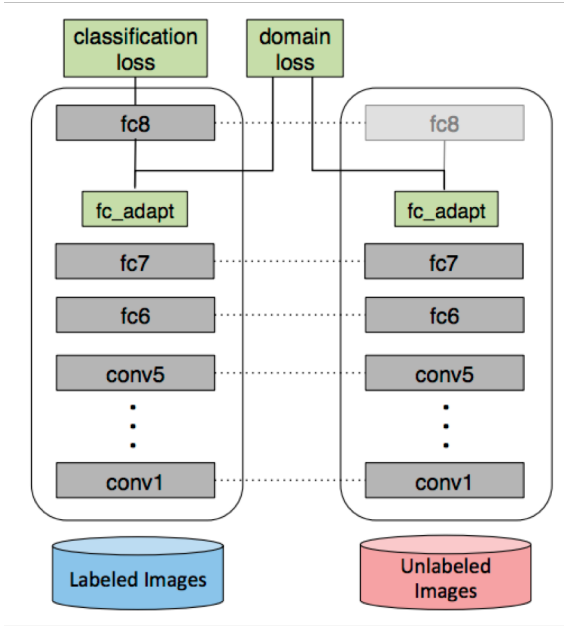
Real world

Augment simulated data with real data [5,6]

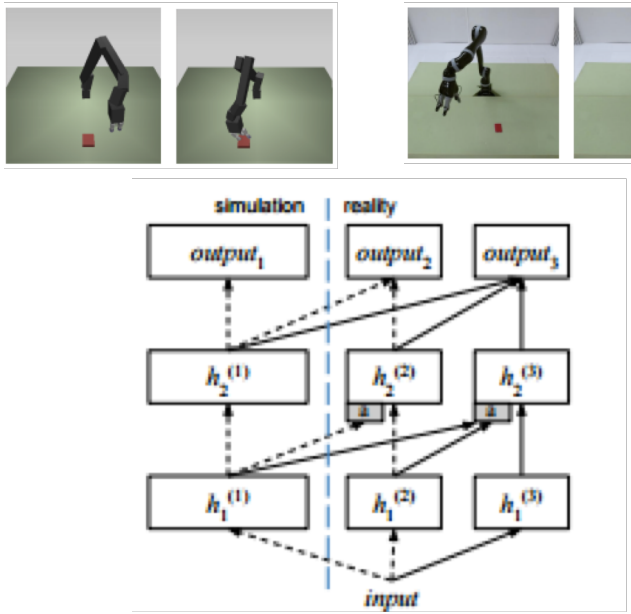
[1] Stephen James, Edward Johns. *3d simulation for robot arm control with deep q-learning (2016)*
[2] Johns, Leutenegger, Davision. Deep learning a grasp function for grasping under gripper pose uncertainty (2016)
[3] Mahler et al, Dex-Net 3.0 (2017)
[4] Koenemann et al. Whole-body model-predictive control applied to the HRP-2 humanoid. (2015)

[5] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. *Playing for data: Ground truth from computer games (2016)*
[6] Bousmalis et al. Using simulation and domain adaptation to improve efficiency of robotic grasping (2017)

Approach 2 – Domain Confusion / Adaptation

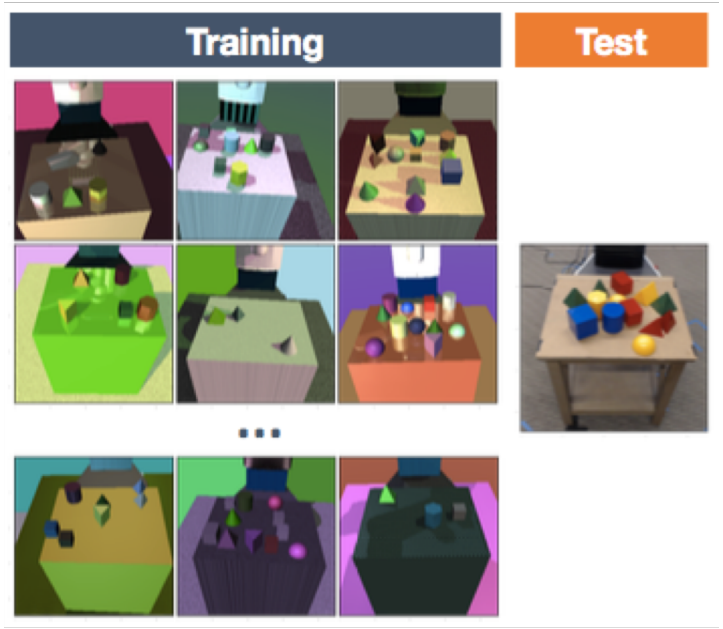


Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, Trevor Darrell. Deep Domain Confusion: Maximizing for Domain Invariance. *arXiv preprint arXiv:1412.3474, 2014*.



Andrei A Rusu, Matej Vecerik, Thomas Rotho¨rl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-real robot learning from pixels with progressive nets. *arXiv preprint arXiv:1610.04286*, 2016.

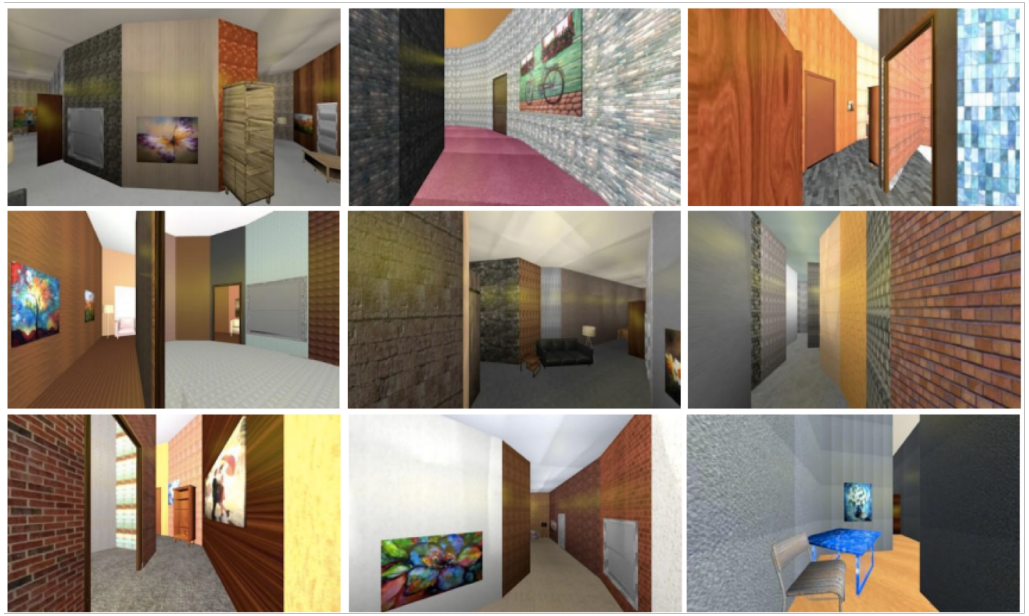
Approach 3 – Domain Randomization



If the model sees enough simulated variation, the real world may look like just the next simulator

Domain Randomization

(cad)^2 rl: Real Single-Image Flight Without a Single Real Image.



- Quadcopter collision avoidance
- ~500 semi-realistic textures, 12 floor plans
- ~40-50% of 1000m trajectories are collision-free

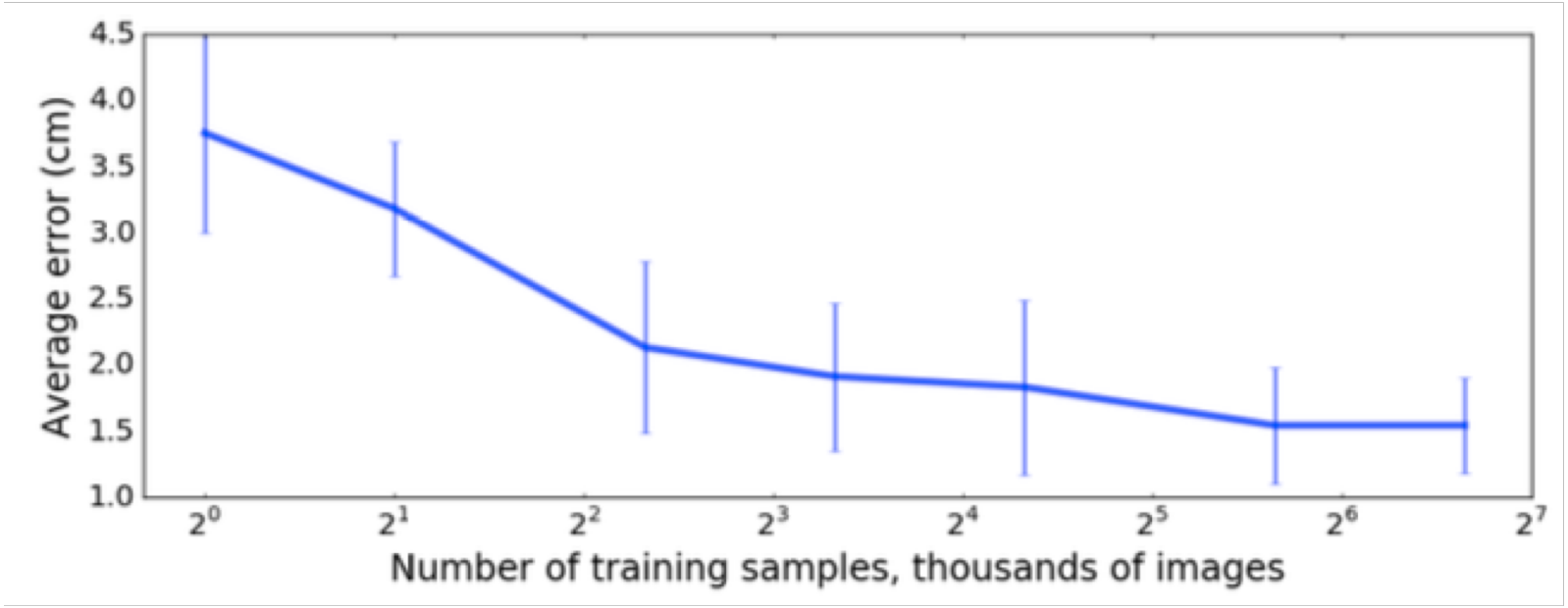
[3] Fereshteh Sadeghi and Sergey Levine. (cad)^2 rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.

Domain Randomization for Pose Estimation

- Precise object pose localization
- 100K images with simple randomly generated textures

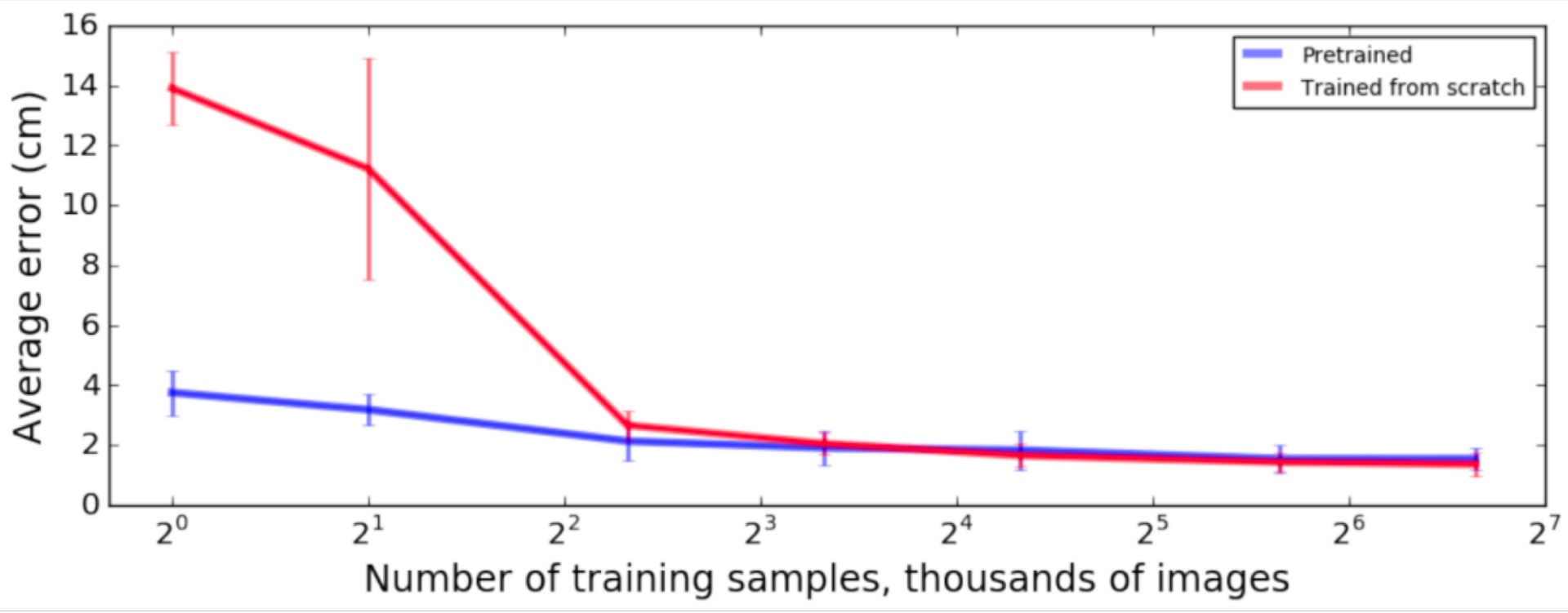
[Tobin, Fong, Ray, Schneider, Zaremba, Abbeel, 2017]

