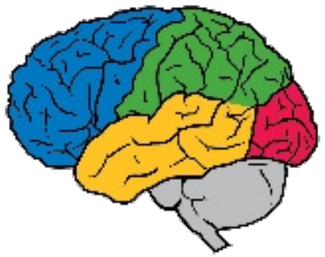
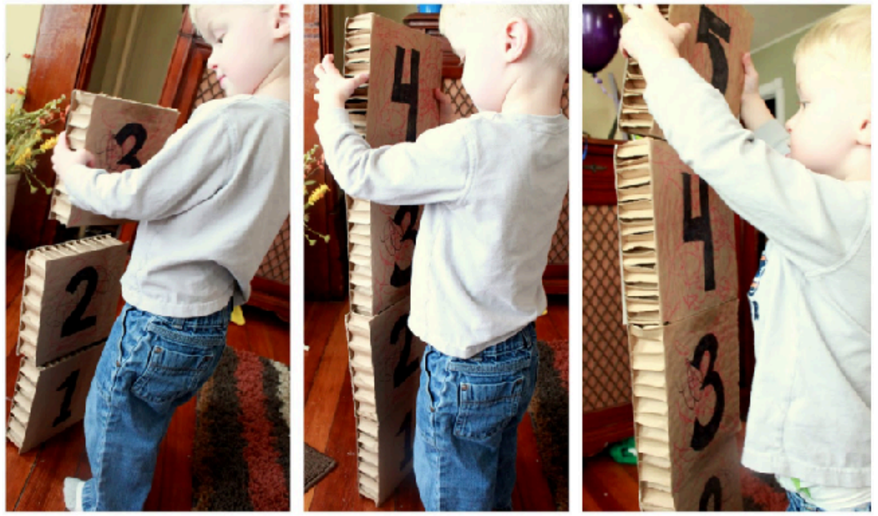
Meta Reinforcement Learning

Chelsea Finn



Why are humans so good at RL?



People have prior experience.

People have an existing representation of the world.

Can we learn a representation under which RL is fast?

Key idea: Explicitly optimize for such a representation

"Learn how to reinforcement learn"

Outline

Meta-RL Problem Formulation & Examples

Method Classes: Recurrent Models, Gradient-Based Models

Challenges & Latest Developments

The Meta-Learning Problem

Supervised Learning:

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

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

Meta Supervised Learning:

Inputs: $\mathcal{D}_{\text{train}}$ $\mathbf{x}_{\text{test}}$ Outputs: $\mathbf{y}_{\text{test}}$ Data: $\{\mathcal{D}_i\}$

$$\{(\mathbf{x}, \mathbf{y})_{1:K}\}$$

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

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

Why is this view useful?

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

Finn & Levine. *Meta-learning and Universality: Deep Representation…* ICLR 2018

Example: Few-Shot Classification

Given 1 example of 5 classes:

training data $\mathcal{D}_{\text{train}}$

Classify new examples

test set $\mathbf{x}_{\text{test}}$

meta-training

$\mathcal{T}_1$

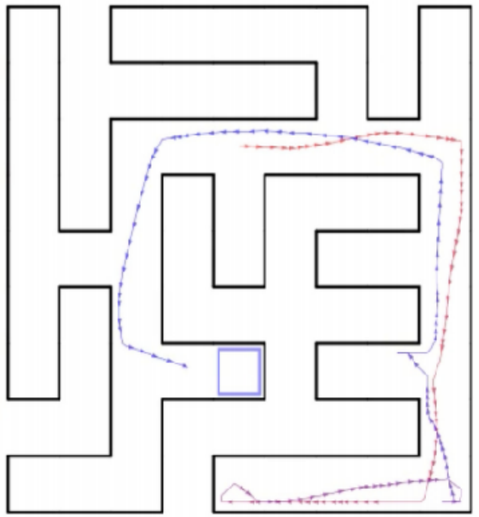
$\mathcal{T}_2$

training classes

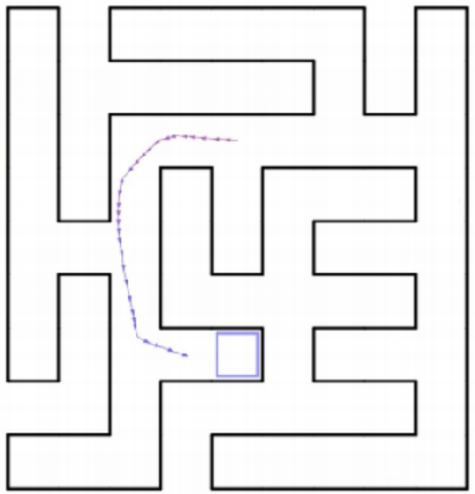
diagram adapted from Ravi & Larochelle '17

