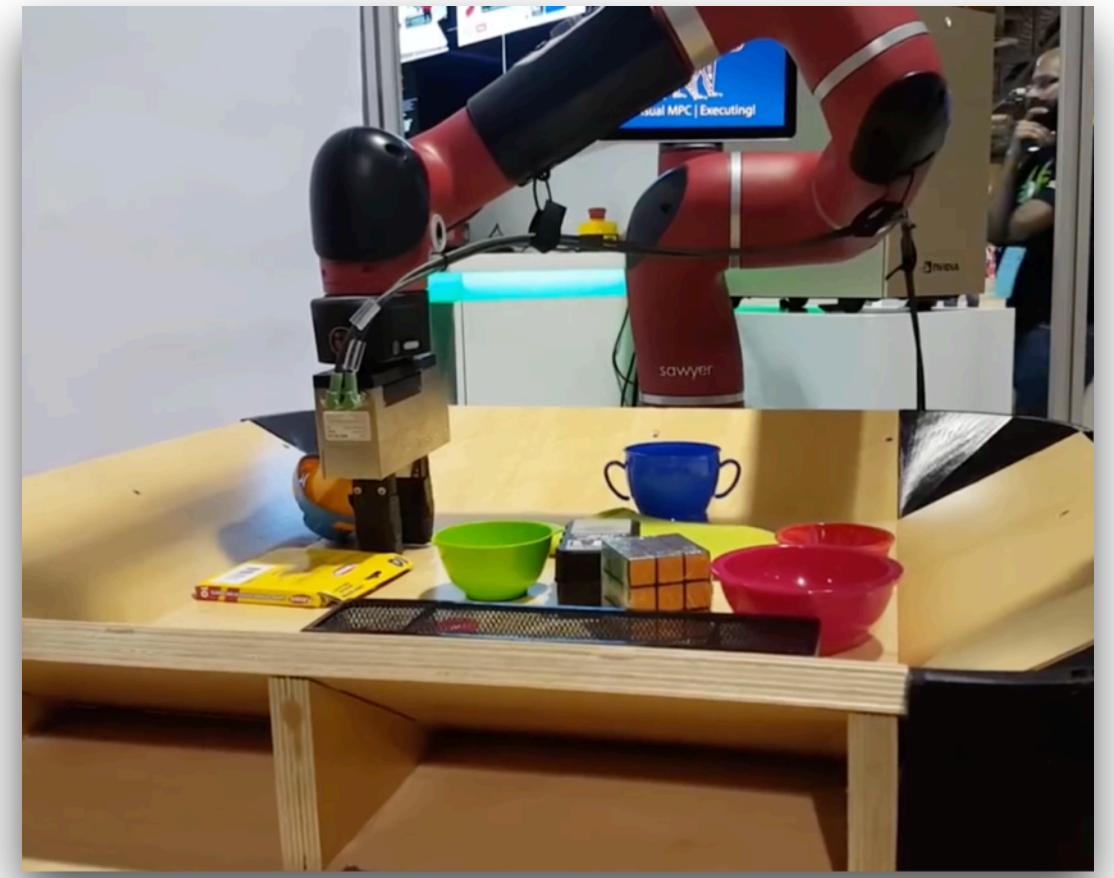


Learning to Adapt to Dynamic, Real-World Environments

Chelsea Finn

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH
UC Berkeley

Google Brain

Stanford

Savva et al. '19

Photorealistic simulators

Sadeghi et al. RSS '17

Randomization

The real world is unmatched.

Unmatched in terms of:	*diversity*	rich, *multi-agent* interactions
	fidelity	*messiness*

Real world will always require ***some amount of adaptation.***

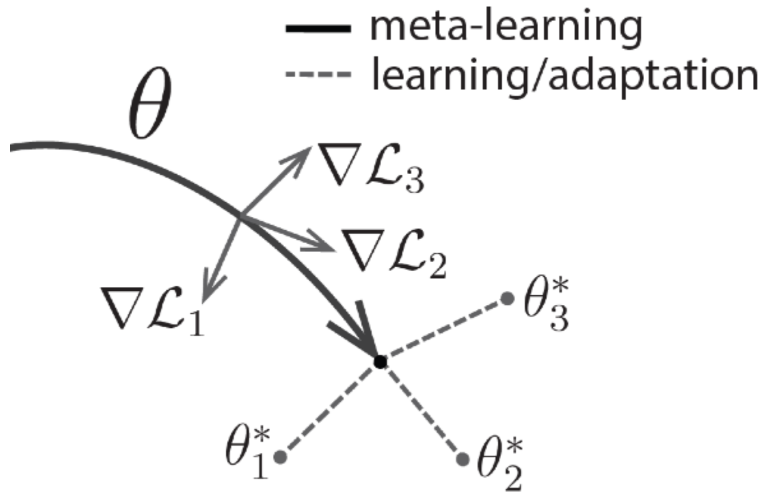
Can robots learn something from *simulation* that can help them adapt *quickly*?

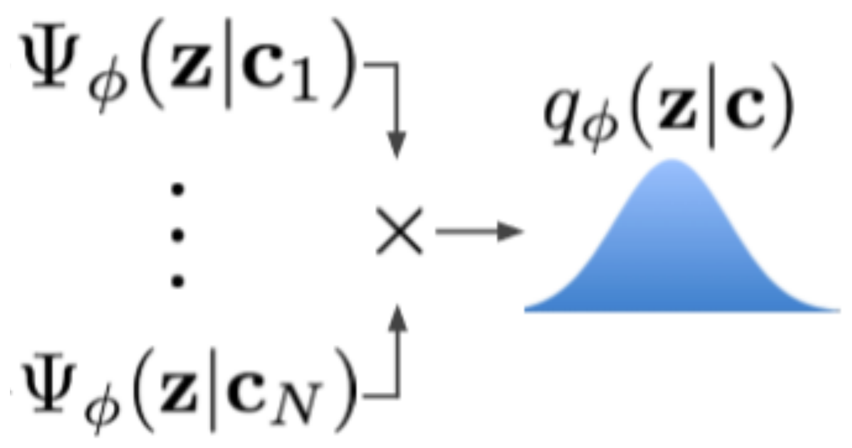
Can robots learn something from *simulation* that can help them adapt *quickly*?

from *other data*

from *past experience*

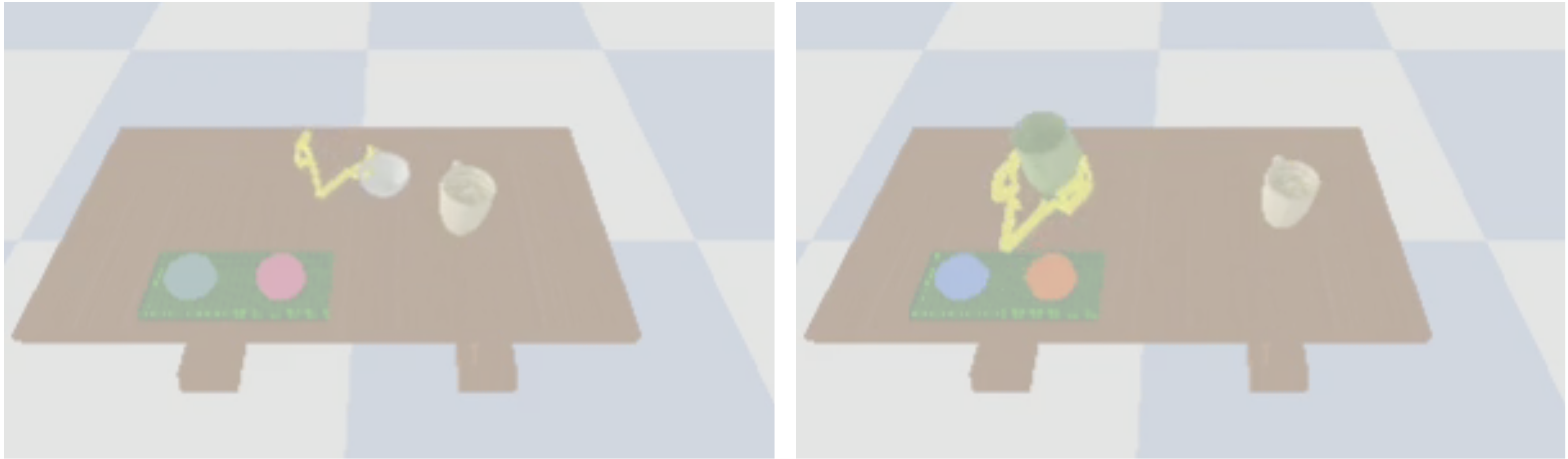
Adaptability is important, regardless of whether you are using simulation.



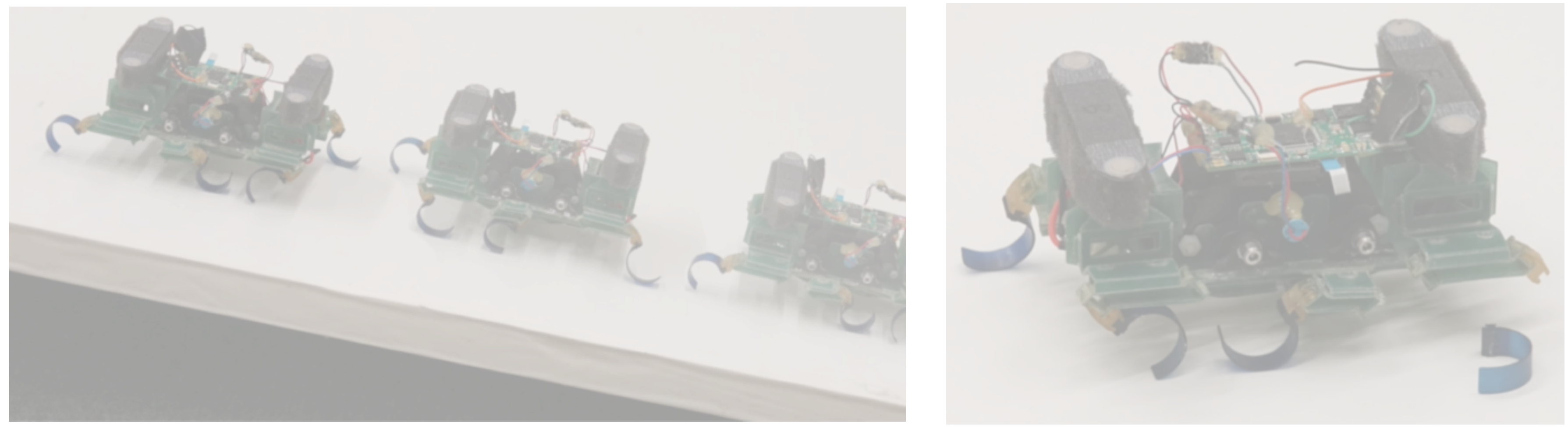


Quick primer on **few-shot meta-learning**

Challenges in applications to robotics:



Meta-learning across **families of manipulation tasks**

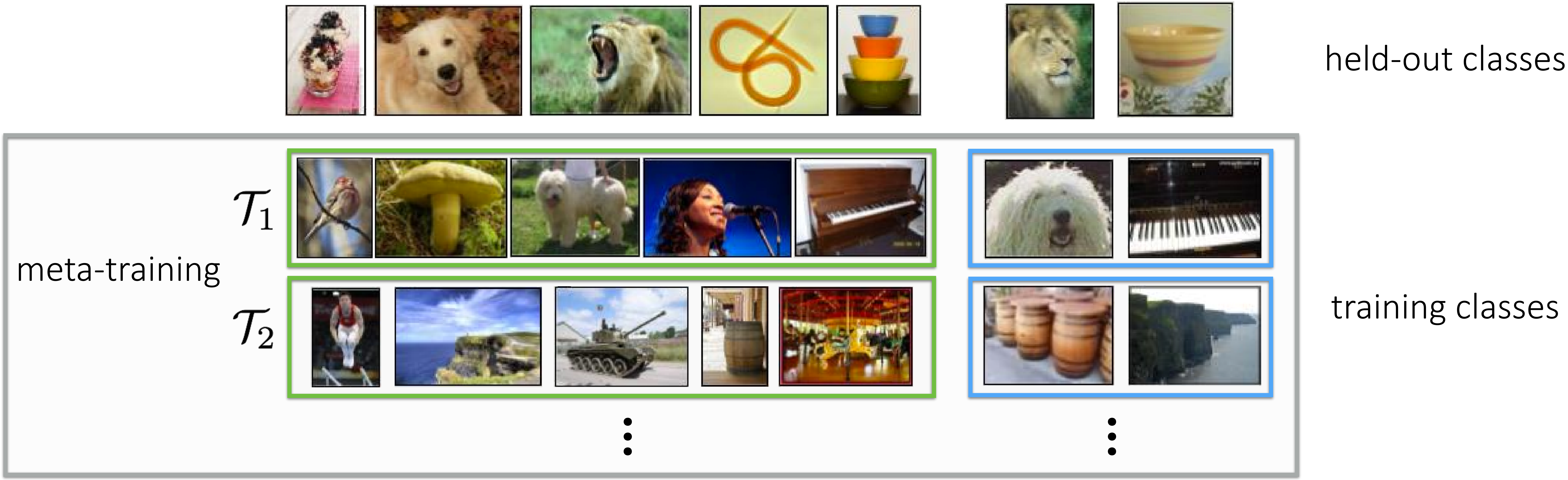


Rapid, ***online*** adaptation to **drastic changes in dynamics**

Example: Few-Shot Image Classification

5-way, **1**-shot image classification (MiniImagenet)

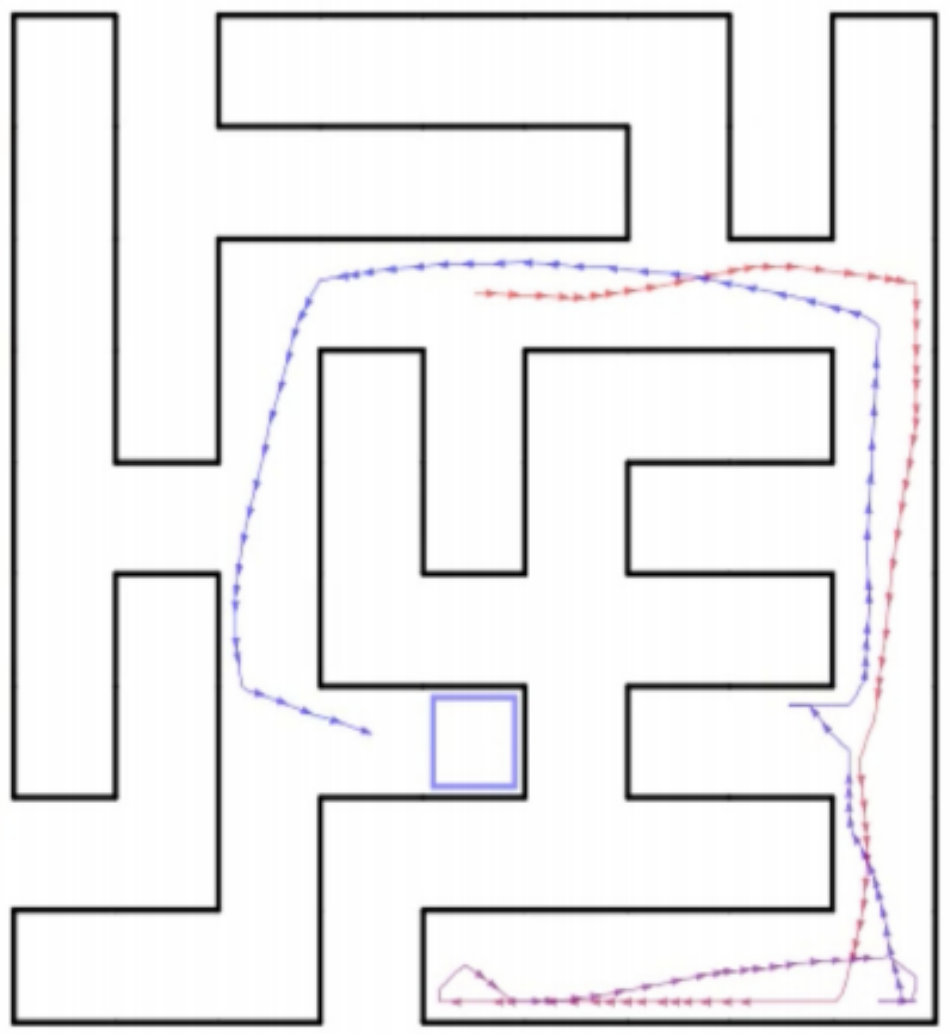
Given 1 example of 5 classes: Classify new examples



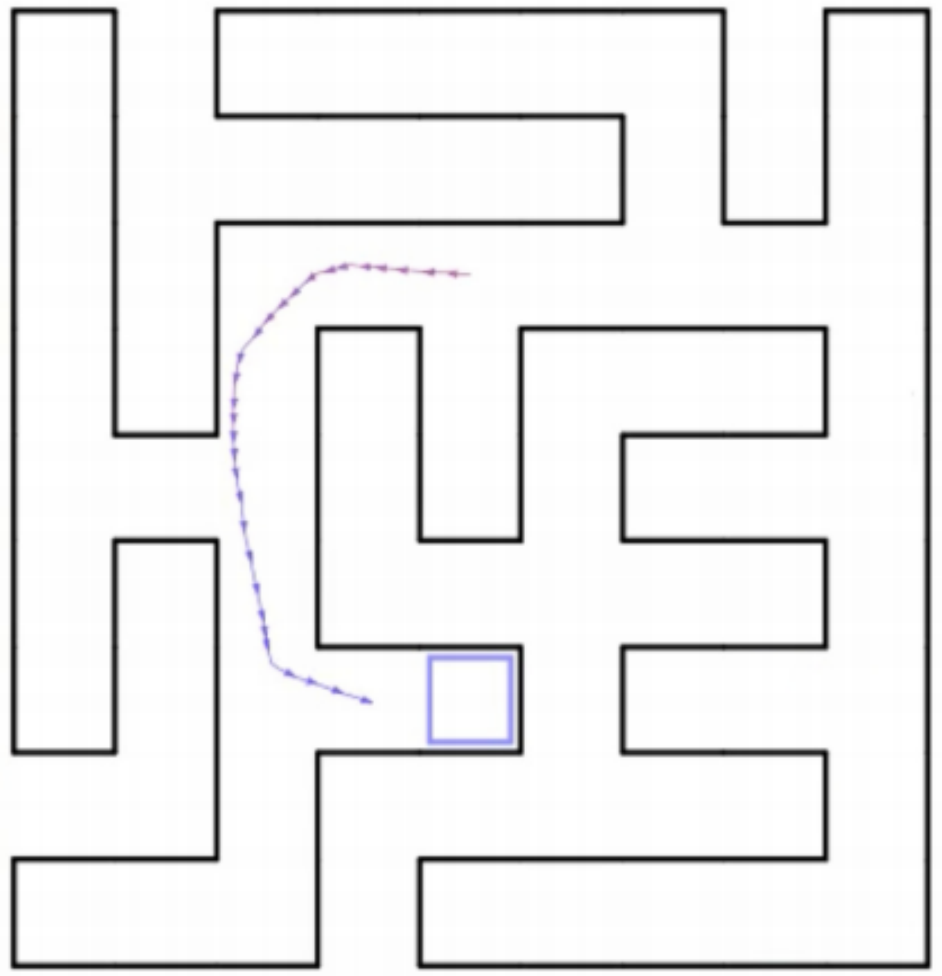
Can replace image classification with: **regression**, **reinforcement learning**, **any ML problem**

