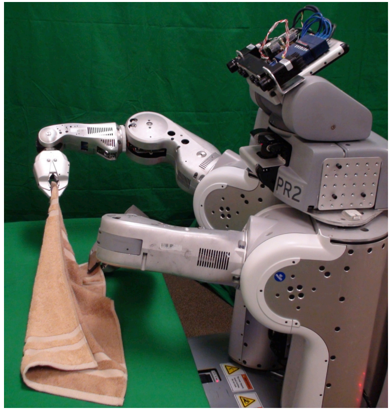
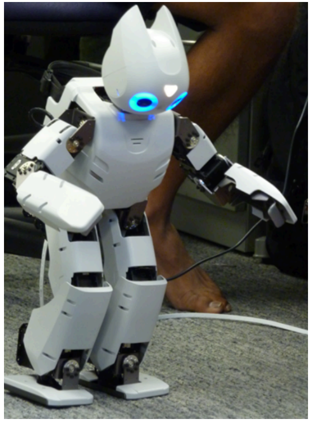
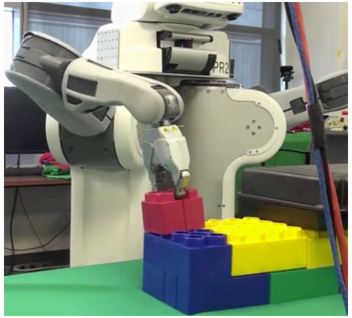






Learning to Learn for Robotic Control

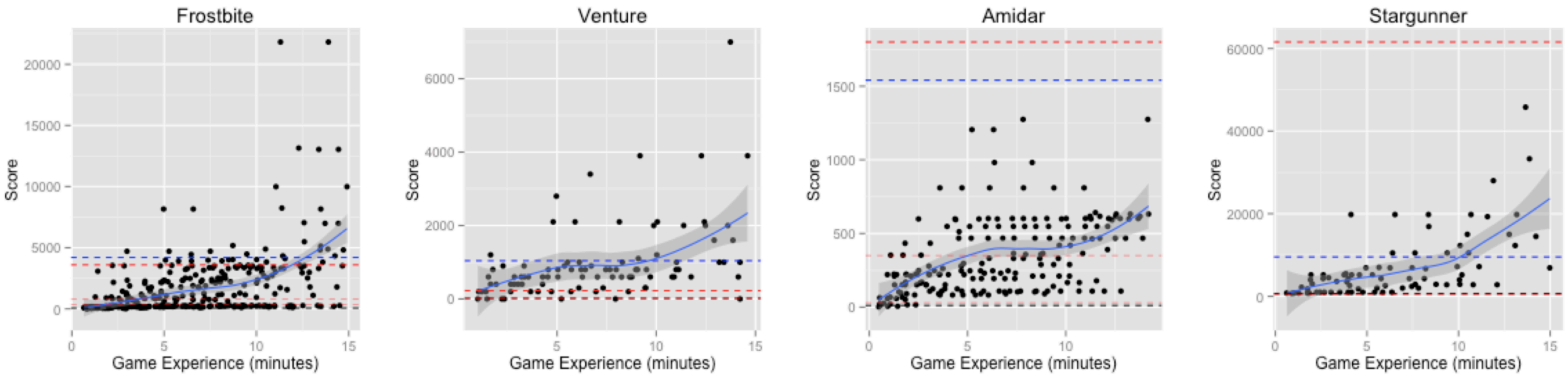
Pieter Abbeel

Embodied Intelligence	AI for robotic automation
UC Berkeley	AI research
Gradescope	AI for grading homework and exams

Research in this talk was done at OpenAI *and* UC Berkeley

Humans vs. DDQN

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

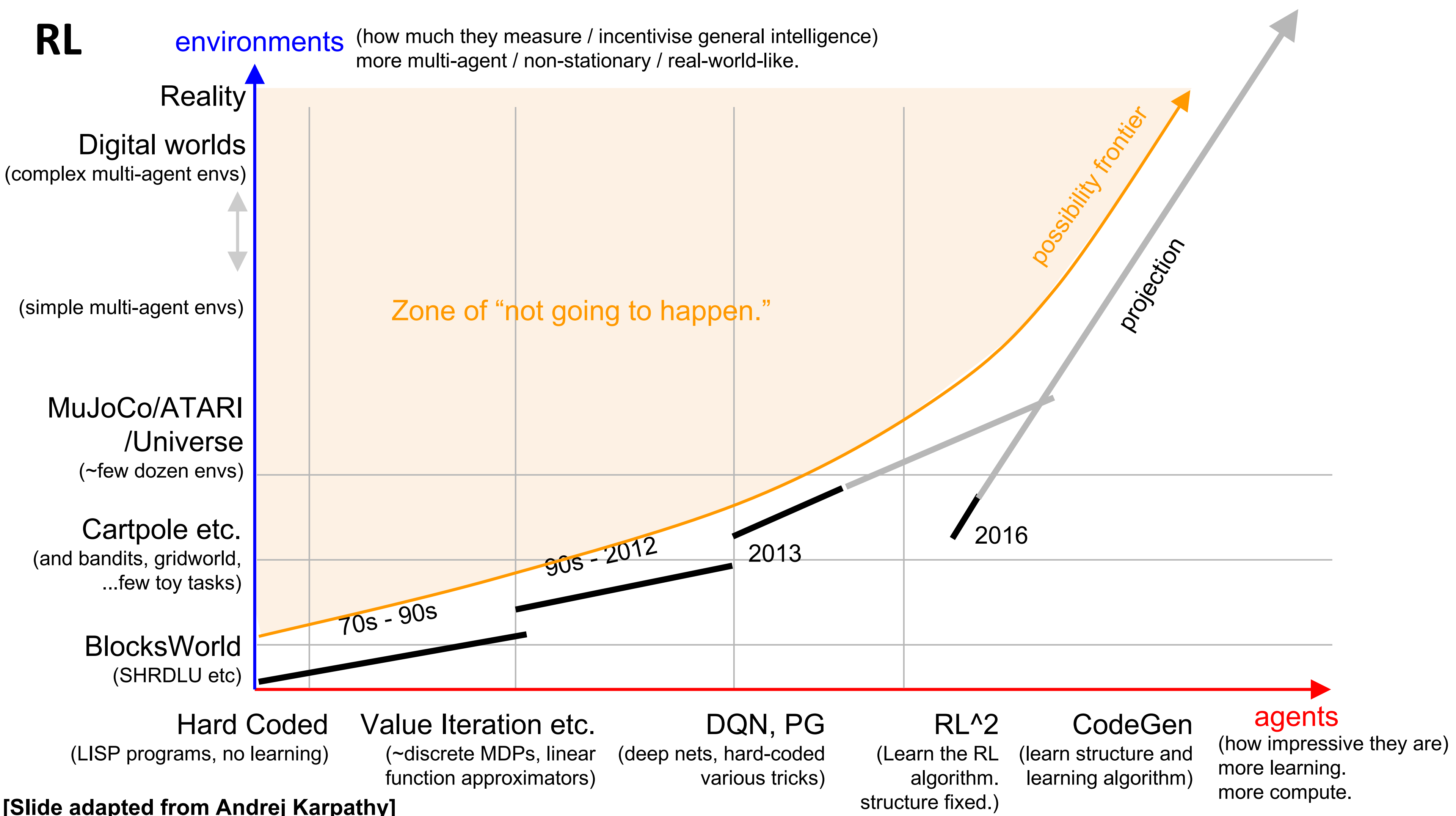
Red dashed lines:

DDQN after 10, 25, 200M frames

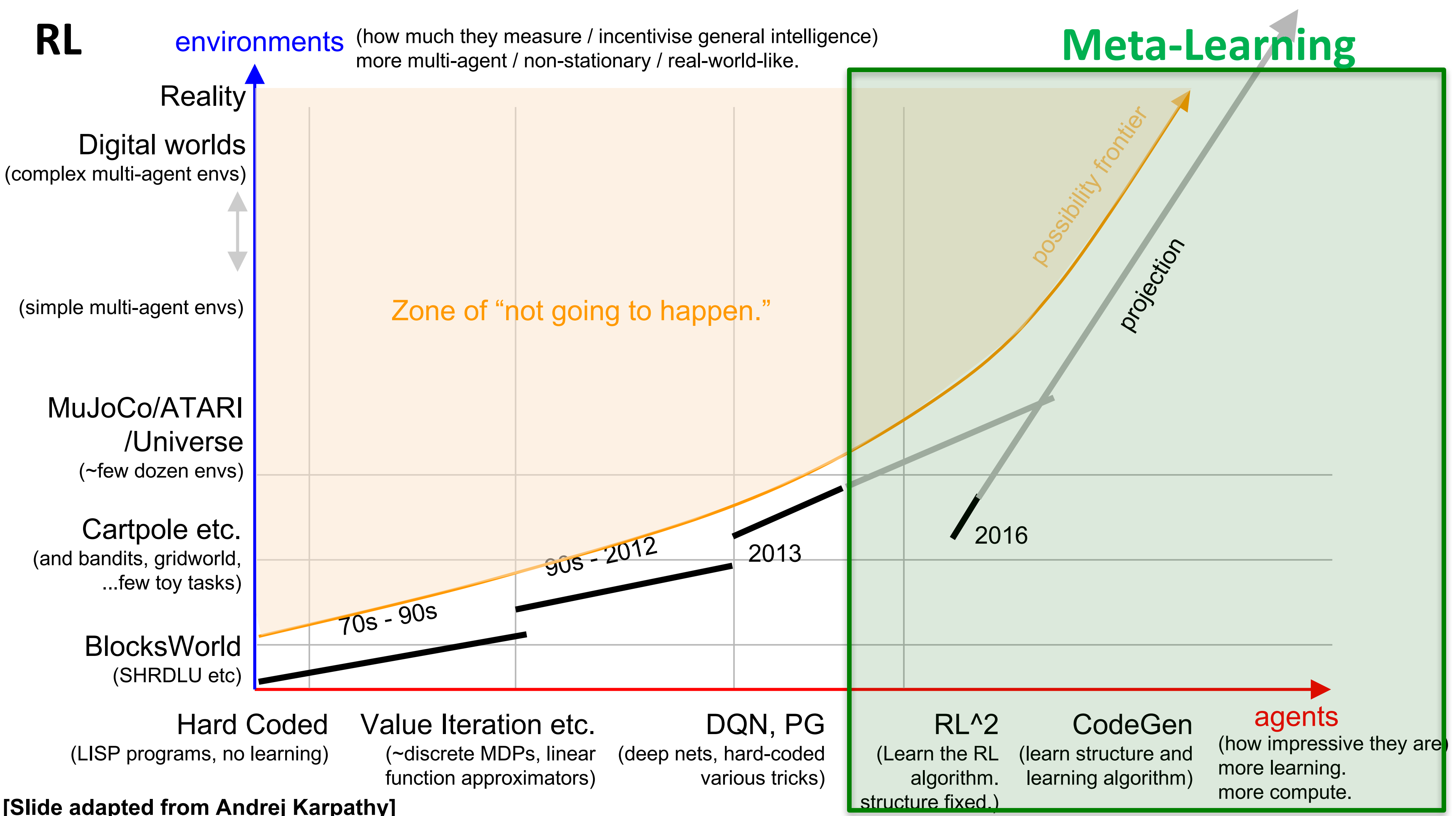
(~ 46, 115, 920 hours)

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

How to bridge this gap?