How does it work? More Data = Better



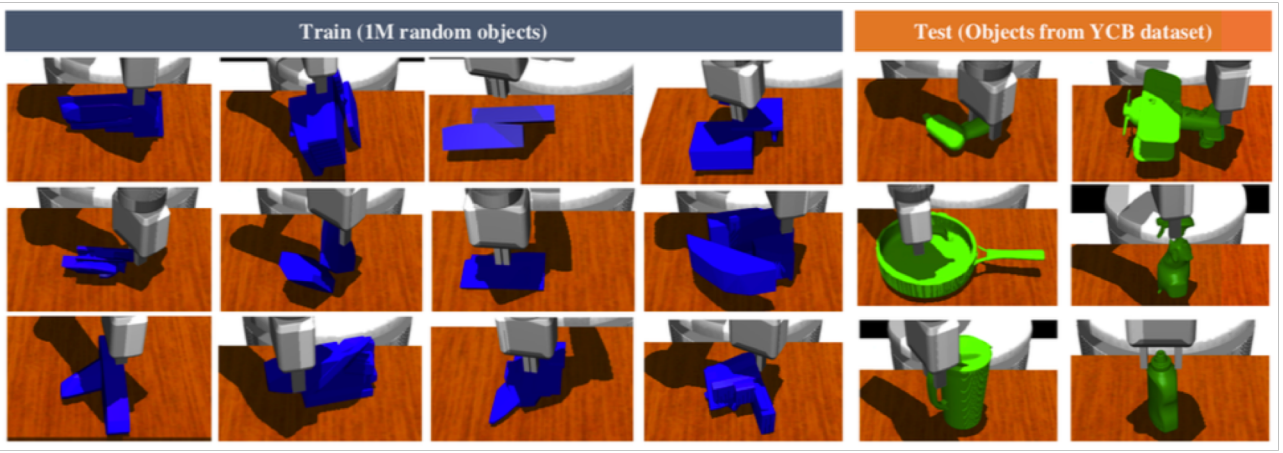
[Tobin, Fong, Ray, Schneider, Zaremba, Abbeel, 2017]

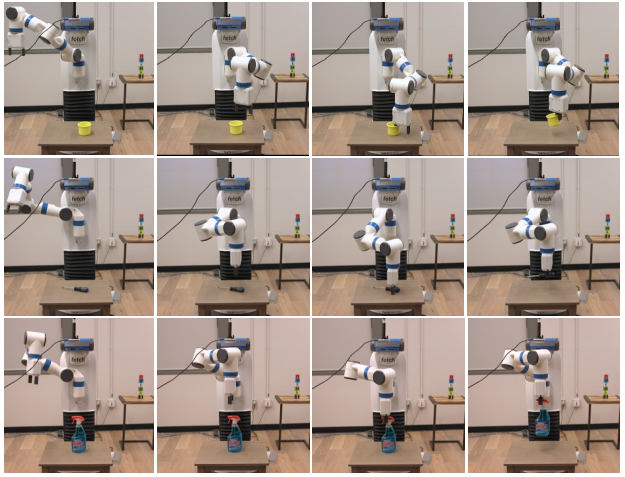
Pre-Training is not Necessary



[Tobin, Fong, Ray, Schneider, Zaremba, Abbeel, 2017]

Domain Randomization for Grasping

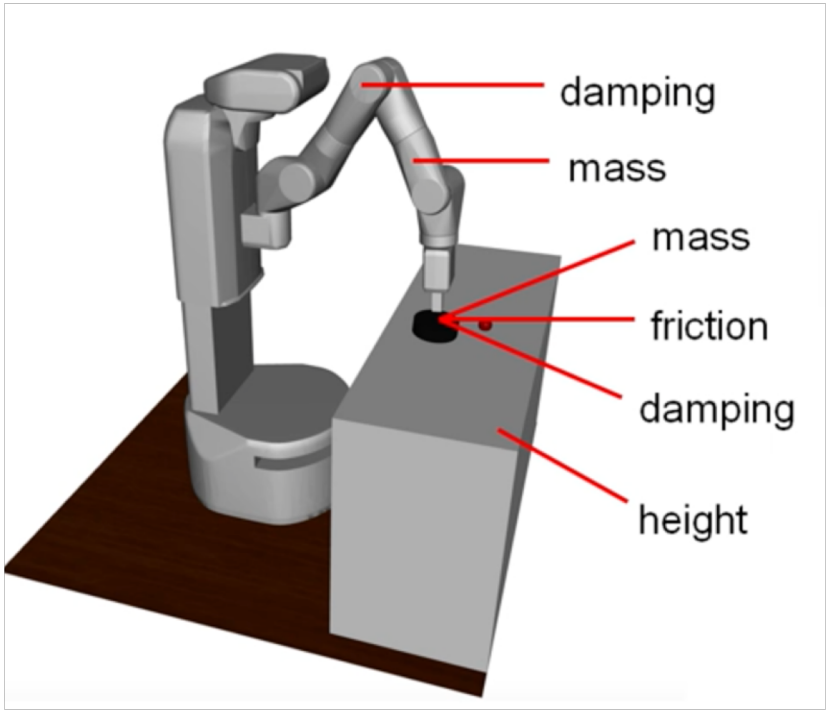




Hypothesis: Training on a diverse array of procedurally generated objects can produce comparable performance to training on realistic object meshes.

[Tobin, Zaremba, Abbeel, 2017] [see also: Levine et al and Goldberg et al]

Domain Randomization + Learning-to-Adapt

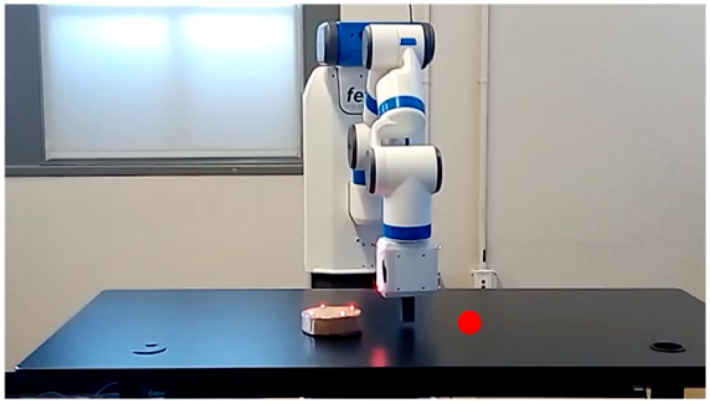


+ action time step, observation noise, action noise

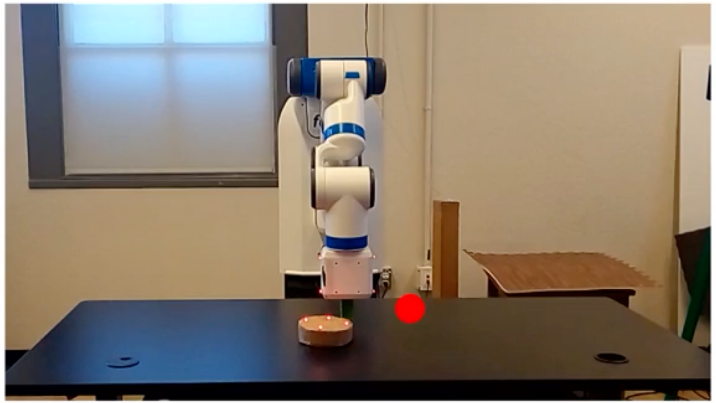
[Peng, Andrychowicz, Zaremba, Abbeel 2017]

Dynamics Randomization

Comparisons



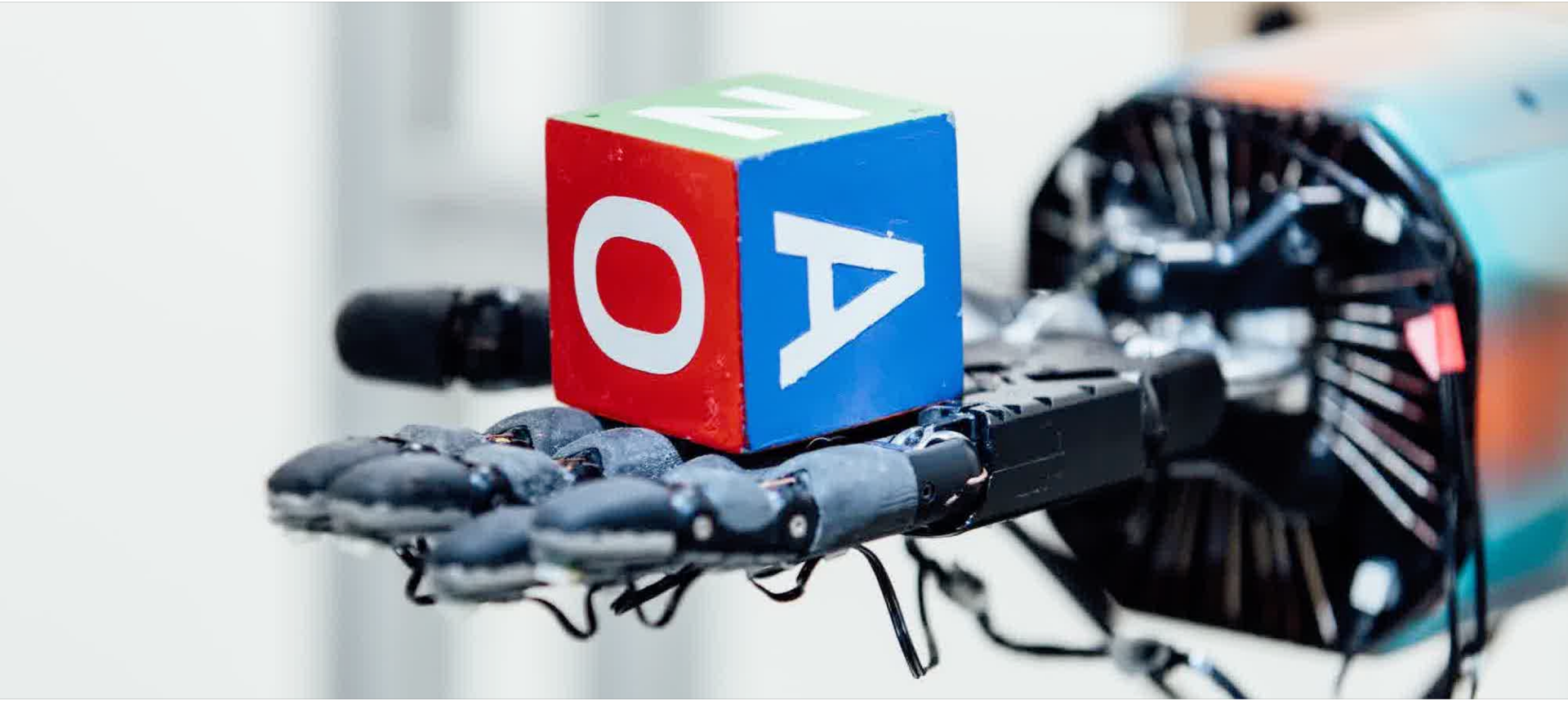
our method



no randomization
during training

[Peng, Andrychowicz, Zaremba, Abbeel 2017]

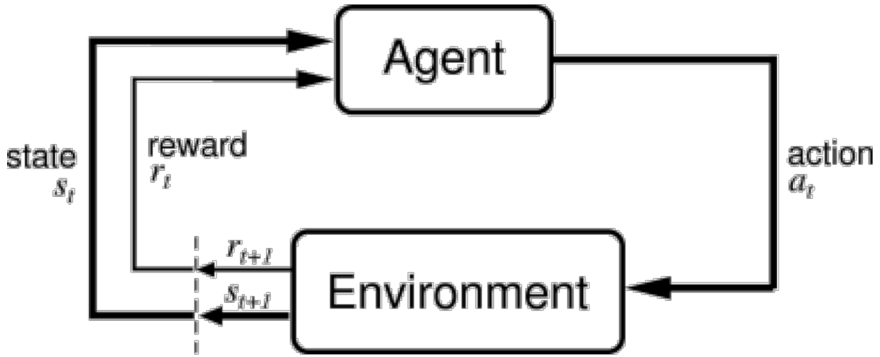
How About a Full Hand?



Many Exciting Challenges in AI for Robotics!