Example: Fast Reinforcement Learning

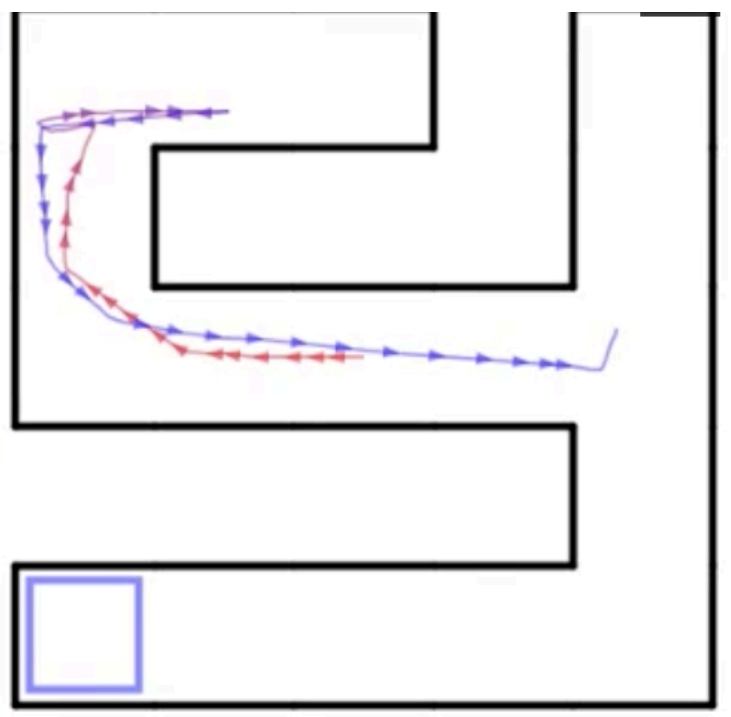
Given a small amount of experience

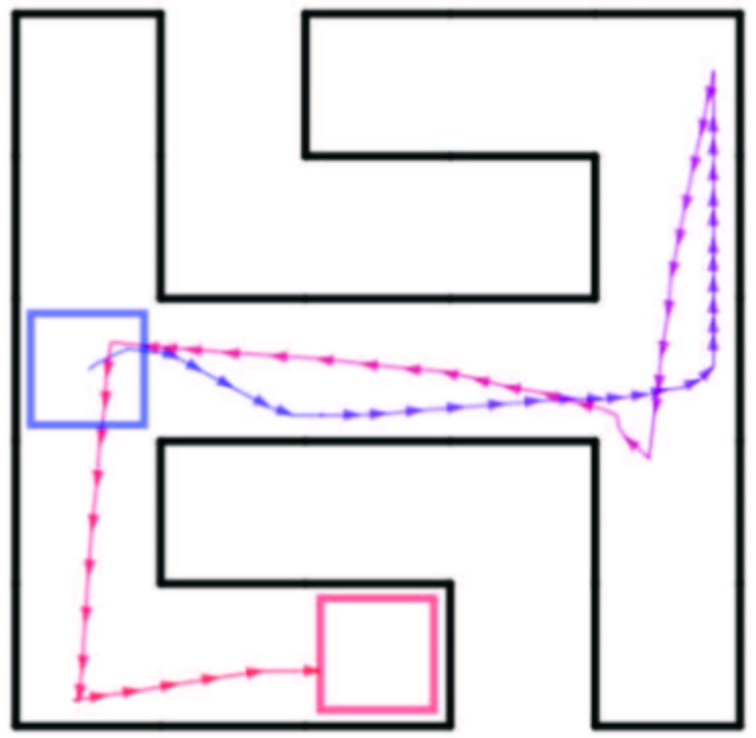


Learn to solve a task



By learning how to learn many other tasks:





...

diagram adapted from Duan et al. '17

The Meta-Learning Problem: The Mechanistic View

Supervised Learning:

Inputs: $\mathbf{x}$ Outputs: $\mathbf{y}$ Data: $\{(\mathbf{x}, \mathbf{y})_i\}$

$$\mathbf{y} = f(\mathbf{x}; \theta)$$

Meta-Supervised Learning:

Inputs: $\underbrace{\mathcal{D}_{\text{train}}}_{\{(\mathbf{x}, \mathbf{y})_{1:K}\}}$ $\mathbf{x}_{\text{test}}$ Outputs: $\mathbf{y}_{\text{test}}$ Data: $\{\mathcal{D}_i\}$

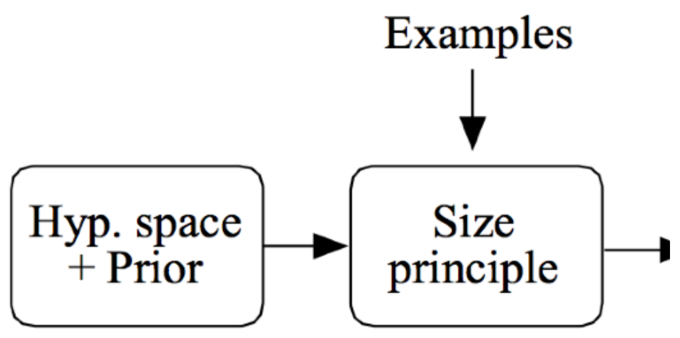
$$\mathbf{y}_{\text{test}} = f(\mathcal{D}_{\text{train}}, \mathbf{x}_{\text{test}}; \theta)$$

$\mathcal{D}_i : \{(\mathbf{x}, \mathbf{y})_j\}$

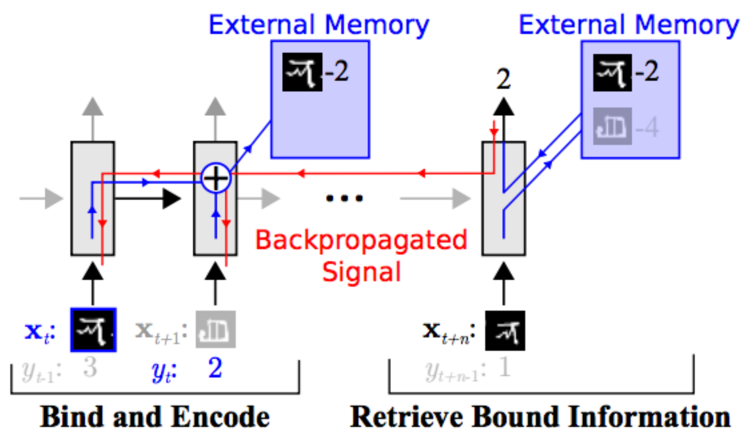
Why is this view useful?

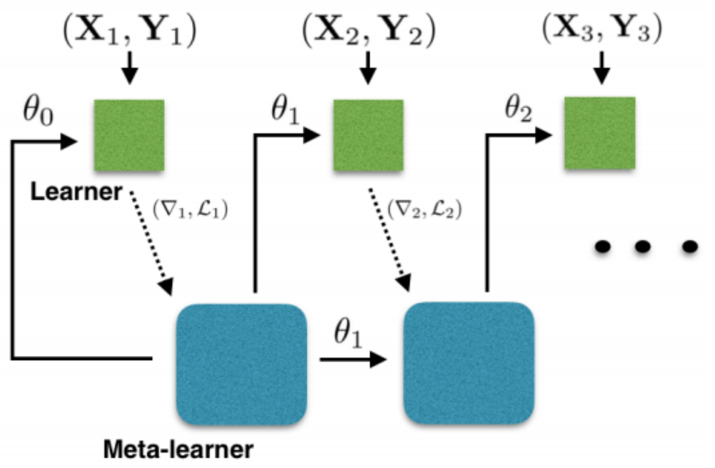
Reduces the problem to the design & optimization of *f*.

Meta-Learning for Few-Shot Learning

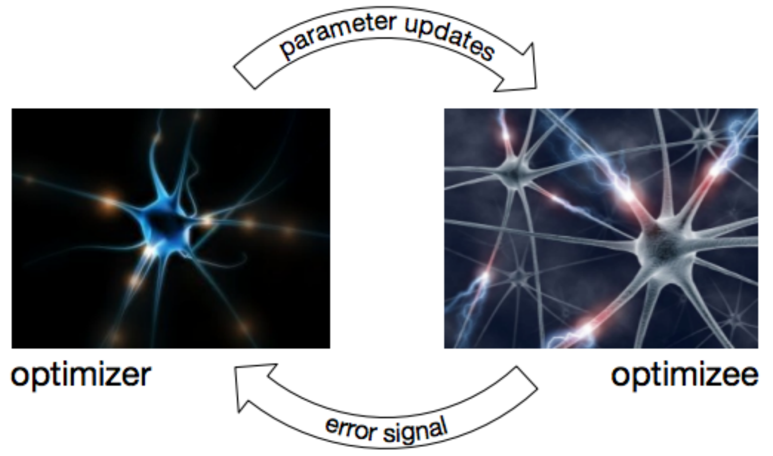


Tenenbaum '99
Fei-Fei et al. '05
Lake et al. '11

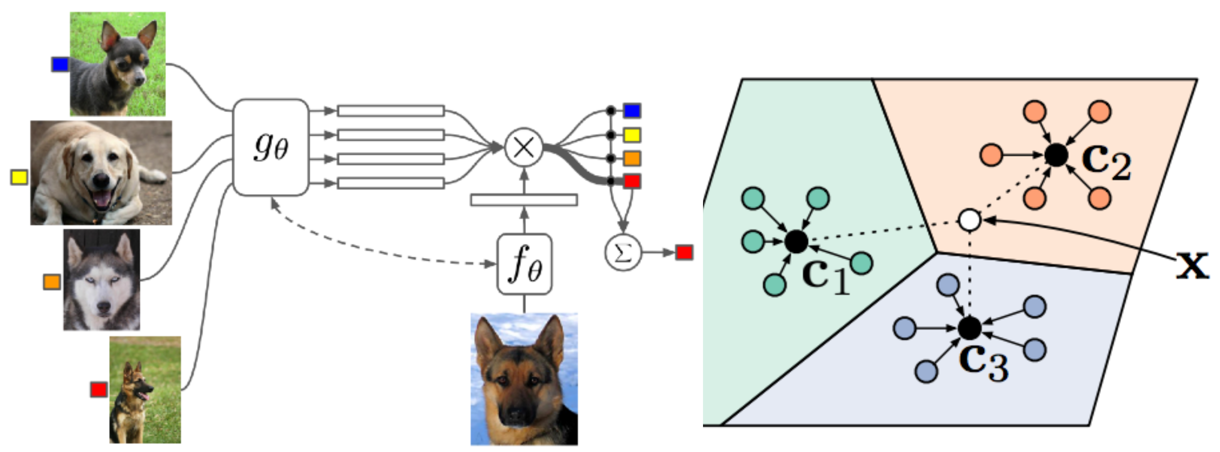




Santoro et al. '16 Ravi & Larochelle '17



Hochreiter et al. '01
Andrychowicz et al. '16
Li & Malik '16



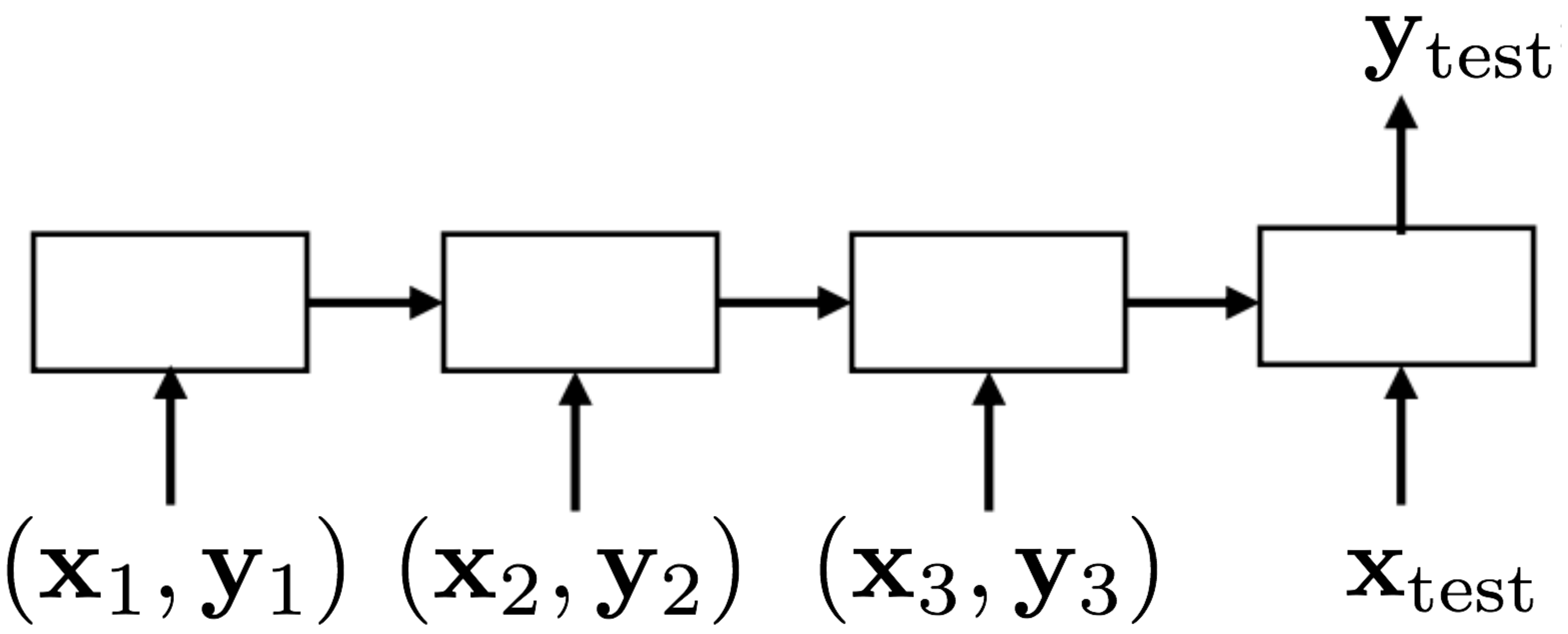
Vinyals et al. '16 Snell et al. '17

and many *many* more approaches

Recurrent network
(LSTM, NTM, Conv)

$$\mathbf{y}^{\text{test}} = f(\mathcal{D}^{\text{train}}, \mathbf{x}^{\text{test}}; \theta)$$

Santoro et al. '16, Duan et al. '17, Wang et al. '17, Munkhdalai & Yu '17, Mishra et al. '17, …



\+ expressive, general
\+ applicable to range of problems

\- complex model for complex task of learning
\- often large data requirements for meta-training

Model-Agnostic Meta-Learning

pretrained parameters

Fine-tuning $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\text{train}}(\theta)$

[test-time]

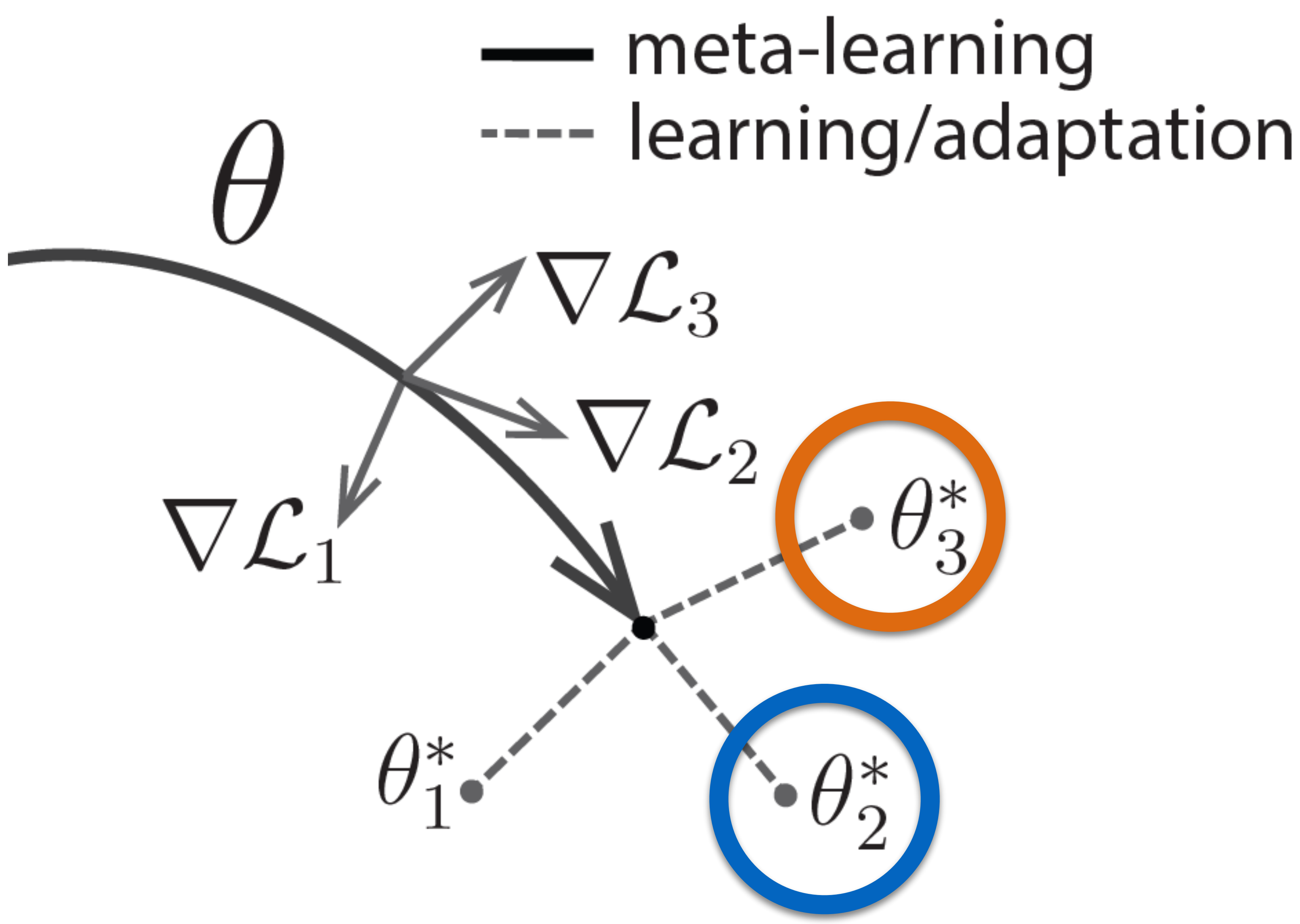
training data for new task

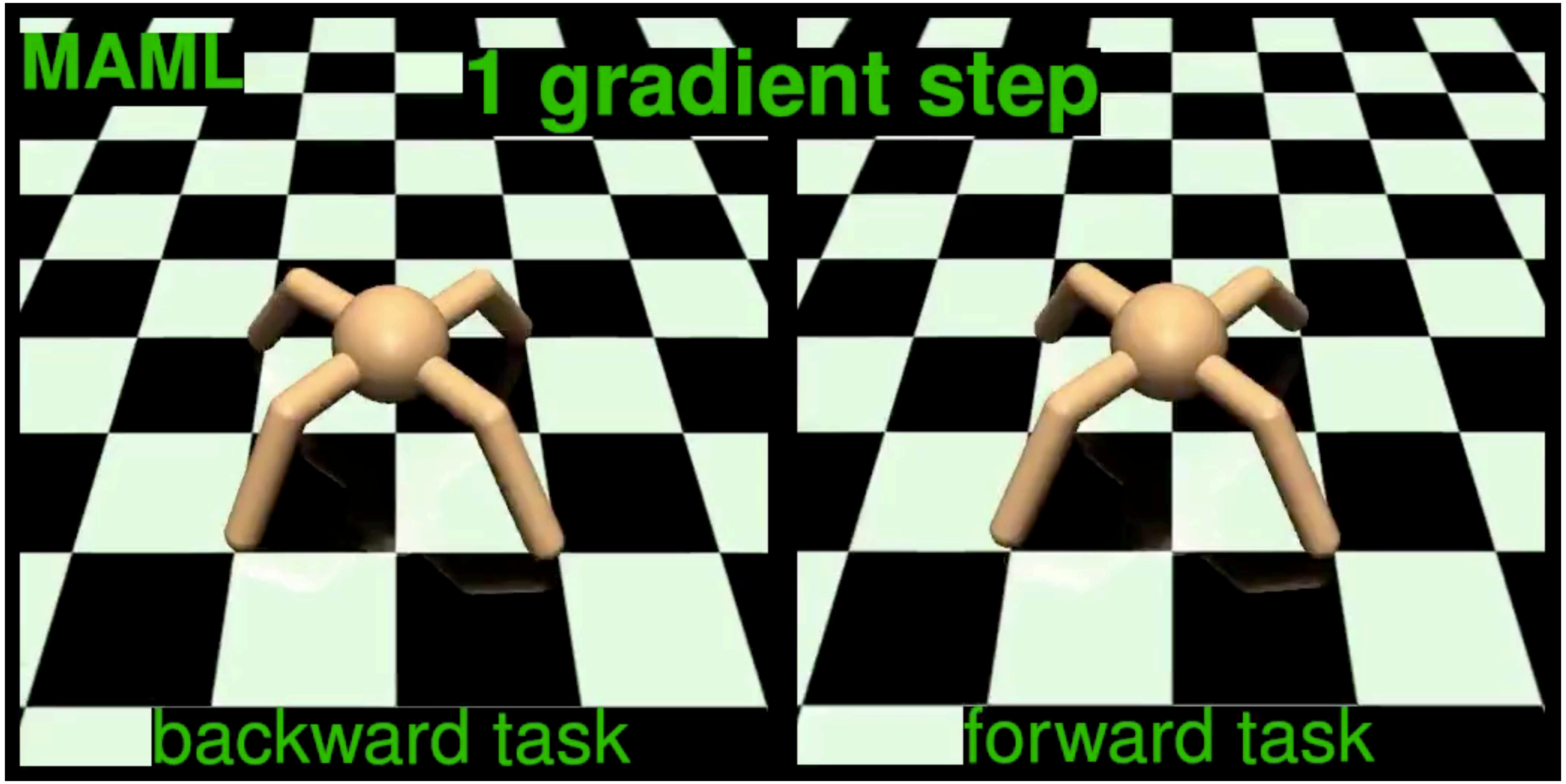
Our method $\min_\theta \sum_{\text{task } i} \mathcal{L}^i_{\text{test}}(\theta - \alpha \nabla_\theta \mathcal{L}^i_{\text{train}}(\theta))$

Key idea: Train over many tasks, to learn parameter vector θ that transfers

Finn, Abbeel, Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. ICML '17

Can we learn a representation under which RL is fast and efficient?