[Slide adapted from Andrej Karpathy]



[Slide adapted from Andrej Karpathy]

Meta Learning for Optimization

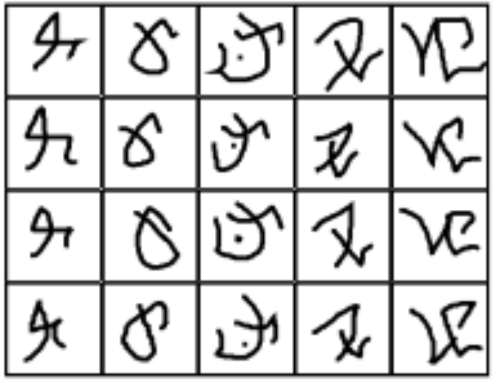
Task distribution: different neural networks, weight initializations, and/or different loss functions

- Bengio et al., (1990) Learning a synaptic learning rule
- Naik et al., (1992) Meta-neural networks that learn by learning
- Hochreiter et al., (2001) Learning to learn using gradient descent
- Younger et al., (2001), Meta learning with back propagation
- Andrychowicz et al., (2016) Learning to learn by gradient descent by gradient descent
- Chen et al., (2016) Learning to Learn for Global Optimization of Black Box Functions
- Wichrowska et al., (2017) Learned Optimizers that Scale and Generalize
- Ke et al., (2017) Learning to Optimize Neural Nets
- Wu et al., (2017) Understanding Short-Horizon Bias in Stochastic Meta-Optimization

Meta Learning for Classification

Task distribution: different classification datasets (input: images, output: class labels)

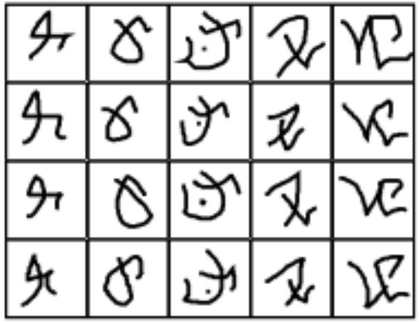
- Hochreiter et al., (2001) Learning to learn using gradient descent
- Younger et al., (2001), Meta learning with back propagation
- Koch et al., (2015) Siamese neural networks for one-shot image recognition
- Santoro et al., (2016) Meta-learning with memory-augmented neural networks
- Vinyals et al., (2016) Matching networks for one shot learning
- Edwards et al., (2016) Towards a Neural Statistician
- Ravi et al., (2017) Optimization as a model for few-shot learning
- Munkhdalai et al., (2017) Meta Networks
- Snell et al., (2017) Prototypical Networks for Few-shot Learning
- Shyam et al., (2017) Attentive Recurrent Comparators
- Finn et al., (2017) Model-Agnostic Meta-Learning for Fast Adaptation of Deep Netwo
- Mehrotra et al., (2017) Generative Adversarial Residual Pairwise Networks for One S
- Mishra et al., (2017) Meta-Learning with Temporal Convolutions
- Li et al., (2017) Meta-SGD: Learning to Learn Quickly for Few Shot Learning
- Finn and Levine, (2017) Meta-Learning and Universality: Deep Representations and Gradient Descent can Approximate any Learning Algorithm
- Anon@OpenReview, (2017) Recasting Gradient-Based Meta-Learning as Hierarchical Bayes



Meta Learning for Generative Models

Task distribution: different unsupervised datasets (e.g. collection of images)

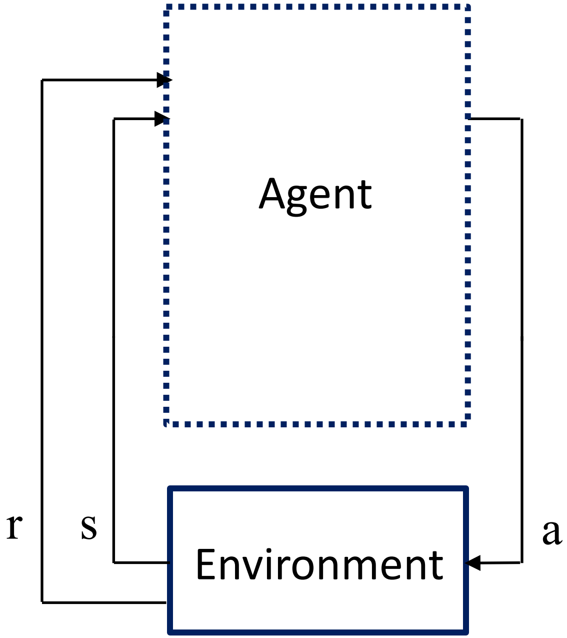
- Rezende et al., (2016) One-Shot Generalization in Deep Generative Models
- Edwards et al., (2016) Towards a Neural Statistician
- Bartunov et al., (2016) Fast Adaptation in Generative Models with Generative Matching Networks
- Bornschein et al., (2017) Variational Memory Addressing in Generative Models
- Reed et al., (2017) Few-shot Autoregressive Density Estimation: Towards Learning to Learn Distributions



Meta-Learning for Control

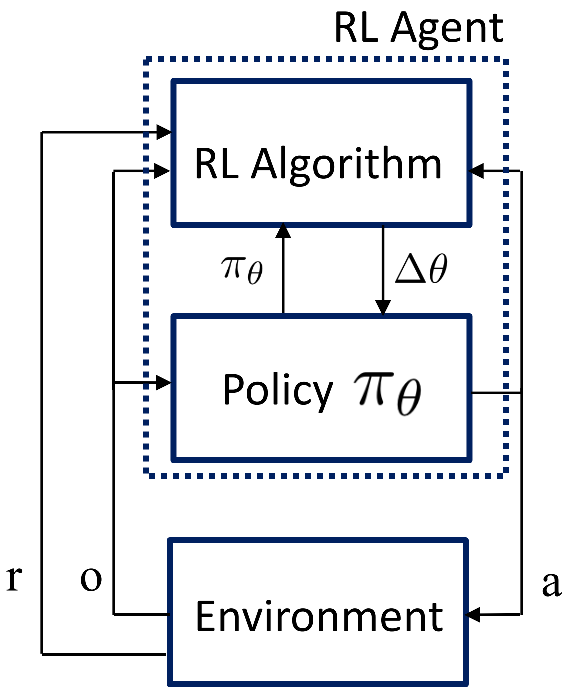
- Learning to Reinforcement Learn
- Learning to Imitate

Reinforcement Learning



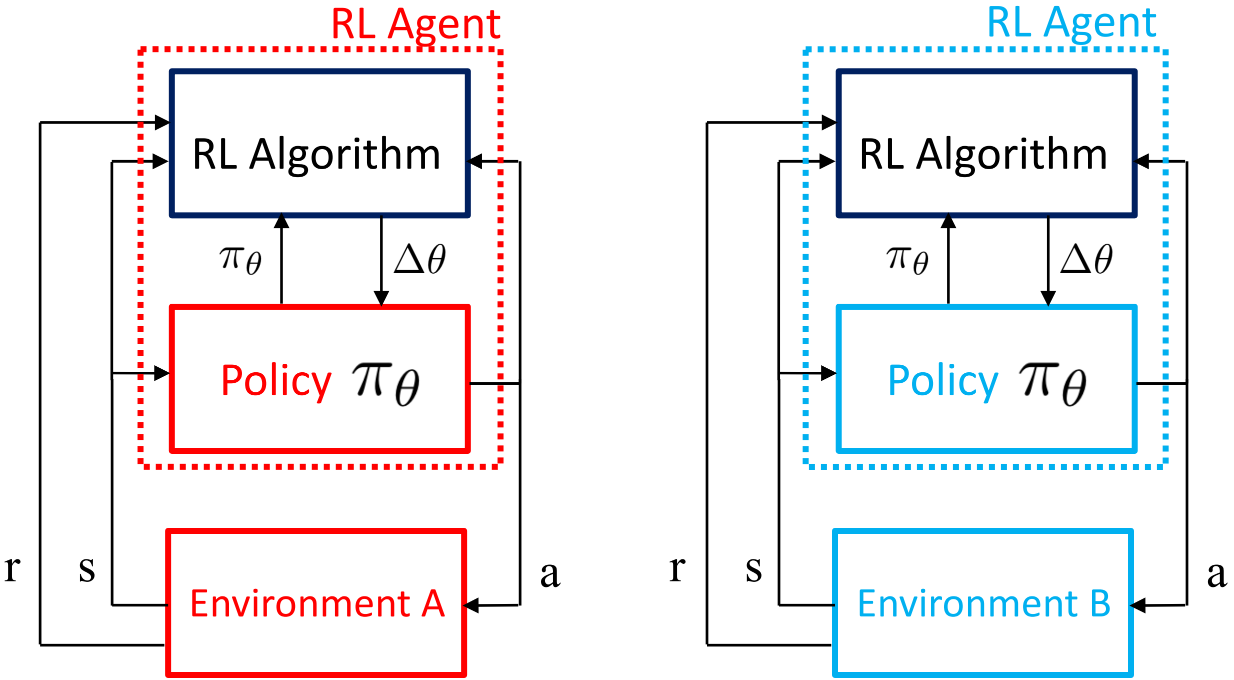
[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016] also: [Wang et al, 2016]

Reinforcement Learning



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

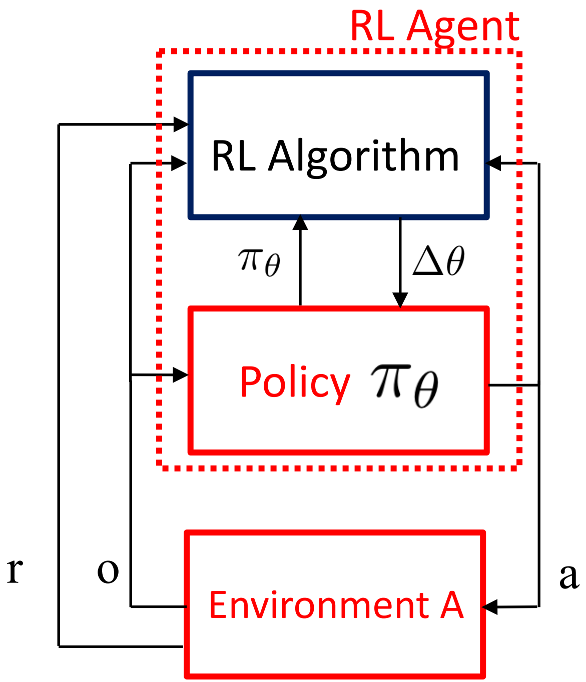
Reinforcement Learning

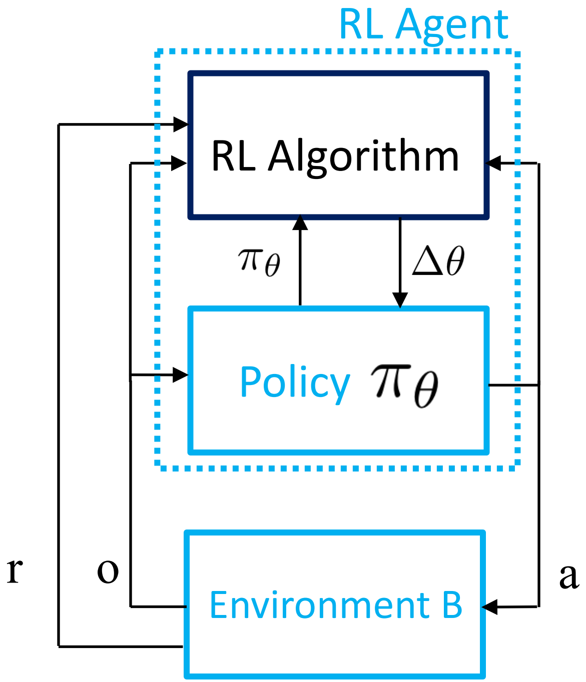


...