- Few-Shot Reinforcement Learning
- Leveraging Simulation
- ***Model-based RL***
- Learning Representations for Exploration
- Few-Shot Imitation Learning

Reinforcement Learning (RL)



$$\max_{\theta} \quad \mathrm{E}[\sum_{t=0}^{H} R(s_t)|\pi_\theta]$$

- RL: agent learns from repeated interaction with environment
- Model-free RL:
  - interaction with real world

  → Improve learned policy or Q function
- Model-based RL:
  - interaction with real world

  → Improve learned environment simulator

  → interaction with learned simulator

  → Improve policy or Q function

[Image credit: Sutton and Barto 1998]

Canonical Model-Based RL

- for iter = 1, 2, …
  - collect data under current policy
  - improve learned simulator from all past data
  - improve policy by RL in learned simulator

Anticipated benefit?
– much better sample efficiency

So why not used all the time?
-- not achieving same asymptotic performance as model-free methods
-- “overfitting” (“model-bias”)

Model-based RL Asymptotic Performance

- Because learned (ensemble of) models imperfect
  - Resulting policy good in simulation(s), but not optimal in real world

- Attempted fix: learn better dynamics model
  - Such efforts have so far proven insufficient

Overfitting in Model-based RL

- Standard overfitting (in supervised learning)
  - Neural network performs well on training data, but poorly on test data
    - E.g. on prediction of s_next from (s, a)

- New overfitting challenge in Model-based RL
  - policy optimization tends to exploit regions where insufficient data is available to train the model, leading to catastrophic failures
  - = "model-bias" (Deisenroth & Rasmussen, 2011; Schneider, 1997; Atkeson & Santamaria, 1997)

Recall: Domain Randomization + Few-Shot RL!

- Key idea:
  - No need to learn an accurate model
  - Suffices to learn a set of models representative of the real world
  - And then run few-shot RL in that set of models

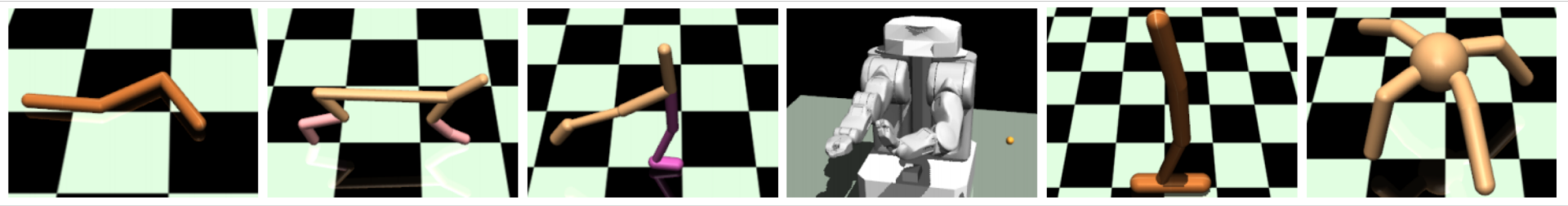
Model-Based RL via Meta Policy Optimization (MB-MPO)

for iter = 1, 2, …

- collect data under current adaptive policies $\pi_{\theta'_1}, \ldots, \pi_{\theta'_K}$
- learn **ENSEMBLE** of K simulators from all past data
- **meta-policy optimization over ENSEMBLE**
  - → **new meta-policy** π_θ
  - → **new adaptive policies** $\pi_{\theta'_1}, \ldots, \pi_{\theta'_K}$

[Clavera*, Rothfuss*, Schulman, Fujita, Asfour, Abbeel, CoRL 2018]

MB-MPO Evaluation



[Clavera*, Rothfuss*, Schulman, Fujita, Asfour, Abbeel, CoRL 2018]

MB-MPO Evaluation

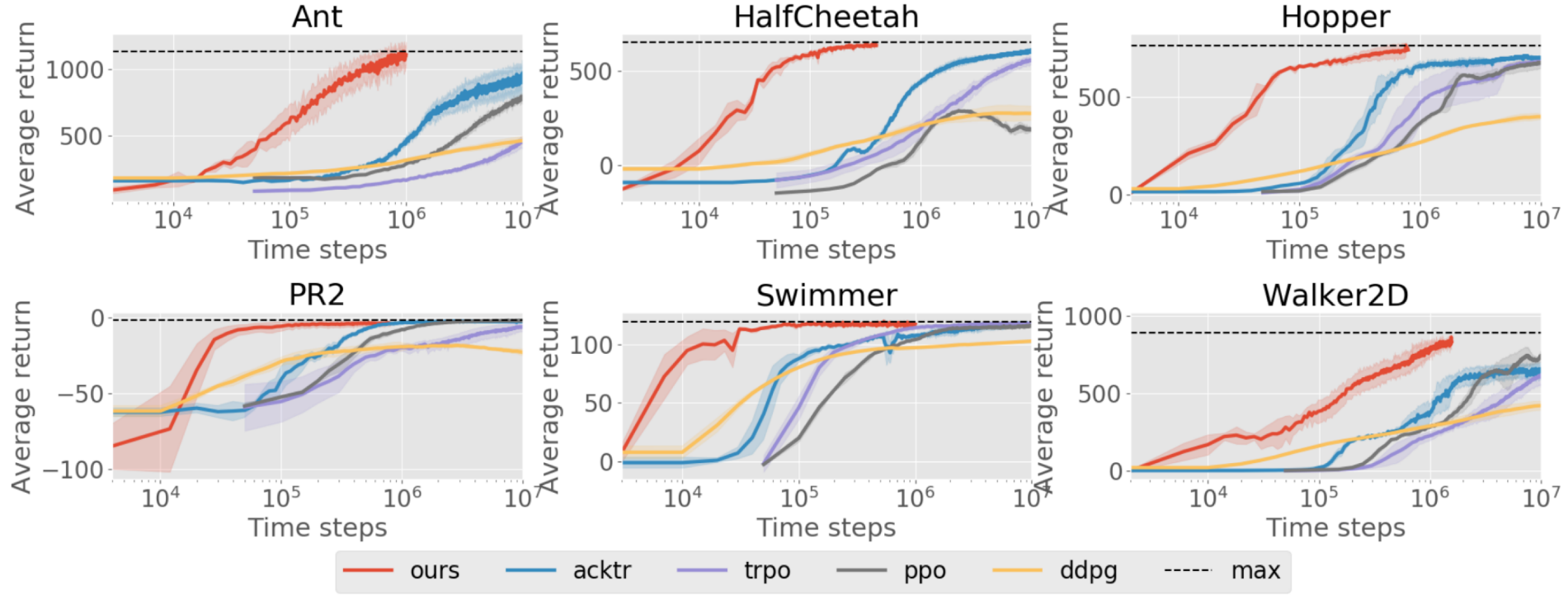
[Clavera*, Rothfuss*, Schulman, Fujita, Asfour, Abbeel, CoRL 2018]

MB-MPO Evaluation

[Clavera*, Rothfuss*, Schulman, Fujita, Asfour, Abbeel, CoRL 2018]

MB-MPO Evaluation

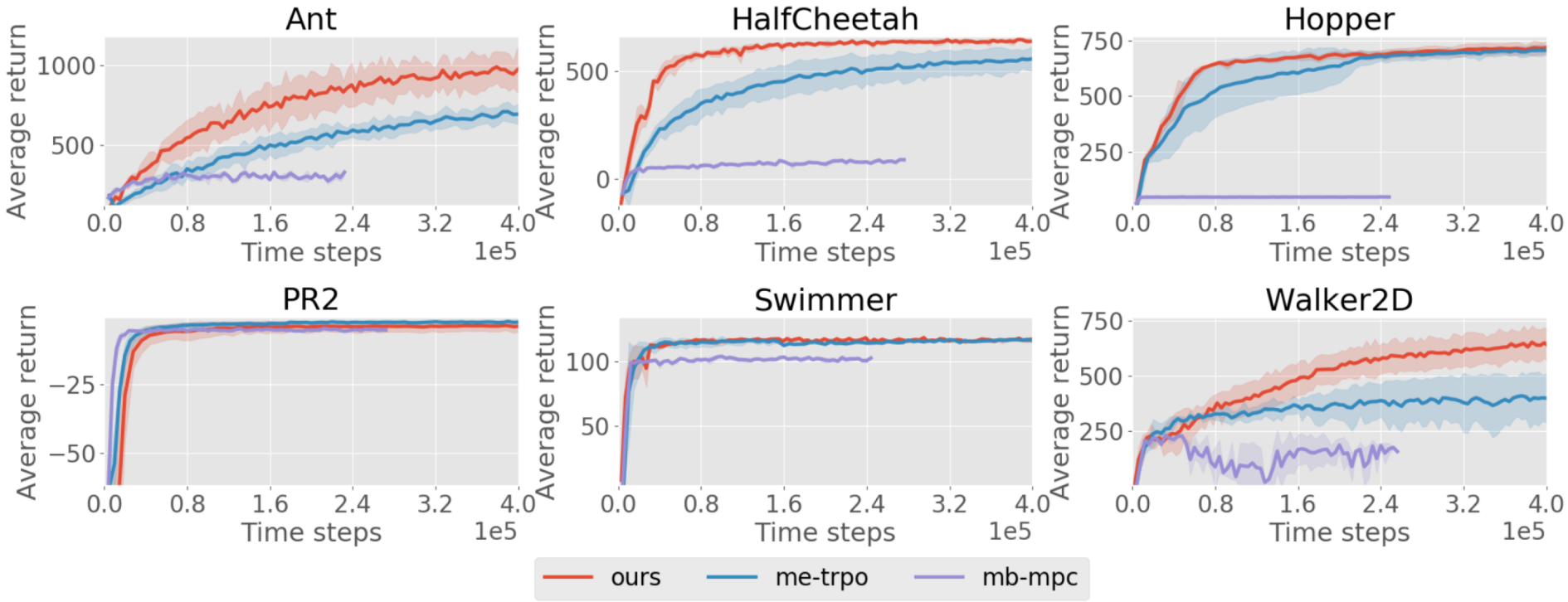
- Comparison with state of the art model-free



[Clavera*, Rothfuss*, Schulman, Fujita, Asfour, Abbeel, CoRL 2018]

MB-MPO Evaluation

- Comparison with state of the art model-based



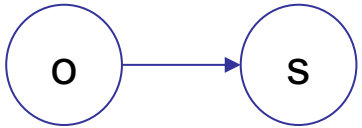
[Clavera*, Rothfuss*, Schulman, Fujita, Asfour, Abbeel, CoRL 2018]

Many Exciting Challenges in AI for Robotics!

- Few-Shot Reinforcement Learning
- Leveraging Simulation
- Model-based RL
- ***Learning Representations for Exploration***
- Few-Shot Imitation Learning

Traditional view on Representation Learning in RL

Typical focus on learning good representations of observations allowing for better learning, modeling, etc



- Embed2Control [Watter et al]
- Deep Predictive Policy Training [Ghadirzadeh et al]
- DARLA [Higgins et al]
- Robotic Priors [Jonschkowski and Brock]
- UPN [Srinivas, Jabri, Finn, et al]
- Causal InfoGAN [Kurutach, Tamar et al]

Especially useful when trying to learn from pixel inputs and complex observations

Alternative View on Representation Learning

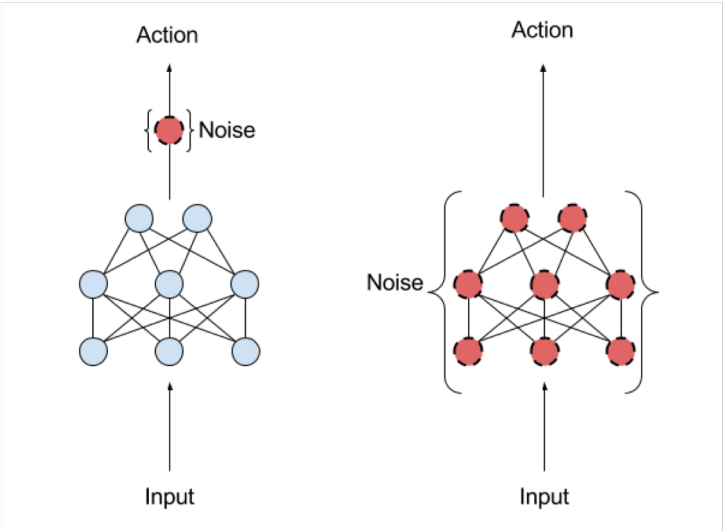
Can we learn representations of things other than the mapping from observations to low level hidden state

- Trajectories
- Behaviors/tasks?
- …

Question: How do we supervise representation learning for these things?

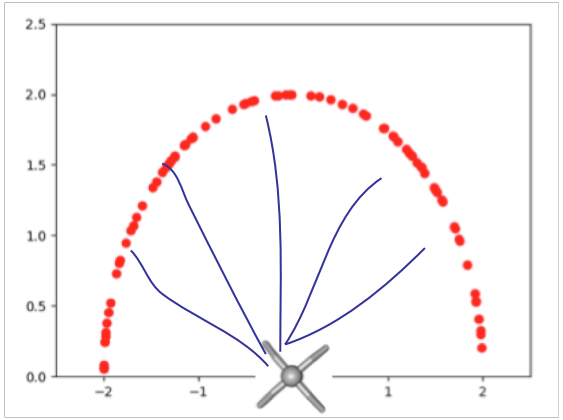
Representation learning for Exploration

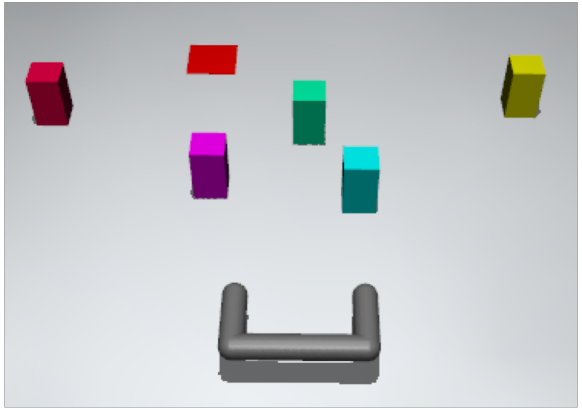
Exploration typically with stochasticity in the space of actions or in parameter space



Can we better inform exploration spaces using prior experience?