Meta-RL Example: Maze Navigation

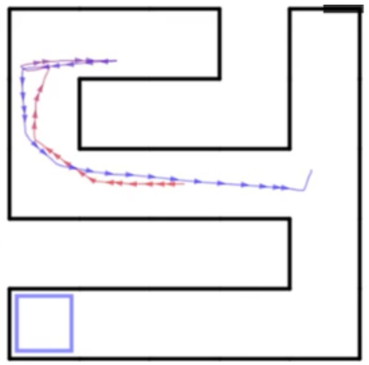
Given a small amount of experience

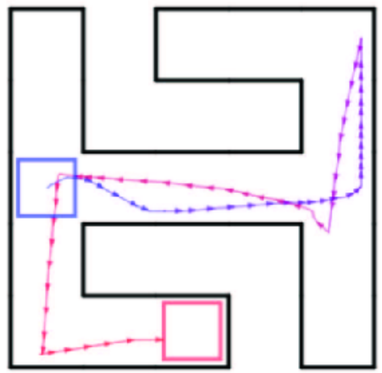


Learn to solve the task



By learning how to learn many other tasks:





…

diagram adapted from Duan et al. ‘17

The Meta Reinforcement Learning Problem

Reinforcement Learning:

Inputs: ~~$\mathbf{x}$~~ $\mathbf{s}_t$ Outputs: ~~$\mathbf{y}$~~ $\mathbf{a}_t$ Data: ~~$\{(\mathbf{x}, \mathbf{y})_i\}$~~

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

$\mathbf{a}_t = \pi(\mathbf{s}_t; \theta)$

$\{(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})\}$

Meta Reinforcement Learning:

Inputs: $\mathcal{D}_{\text{train}}$ $\mathbf{s}_t$ Outputs: $\mathbf{a}_t$ Data: $\{\mathcal{D}_i\}$

k rollouts from π

$\mathbf{a}_t = f(\mathcal{D}_{\text{train}}, \mathbf{s}_t; \theta)$

dataset of datasets collected for each task

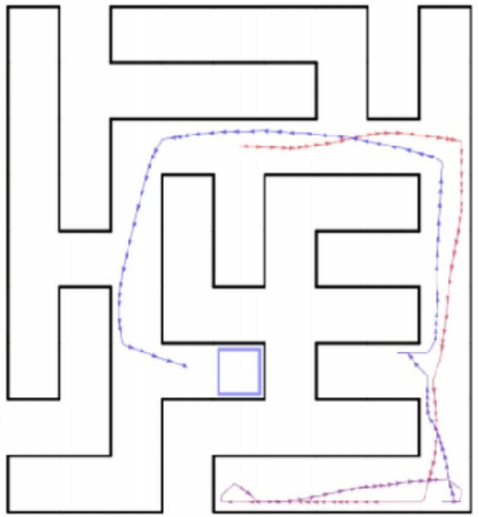
Design & optimization of *f* *and* collecting appropriate data (learning to explore)

Finn. *Learning to Learn with Gradients.* PhD Thesis 2018

Meta-RL Example: Maze Navigation

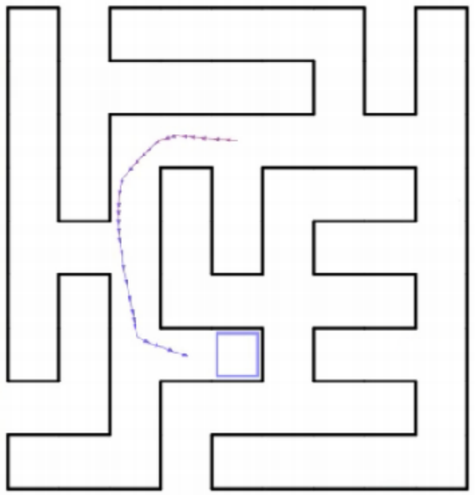
[meta] test time

Given a small amount of experience



$\mathcal{D}_{\text{train}}$

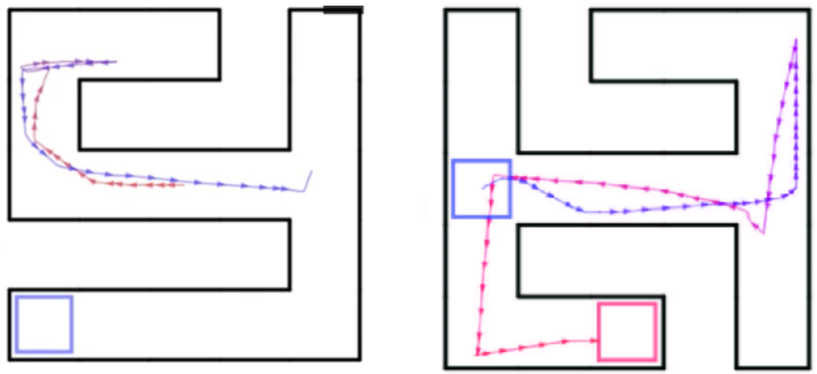
Learn to solve the task



$\mathbf{s}_t \rightarrow \mathbf{a}_t$

[meta] train time

By learning how to learn many other tasks:



$\{\mathcal{D}_i\} \sim \{\mathcal{T}_i\}$

…

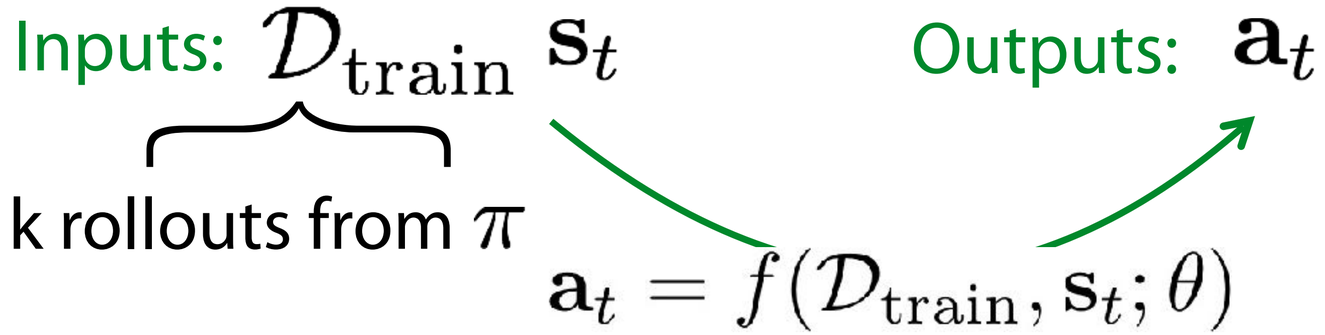
meta-training tasks

diagram adapted from Duan et al. ‘17

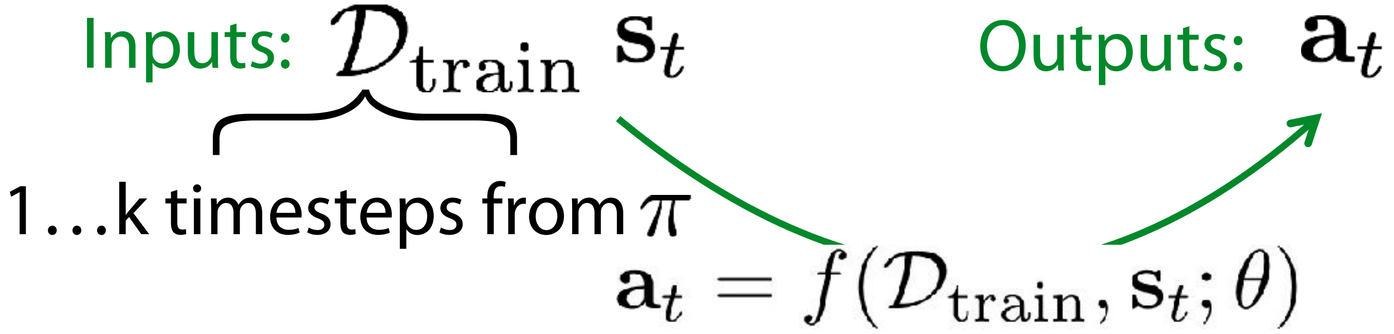
The Meta Reinforcement Learning Problem

Meta Reinforcement Learning:

Episodic Variant



Online Variant



Outline

Meta-RL Problem Formulation & Examples

Method Classes: Recurrent Models, Gradient-Based Models

Challenges & Latest Developments

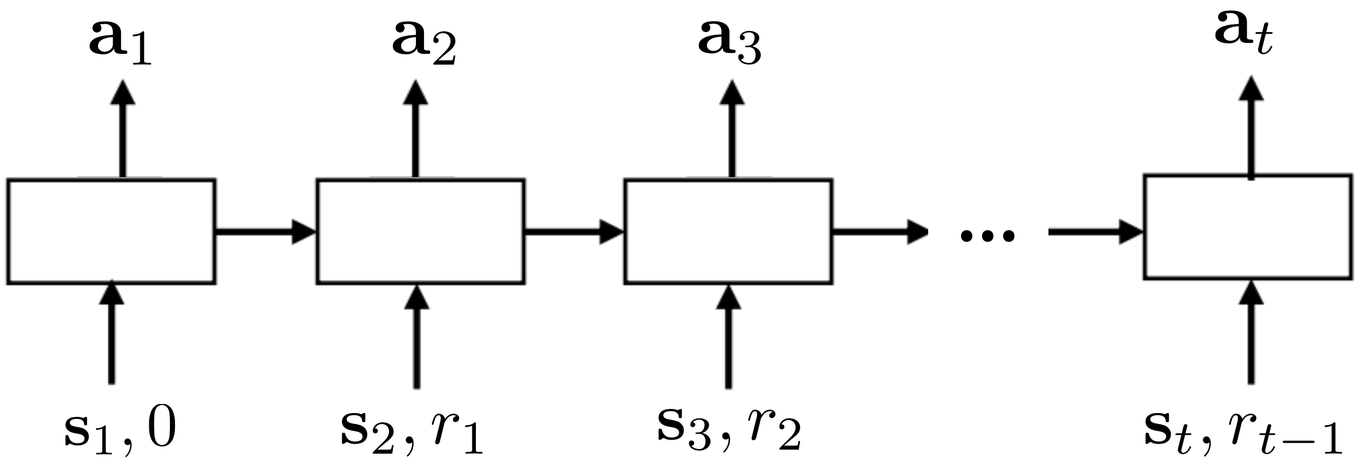
Design of *f*?