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

Reinforcement Learning





Traditional RL research:

- Human experts develop the RL algorithm
- After many years, still no RL algorithms nearly as good as humans…

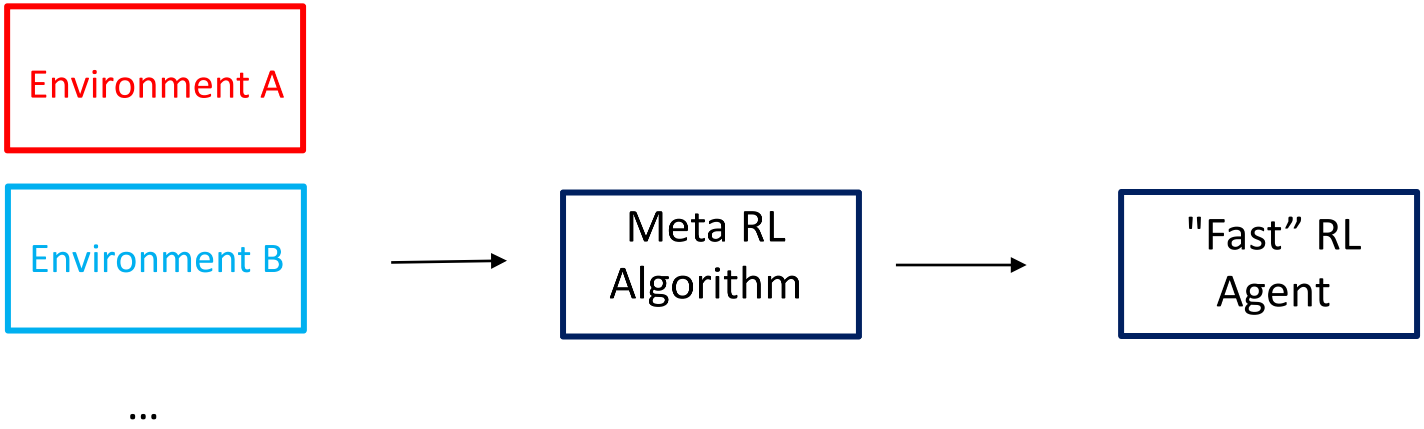
Alternative:

- Could we learn a better RL algorithm?
- Or even learn a better entire agent?

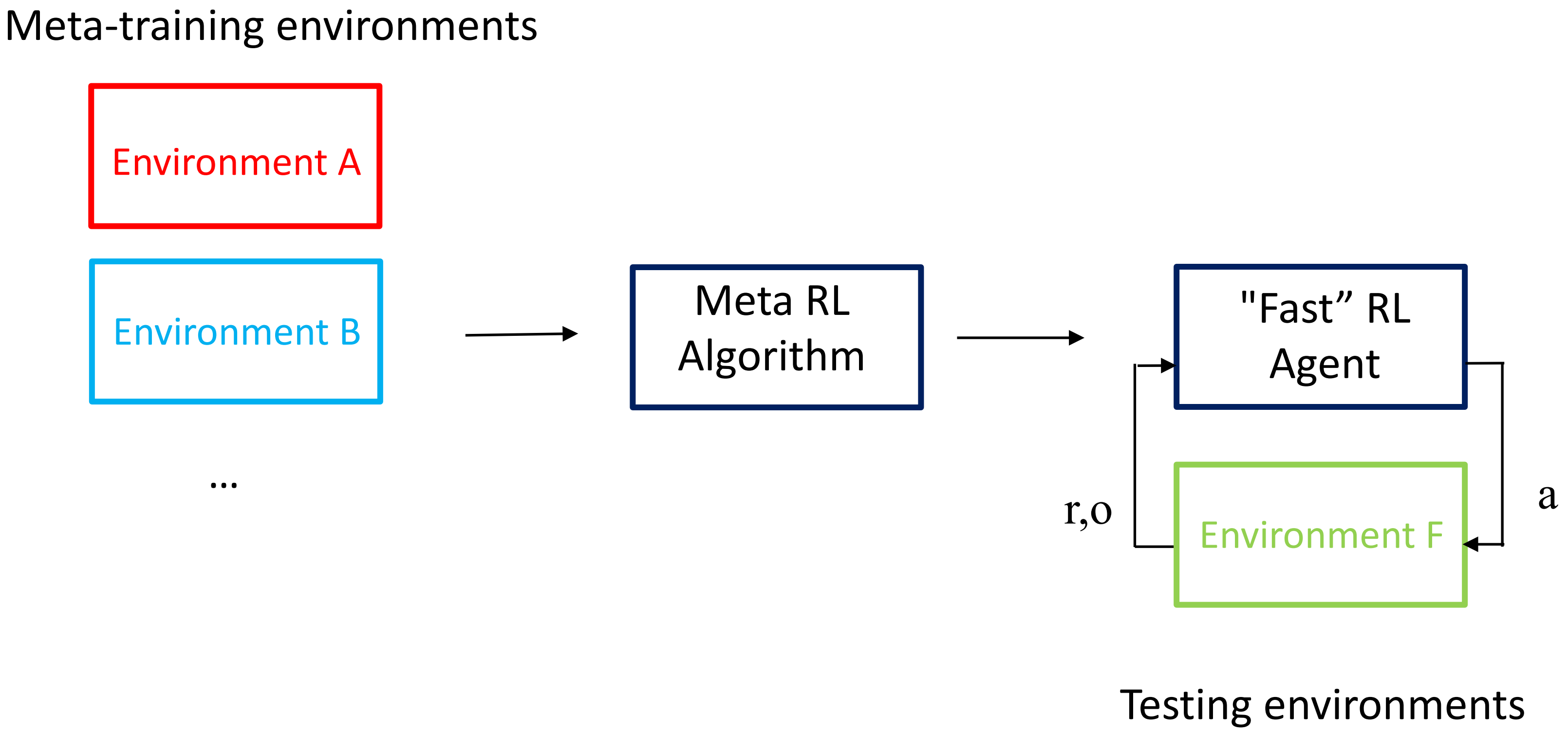
[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016] also: [Wang et al, 2016]

Meta-Reinforcement Learning

Meta-training environments



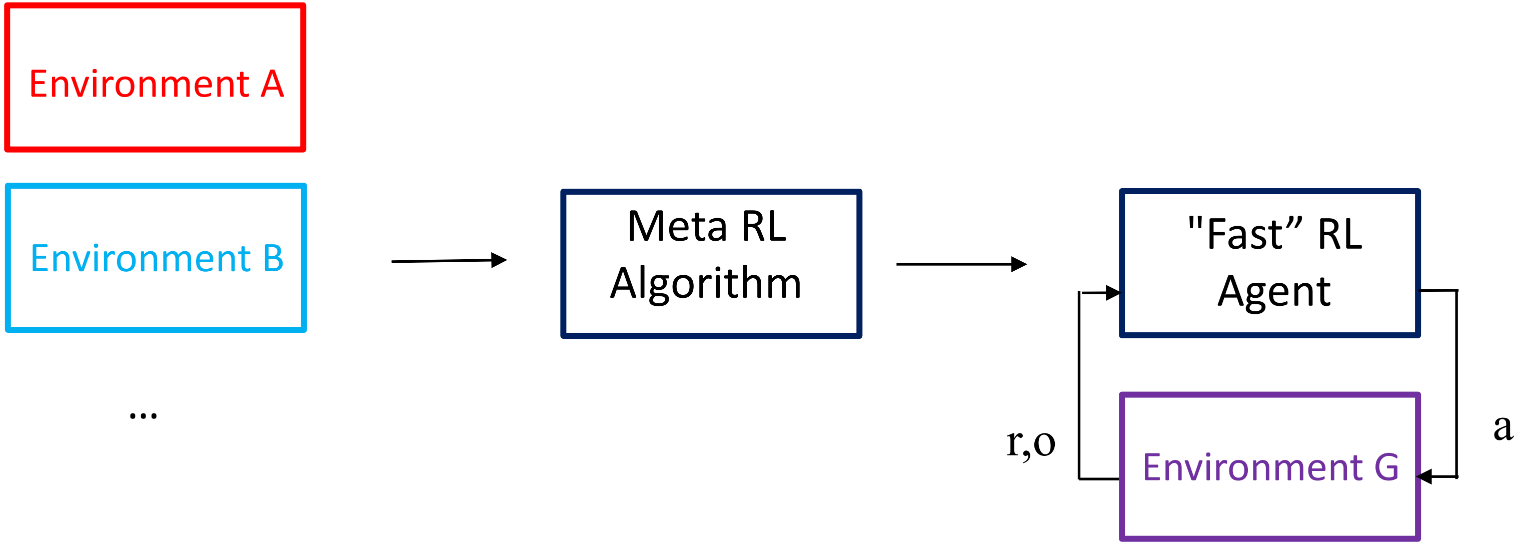
Meta-Reinforcement Learning



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

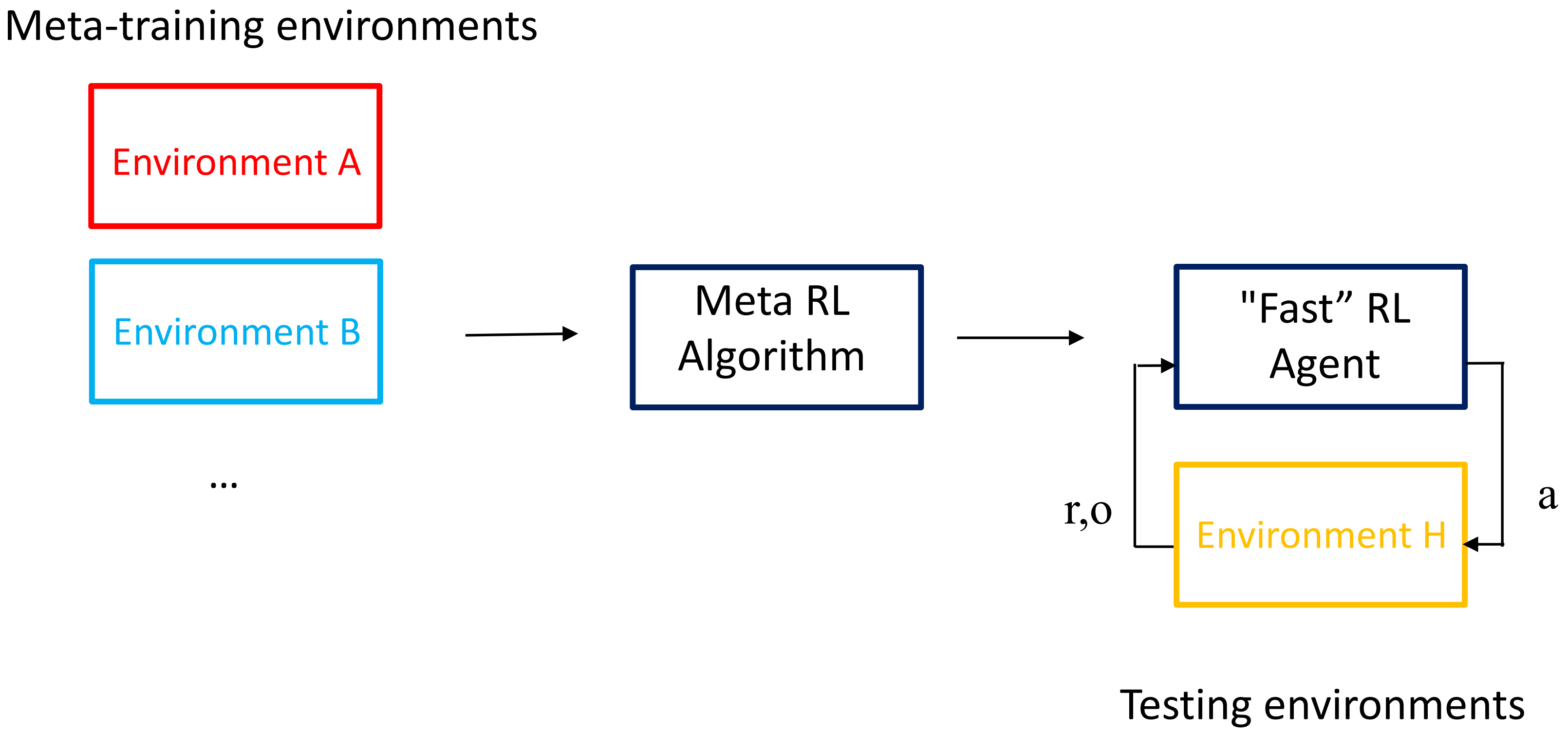
Meta-Reinforcement Learning

Meta-training environments

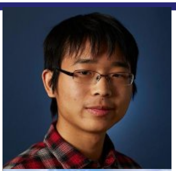
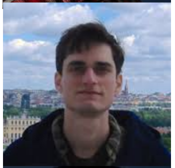
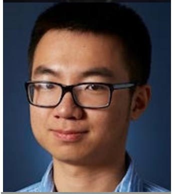