Why is this important?



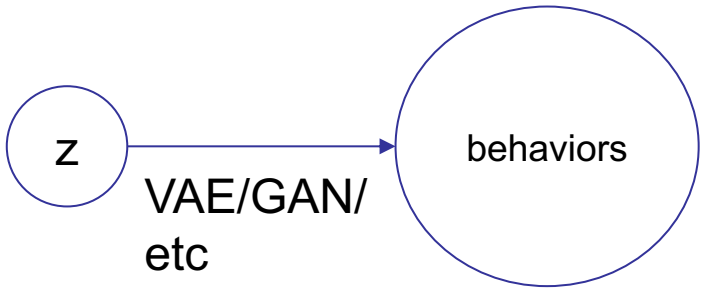


More efficient exploration for test tasks based on prior knowledge

Latent Exploration Spaces

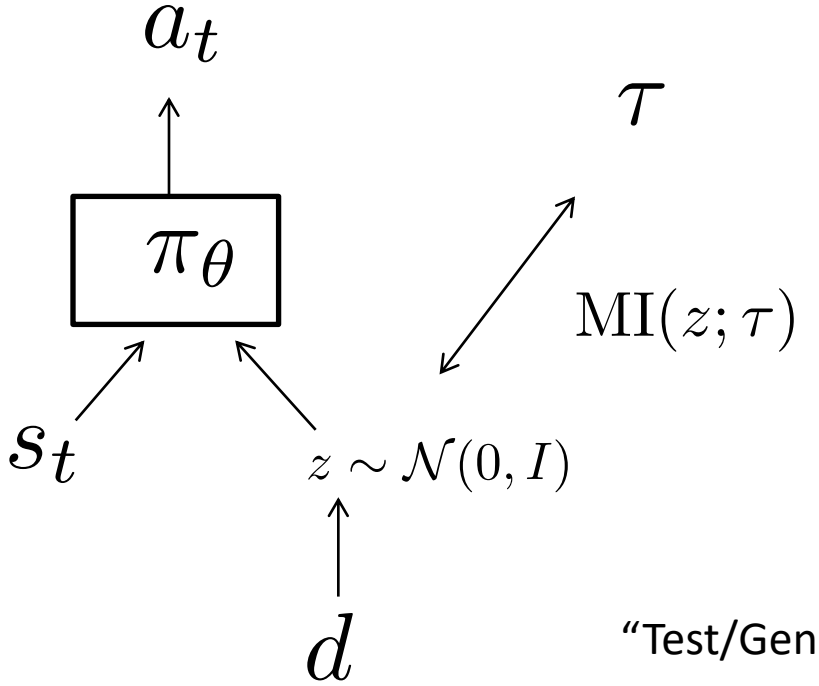
Can we learn better spaces to inject noise instead of simply into action/parameter space?

- What if we learned a latent space of behaviors which spans the input task distribution? i.e can we learn a generative model of behaviors and use it to generate novel exploratory behavior



Approach 1: Hausman et al 2018

Train against many tasks
(each task indexed by d):

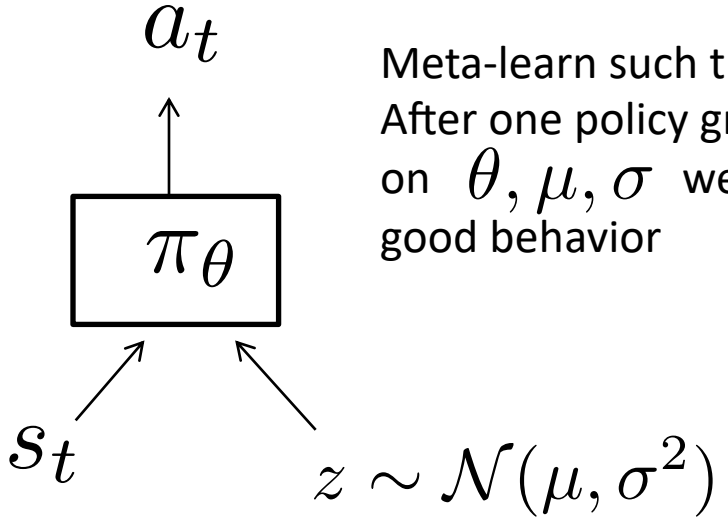


"Test/Generalize" in z-space

Learning an Embedding Space for Transferrable Skills
Hausman, Springenberg, Wang, Heess, Riedmiller, ICLR 2018

Approach 2: Gupta et al 2018

Train against many tasks



Meta-learn such that:
After one policy gradient update on θ, μ, σ we get good behavior

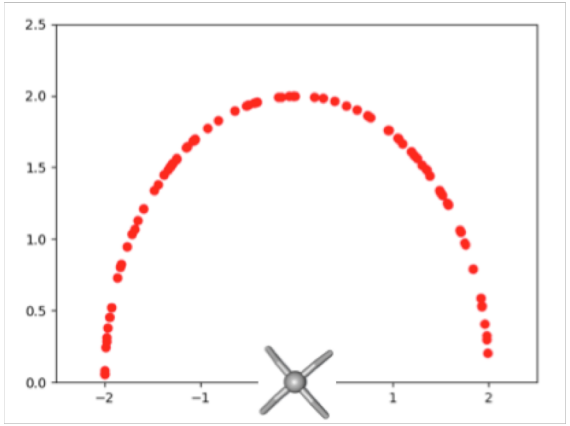
"Test/Generalize" in z-space

Model-Agnostic Exploration with Structured Noise (MAESN)
Gupta, Mendonca, Liu, Abbeel, Levine, 2018

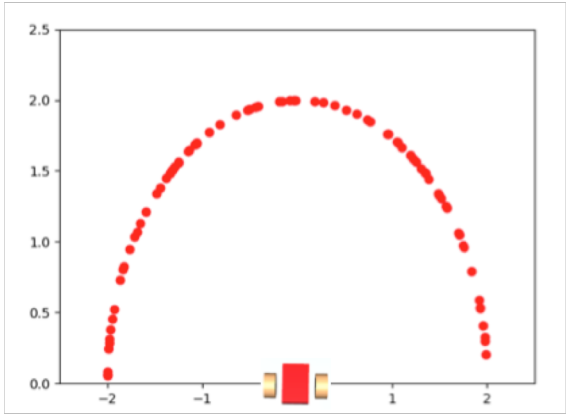
Experiments

3 task families:

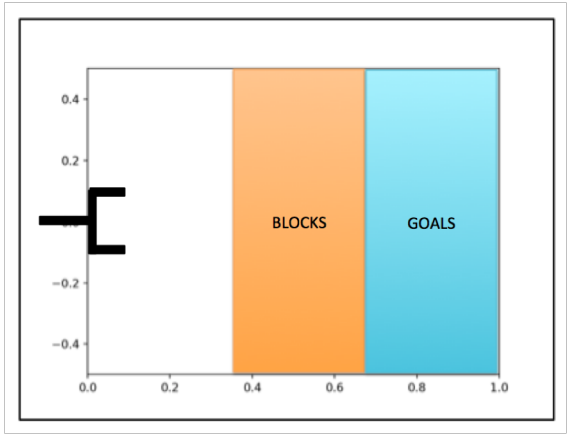
Ant



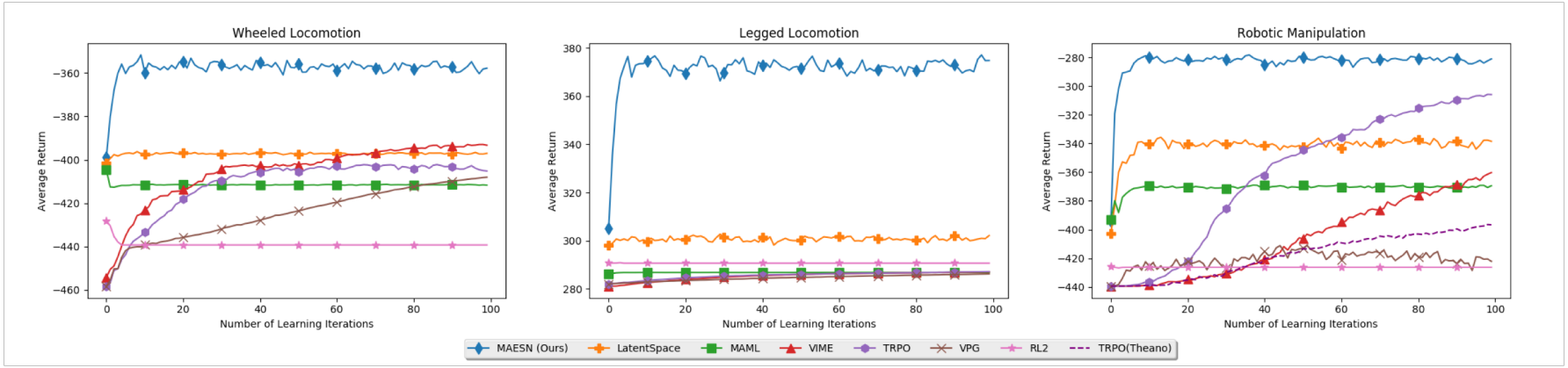
Wheeled Locomotion

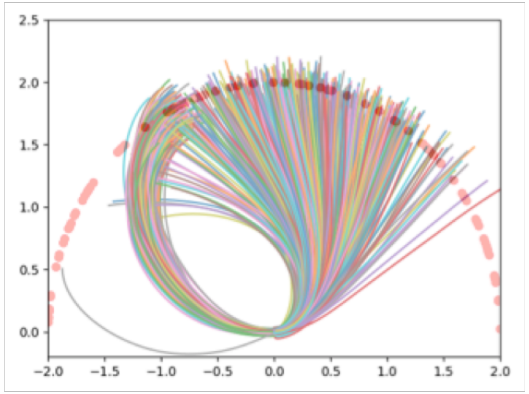


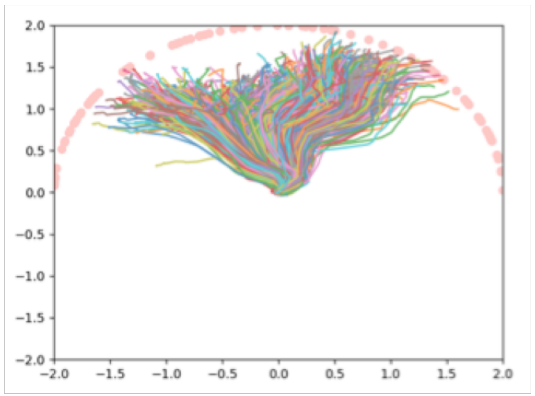
Block Manipulation

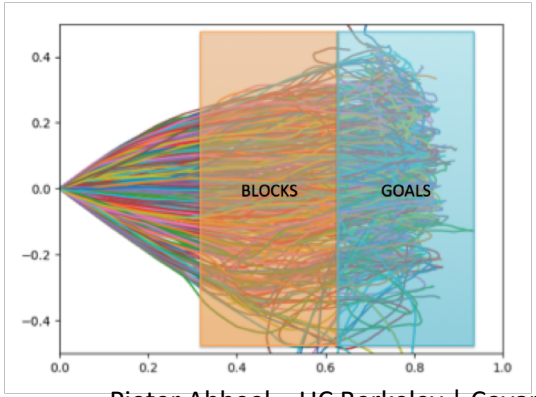


Experiments

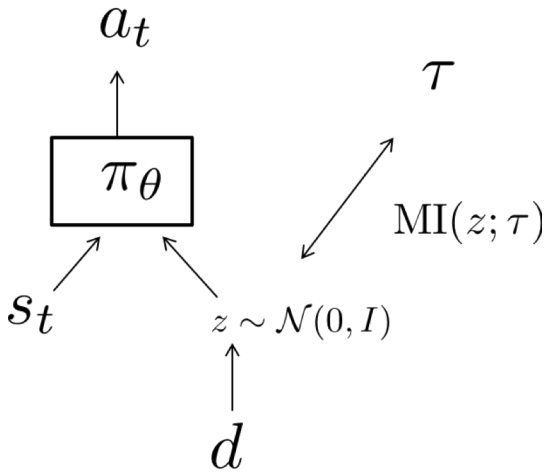




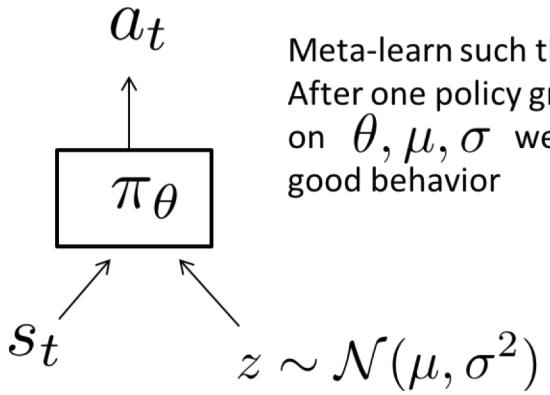




Recap so far



Hausman et al 2018

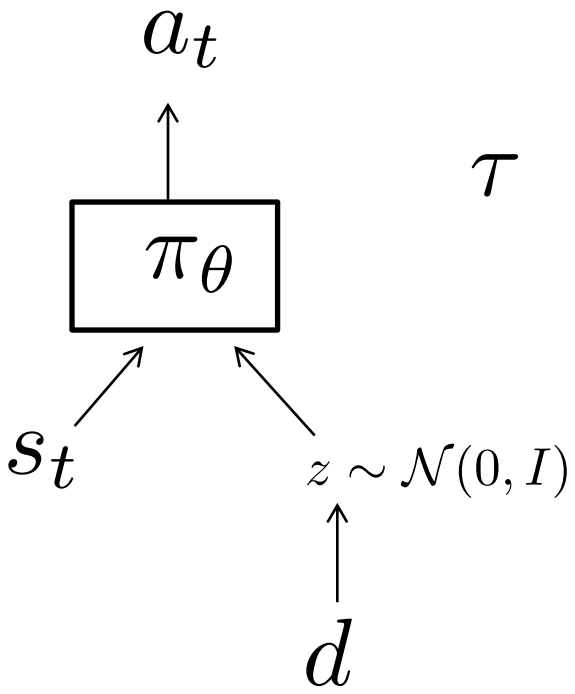


Meta-learn such that:
After one policy gradient update on θ, μ, σ we get good behavior

Gupta et al, 2018

How about unsupervised skill discovery?