$\mathcal{D}_{\text{train}}$ $\mathbf{s}_t$ ⟶ $\mathbf{a}_t$

Recurrent network
(LSTM, NTM, Conv)

$$\mathbf{a}_t = f(\mathcal{D}_{\text{train}}, \mathbf{s}_t; \theta)$$

Santoro et al. '16, **Duan et al. '17**, **Wang et al. '17**, Mishra et al. '17, …



Difference compared to using a recurrent policy?

Hidden state maintained **across episodes** within a task!

Case Study: Simple Neural Attentive Meta-Learner

Mishra et al, ICLR '18

A Simple Neural Attentive Meta-Learner

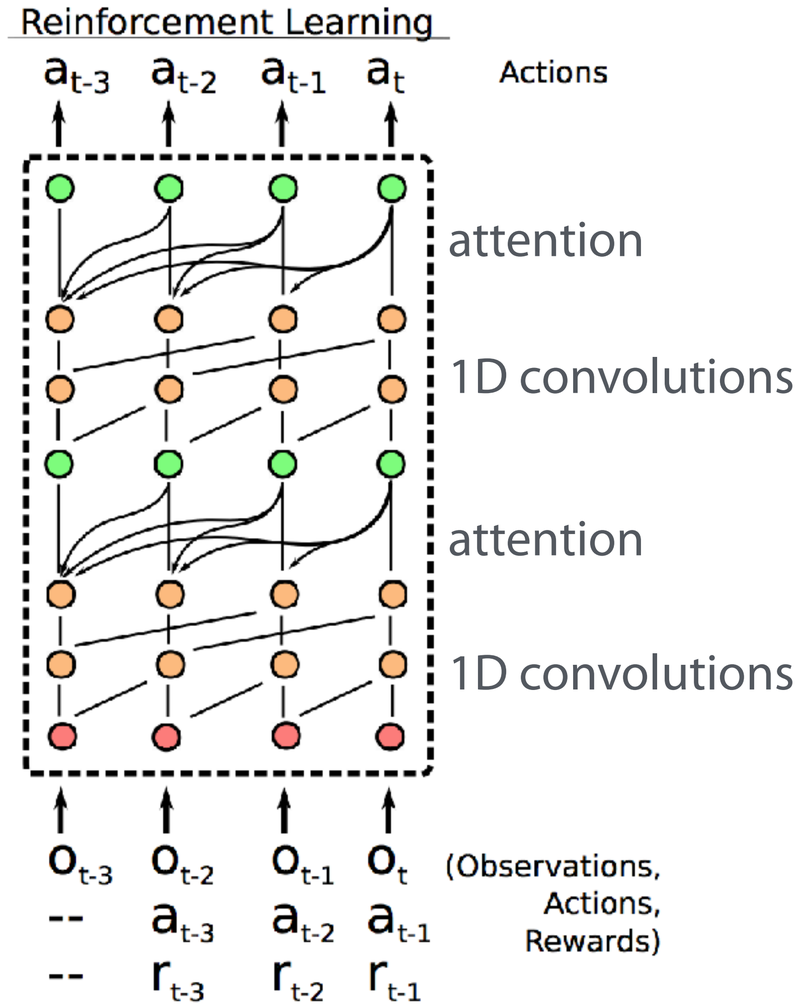
Nikhil Mishra *† **Mostafa Rohaninejad*** **Xi Chen**† **Pieter Abbeel**†

UC Berkeley, Department of Electrical Engineering and Computer Science
Embodied Intelligence
{nmishra, rohaninejadm, c.xi, pabbeel}@berkeley.edu