Testing environments

Meta-Reinforcement Learning



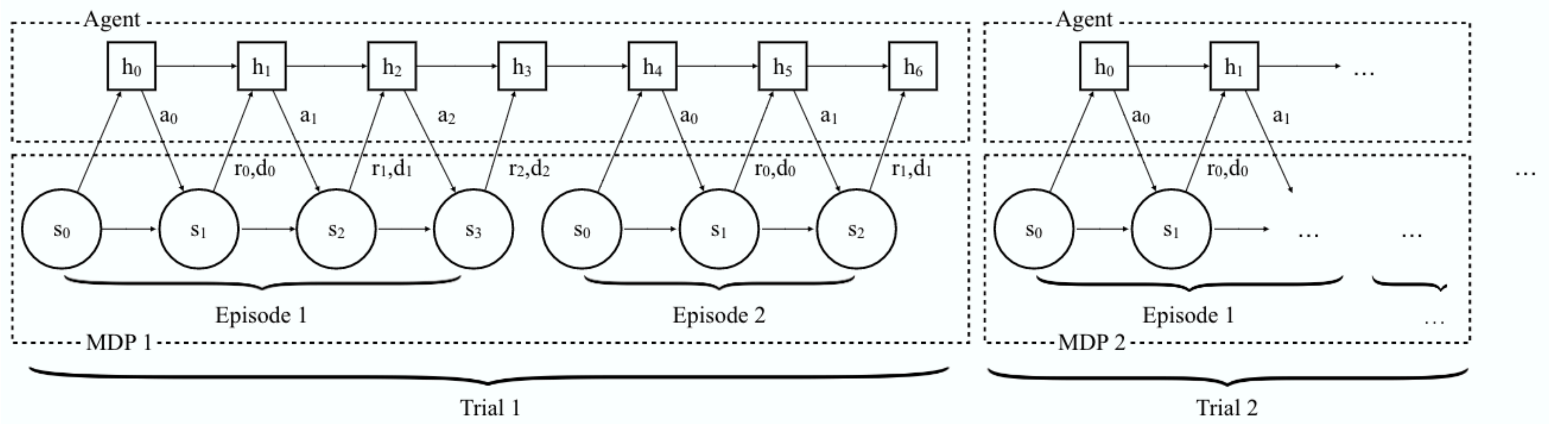
Formalizing Learning to Reinforcement Learn

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

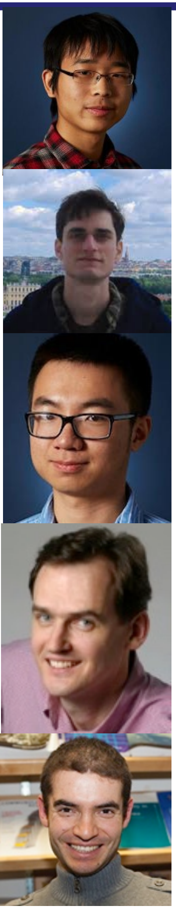
M : sample environment

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



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

Formalizing Learning to Reinforcement Learn



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

M : sample MDP

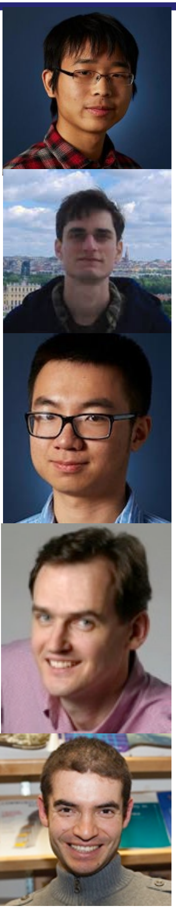
$\tau_M^{(k)}$: k'th trajectory in MDP M

Meta-train:

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

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

Representing RLagent_θ: RL2



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

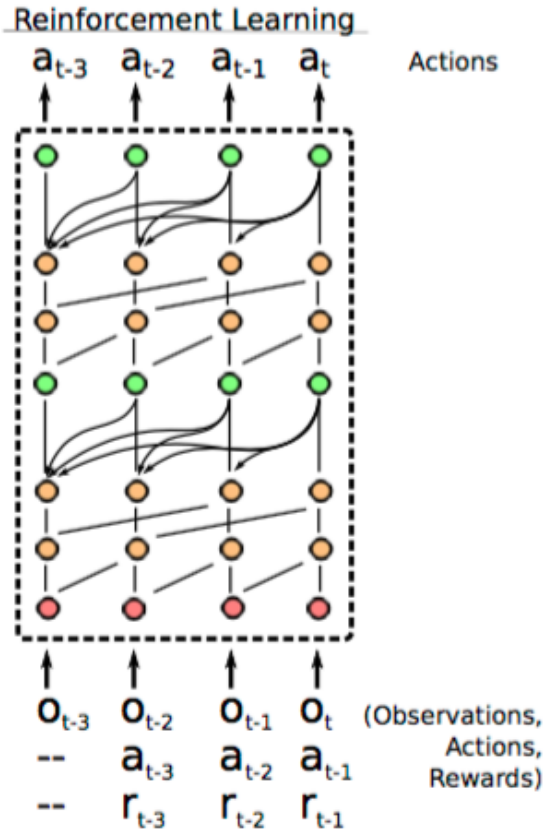
- RLagent = RNN = generic computation architecture
  - different weights in the RNN means different RL algorithm and prior
  - different activations in the RNN means different current policy
  - meta-train objective can be optimized with an existing (slow) RL algorithm

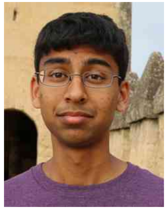
[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016] also: [Wang et al, 2016]

Representing $\mathrm{RLagent}_\theta$: SNAIL

- Like RL2 but:

replace the LSTM with dilated temporal convolution (like wavenet) + attention





[Wavenet: van den Oord et al, 2016]

[Attention-is-all-you-need: Vaswani et al, 2017]

[Mishra*, Rohaninejad*, Chen, Abbeel, 2017]

Representing $\mathrm{RLagent}_\theta$: 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: $\theta' \leftarrow \theta \; - \alpha \nabla_\theta \mathcal{L}_{\mathrm{tr}}(\theta)$

train data for new task

MAML training: $$\min_\theta \sum_{\mathrm{tasks}} \mathcal{L}_{\mathrm{val}}\left(\theta - \alpha \nabla_\theta \mathcal{L}_{\mathrm{train}}(\theta)\right)$$

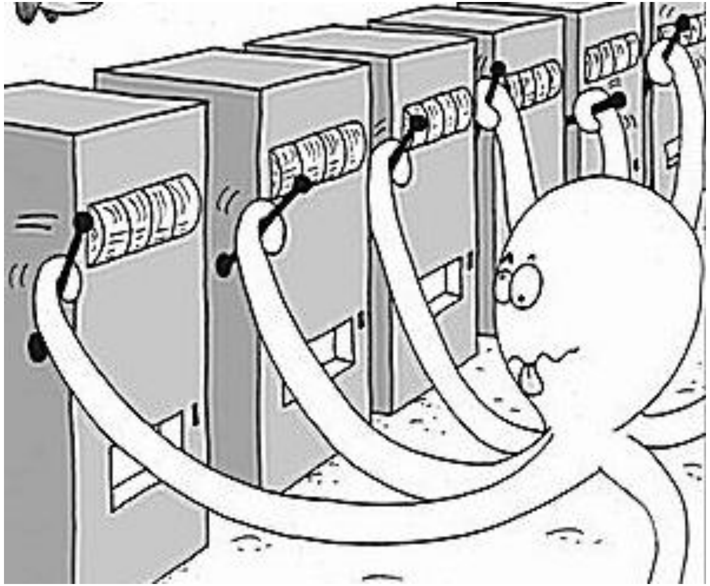
finetuned parameters


[Finn, Abbeel, Levine ICML 2017]

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]

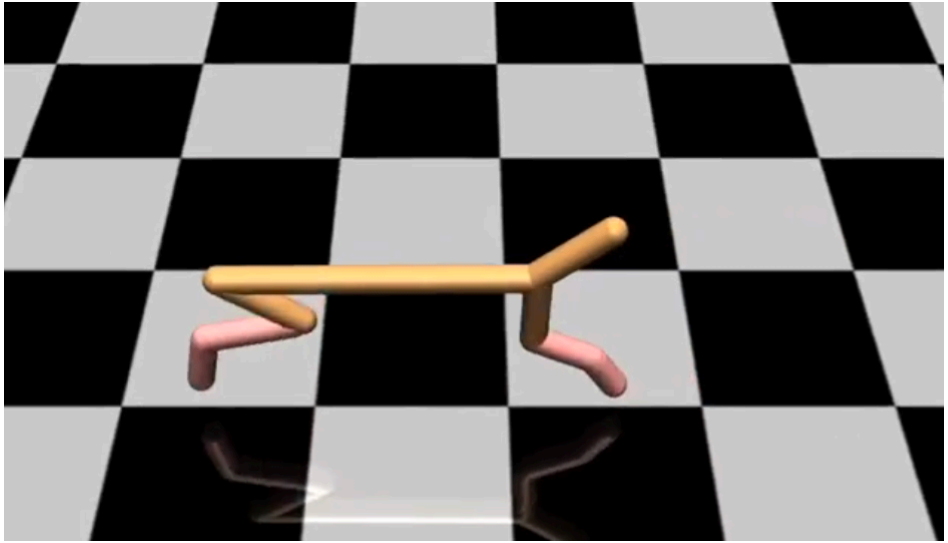
Bandits

Setup (N, K)	Method				
	Gittins (optimal as $N \to \infty$)	Random	RL^2	MAML	SNAIL (ours)
10, 5	**6.6**	5.0	**6.7**	6.5 ± 0.1	$\mathbf{6.6 \pm 0.1}$
10, 10	**6.6**	5.0	**6.7**	$\mathbf{6.6 \pm 0.1}$	$\mathbf{6.7 \pm 0.1}$
10, 50	6.5	5.1	**6.8**	$\mathbf{6.6 \pm 0.1}$	$\mathbf{6.7 \pm 0.1}$
100, 5	**78.3**	49.9	**78.7**	67.1 ± 1.1	$\mathbf{79.1 \pm 1.0}$
100, 10	**82.8**	49.9	**83.5**	70.1 ± 0.6	$\mathbf{83.5 \pm 0.8}$
100, 50	**85.2**	49.8	**84.9**	70.3 ± 0.4	$\mathbf{85.1 \pm 0.6}$
500, 5	**405.8**	249.8	401.5	–	$\mathbf{408.1 \pm 4.9}$
500, 10	**437.8**	249.0	**432.5**	–	$\mathbf{432.4 \pm 3.5}$
500, 50	**463.7**	249.6	438.9	–	442.6 ± 2.5
1000, 50	**944.1**	499.8	847.43	–	889.8 ± 5.6