Unsupervised Skill Discovery



- SSN4HRL (Florensa, Duan, Abbeel, 2016)

$$\mathrm{MI}(d; \tau)$$

- Variational Intrinsic Control (Gregor et al 2016)

$$\mathrm{MI}(z; s_H)$$

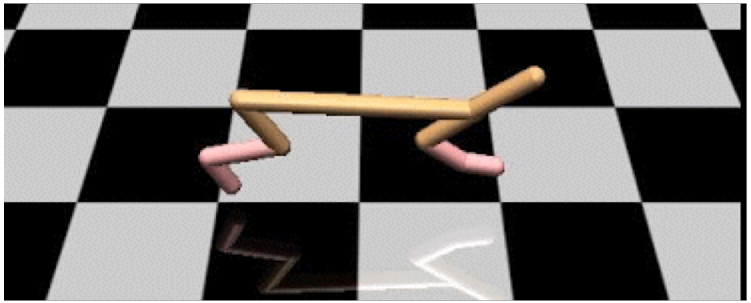
- Diversity is all you need (Eysenbach et al 2018)

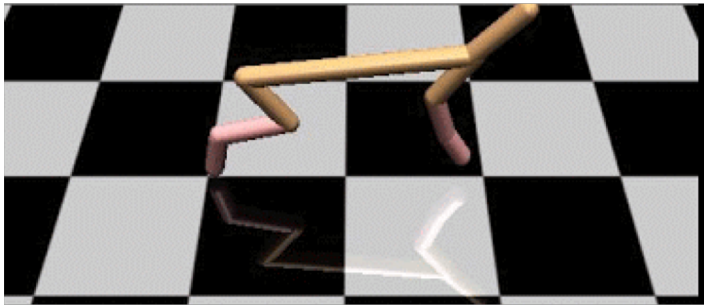
$$\sum_t \mathrm{MI}(z; s_t)$$

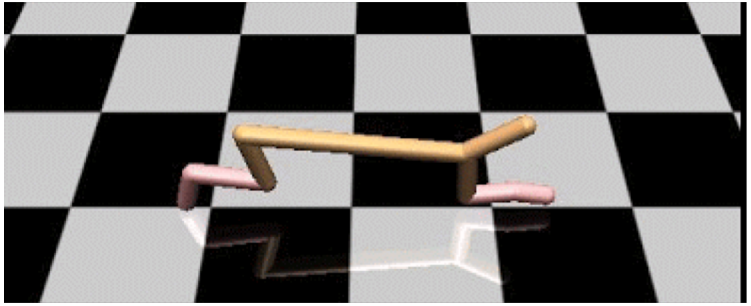
- VALOR (Achiam et al 2018)

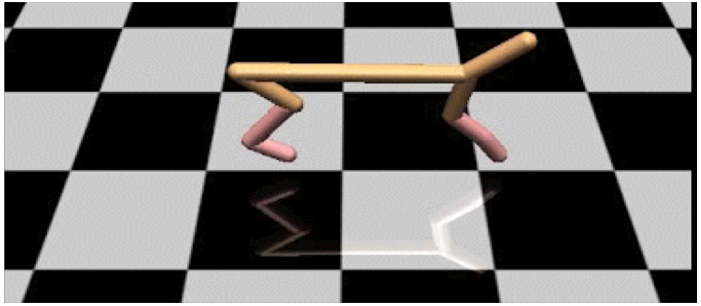
$$\mathrm{MI}(z; \tau)$$

Example videos (from Eysenbach et al)









Many Exciting Challenges in AI for Robotics!

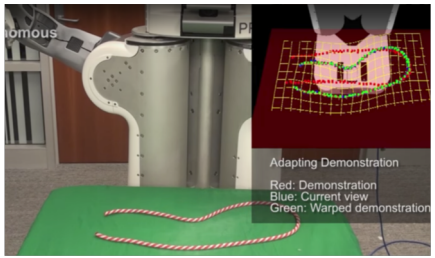
- Few-Shot Reinforcement Learning
- Leveraging Simulation
- Model-based RL
- Learning Representations for Planning
- ***Few-Shot Imitation Learning***

Imitation Learning in Robotics

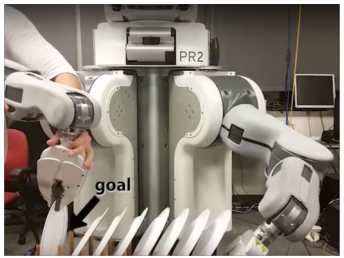
[Abbeel et al. 2008]

[Kolter et al. 2008]

[Ziebart et al. 2008]

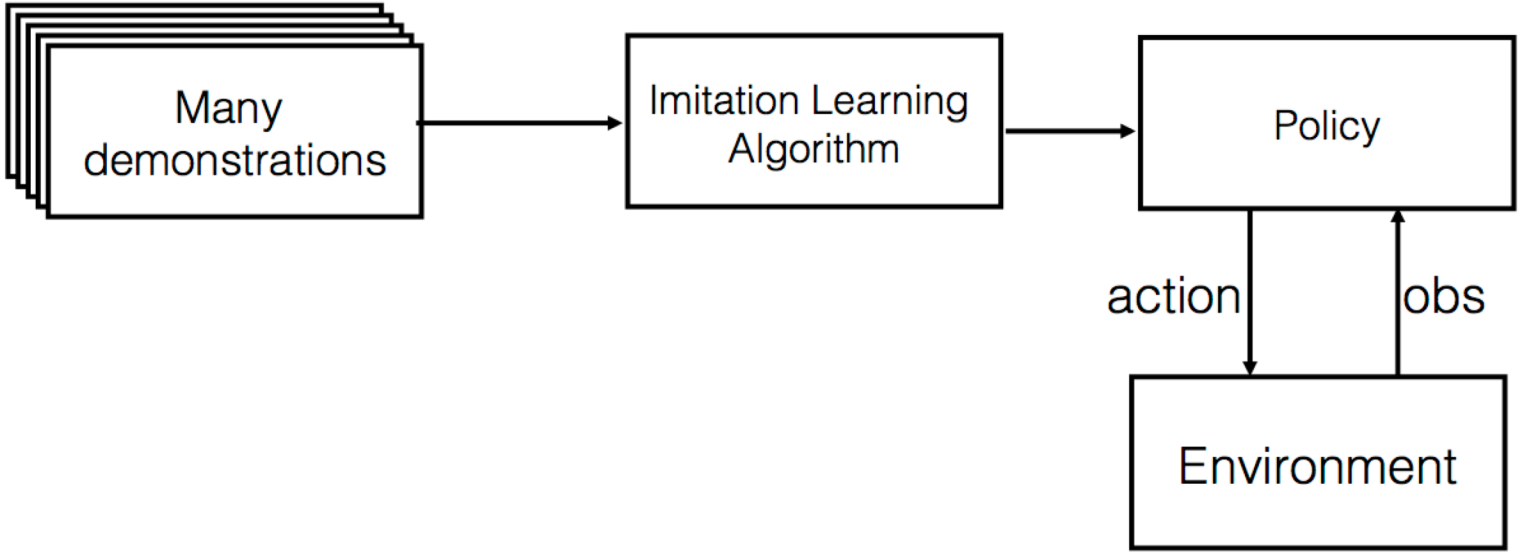


[Schulman et al. 2013]

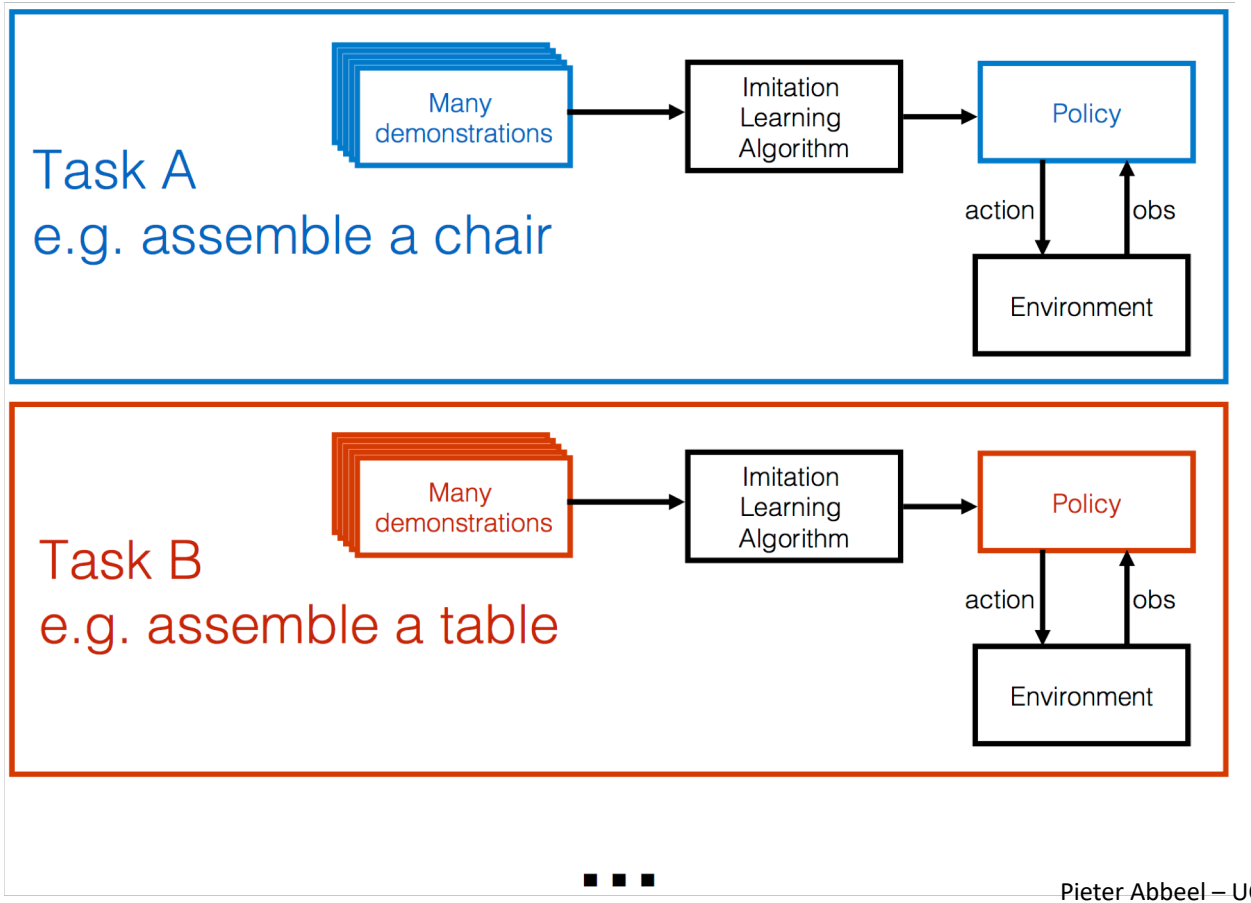


[Finn et al. 2016]