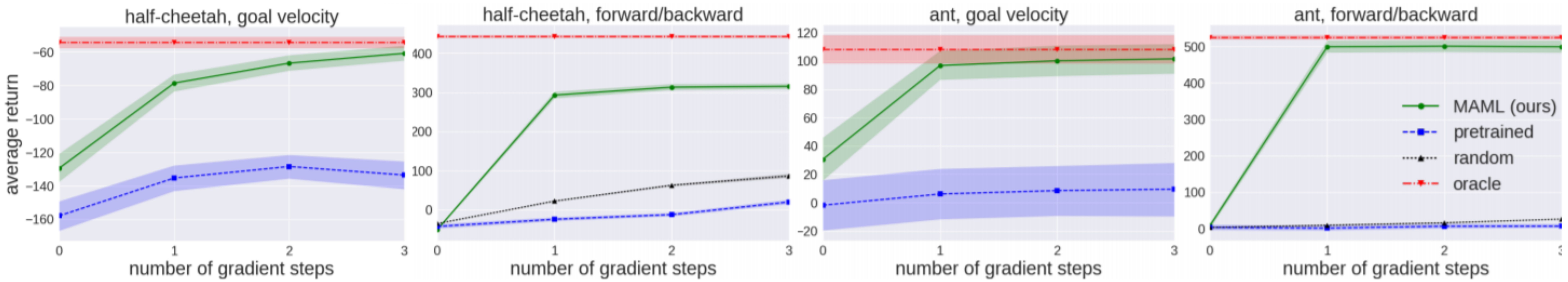




two tasks: running backward, running forward

Finn, Abbeel, Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. ICML '17

The Efficiency Challenge with Meta-RL



Finn et al., Model-Agnostic Meta-Learning. ‘17

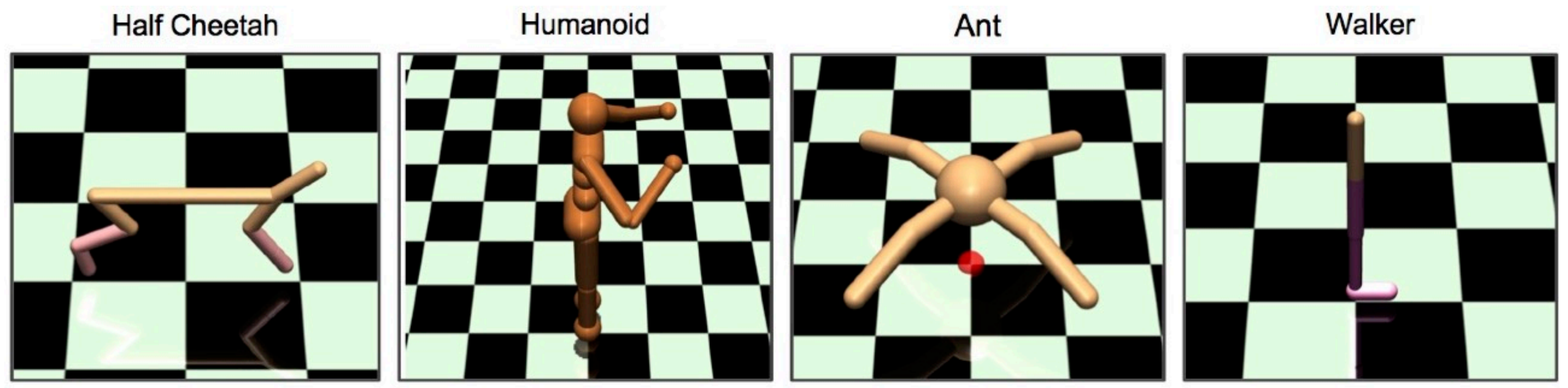
excellent “meta-test-time” learning efficiency

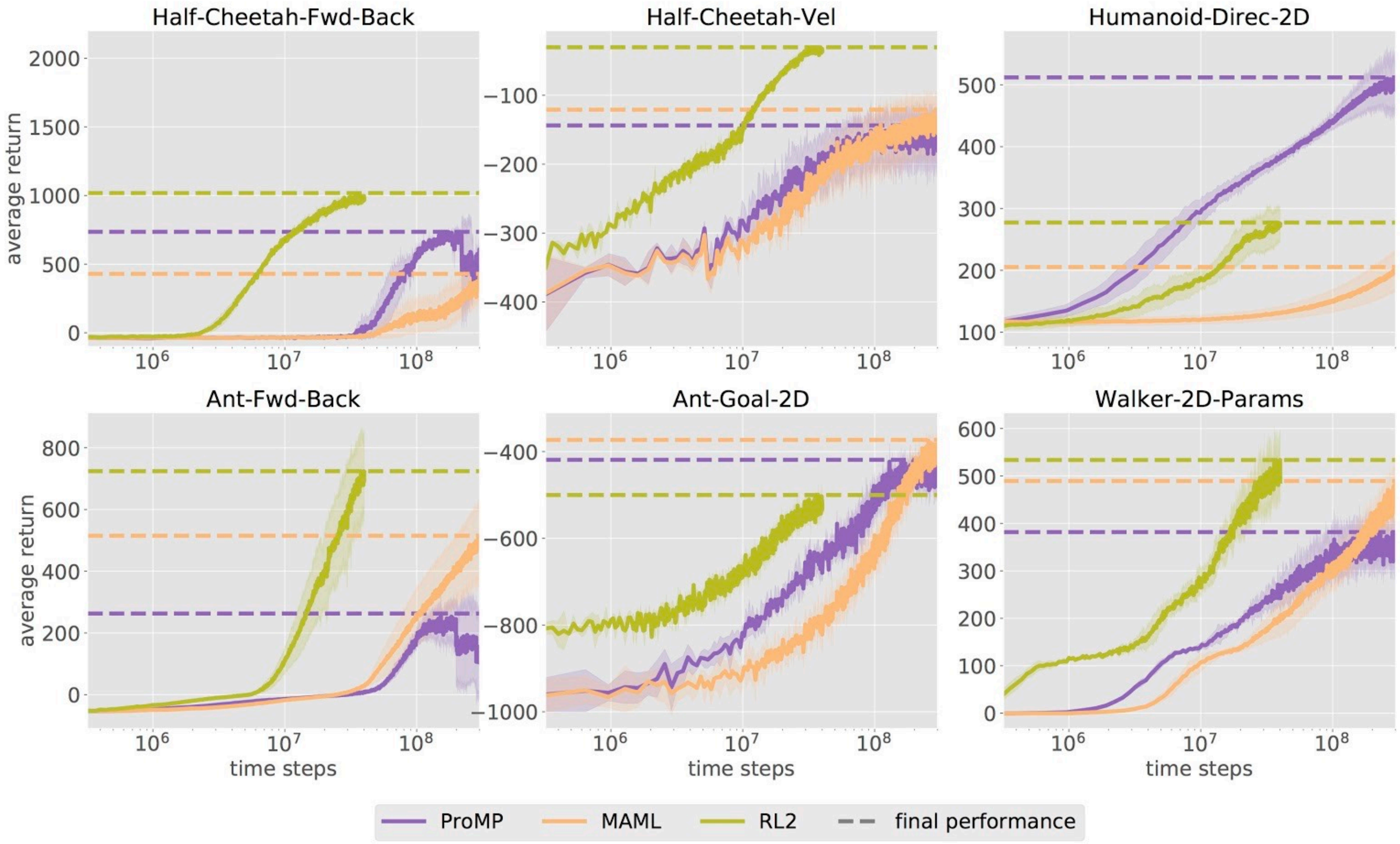
but how long did it take to **meta-train**?

100s of millions of steps

(about one month if it was in real time…)

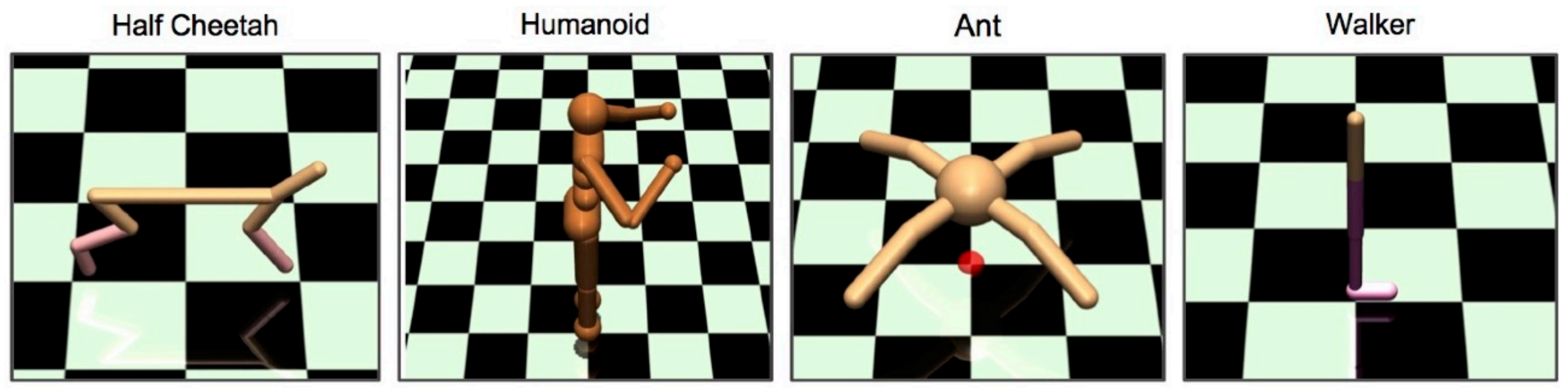
PEARL: Sample-Efficient Meta-RL

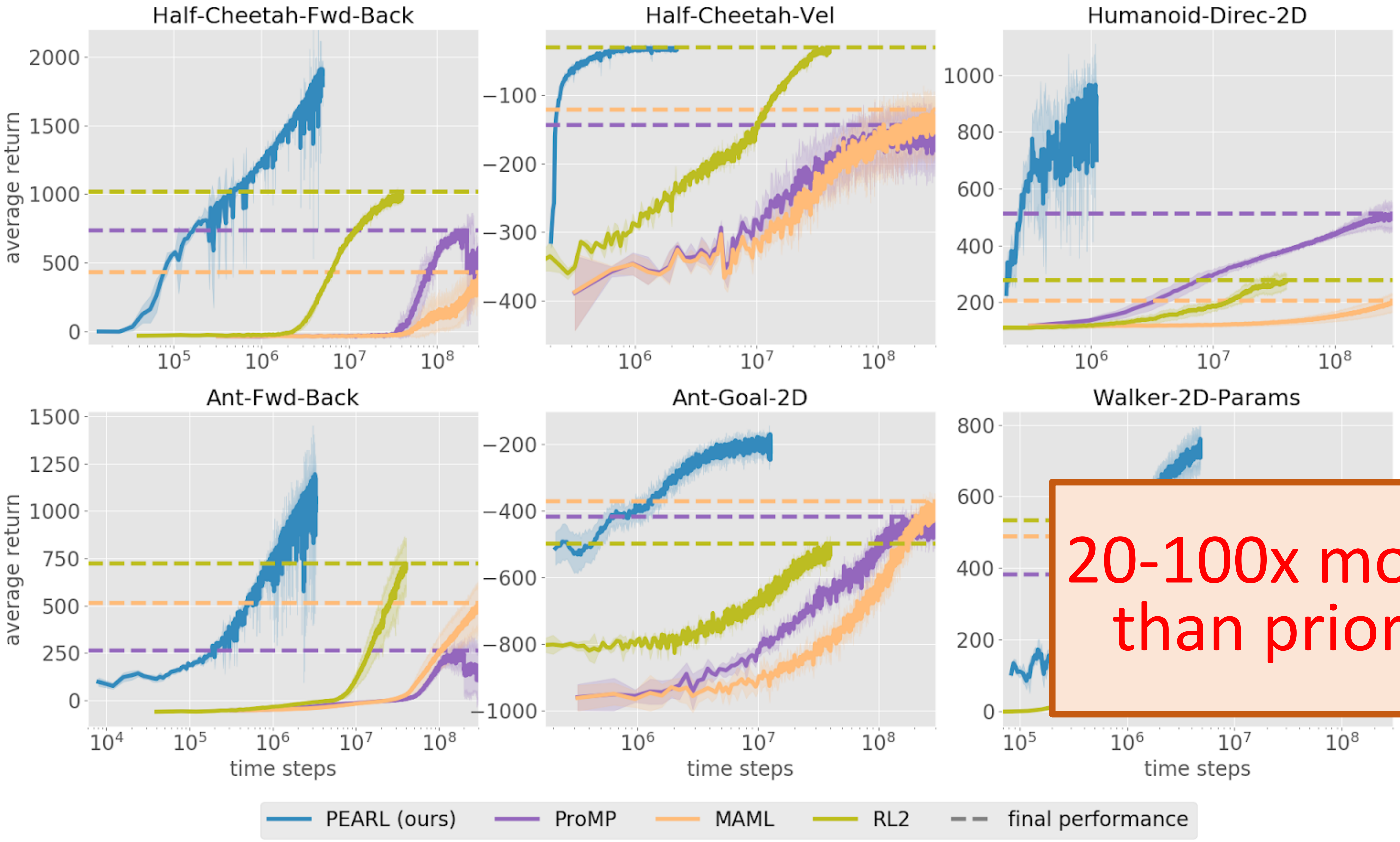




Rakelly*, Zhou*, Quillen, Finn, Levine. Efficient Off-Policy Meta-Reinforcement learning via Probabilistic Context Variables

PEARL: Sample-Efficient Meta-RL





20-100x more efficient than prior methods

Rakelly*, Zhou*, Quillen, Finn, Levine. Efficient Off-Policy Meta-Reinforcement learning via Probabilistic Context Variables

How does it work?

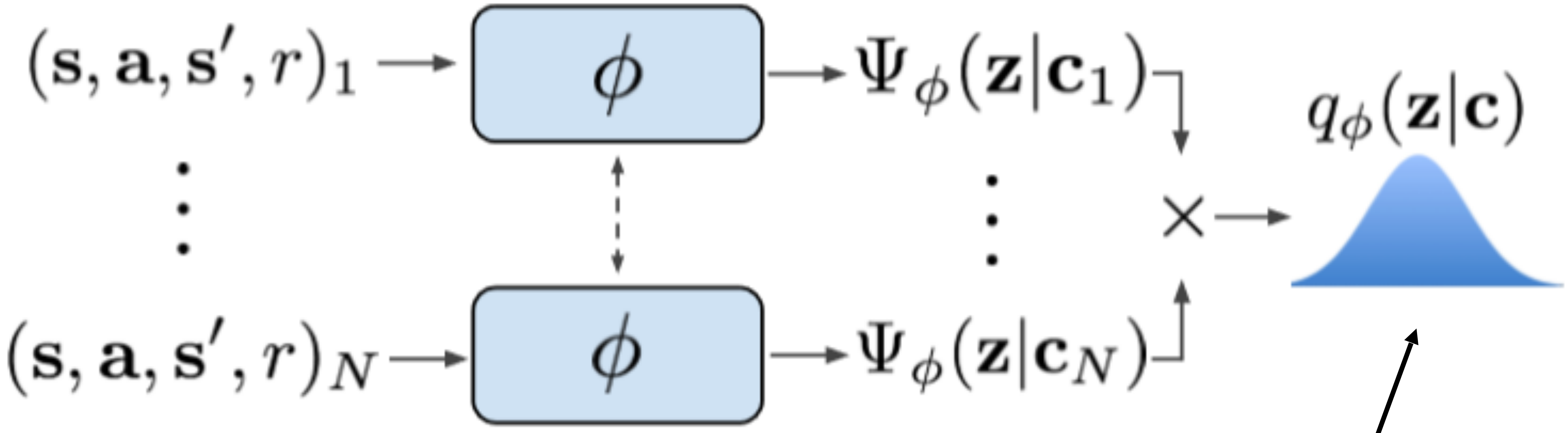
Idea 1: use stochastic latent context to represent task-relevant knowledge

$\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$

encapsulates information policy
needs to solve current task

learning a task = inferring $\mathbf{z}$

from *context* $(\mathbf{s}_1, \mathbf{a}_1, \mathbf{s}_2, r_1), (\mathbf{s}_2, \mathbf{a}_2, \mathbf{s}_3, r_2), \ldots$



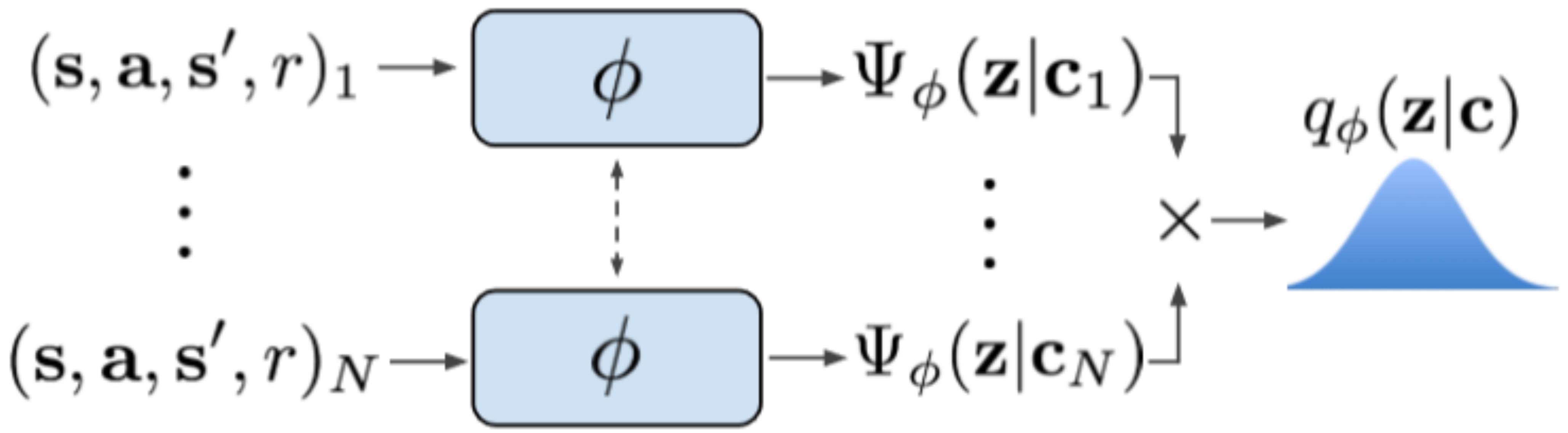
models our uncertainty about
how the task should be solved