[Mishra*, Rohaninejad*, Chen, Abbeel, 2017]

Evaluation: Locomotion – Half Cheetah

- Task – reward based on target running direction + speed



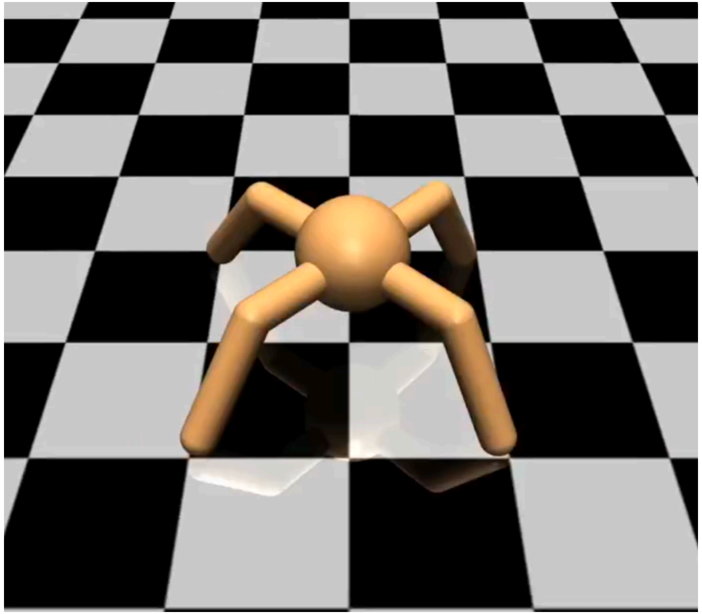
Evaluation: Locomotion – Half Cheetah

- Task – reward based on target running direction + speed

- Result of meta-training = a single agent (the “fast RL agent”), which masters each task almost instantly within 1st episode

Evaluation: Locomotion – Ant

- Task – reward based on target running direction + speed



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

Evaluation: Locomotion – Ant

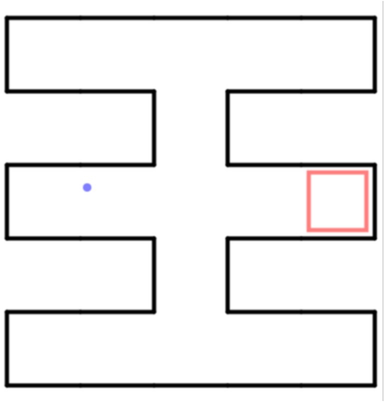
- Task – reward based on target running direction + speed

- Result of meta-training = a single agent (the "fast RL agent"), which masters each task almost instantly within 1st episode

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

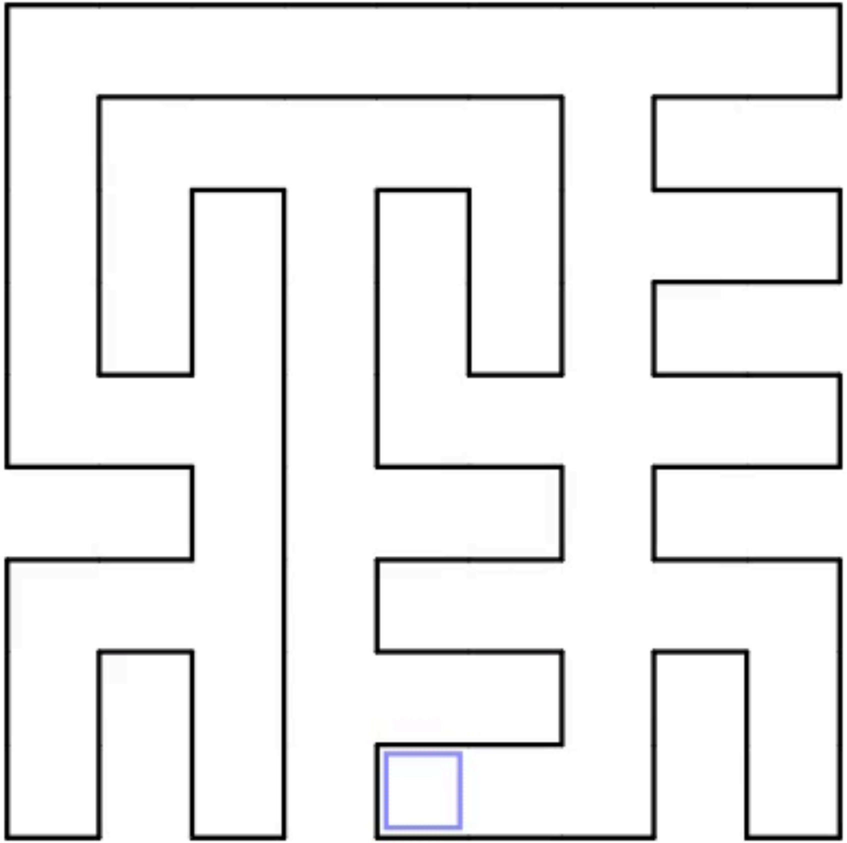
Agent's view



Maze

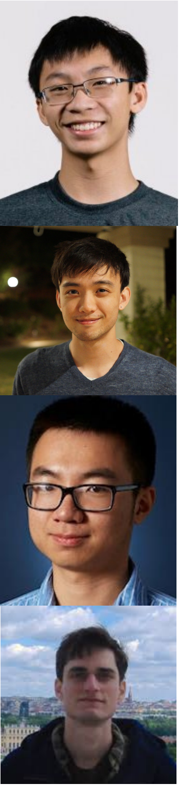
Related work: Mirowski, et al, 2016; Jaderberg et al, 2016; Mnih et al, 2016; Wang et al, 2016

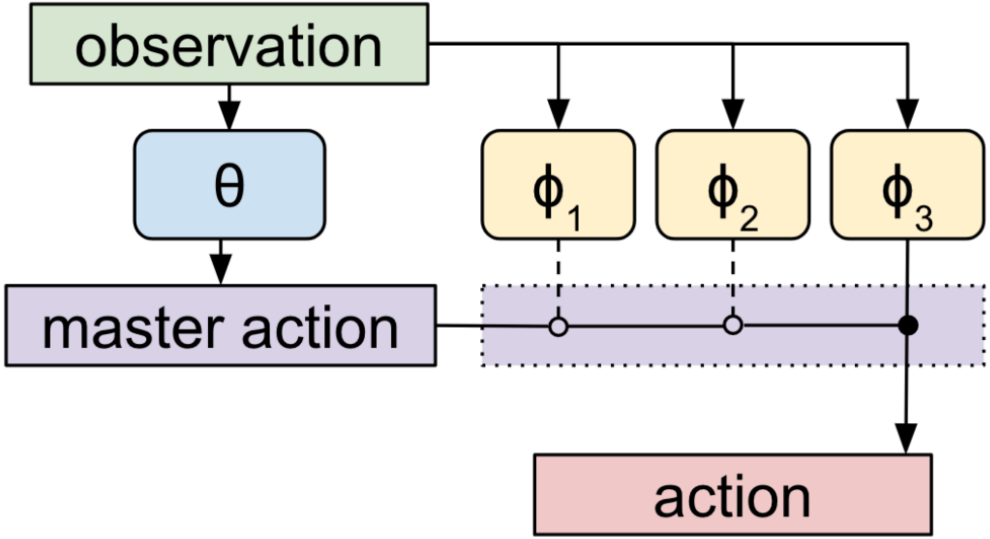
Agent Dropped in New Maze



[Mishra*, Rohaninejad*, Chen, Abbeel, 2017]

Meta-Learning Shared Hierarchies





Goal: find subpolicies that enable fast learning of master policy θ

[Frans, Ho, Chen, Abbeel, Schulman, 2017]

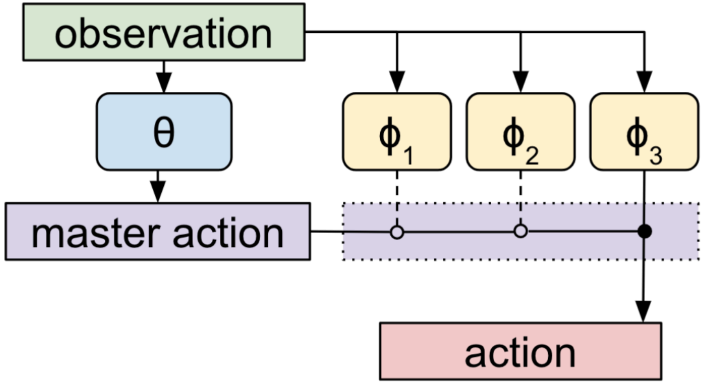
Meta-Learning Shared Hierarchies

RL2 Meta-Learning Objective:

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

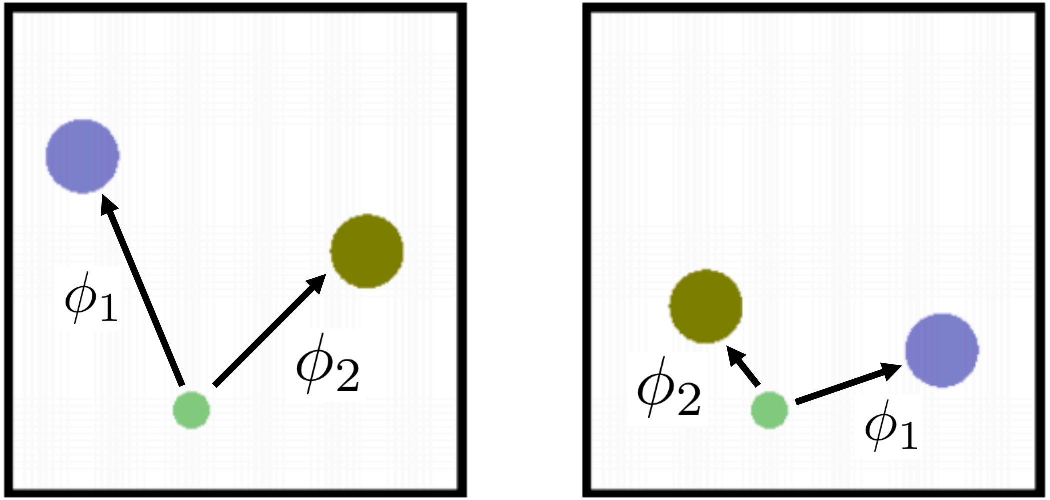
MLSH Meta-Learning Objective:

$$\max_\phi \mathbb{E}_{\theta_0} \mathbb{E}_M \mathbb{E}_{\tau_M^{(k)}} \left[\sum_{k=1}^{K} R(\tau_M^{(k)}) \mid \phi, \mathrm{RLagent}_{\theta_0} \right]$$



= find a set of subpolicies that enable fast learning of the master policy

MLSH -- Experiment 1: Moving Bandits



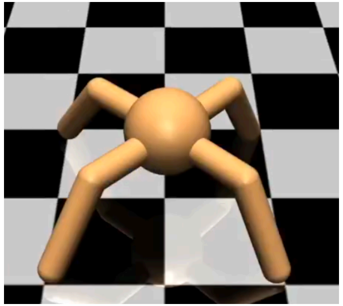
...