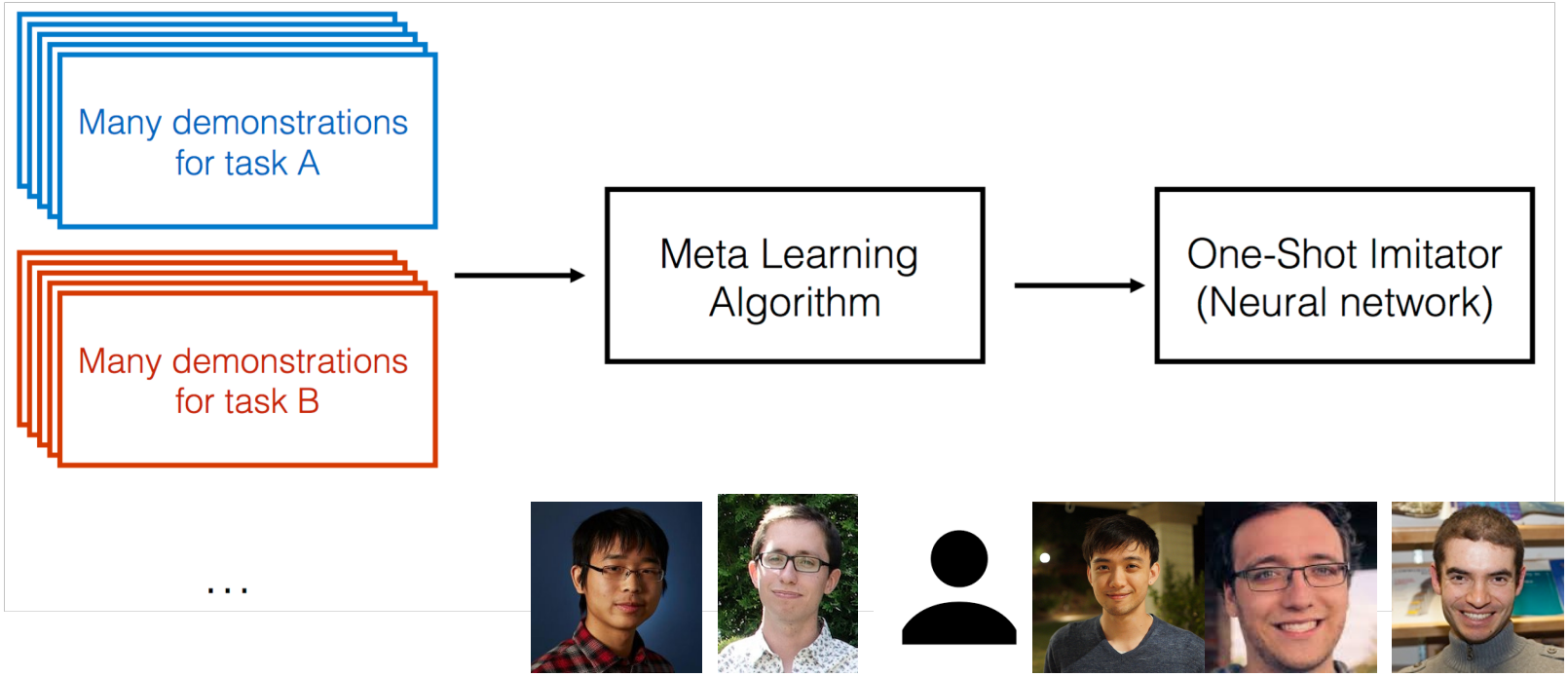
Imitation Learning



Imitation Learning

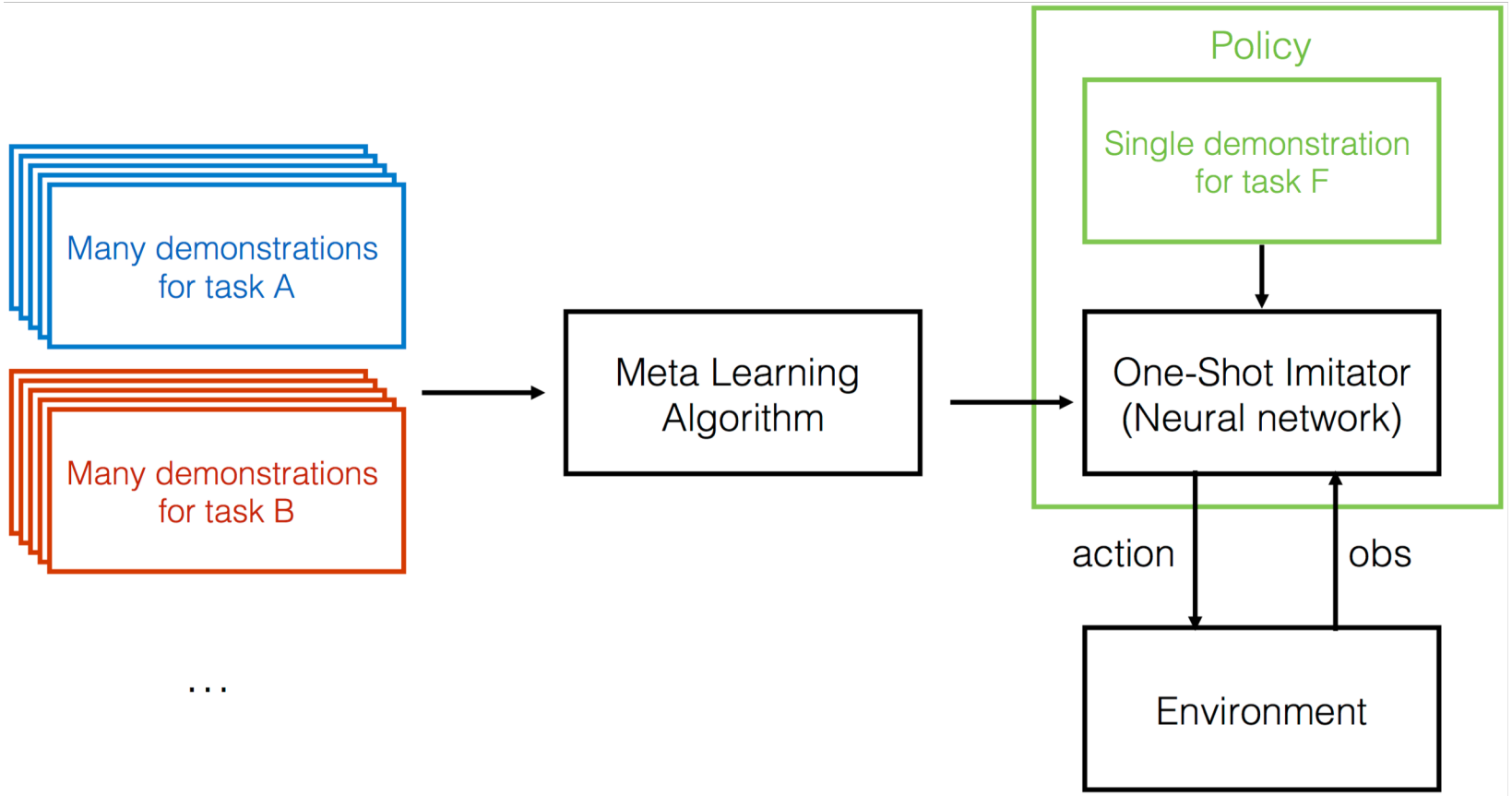


One-Shot Imitation Learning



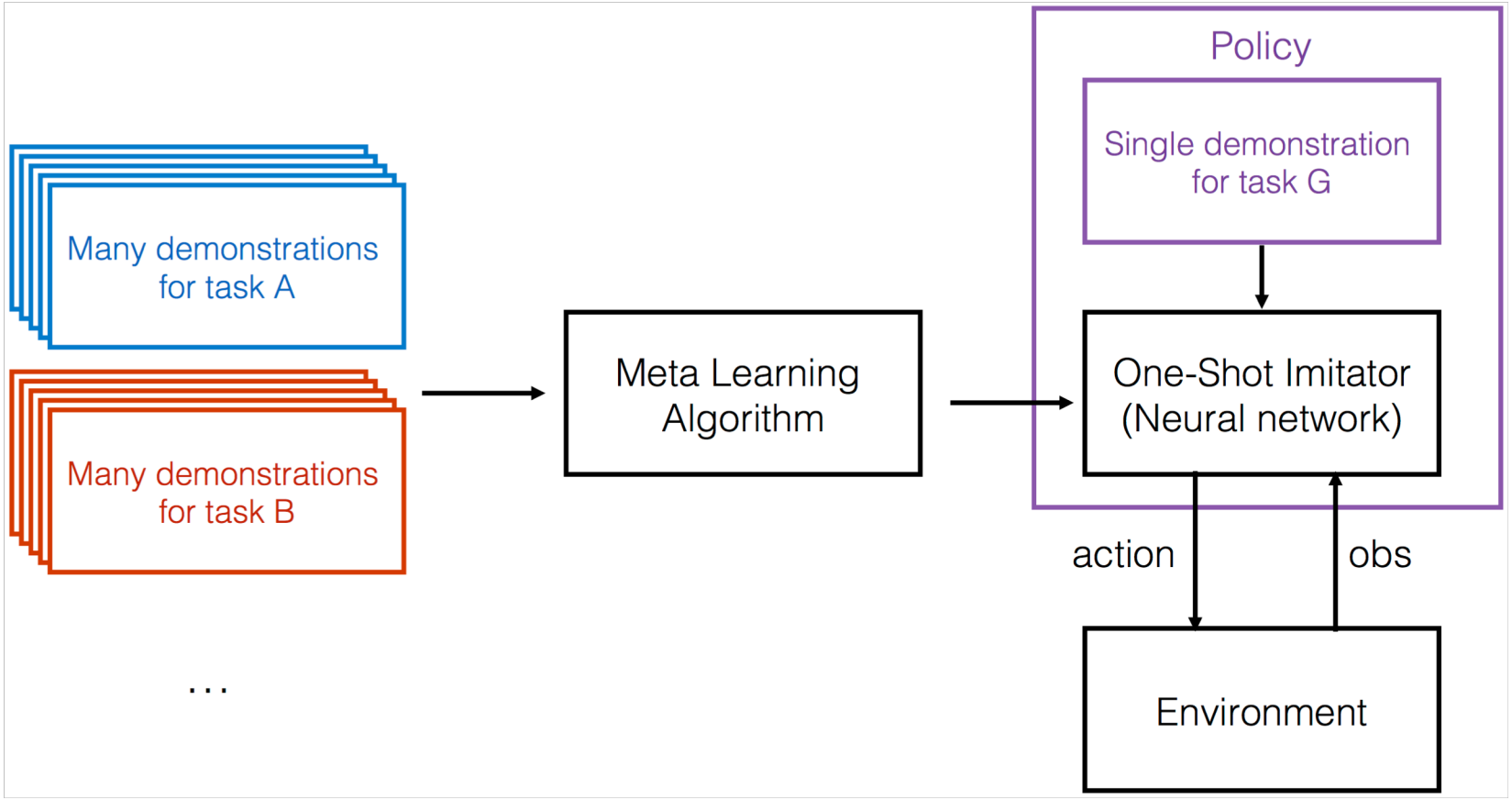
[Duan, Andrychowicz, Stadie, Ho, Schneider, Sutskever, Abbeel, Zaremba, 2017]

One-Shot Imitation Learning



[Duan, Andrychowicz, Stadie, Ho, Schneider, Sutskever, Abbeel, Zaremba, 2017]

One-Shot Imitation Learning



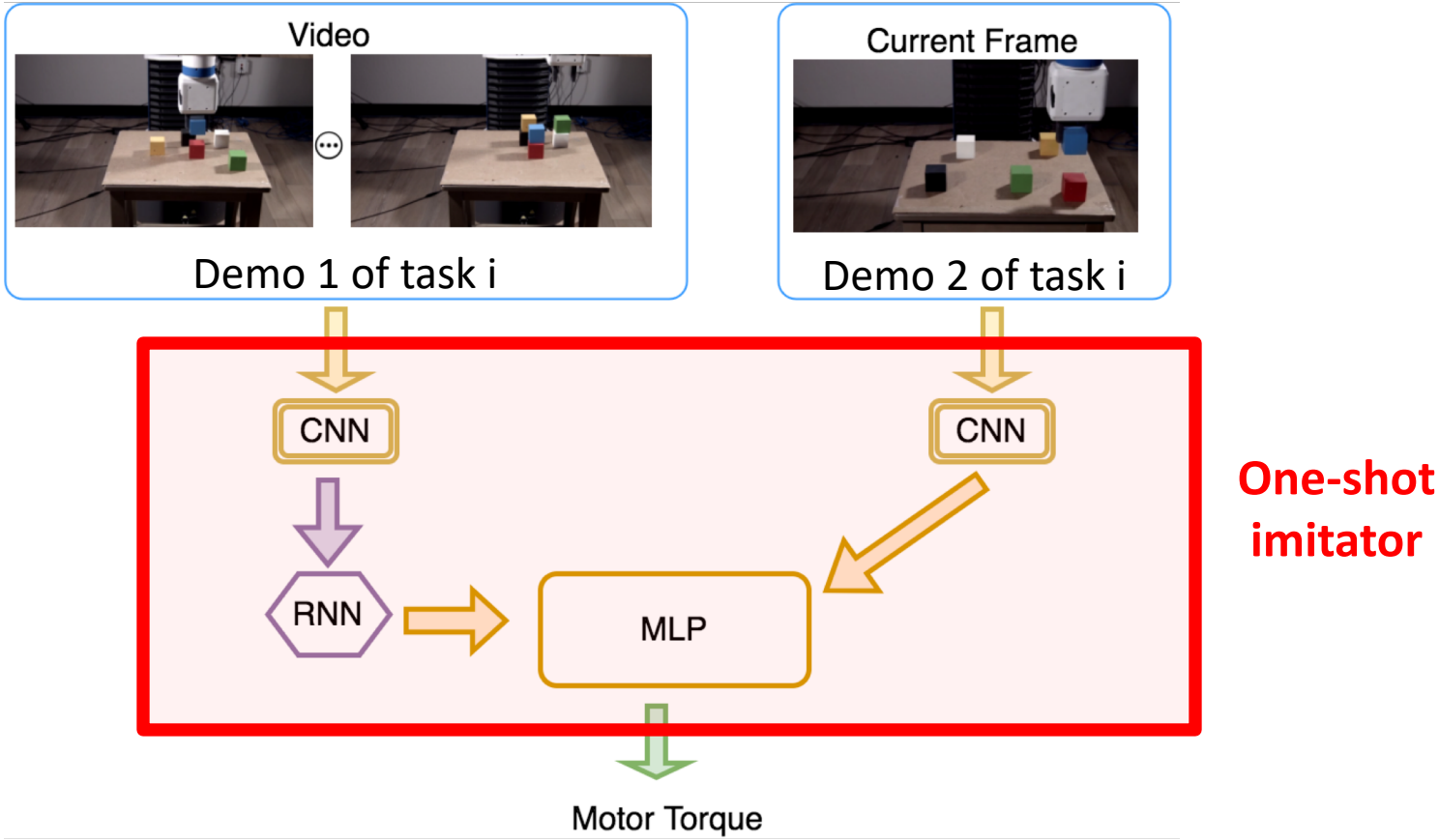
[Duan, Andrychowicz, Stadie, Ho, Schneider, Sutskever, Abbeel, Zaremba, 2017]