(turns out to be crucial for exploration)

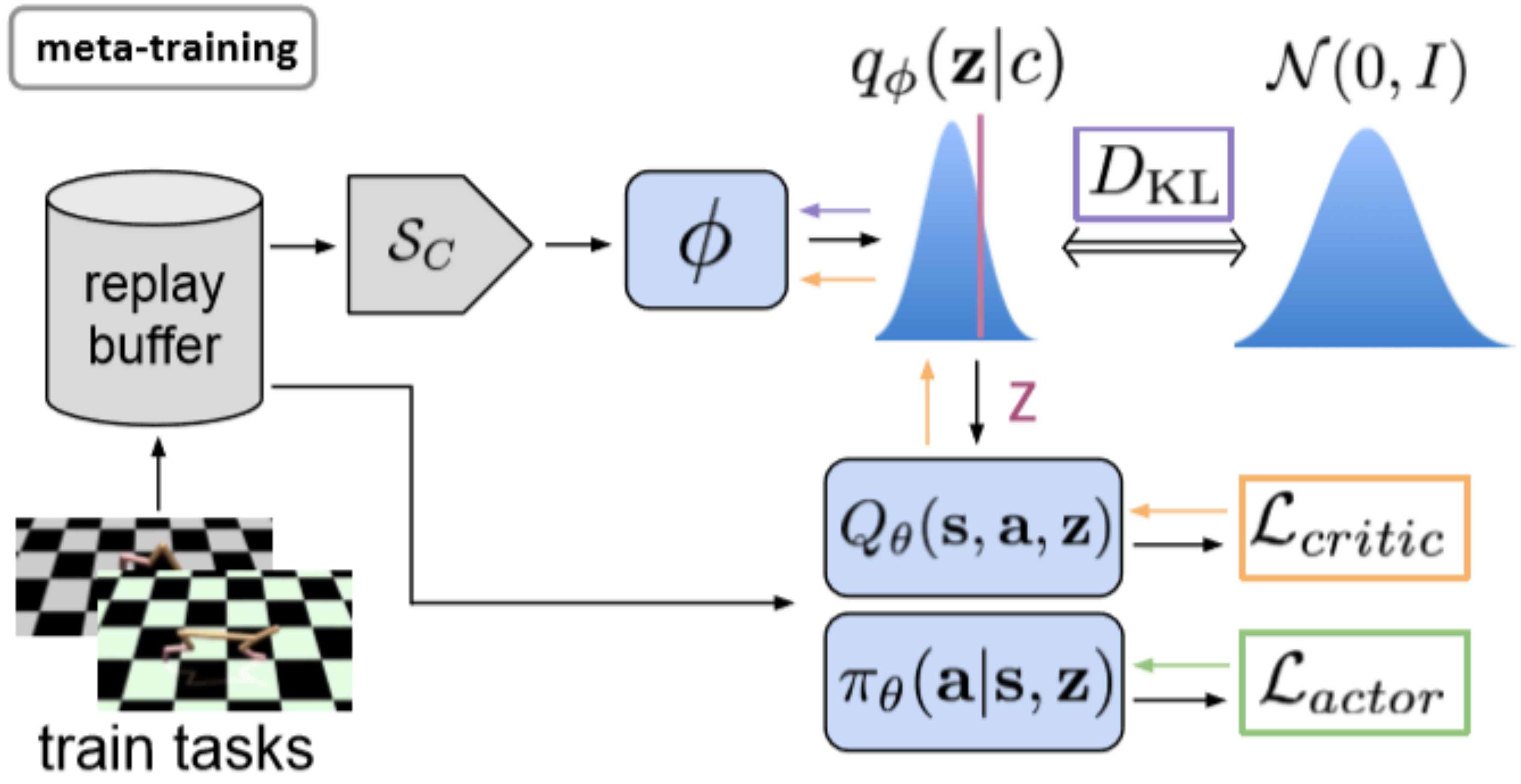
Rakelly*, Zhou*, Quillen, Finn, Levine. Efficient Off-Policy Meta-Reinforcement learning via Probabilistic Context Variables

How does it work?

Idea 1: use stochastic latent context to represent task-relevant knowledge



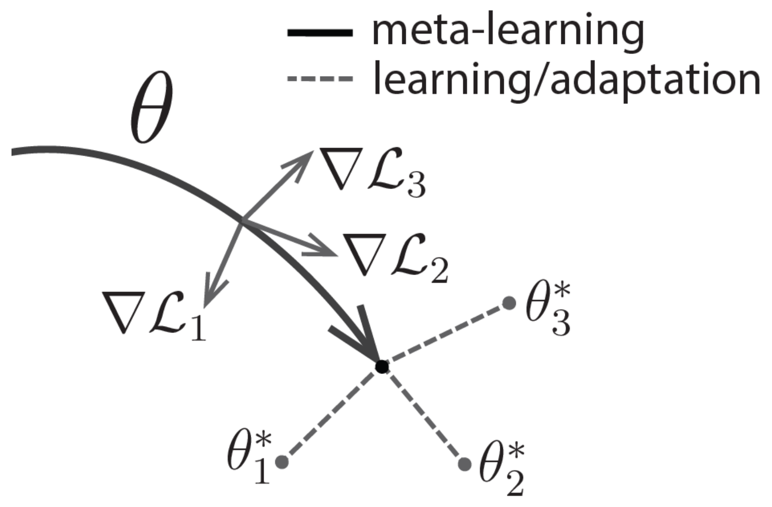
Idea 2: use **efficient off-policy model-free RL** for meta-training

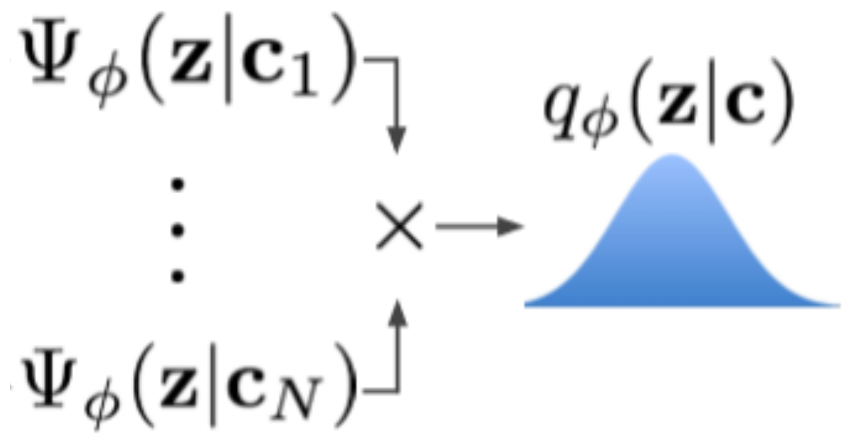


meta-train with soft actor-critic (SAC), state-of-the-art off-policy RL method

Rakelly*, Zhou*, Quillen, Finn, Levine. Efficient Off-Policy Meta-Reinforcement learning via Probabilistic Context Variables

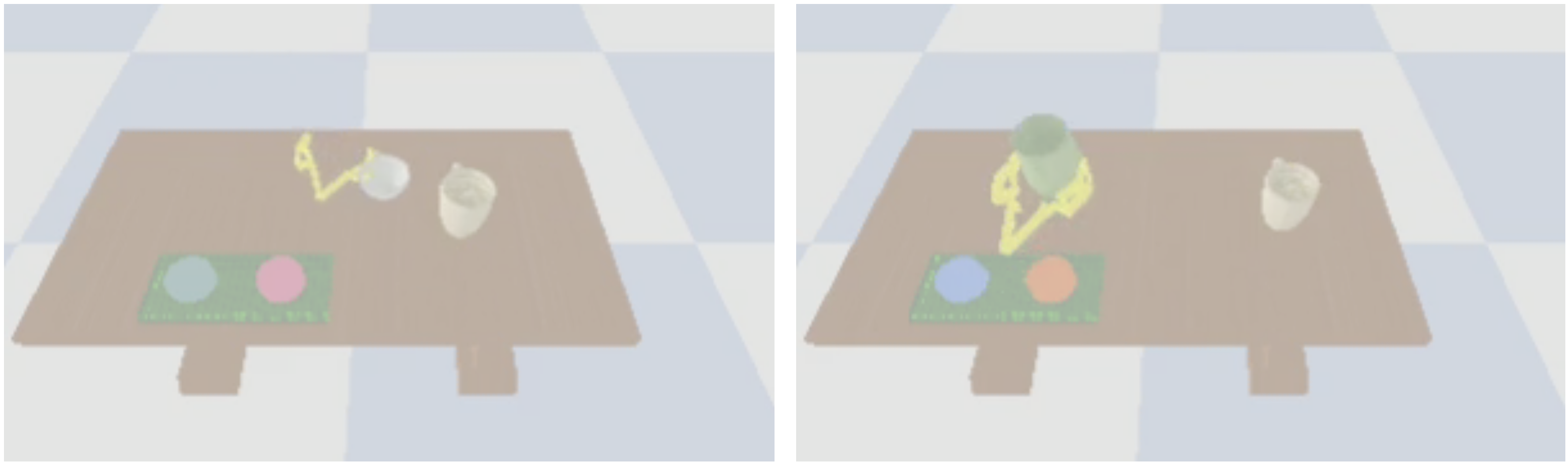
Can robots learn something that can help them adapt *quickly*?



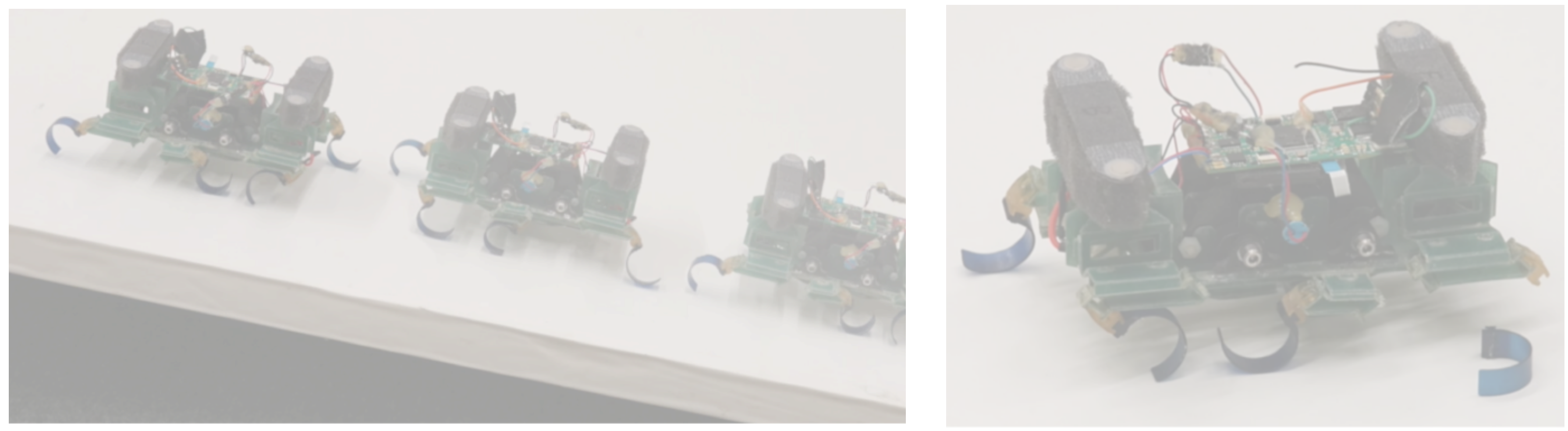


Primer on **few-shot meta-learning**

Challenges in applications to robotics:

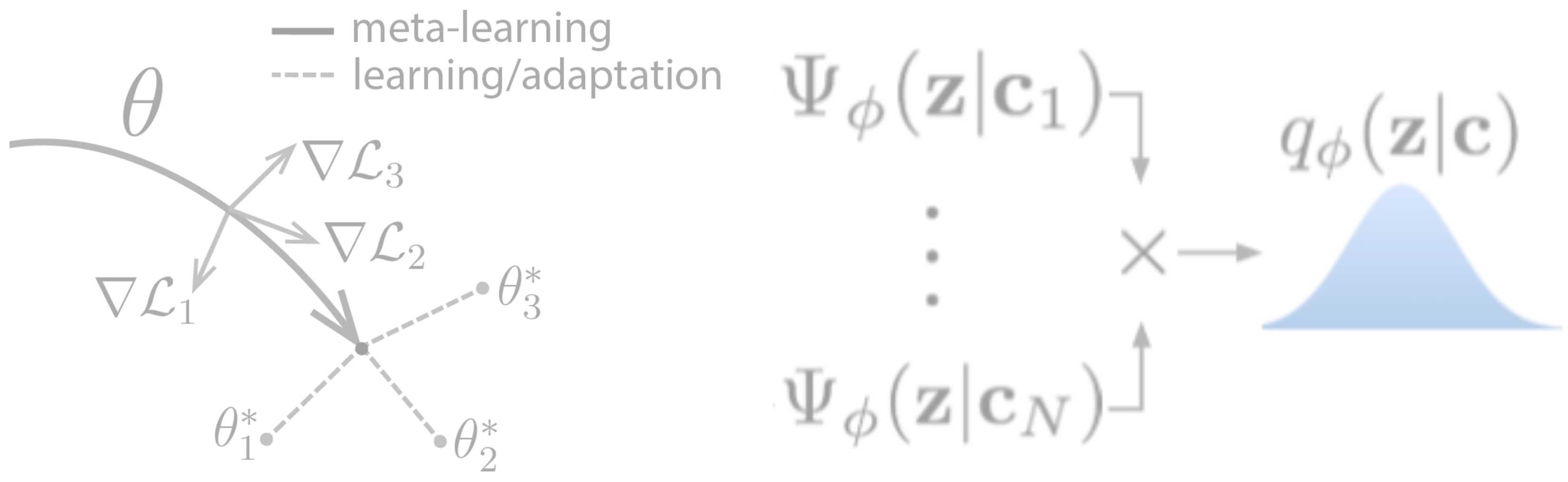


Meta-learning across **families of manipulation tasks**



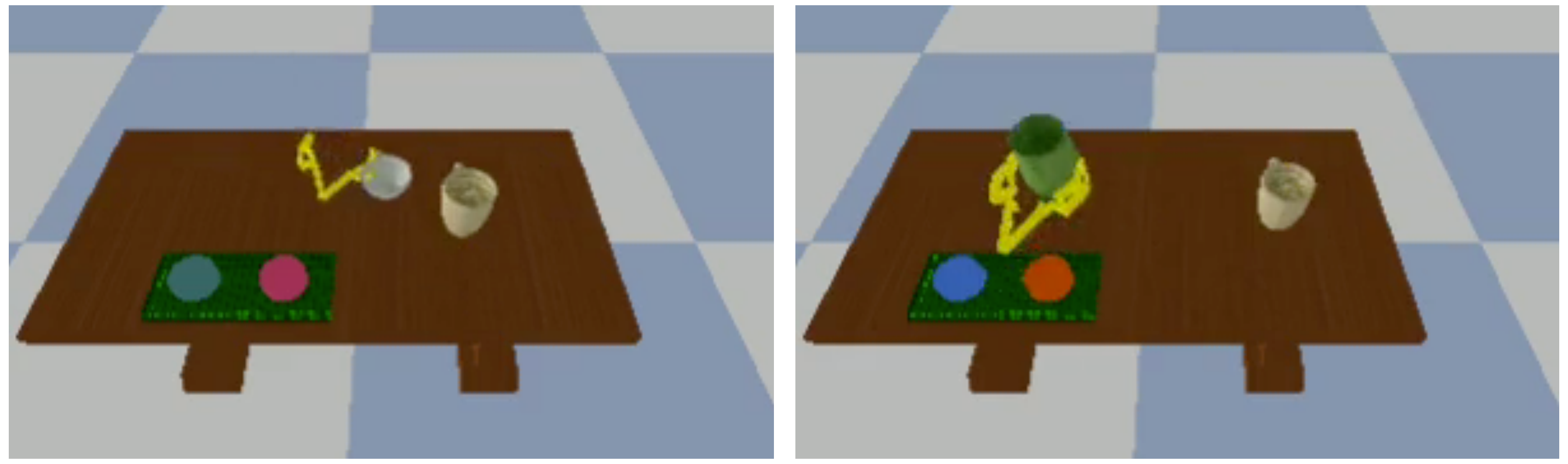
Rapid, ***online*** adaptation to **drastic changes in dynamics**

Can robots learn something that can help them adapt *quickly*?