Hope for

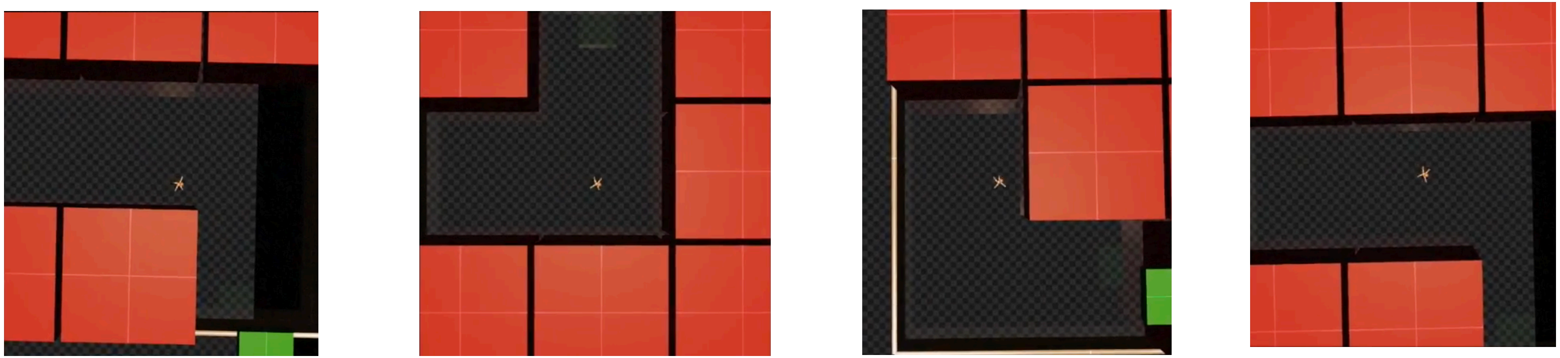
- Learned subpolicies: low level control for each of the targets
- High level policy: standard bandit problem

Episode Duration = 50, Subpolicy Duration = 10

Experiment 2: Maze Navigation



- Episode duration = 1000
- Subpolicy duration = 200



...

Discovered Three Gaits

MLSH agent was trained on nine separate mazes.
It discovered sub-policies for upwards, rightwards, and downwards movement.

Meta Learning for RL

Task distribution: different environments

- Schmidhuber. Evolutionary principles in self-referential learning. (1987)
- Wiering, Schmidhuber. Solving POMDPs with Levin search and EIRA. (1996)
- Schmidhuber, Zhao, Wiering. Shifting inductive bias with success-story algorithm, adaptive Levin search, and incremental self-improvement. (MLJ 1997)
- Schmidhuber, Zhao, Schraudolph. Reinforcement learning with self-modifying policies (1998)
- Zhao, Schmidhuber. Solving a complex prisoner's dilemma with self-modifying policies. (1998)
- Schmidhuber. A general method for incremental self-improvement and multiagent learning. (1999)
- Singh, Lewis, Barto. Where do rewards come from? (2009)
- Singh, Lewis, Barto. Intrinsically Motivated Reinforcement Learning: An Evolutionary Perspective (2010)
- Niekum, Spector, Barto. Evolution of reward functions for reinforcement learning (2011)
- Duan et al., (2016) RL2: Fast Reinforcement Learning via Slow Reinforcement Learning
- Wang et al., (2016) Learning to Reinforcement Learn
- Finn et al., (2017) Model-Agnostic Meta-Learning
- Mishra, Rohinenjad et al., (2017) Simple Neural AttentIve meta-Learner
- Frans et al., (2017) Meta-Learning Shared Hierarchies

Meta-Learning for Control

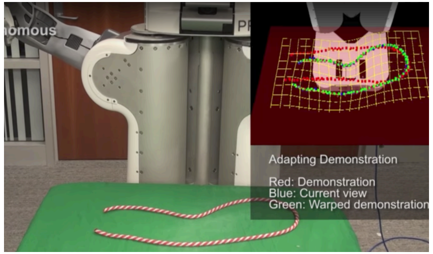
- Learning to Reinforcement Learn
- Learning to Imitate

Imitation Learning in Robotics

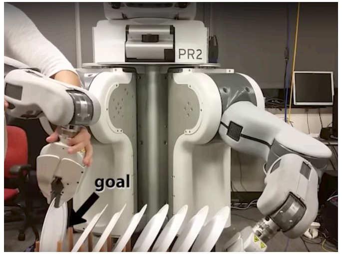
[Abbeel et al. 2008]

[Kolter et al. 2008]

[Ziebart et al. 2008]

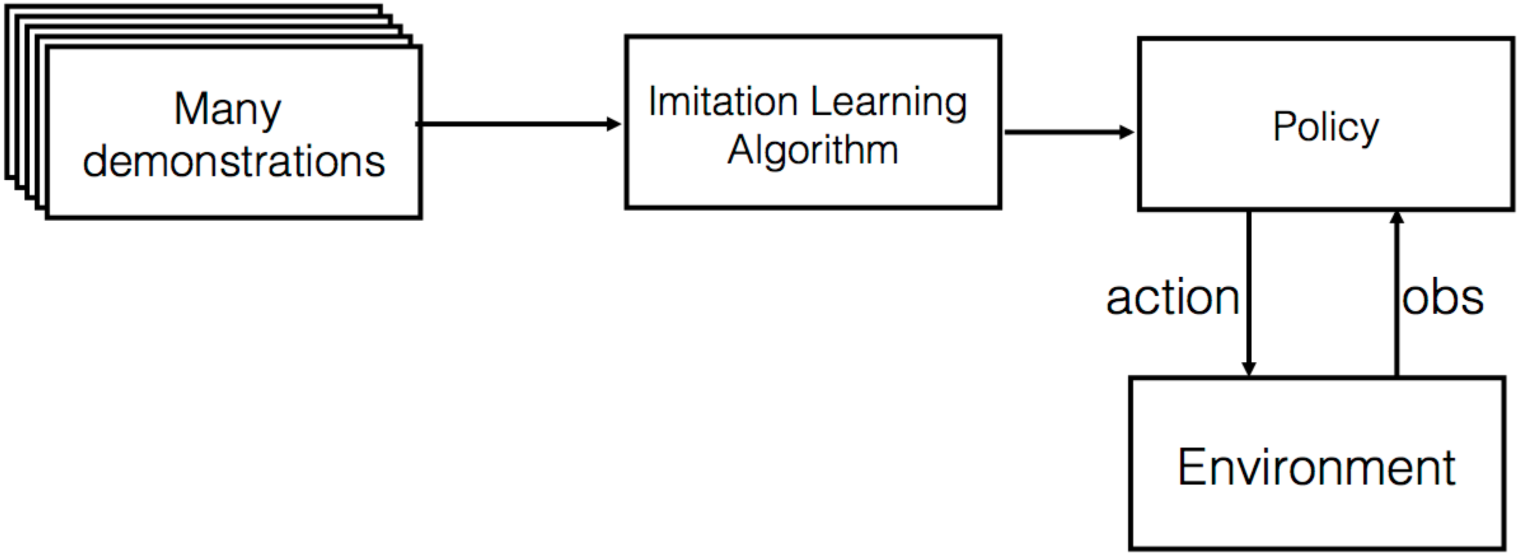


[Schulman et al. 2013]

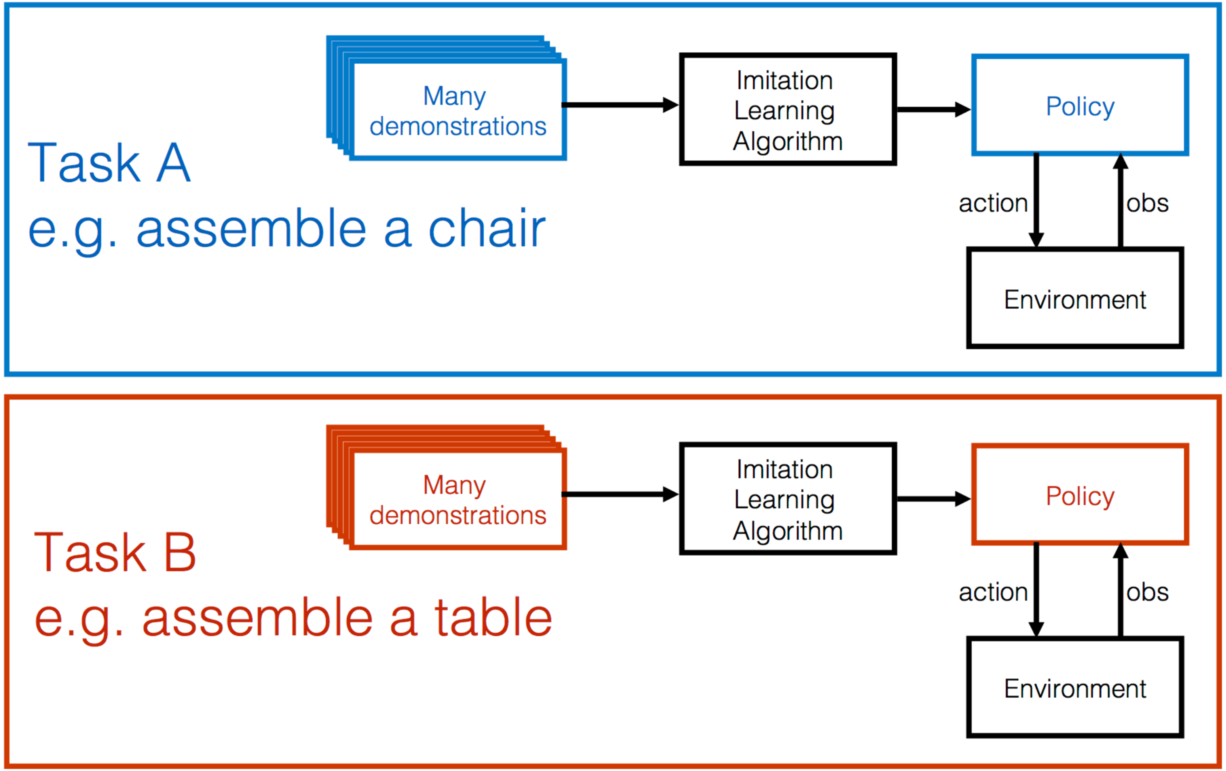


[Finn et al. 2016]