Two One Shot-Imitation Approaches

- Learn an RNN that knows how to parse a single demo into what it means to imitate [Duan et al, NIPS 2017]

- Behavioral cloning MAML [Finn*, Yu*, et al, CoRL 2017]

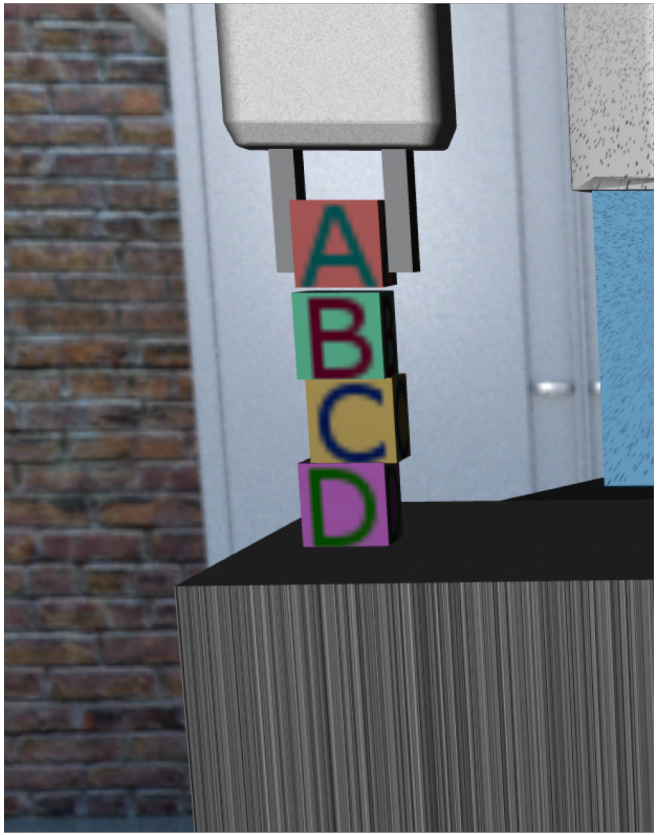
Learning a One-Shot Imitator



[Figure credit: Bradly Stadie]

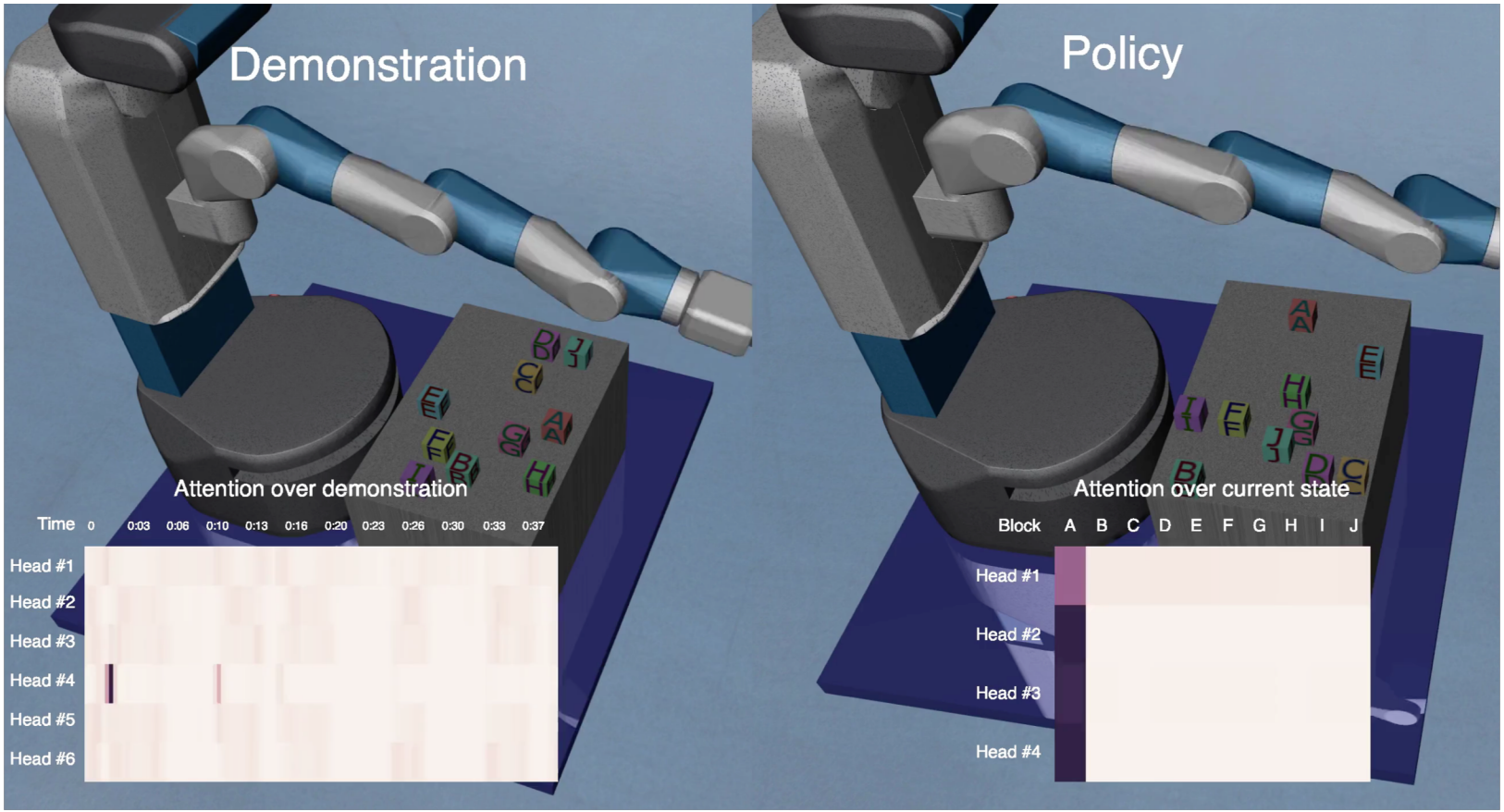
Proof-of-concept: Block Stacking

- Each task is specified by a desired final layout
  - Example: abcd
    - “Place c on top of d, place b on top of c, place a on top of b.”



[Duan, Andrychowicz, Stadie, Ho, Schneider, Sutskever, Abbeel, Zaremba, NIPS 2017]

Evaluation

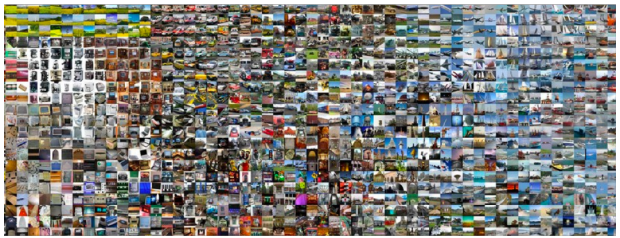


[Duan, Andrychowicz, Stadie, Ho, Schneider, Sutskever, Abbeel, Zaremba, NIPS 2017]

Model-Agnostic Meta-Learning (MAML)

- Starting observation:
  - Computer vision practice:
    - Train on ImageNet [Deng et al. '09]
    - Fine-tune on actual task

    works really well! [Decaf: Donahue et al. '14; …]



- Questions:
  - How to generalize this to behavior learning?
  - And can we explicitly train end-to-end for being maximally ready for efficient fine-tuning?

Model-Agnostic Meta-Learning (MAML)

<u>Key idea:</u> End-to-end learning of parameter vector θ that is good init for fine-tuning for many tasks

finetuned parameters

pretrained parameters

MAML test time: fine-tuning: $\theta' \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\text{tr}}(\theta)$

train data for new task

MAML training: $\min_\theta \sum_{\text{tasks } i} \mathcal{L}^{(i)}_{\text{val}}(\underbrace{\theta - \alpha \nabla_\theta \mathcal{L}^{(i)}_{\text{train}}(\theta)}_{\theta^{(i)\prime}})$

finetuned parameters $\theta^{(i)\prime}$

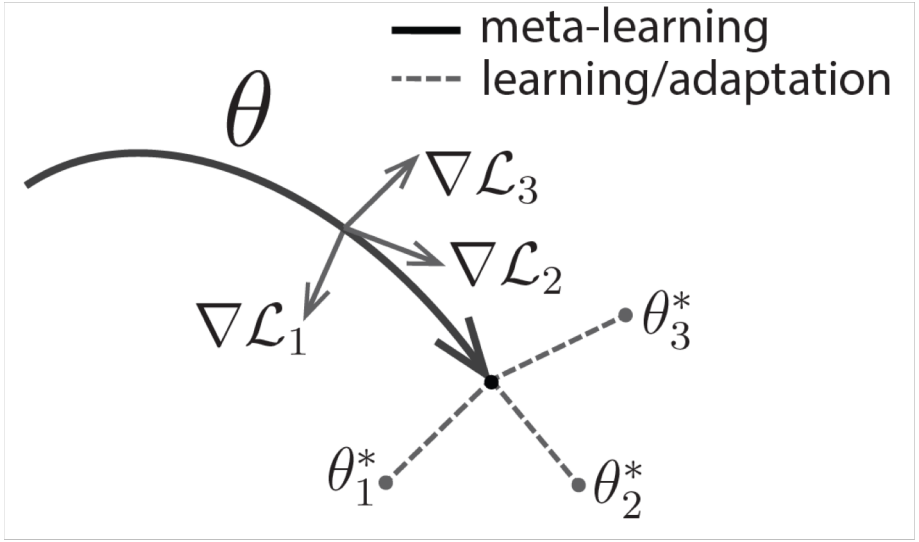
[Finn, Abbeel, Levine ICML 2017]

Model-Agnostic Meta-Learning (MAML)

$$\min_{\theta} \sum_{\text{tasks } i} \mathcal{L}^{(i)}_{\text{val}}(\theta - \alpha \nabla_{\theta} \mathcal{L}^{(i)}_{\text{train}}(\theta))$$

θ parameter vector being meta-learned

θ^*_i optimal parameter vector for task i



Learning a One-Shot Imitator with MAML

- Meta-learning loss:

$$\min_{\theta} \sum_{\text{tasks } i} \mathcal{L}^{(i)}_{\text{val}}(\theta - \alpha \nabla_{\theta} \mathcal{L}^{(i)}_{\text{train}}(\theta))$$

- Task loss = behavioral cloning loss: [Pomerleau'89,Sammut'92]

$$\mathcal{L}(\theta) = \sum_{t} \| \pi_{\theta}(o_t) - a^*_t \|^2$$

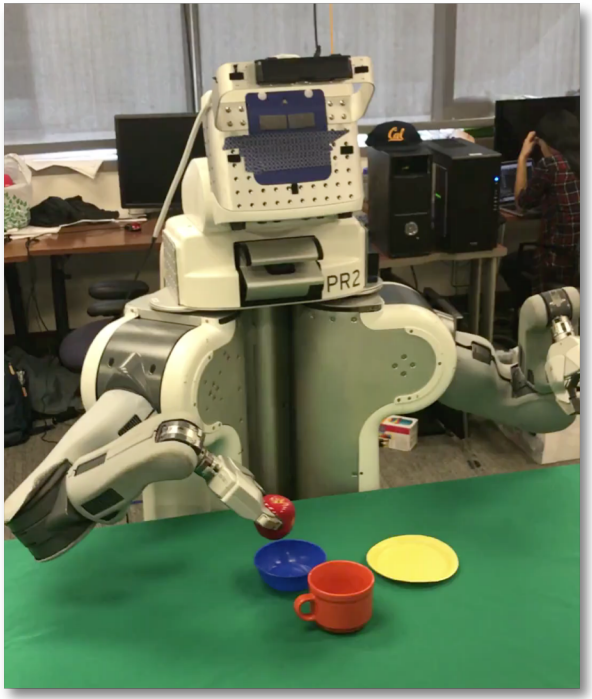
[Finn*, Yu*, Zhang, Abbeel, Levine, CoRL 2017]

Object placing from pixels

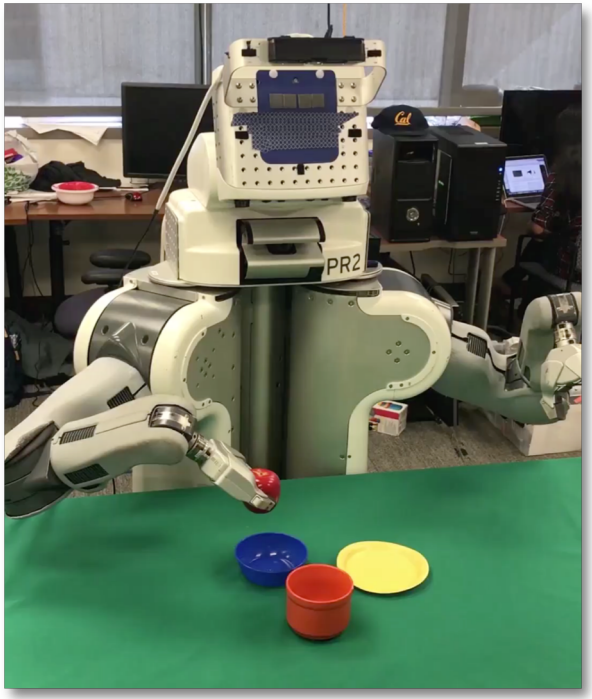
subset of training objects

held-out test objects

input demo
(via teleoperation)



resulting policy



[real-time execution]

Finn*, Yu*, Zhang, Abbeel, Levine CoRL '17

Can We Learn from Just Video?