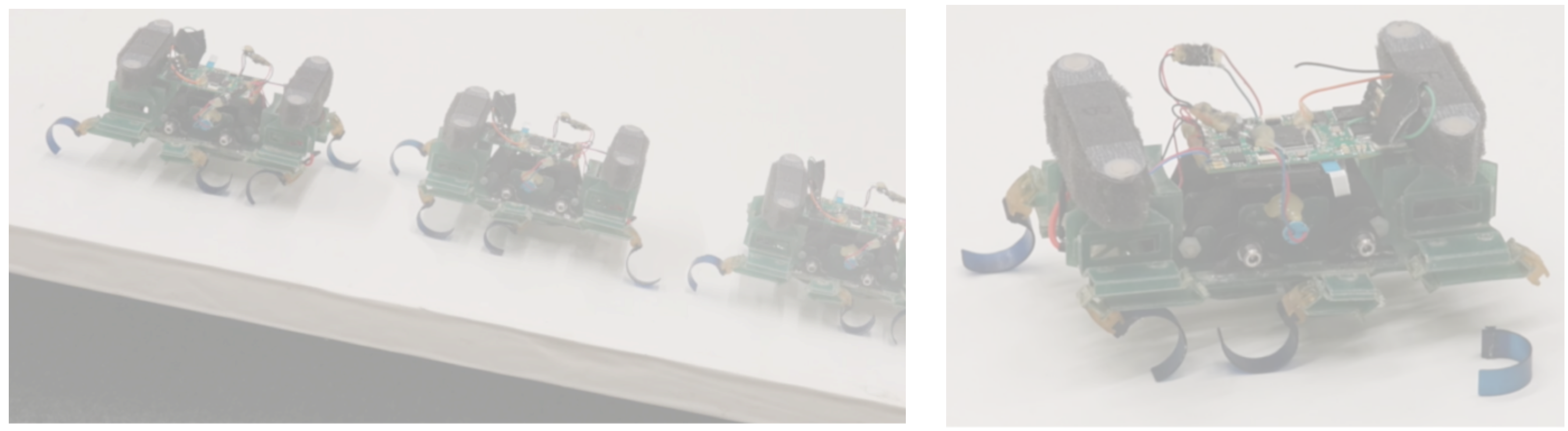


Primer on **few-shot meta-learning**

Challenges in applications to robotics:



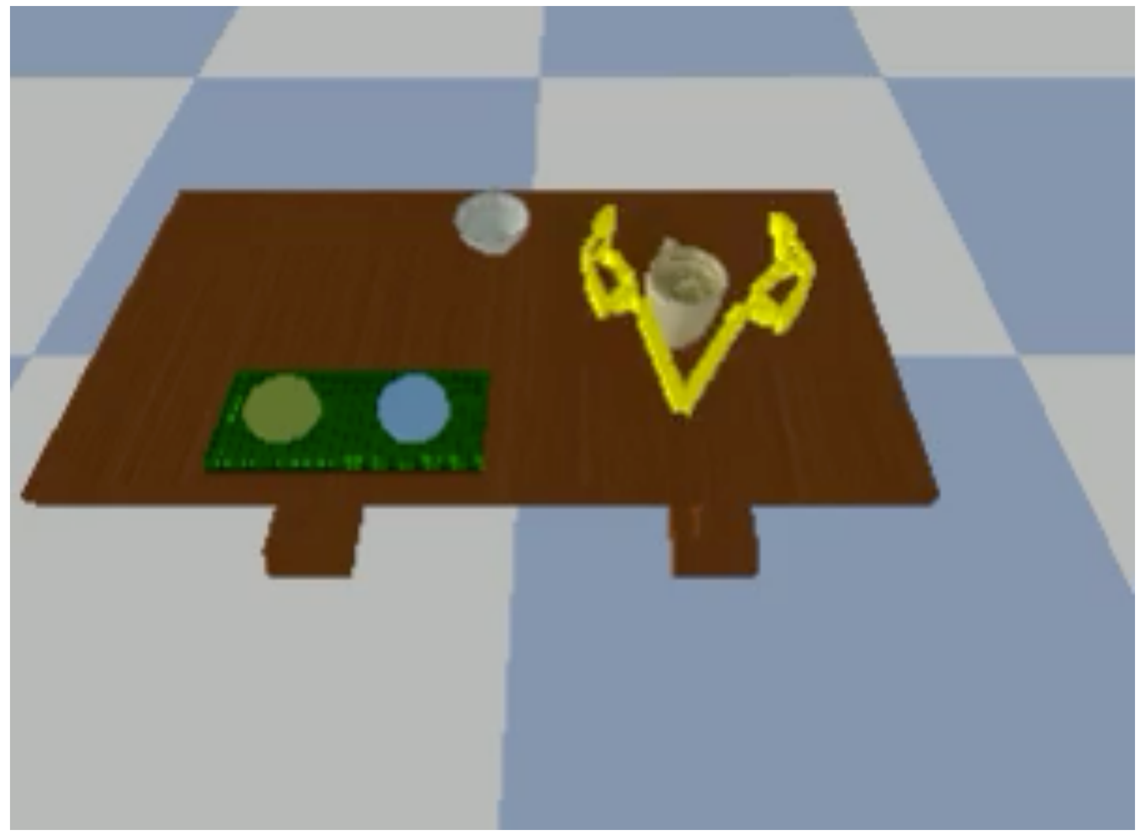
Meta-learning across **families of manipulation tasks**



Rapid, ***online*** adaptation to **drastic changes in dynamics**

Can we meta-learn **across task families**?

Space of manipulation tasks



- grasping objects
- pressing buttons
- sliding objects
- stacking two objects

Goal: Learn a new variation of one of these task families with a **small number of trials** & **sparse rewards**

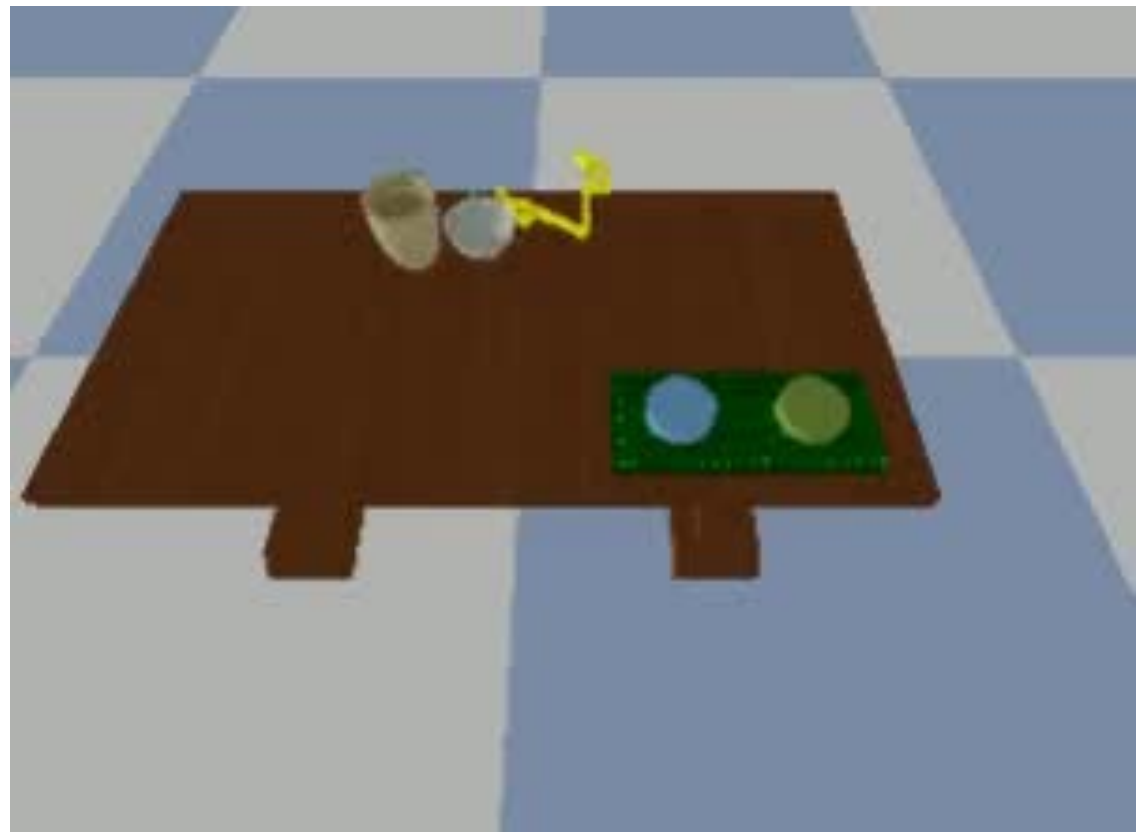
Problem: Robot will have to explore **every possible task**.

This work: Can we learn from **one demonstration** & **a few trials**?
(to convey the task) (to figure out how to solve it)

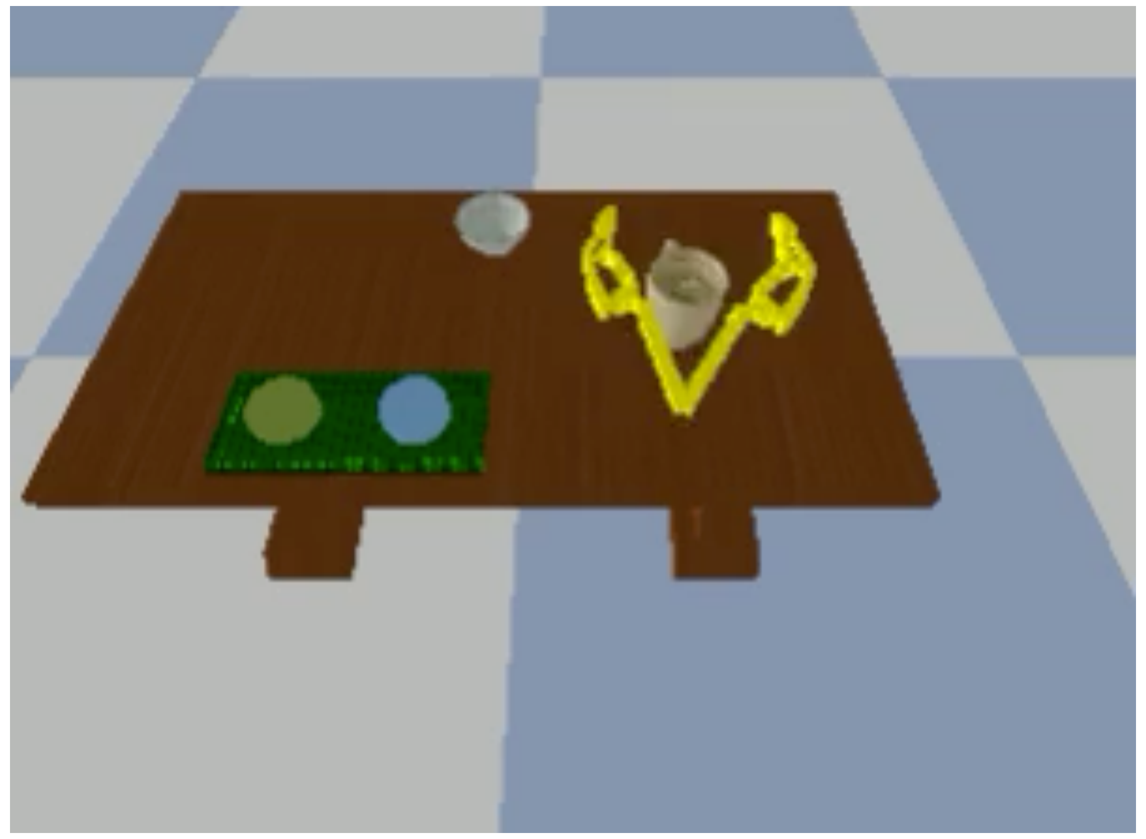
Zhao, Jang, Kappler, Herzog, Khansari, Bai, Kalakrishnan, Levine, Finn. Watch-Try-Learn. '19

Can we learn from one demonstration & a few trials?

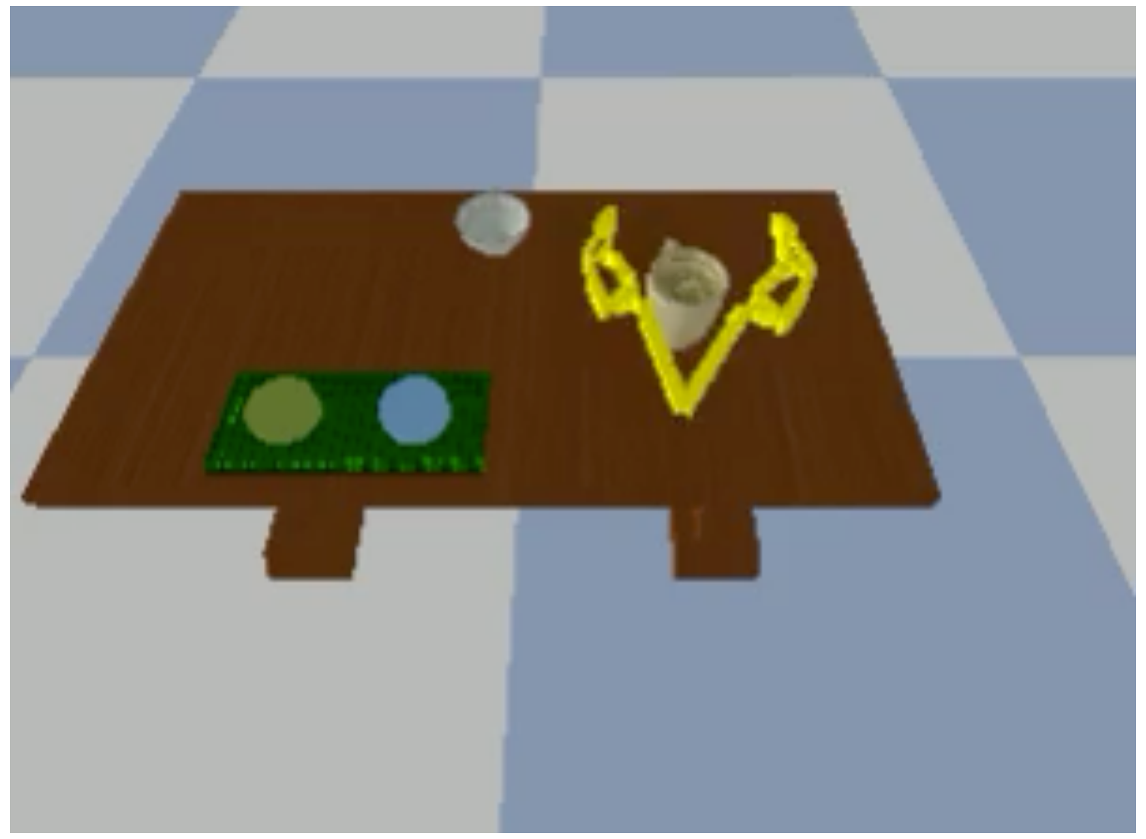
Watch one task demonstration



Try task in new situation



Learn from demo & trial to solve task



How can we train for this in a scalable way?

1. Collect a **few** demonstrations for **many** different tasks
2. Train a **one-shot imitation learning** policy.
3. Collect trials for each task by running one-shot imitation policy.
   [batch off-policy collection]
4. Train **“re-trial” policy** through imitation objective. $\mathcal{D}_{\text{train}}$: demo + trial(s)

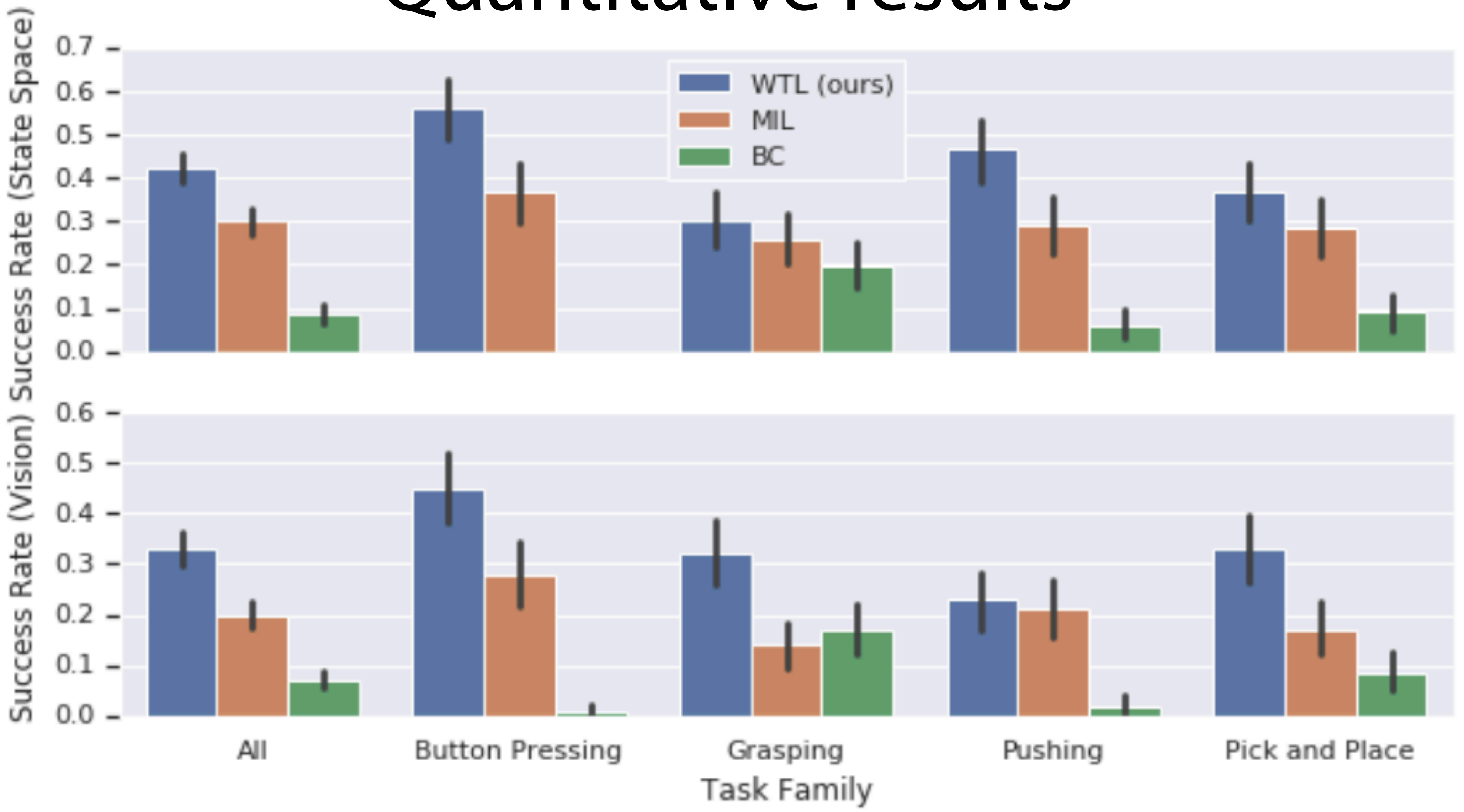
Zhao, Jang, Kappler, Herzog, Khansari, Bai, Kalakrishnan, Levine, Finn. Watch-Try-Learn. ‘19

Experiments

Compare:

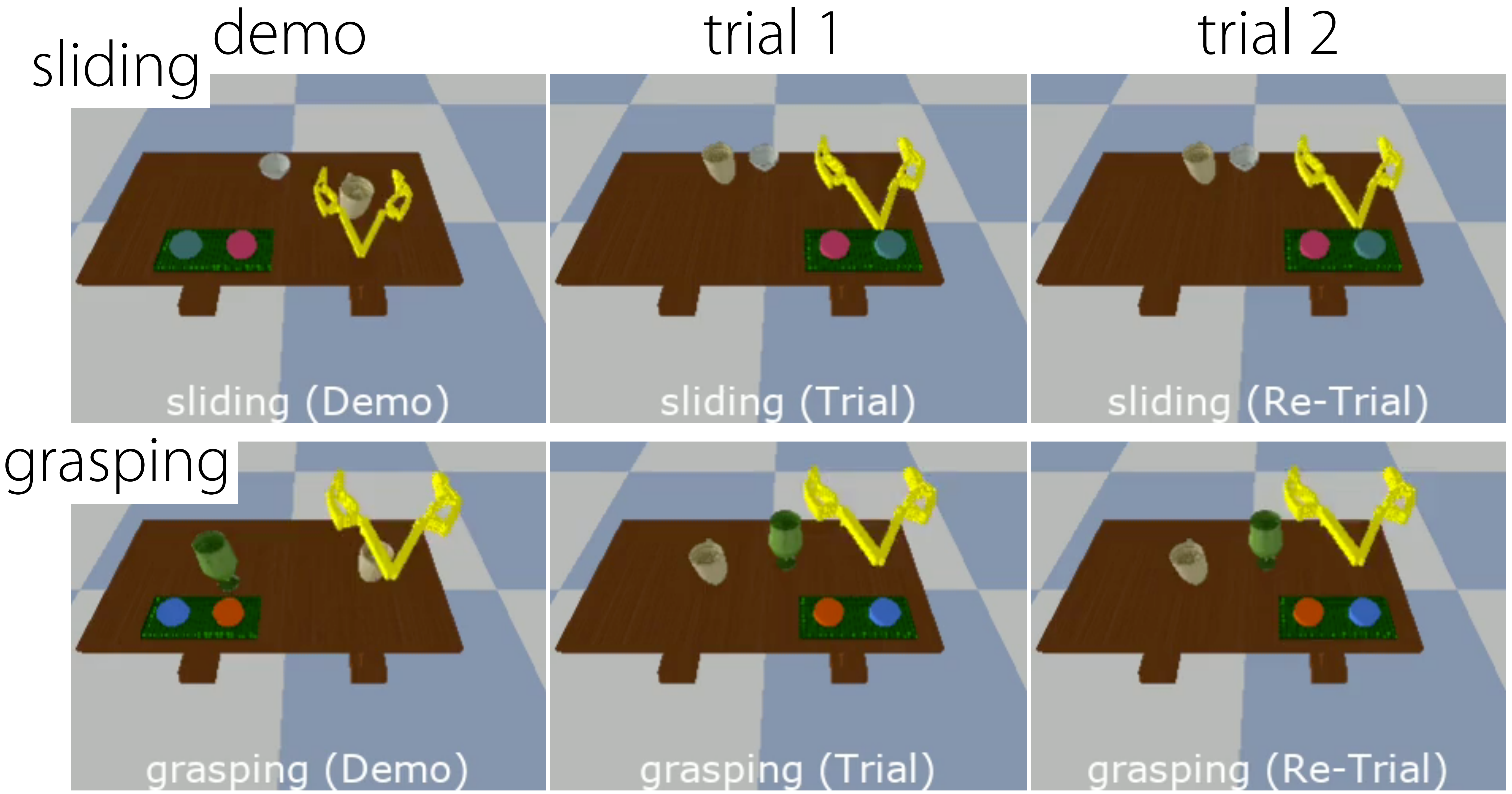
- Watch-Try-Learn (one trial + one demo)
- meta-reinforcement learning (only use trials)
- meta imitation learning (only use demonstration)
- behavior cloning across all tasks (no meta-learning)