Simple Neural Attentive Meta-Learner

Mishra et al, ICLR '18

interleave 1D convolutions and attention

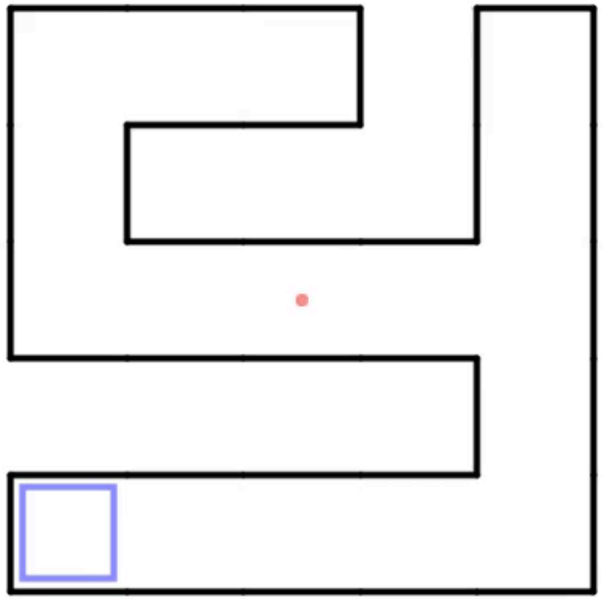


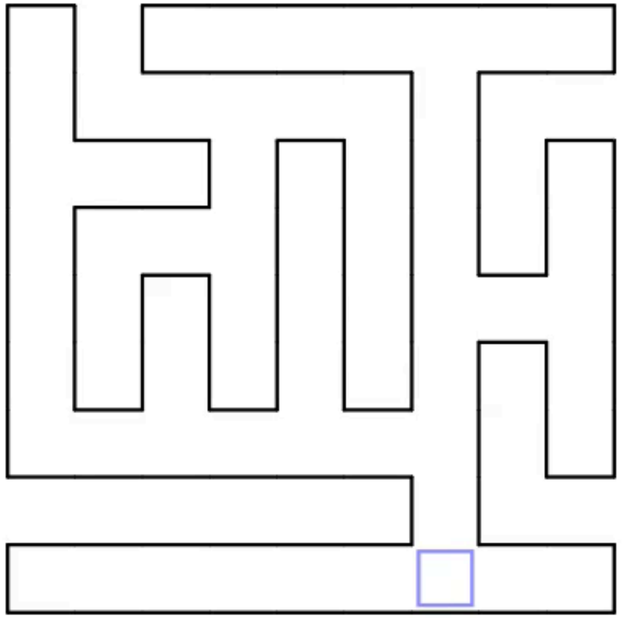
Simple Neural Attentive Meta-Learner

Mishra et al, ICLR '18

Experiment: Learning to visually navigate a maze

- train on 1000 small mazes
- test on held-out small mazes and large mazes



Simple Neural Attentive Meta-Learner

Mishra et al, ICLR '18

Experiment: Learning to visually navigate a maze

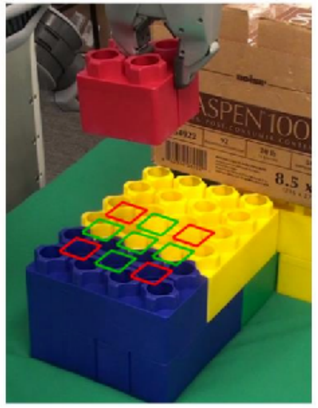
- train on 1000 small mazes
- test on held-out small mazes and large mazes

Method	Small Maze		Large Maze	
	Episode 1	Episode 2	Episode 1	Episode 2
Random	188.6 ± 3.5	187.7 ± 3.5	420.2 ± 1.2	420.8 ± 1.2
LSTM	52.4 ± 1.3	39.1 ± 0.9	180.1 ± 6.0	150.6 ± 5.9
SNAIL (ours)	$\mathbf{50.3 \pm 0.3}$	$\mathbf{34.8 \pm 0.2}$	$\mathbf{140.5 \pm 4.2}$	$\mathbf{105.9 \pm 2.4}$

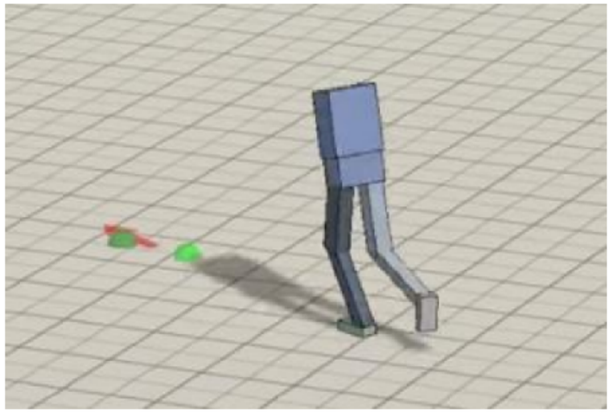
Table 5: Average time to find the goal on each episode