- Recall, meta-learning loss:

$$\min_{\theta} \sum_{\text{tasks } i} \mathcal{L}^{(i)}_{\text{val}}(\theta - \alpha \nabla_{\theta} \mathcal{L}^{(i)}_{\text{train}}(\theta)) \qquad \text{with} \qquad \mathcal{L}(\theta) = \sum_{t} \| \pi_{\theta}(o_t) - a^*_t \|^2$$

- Key idea: different loss for "val" and "train"

$$\mathcal{L}_{\text{val}}(\theta) = \sum_{t} \| \pi_{\theta}(o_t) - a^*_t \|^2$$

"val" only needed during meta-training, and continues to assume access to action taken

$$\mathcal{L}_{\text{train}}(\theta) = \sum_{t} \| f_{\theta}(o_t) - o^*_{t+1} \|^2$$

"train" doesn't require access to action taken, so at meta-testing video suffices

[Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine, RSS 2018]

Can We Learn from Just Video?

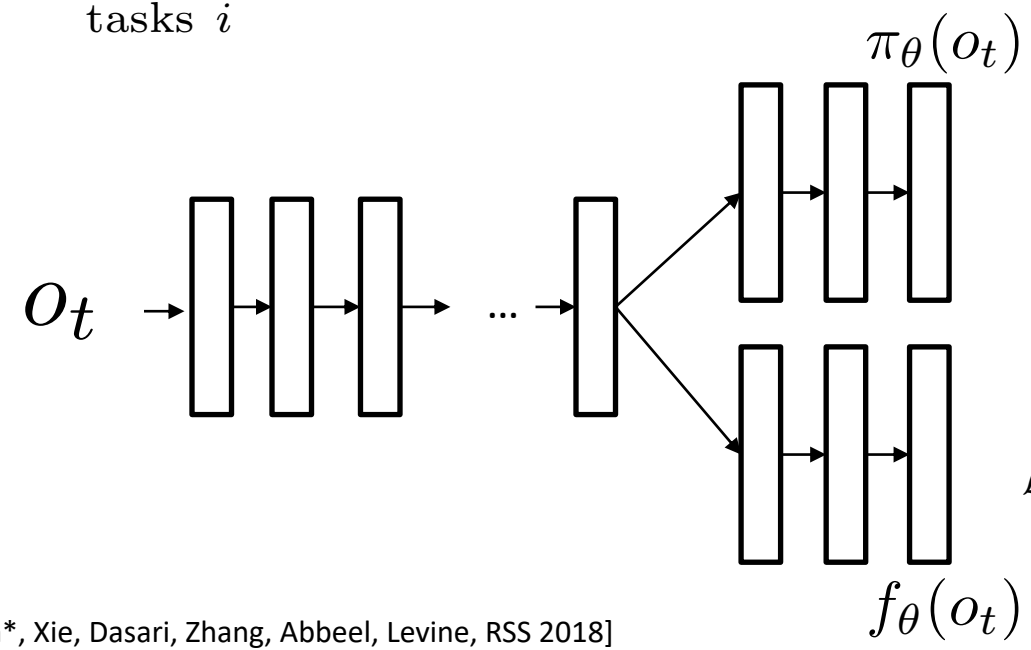
- Recall, meta-learning loss:

$$\min_{\theta} \sum_{\text{tasks } i} \mathcal{L}^{(i)}_{\text{val}}(\theta - \alpha \nabla_{\theta} \mathcal{L}^{(i)}_{\text{train}}(\theta))$$

with

$$\mathcal{L}_{\text{val}}(\theta) = \sum_t \|\pi_\theta(o_t) - a^*_t\|^2$$

$$\mathcal{L}_{\text{train}}(\theta) = \sum_t \|f_\theta(o_t) - o^*_{t+1}\|^2$$

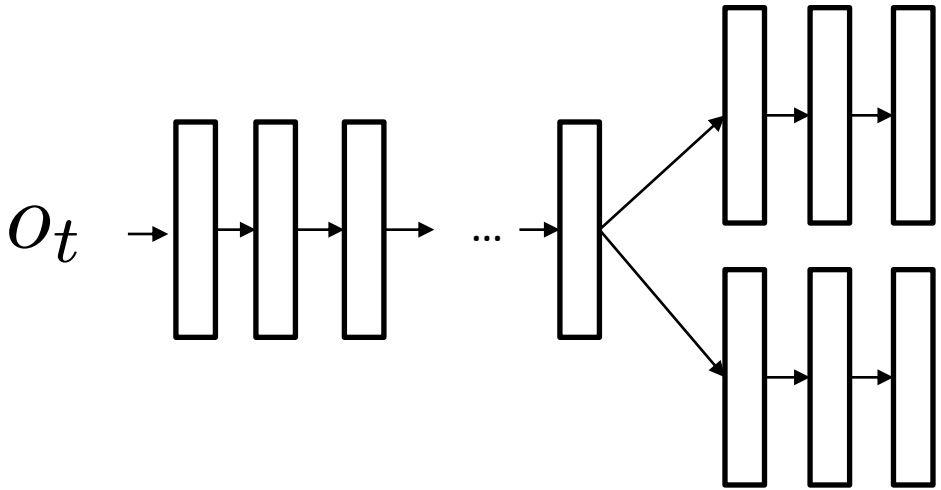


$$\mathcal{L}_{\text{val}}(\theta) = \sum_t \|\pi_\theta(o_t) - a^*_t\|^2$$

$$\mathcal{L}_{\text{train}}(\theta) = \sum_t \|f_\theta(o_t) - o^*_{t+1}\|^2$$

[Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine, RSS 2018]

Can We Learn from Just Video?



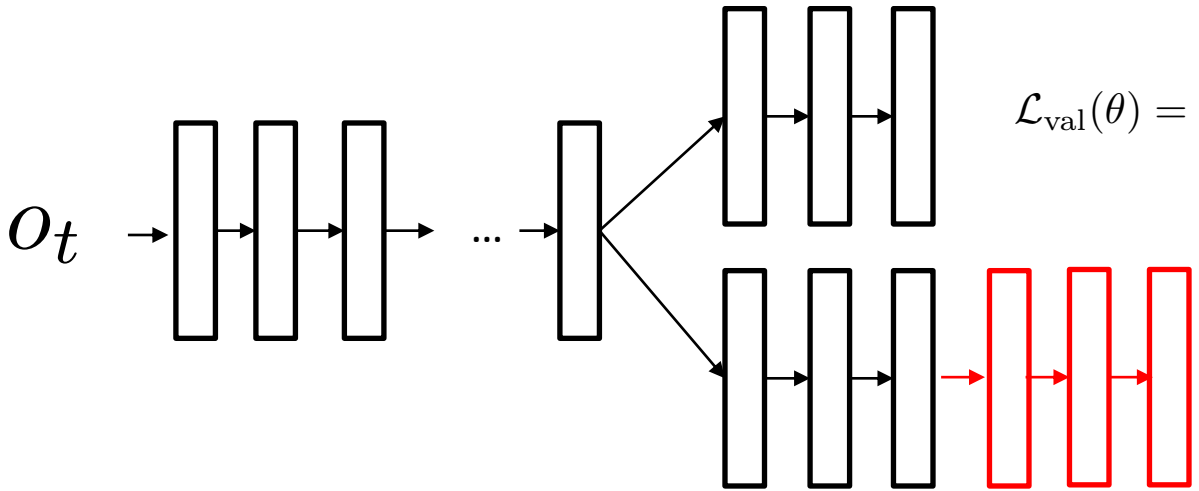
$$\mathcal{L}_{\text{val}}(\theta) = \sum_t \|\pi_\theta(o_t) - a_t^*\|^2$$

$$\mathcal{L}_{\text{train}}(\theta) = \sum_t \|f_\theta(o_t) - o_{t+1}^*\|^2$$

Issue: direct pixel level prediction not a great loss function…

[Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine, RSS 2018]

Can We Learn from Just Video?



$$\mathcal{L}_{\text{val}}(\theta) = \sum_t \|\pi_\theta(o_t) - a_t^*\|^2$$

$$\mathcal{L}_\psi(f_\theta(o_t), o_{t+1}^*)$$

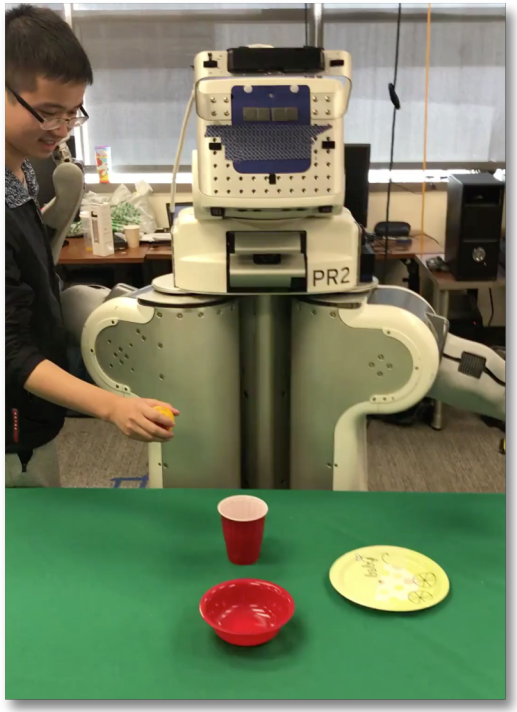
~~$$\mathcal{L}_{\text{train}}(\theta) = \sum_t \|f_\theta(o_t) - o_{t+1}^*\|^2$$~~

$\mathcal{L}_\psi$ can be thought of as (learned) Discriminator in GANs

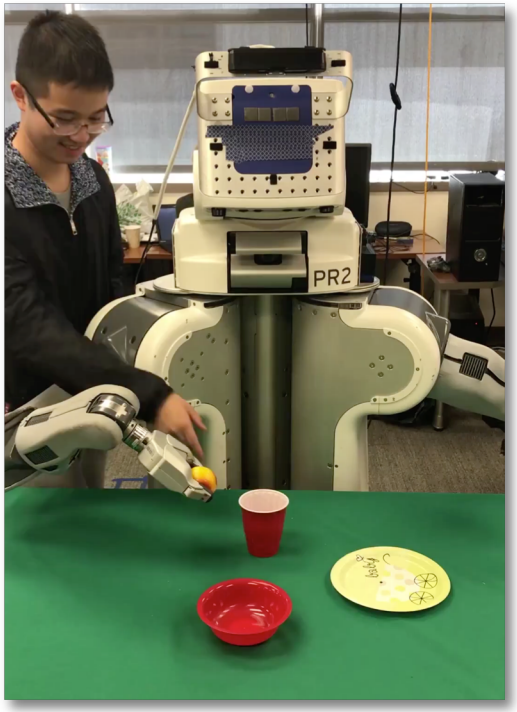
[Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine, RSS 2018]

One-shot imitation from human video

input human demo



one-shot imitation policy



[Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine, RSS 2018]

Many Exciting Challenges in AI for Robotics!

- Few-Shot Reinforcement Learning
- Leveraging Simulation
- Model-based RL
- Learning Representations for Exploration
- Few-Shot Imitation Learning

- Wider distribution over tasks?
- Laborious to build sims...
- Visual input? Longer time scales?
- Wider distribution over tasks?
- Wider distribution over tasks? Natural language?

covariant.ai

The New York Times

A.I. Researchers Leave Elon Musk Lab to Begin Robotics Start-Up

By CADE METZ NOV. 6, 2017

From left, Peter Chen, chief executive of Embodied Intelligence; Pieter Abbeel, president and chief scientist; Rocky Duan, chief technology officer; and Tianhao Zhang, research scientist. Their new company is working on ways to teach robots to learn on their own rather than be painstakingly programmed.
Amy Harrity for The New York Times

EMERYVILLE, Calif. — During a recent speech at the University of California, Berkeley, Pieter Abbeel played a video clip of a robot doing housework.

- Existing automation largely relies on highly precise, repeated motion. This limits applicability.
- We build AI that drastically expands use cases that can be tackled by robots
- Industries: logistics, warehousing, shipping, food, car manufacturing, electronics manufacturing, …
- Looking forward to chatting with new potential partners!

→ pabbeel@covariant.ai

Thank you

pabbeel@cs.berkeley.edu