Quantitative results



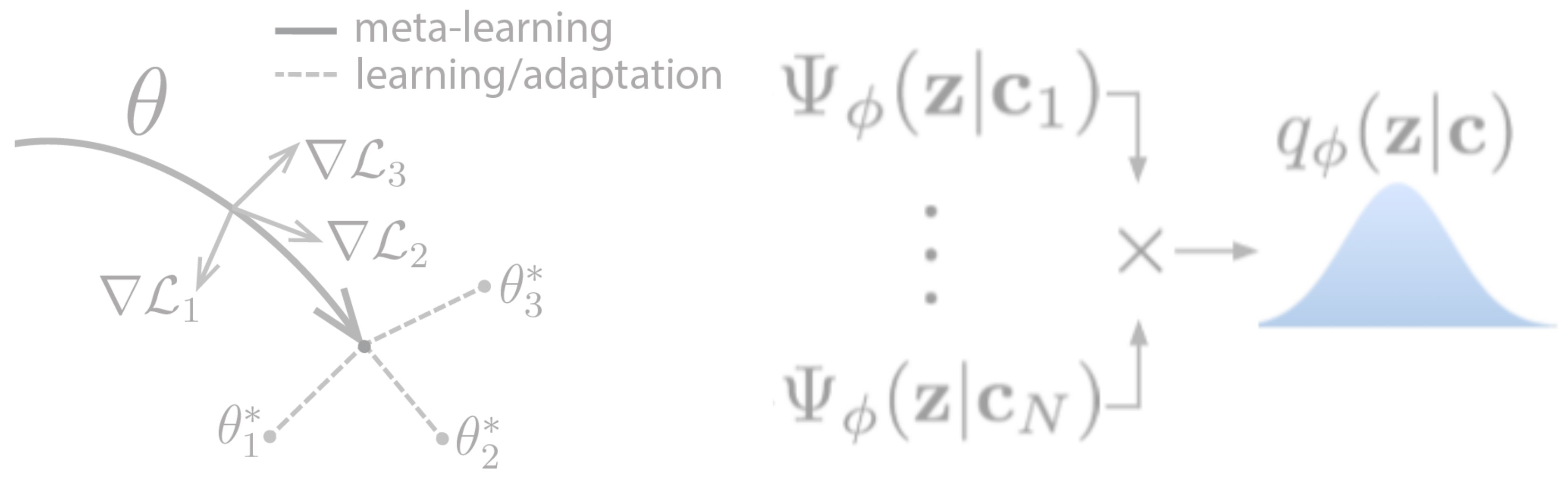
- WTL learns across **4 distinct task families**
- significantly outperforms using only trials or only demos

Qualitative examples



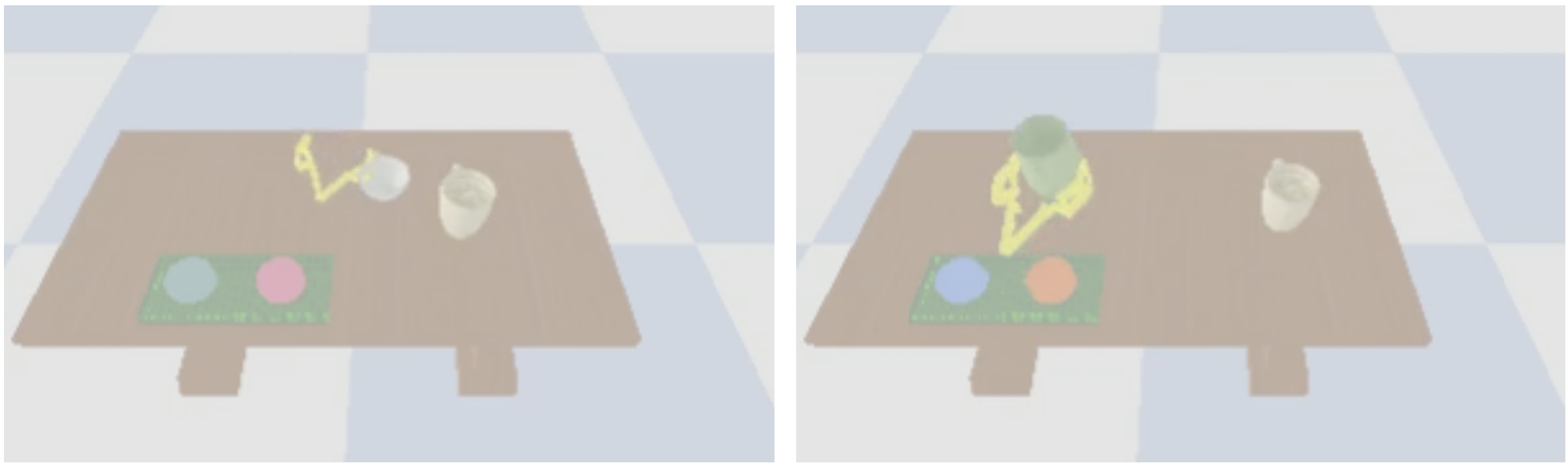
Reinforcement learning from BC initialization requires **900 trials** to match performance of WTL.

Can robots learn something that can help them adapt *quickly*?

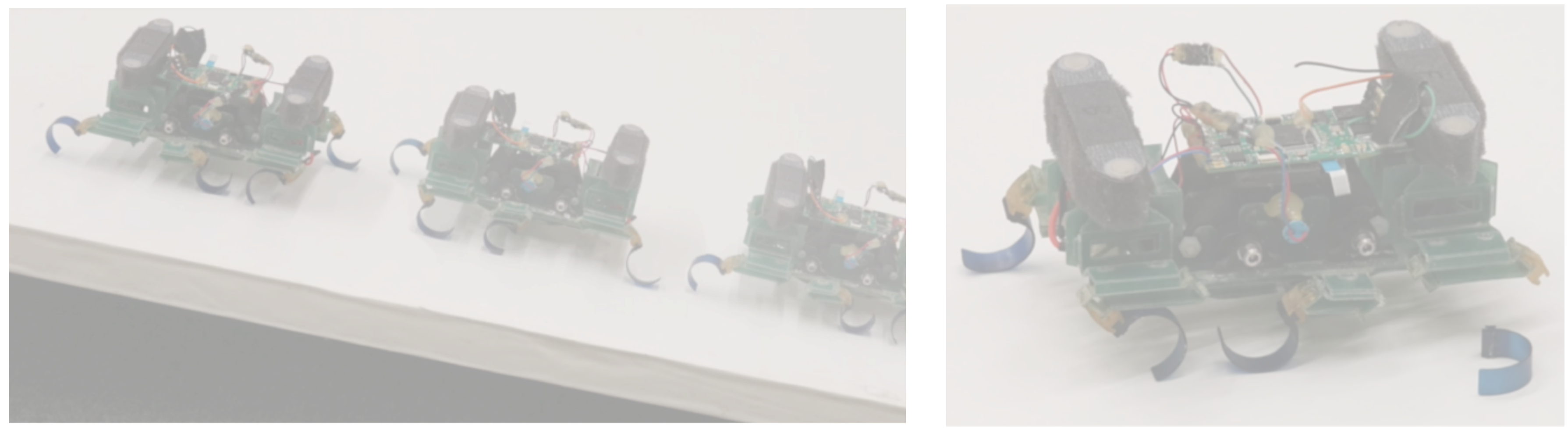


Primer on **few-shot meta-learning**

Challenges in applications to robotics:

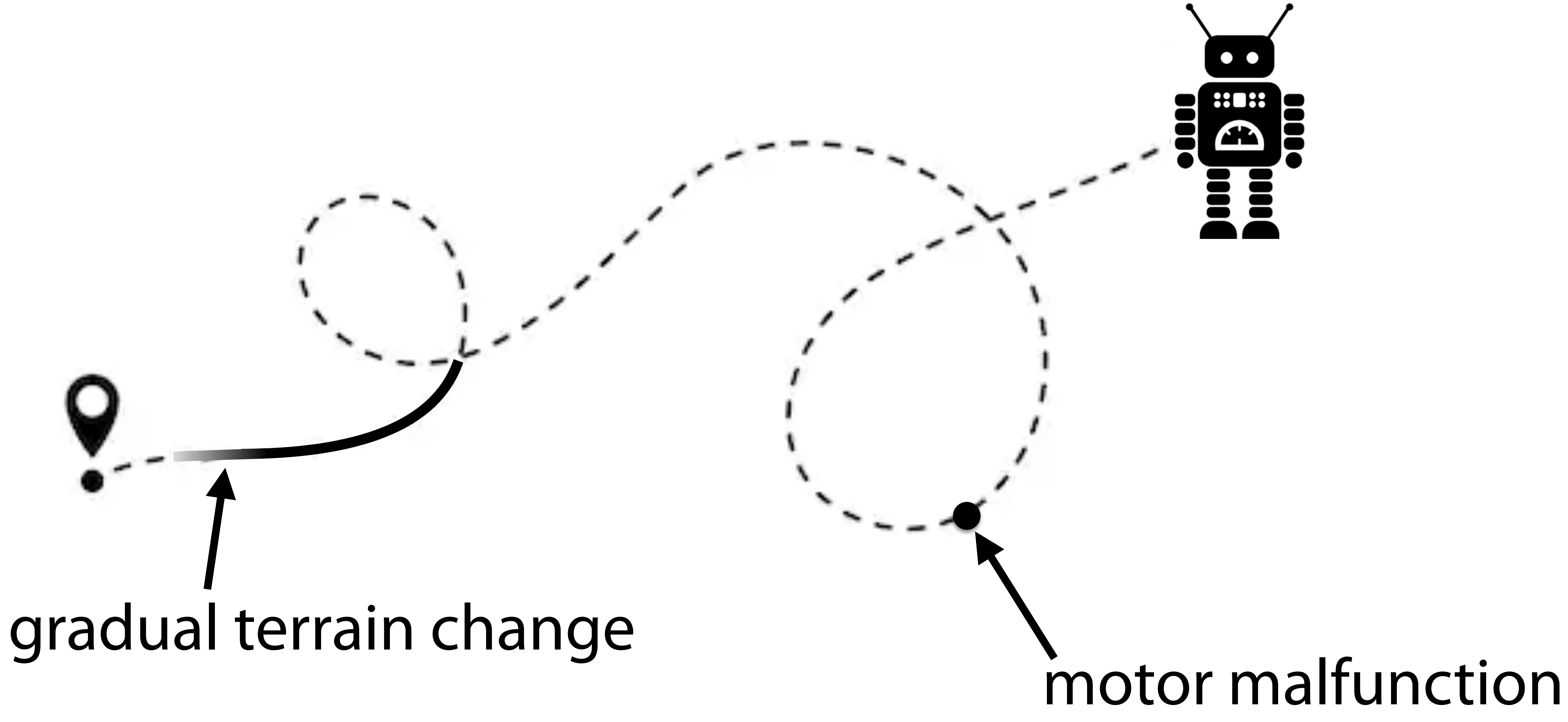


Meta-learning across **families of manipulation tasks**



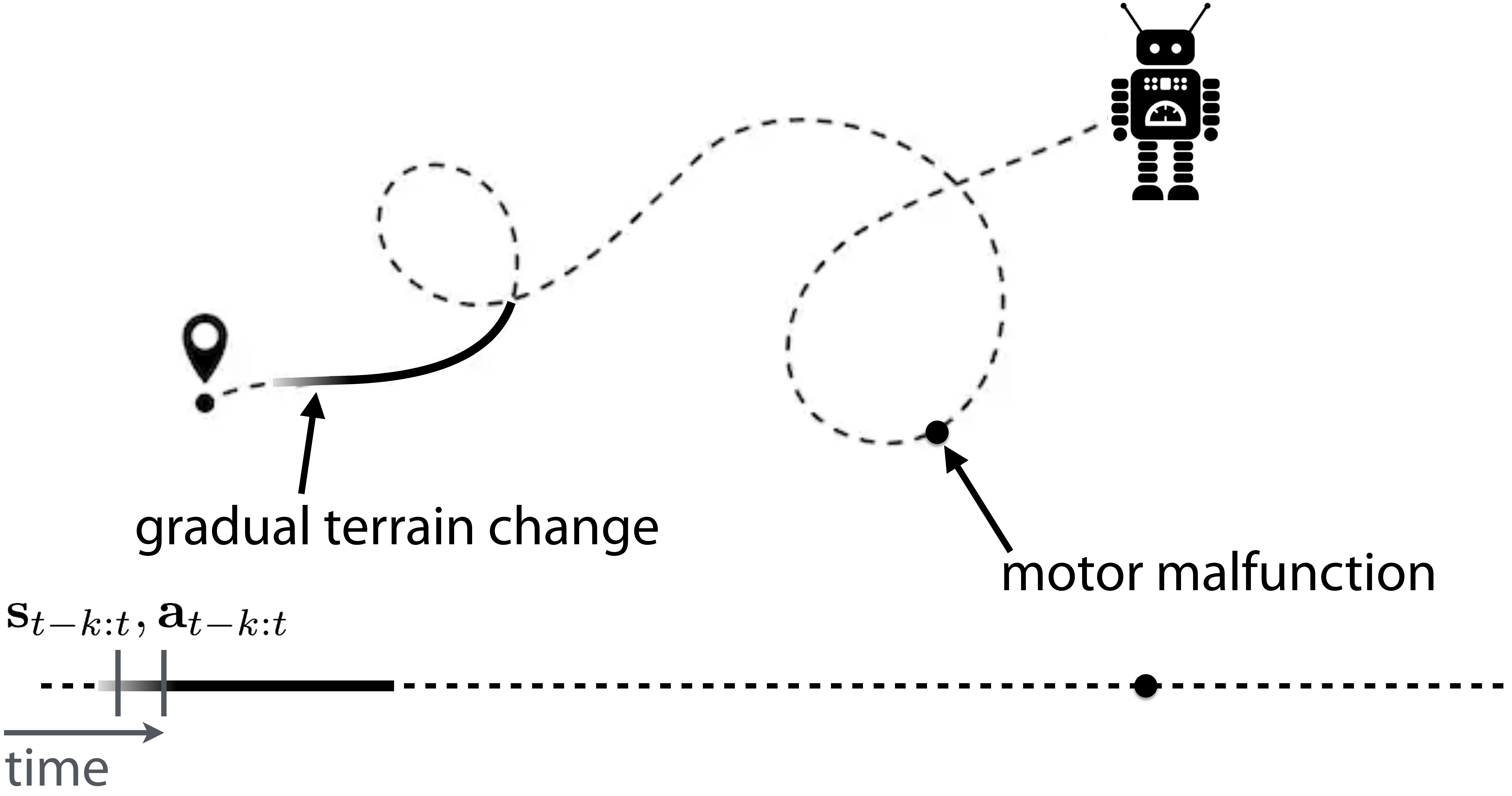
Rapid, ***online*** adaptation to **drastic changes in dynamics**

Goal: learn to adapt model quickly to new environments



Nagabandi*, Clavera*, Liu, Fearing, Abbeel, Levine, Finn. Learning to Adapt in Dynamic Real-World Environments through Meta-RL

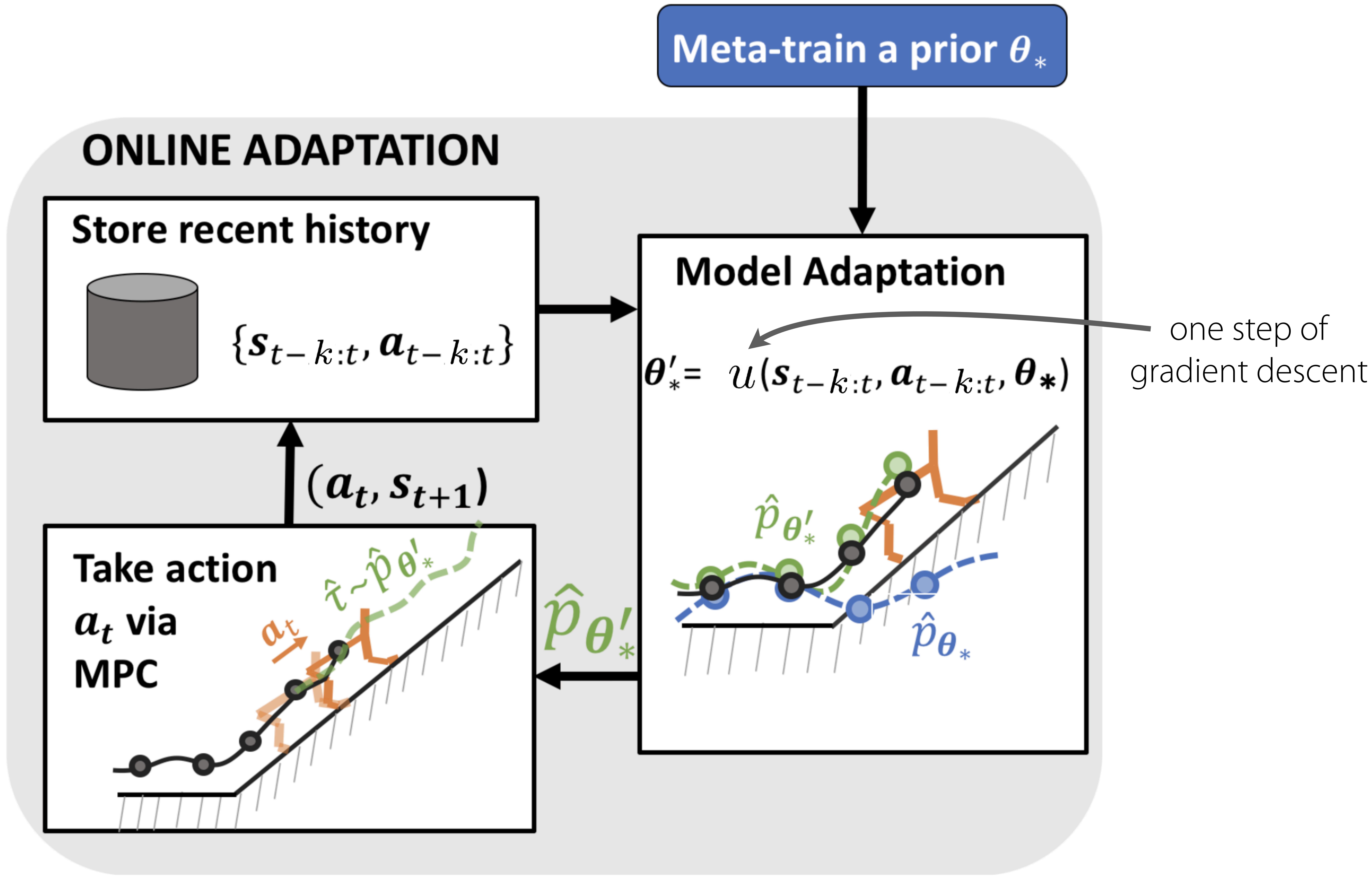
Goal: learn to adapt model quickly to new environments