Imitation Learning

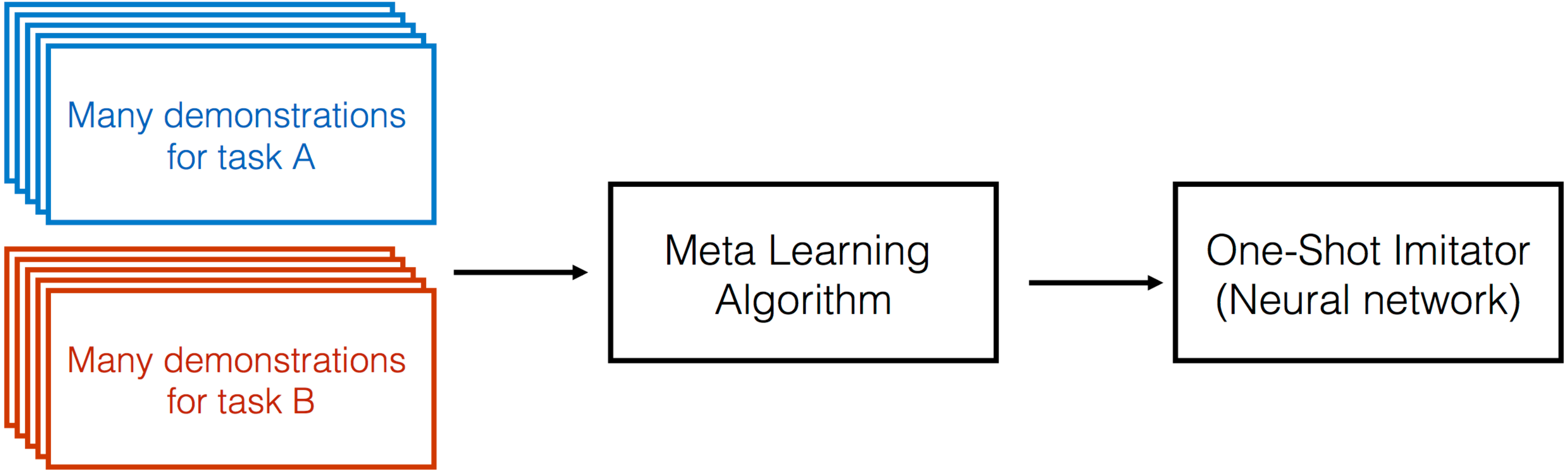


Imitation Learning

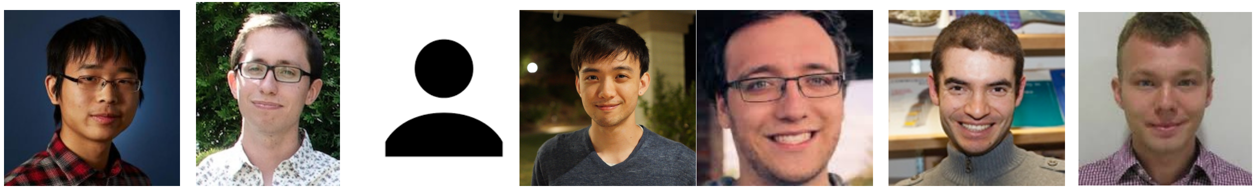


…

One-Shot Imitation Learning

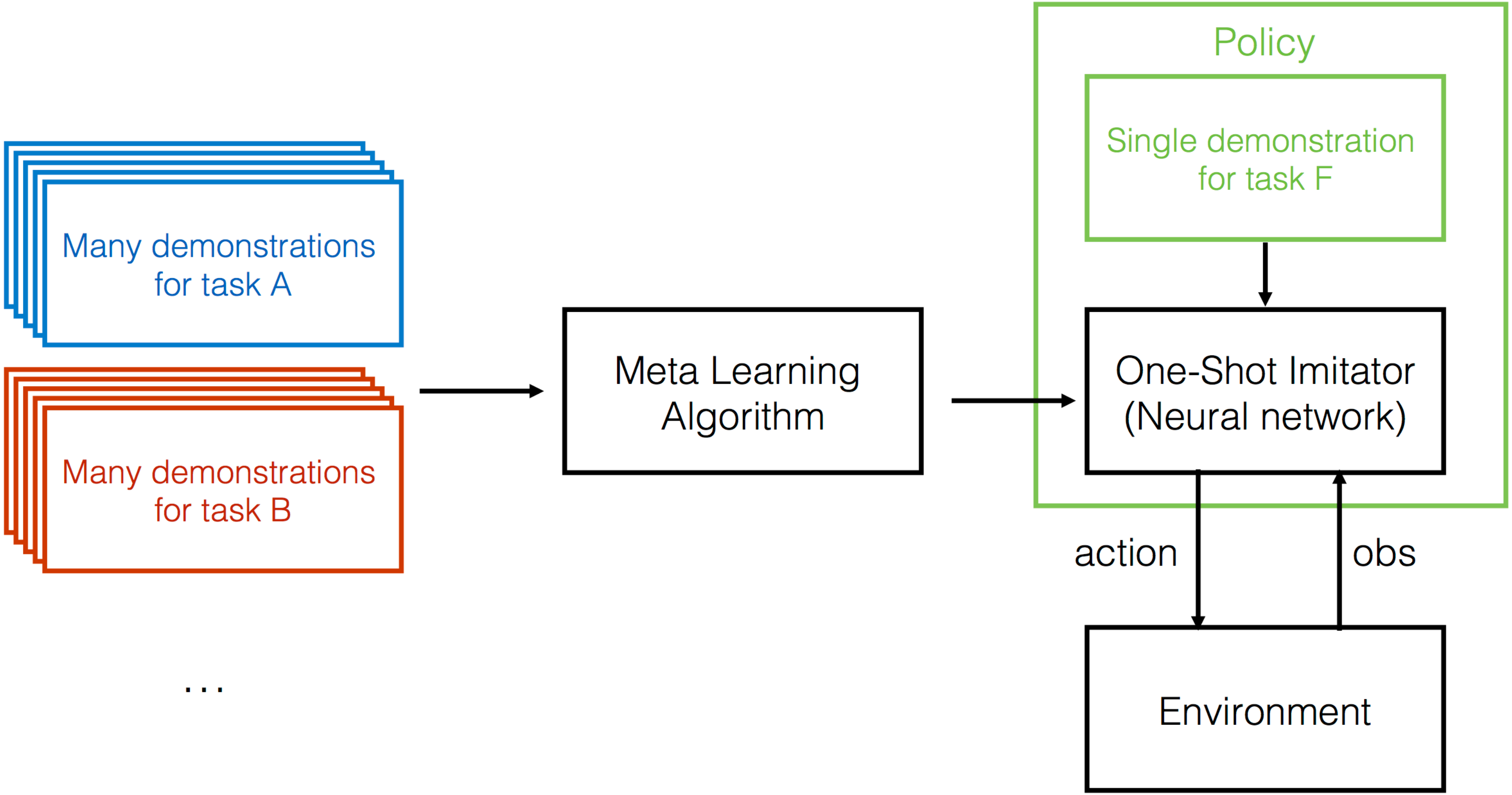


. . .



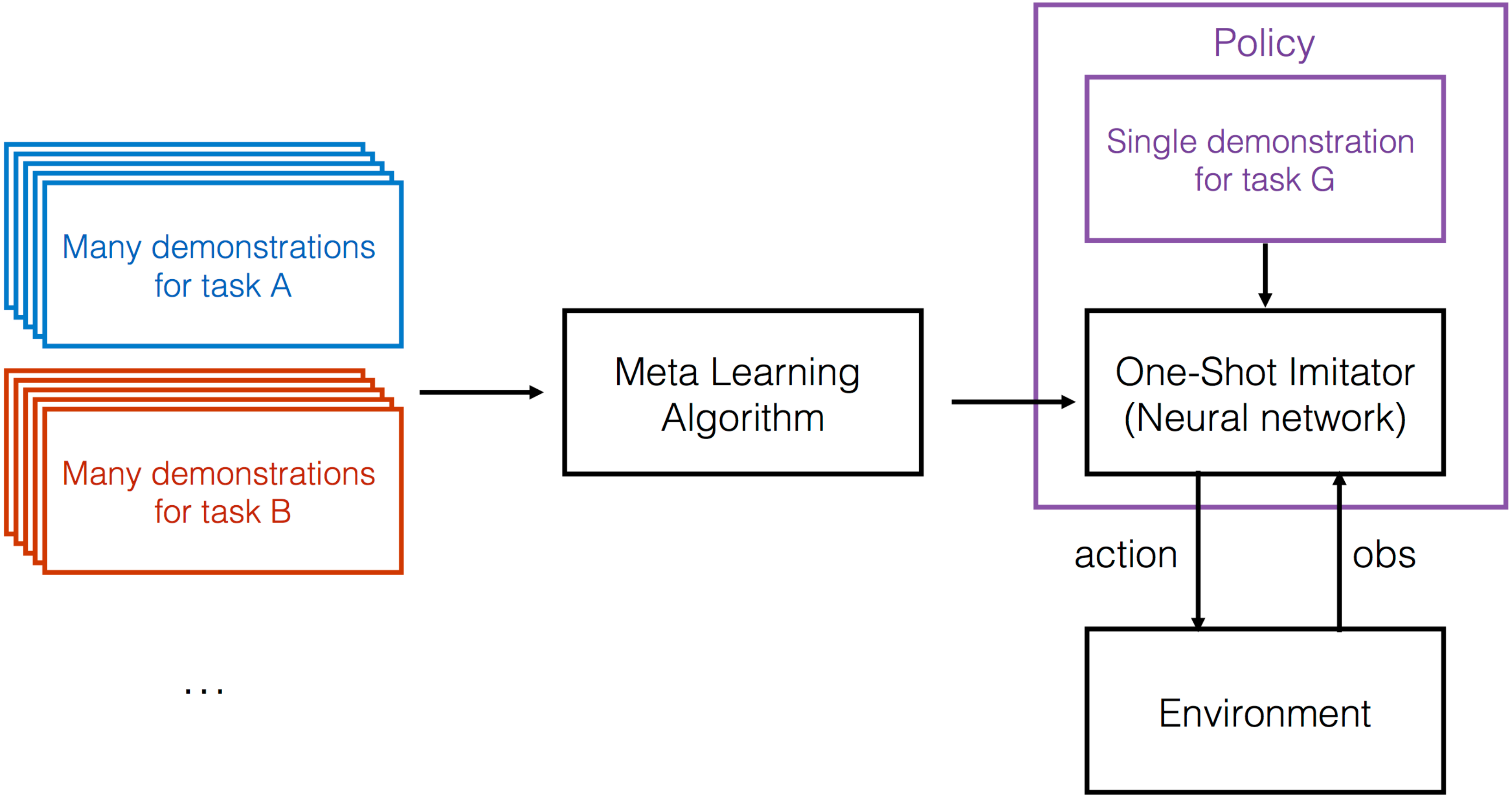
[Duan, Andrychowicz, Stadie, Ho, Schneider, Sutskever, Abbeel, Zaremba, 2017]