Digression: Connection to Contextual Policies

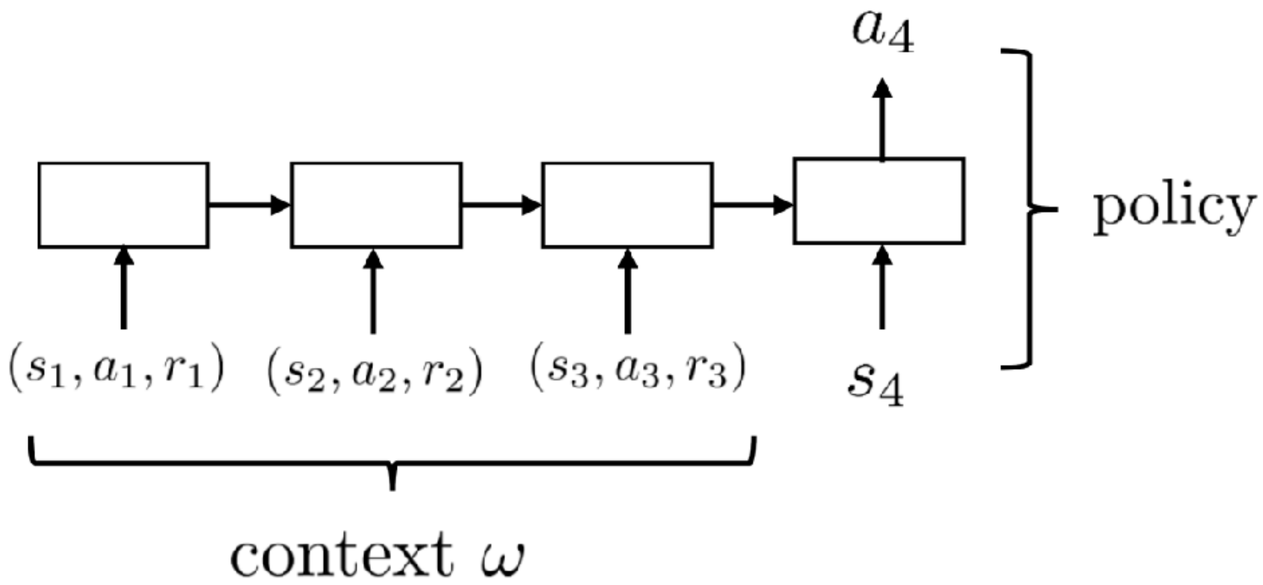
contextual policy: $\pi_\theta(\mathbf{a}|\mathbf{s}, \omega)$



ω: stack location



ω: walking direction



Contextual policy with experience as context.

What about **goal-conditioned policies** / **value functions**?

- rewards are a strict generalization of goals
- meta-RL objective is to ***adapt*** new tasks vs. ***generalize*** to new goals

(**k-shot** vs. **0-shot**)

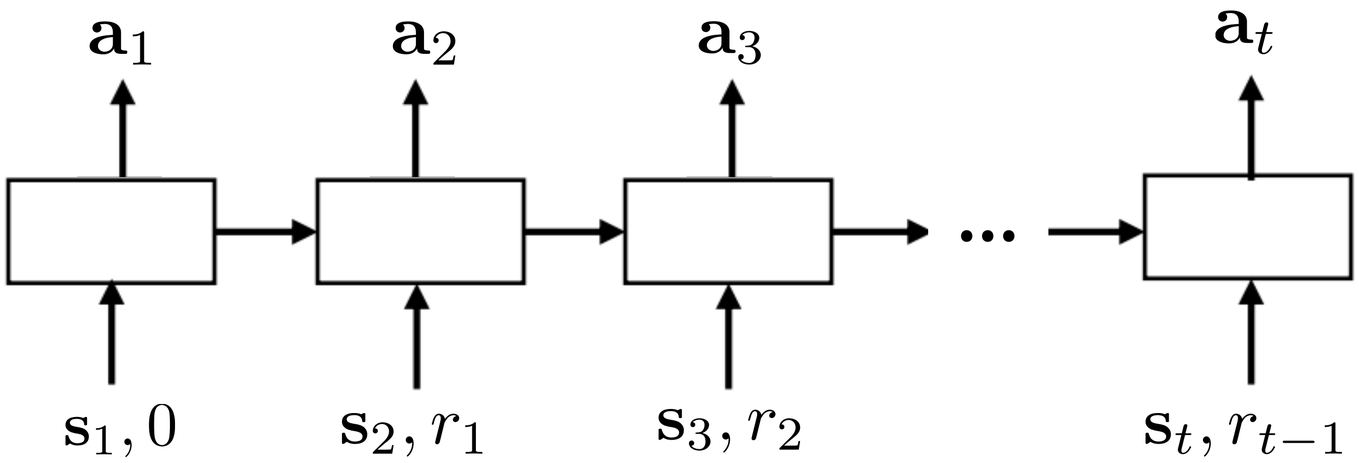
Design of *f* ?

$\mathcal{D}_{\text{train}}\ \mathbf{x}_{\text{test}} \longrightarrow \mathbf{y}_{\text{test}}$