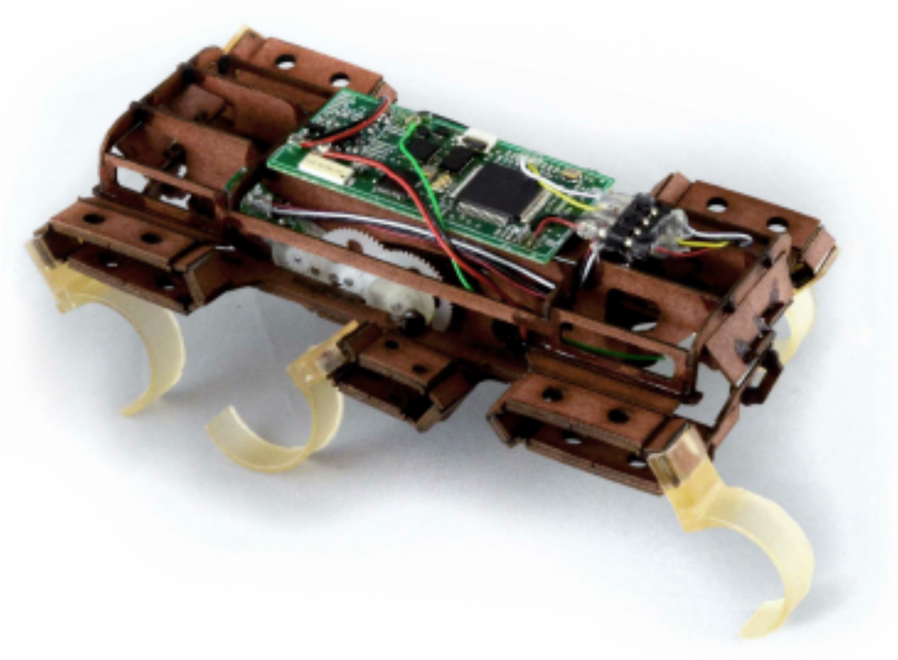
online adaptation = few-shot learning

tasks are **temporal slices** of experience

Nagabandi*, Clavera*, Liu, Fearing, Abbeel, Levine, Finn. Learning to Adapt in Dynamic Real-World Environments via Meta-RL. ICLR'19

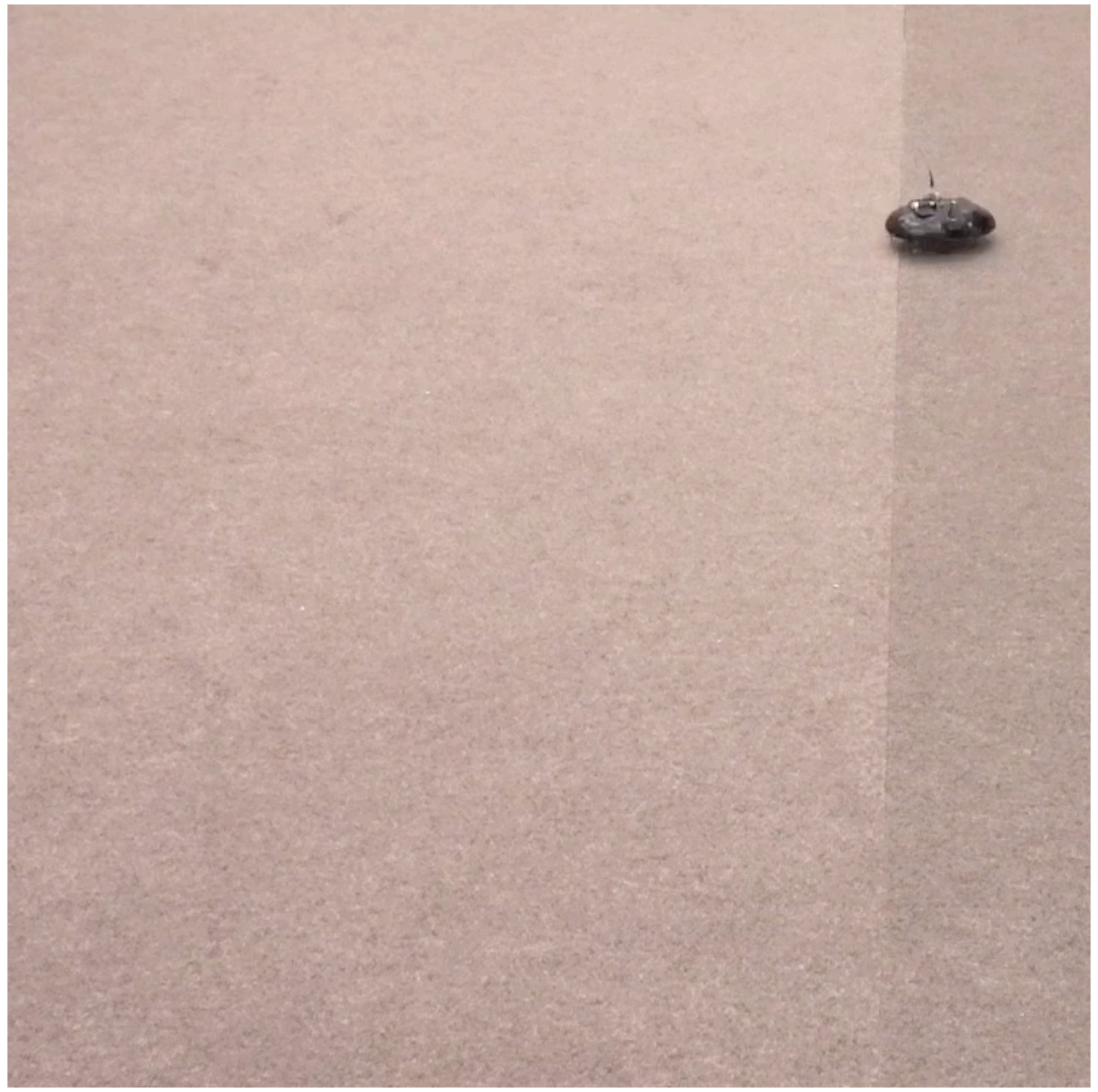


Nagabandi*, Clavera*, Liu, Fearing, Abbeel, Levine, Finn. Learning to Adapt in Dynamic Environments through Meta-RL. ICLR'19



VelociRoACH Robot

Meta-train on **variable terrains**



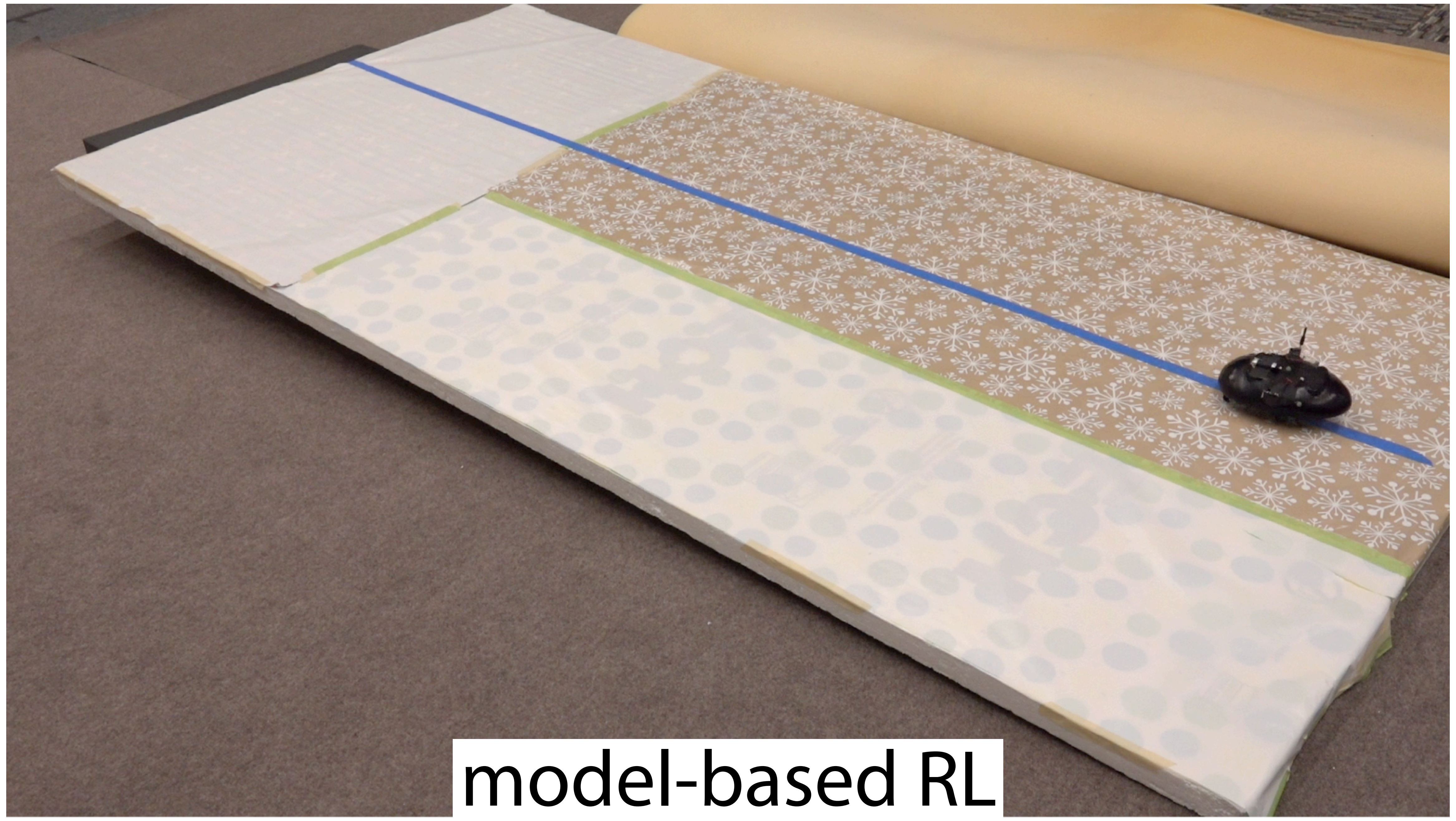
Meta-test with **slope, missing leg, payload, calibration errors**

Nagabandi*, Clavera*, Liu, Fearing, Abbeel, Levine, Finn. Learning to Adapt in Dynamic Environments through Meta-RL. ICLR'19

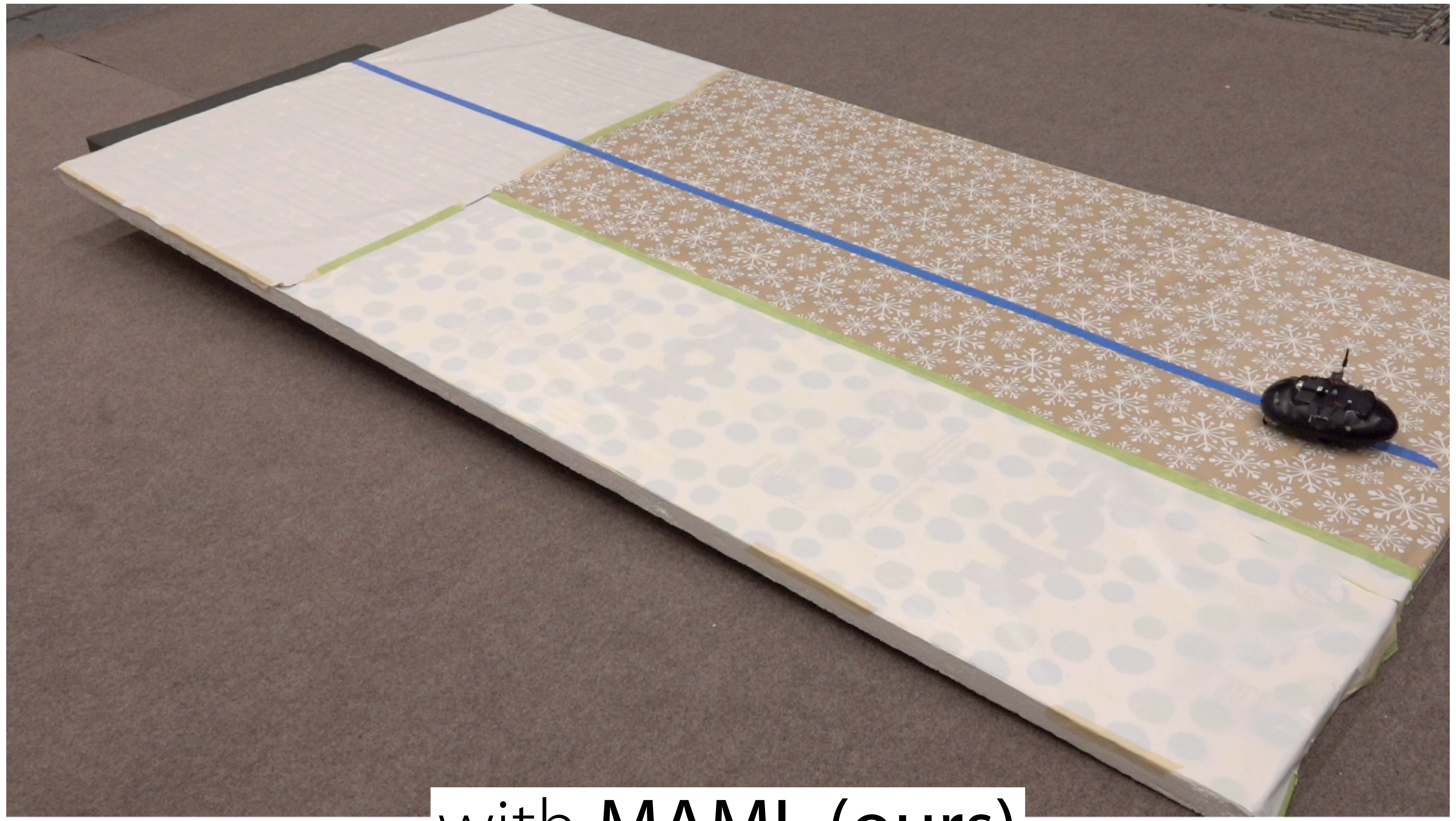
VelociRoACH Robot

Meta-train on **variable terrains**

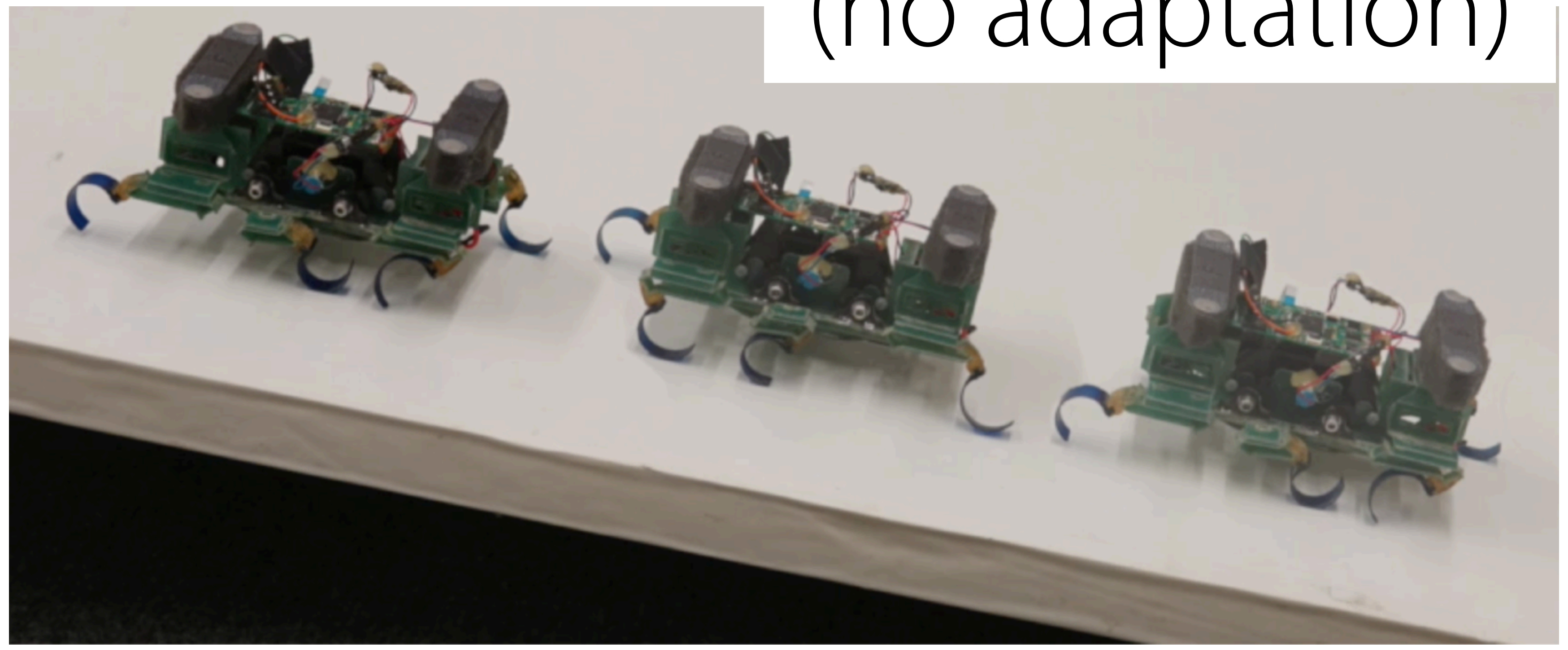
Meta-test with **<u>slope</u>, missing leg, payload, calibration errors**



model-based RL
(no adaptation)