One-Shot Imitation Learning



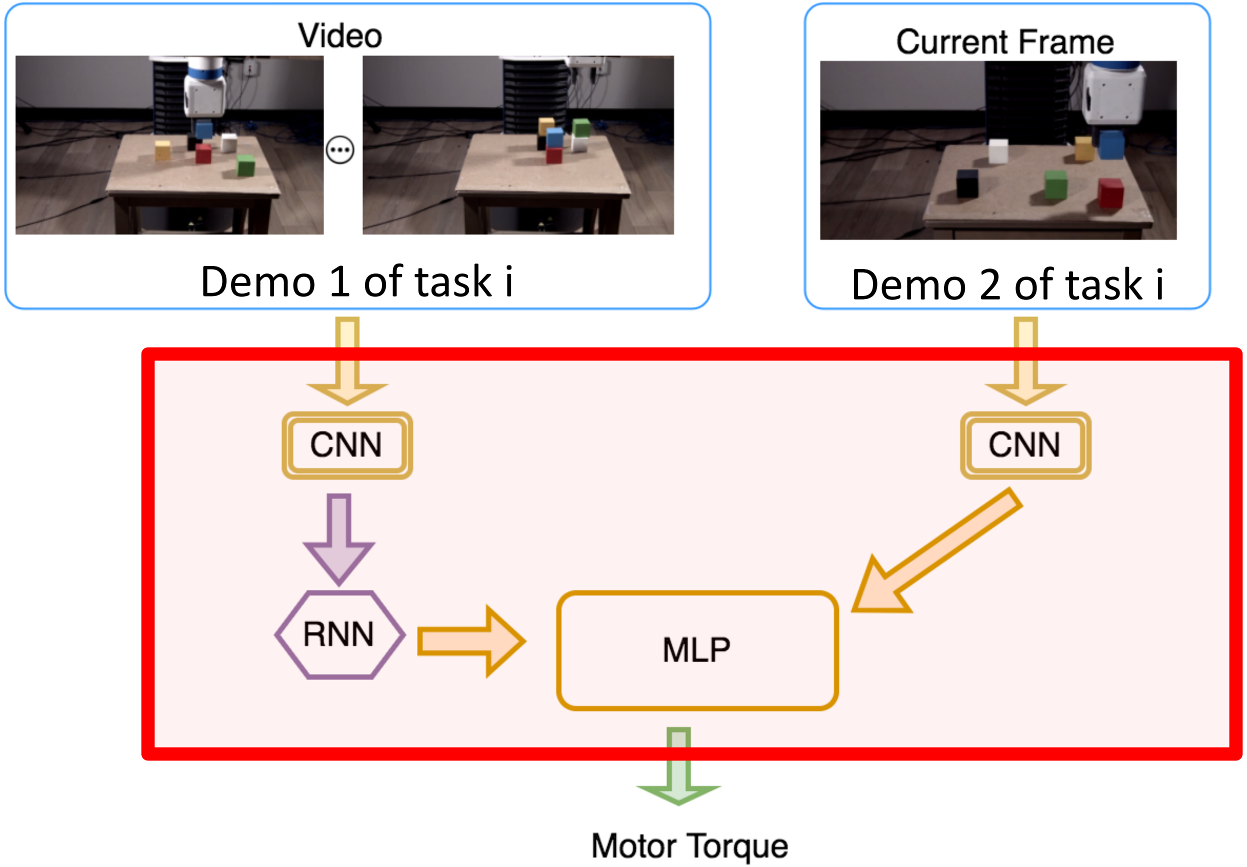
[Duan, Andrychowicz, Stadie, Ho, Schneider, Sutskever, Abbeel, Zaremba, 2017]

One-Shot Imitation Learning



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

Learning a One-Shot Imitator

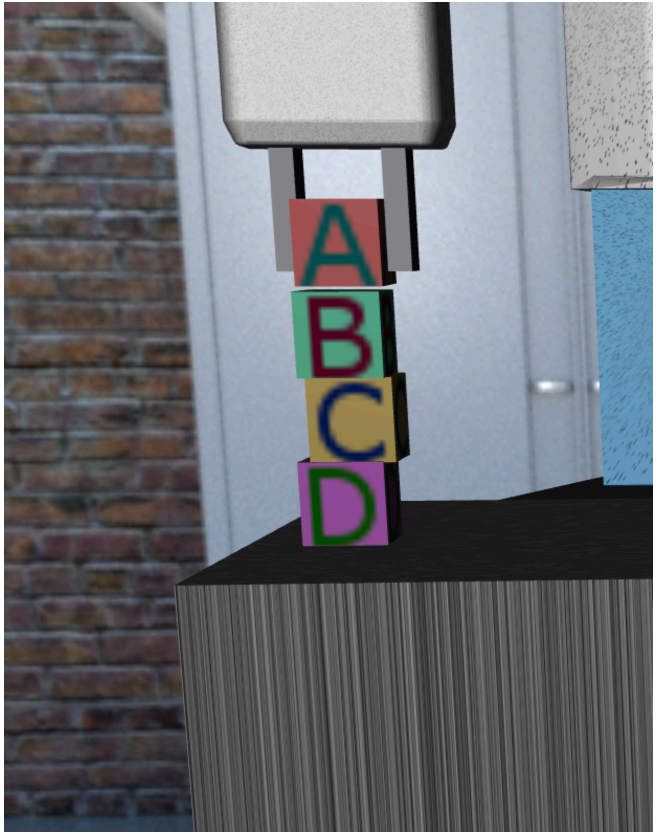


One-shot imitator

[Figure credit: Bradly Stadie]

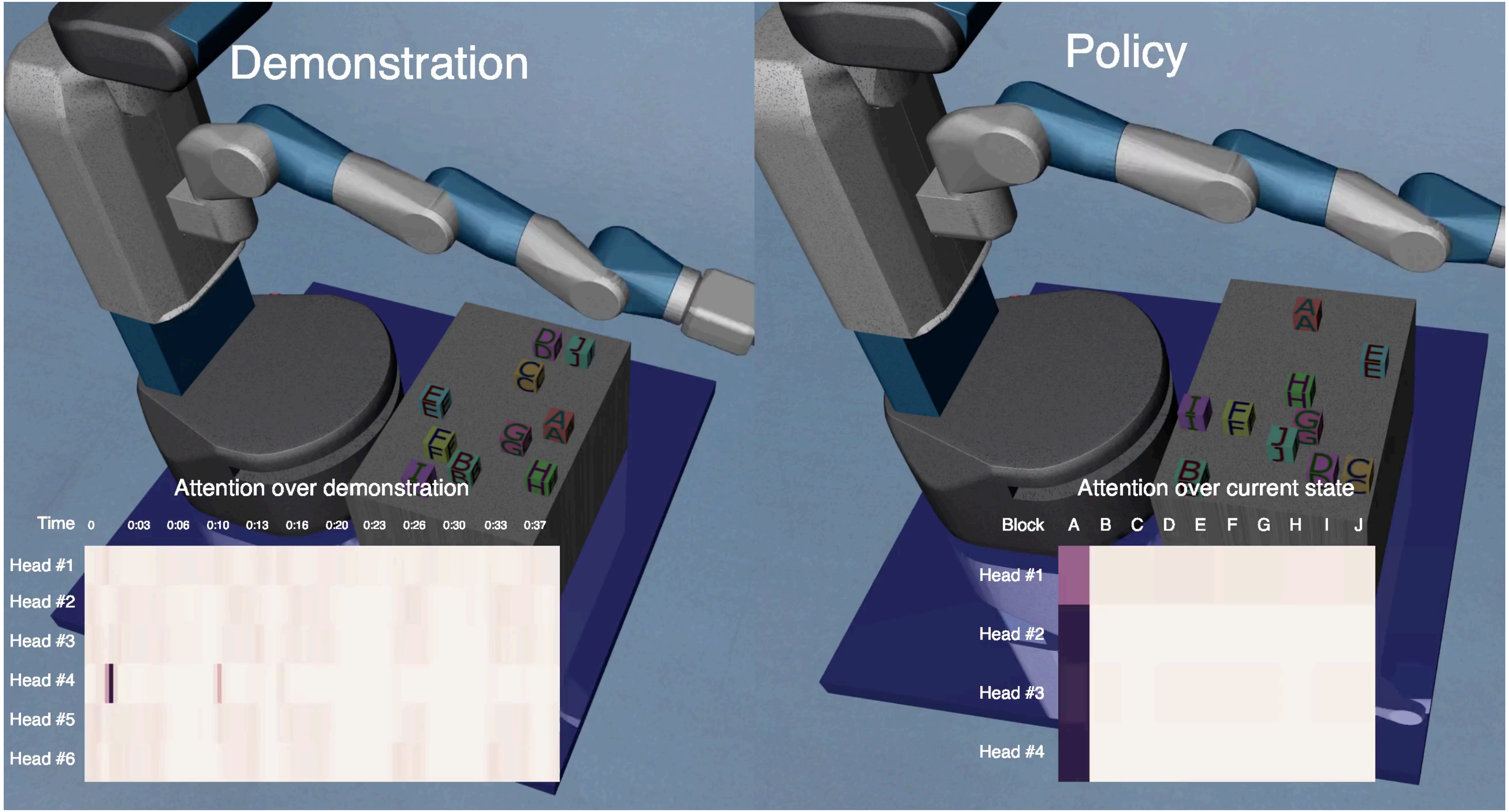
Proof-of-concept: Block Stacking

- 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, 2017]

Evaluation



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

Learning a One-Shot Imitator with MAML

- Meta-learning loss:

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

- 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, 2017]

Robot Experiments: Learning to Place

- Meta-training targets / objects

- Meta-testing targets / objects

1,300 demonstrations for meta-training

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

Robot Experiments: Learning to Place

1 demo

imitation

Succes rate: 90%

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

Robot Experiments: Learning to Place

1 demo

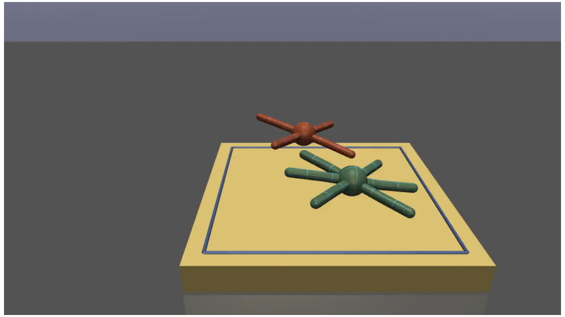
imitation

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

Succes rate: 90%

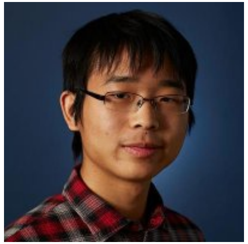
Current Directions

- Architectures for meta RL and imitation agents
  - Neural
  - Code
- Lifelong Learning
  - Non-stationary environments
  - Competition




Chelsea Finn [3,6]


Sergey Levine [3,6]


Yan Duan [1,5]


John Schulman [1,4,5]


Xi Chen [1,2,4]


Peter Bartlett [1]


Ilya Sutskever [1,5,7]


Marcin Andrychowicz [5,9]


Kevin Frans [4]


Jonathan Ho [4,5]


Jonas Schneider [5,8, 9]


Wojciech Zaremba [5,8,9]


Josh Tobin [8,9]


Rachel Fong [8,9]

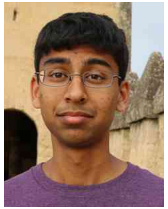

Alex Ray [8,9]


Bradly Stadie [5]


Peter Welinder [9]


Bob McGrew [9]


Filip Wolski [9]


Nikhil Mishra [3]


M. Rohaninejad [3]


Tianhe Yu [5]


Maruan Al-Shedivat [7]


Trapit Bansal [7]


Yura Burda [7]


Igor Mordatch [7]

[1] *RL2*, Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016
[2] *Simple Neural Attentive Meta-Learner*, Mishra*, Rohaninejad*, Chen, Abbeel, 2017
[3] *MAML*, Finn, Abbeel, Levine, 2017
[4] *Meta-Learning Shared Hierarchies*, Frans, Ho, Chen, Abbeel, Schulman, 2017
[5] *One-Shot Imitation*, Duan, Andrychowicz, Stadie, Ho, et al, 2017
[6] *One-Shot Visual Imitation Learning*, Finn*, Yu*, Zhang, Abbeel, Levine, 2017
[7] *Continuous Adaptation*, Al-Shedivat, Bansal, Burda, Sutskever, Mordatch, Abbeel, 2017
[8] *Domain Randomization for Transferring Deep Neural Nets from Sim to Real World*, Tobin, Fong, Ray, Schneider, Zaremba, Abbeel, 2017
[9] *Hindsight Experience Replay*, Andrychowicz, Wolski, Ray, Schneider, Fong, Welinder, McGrew, Tobin, Abbeel, Zaremba, 2017