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, …



+ general & expressive
+ a variety of design choices in architecture

-- complex model for complex task of learning
-- impractical data requirements

pretrained parameters

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

[test-time]

training data for new task

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

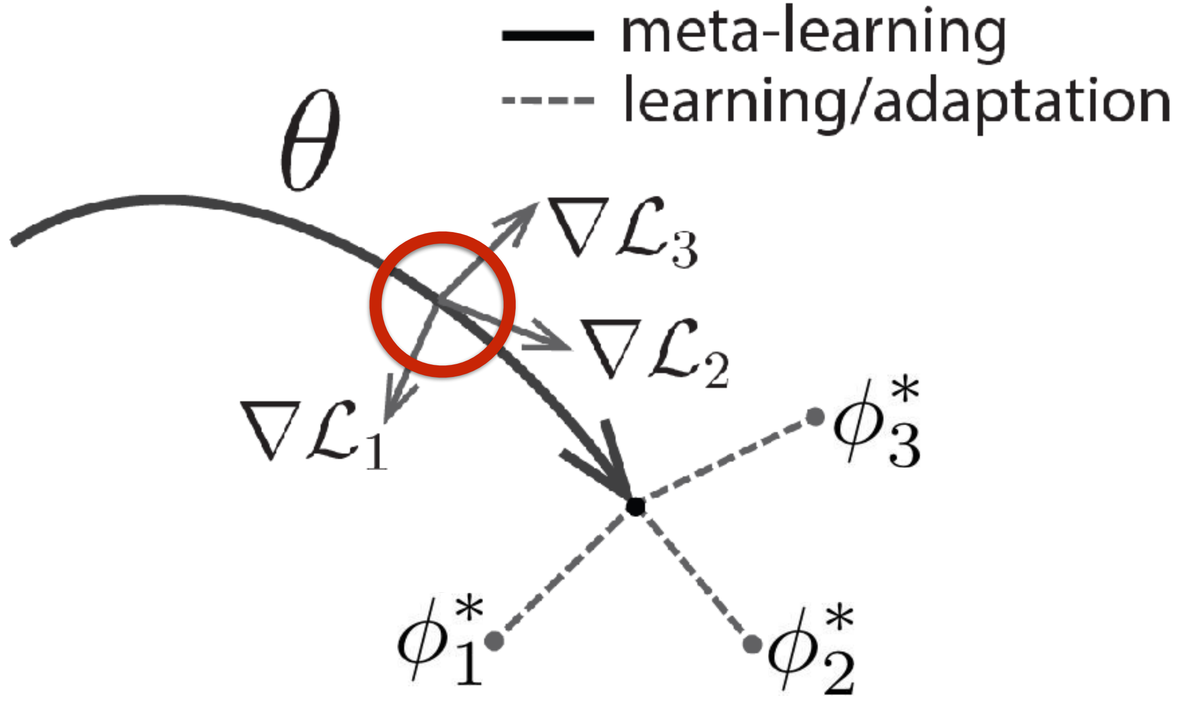
Key idea: Train over many tasks, to learn parameter vector θ that transfers

Finn, Abbeel, Levine. *Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks*. ICML 2017