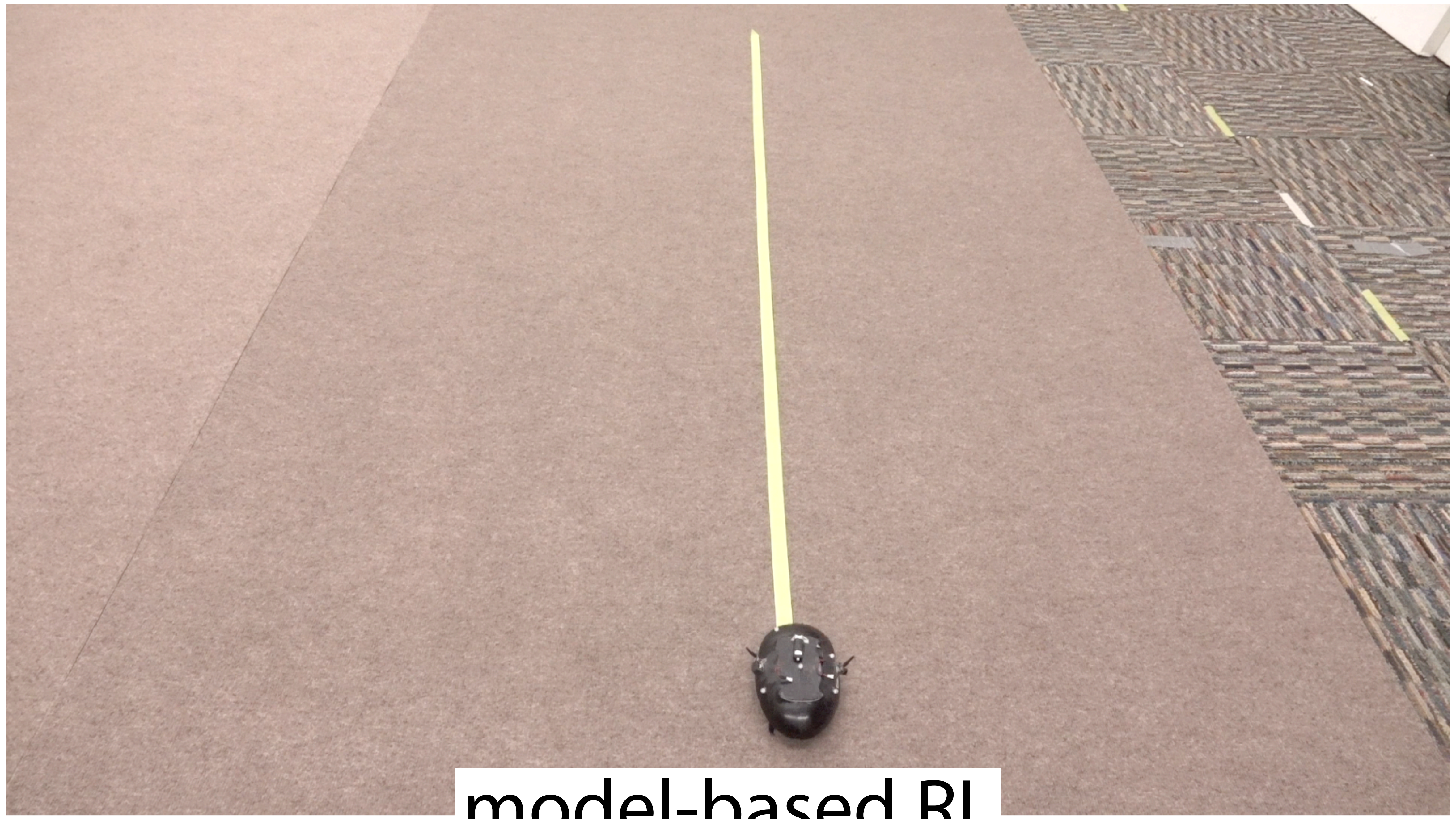
with **MAML (ours)**



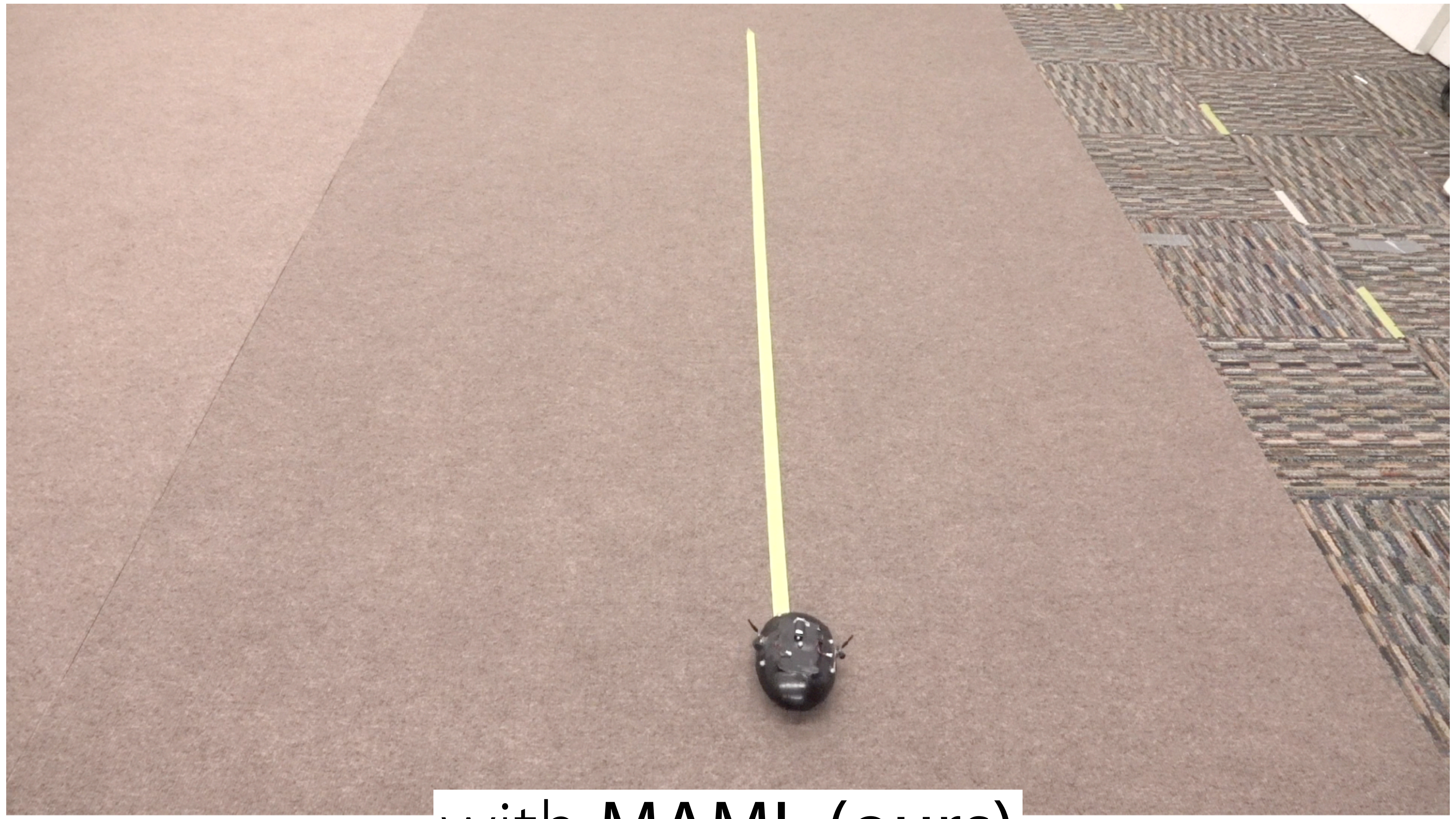
Nagabandi*, Clavera*, Liu, Fearing, Abbeel, Levine, Finn. Learning to Adapt in Dynamic Environments through Meta-RL. ICLR'19

VelociRoACH Robot

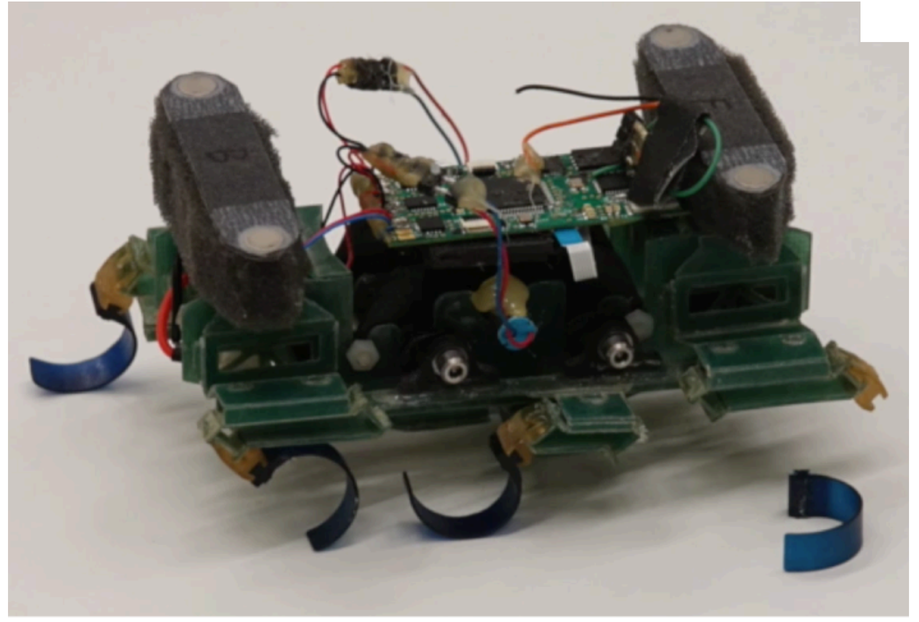
Meta-train on **variable terrains** **Meta-test** with **slope**, <u>**missing leg**</u>, **payload**, **calibration errors**



model-based RL
(no adaptation)

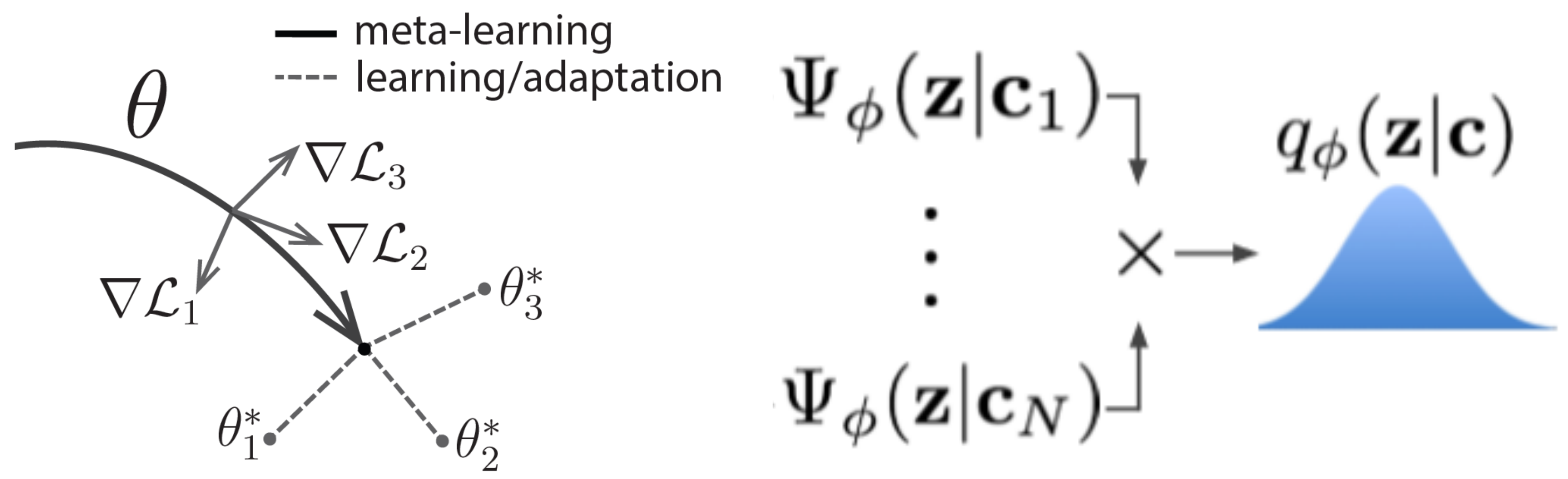


with **MAML (ours)**



Nagabandi*, Clavera*, Liu, Fearing, Abbeel, Levine, Finn. Learning to Adapt in Dynamic Environments through Meta-RL. ICLR '19

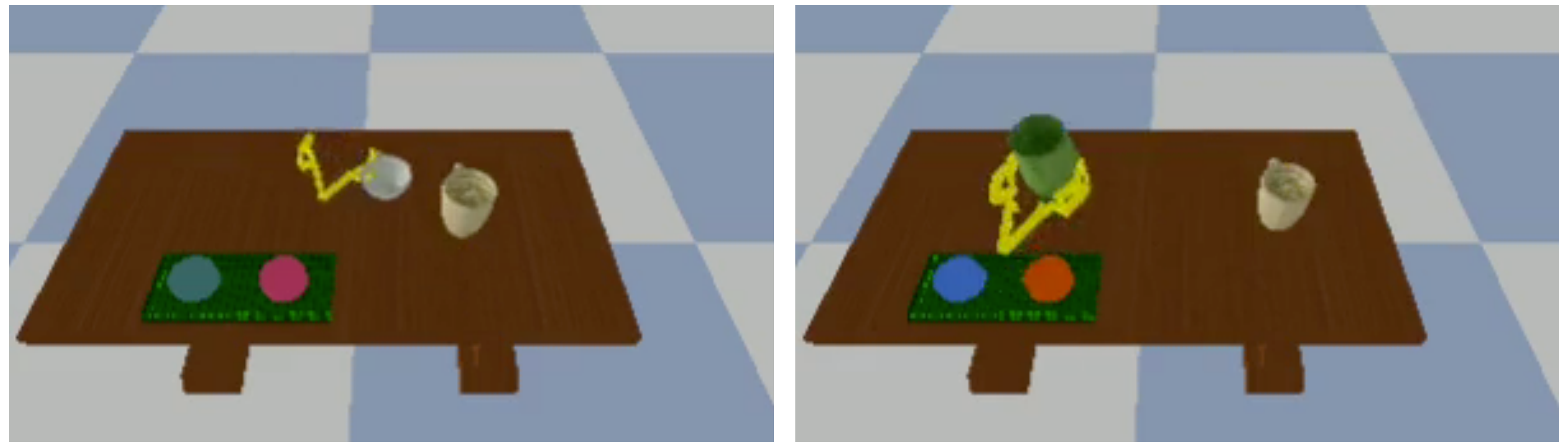
Can robots learn something that can help them adapt *quickly*?



Quick primer on **few-shot meta-learning**

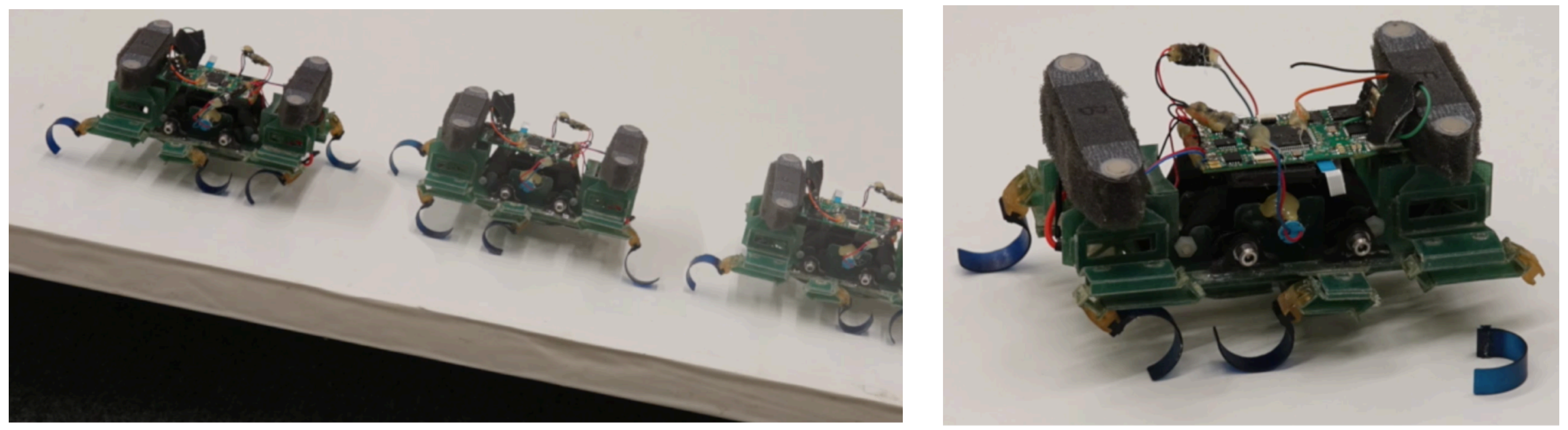
(and it's extension to RL)

Challenges in applications to robotics:



Adapt to **new vision-based manipulation task**

from only 1 demo & 1 trial



Adapt *online* to **drastic changes in dynamics**

Key takeaway: Leverage **previous data** to **optimize for fast adaptation**

Closing Thoughts on Simulation to Real-World Transfer

What is simulators useful for:

algorithm development

short-horizon wins (~3 yr)

What it is not useful for:

autonomous learning without human expertise

better performance in the long run (3+ yrs)

Typical sim2real pipeline:

1. Identify real task
2. Hand design a simulator and/or randomization parameters for that task
3. Optimize for behavior in sim.
4. Try out behavior in the real world.

iterate

Defeats the point of reinforcement learning!

(the *autonomous* acquisition of a breadth of skills)

Sim2Real Counterargument: We will design better and better simulators of the world

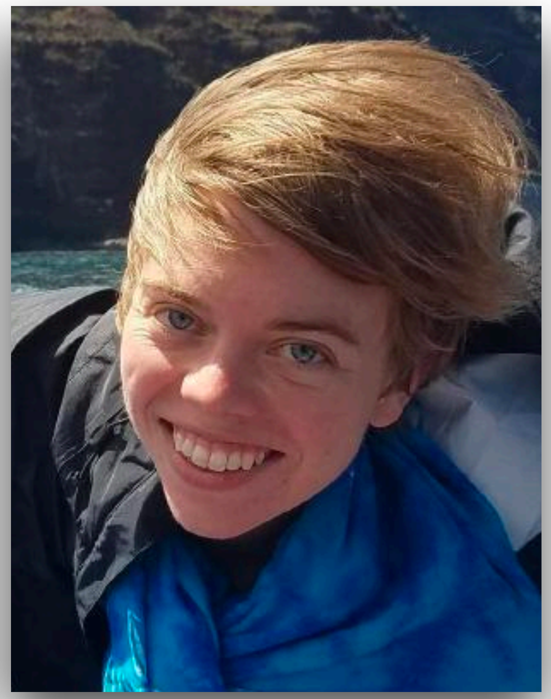
Computer vision: design better features?
Go: incorporate human gameplay?
Machine translation: incorporate grammar?

Learning from data is what consistently wins.

Collaborators & Students

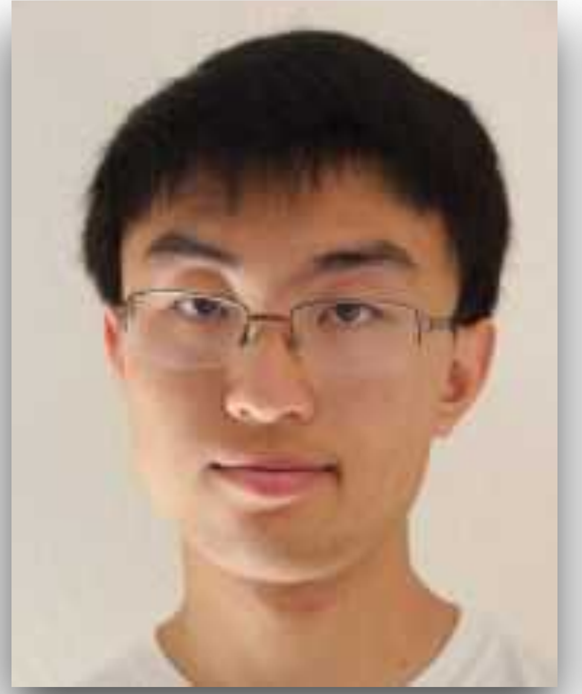
Sergey Levine

Kate Rakelly



Deirdre Quillen

Aurick Zhou



Pieter Abbeel

Anusha Nagabandi

Ignasi Clavera

Simin Liu

Allan Zhou

Eric Jang

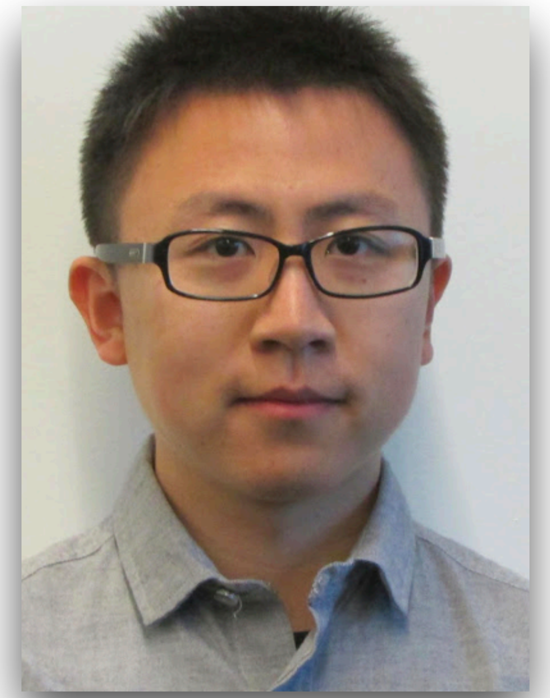
Daniel Kappler

Alex Herzog

Paul Wohlhart

Mohi Khansari

Yunfei Bai



Mrinal Kalakrishnan

Papers, data, and **code** linked at: people.eecs.berkeley.edu/~cbfinn

Questions?