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

θ parameter vector being meta-learned

ϕ_i^* optimal parameter vector for task i

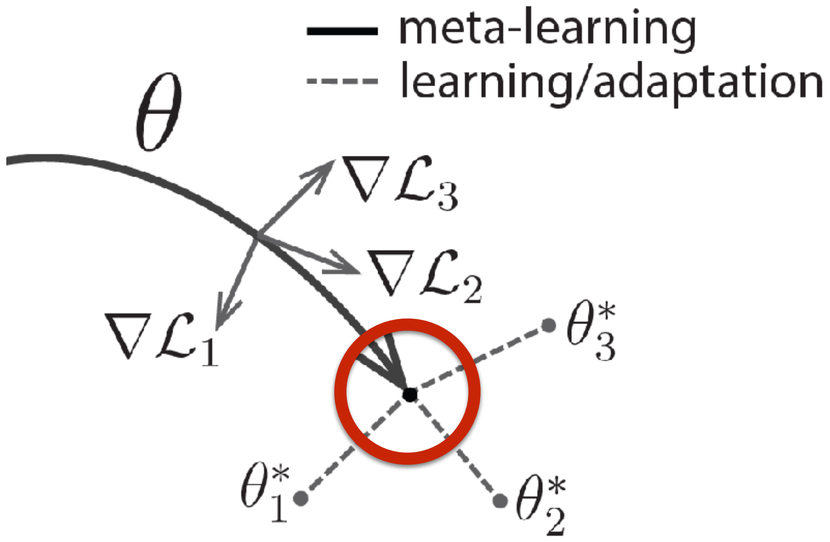


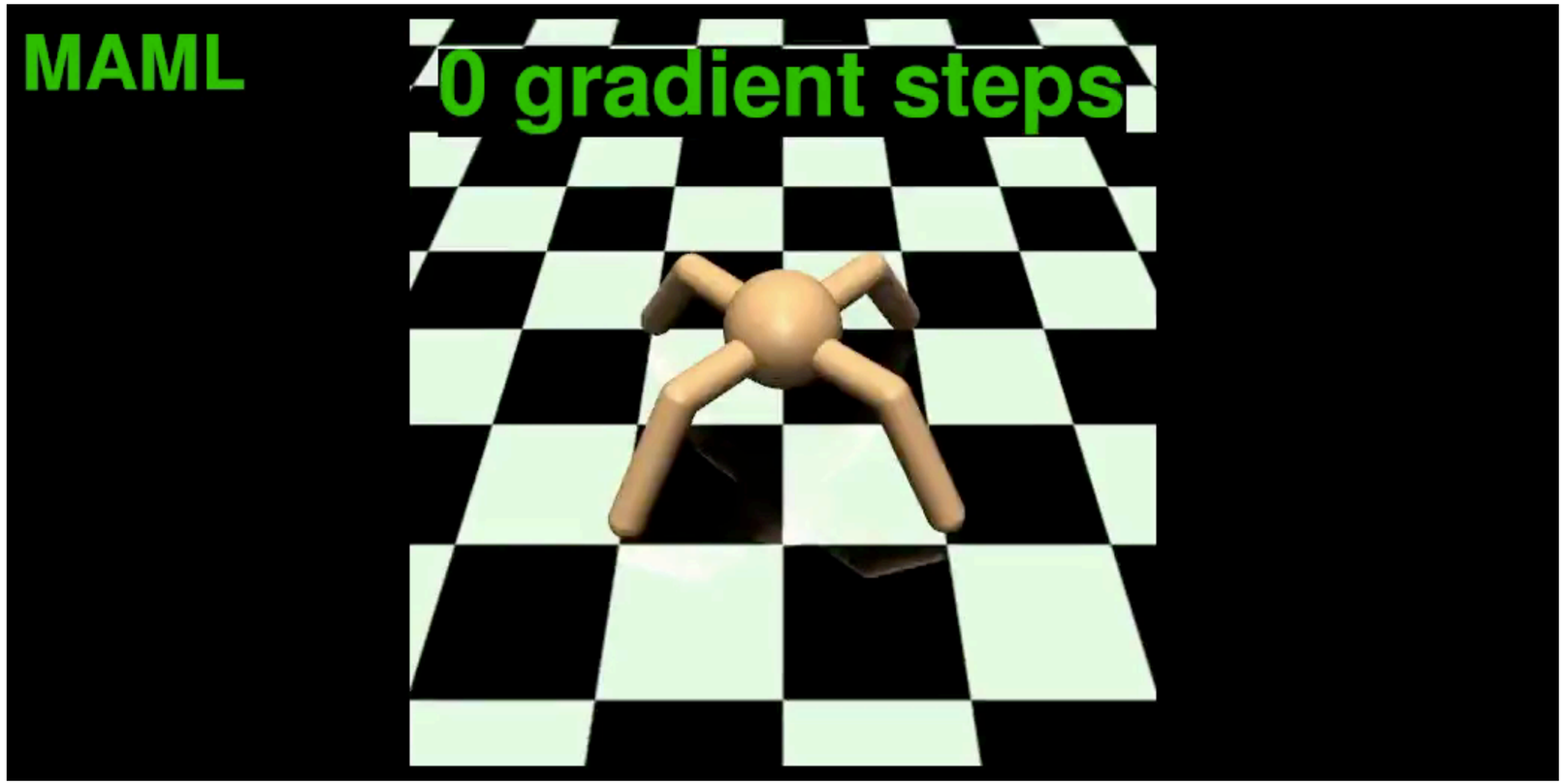
Model-**A**gnostic **M**eta-**L**earning

Can we learn a representation under which RL is fast?

Finn, Abbeel, Levine. *Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks.* ICML 2017

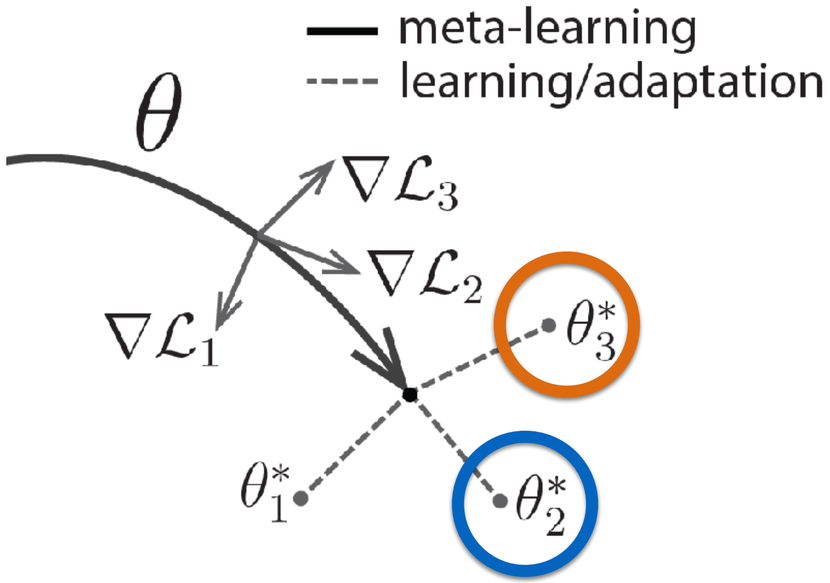
Fast Adaptation in Reinforcement Learning

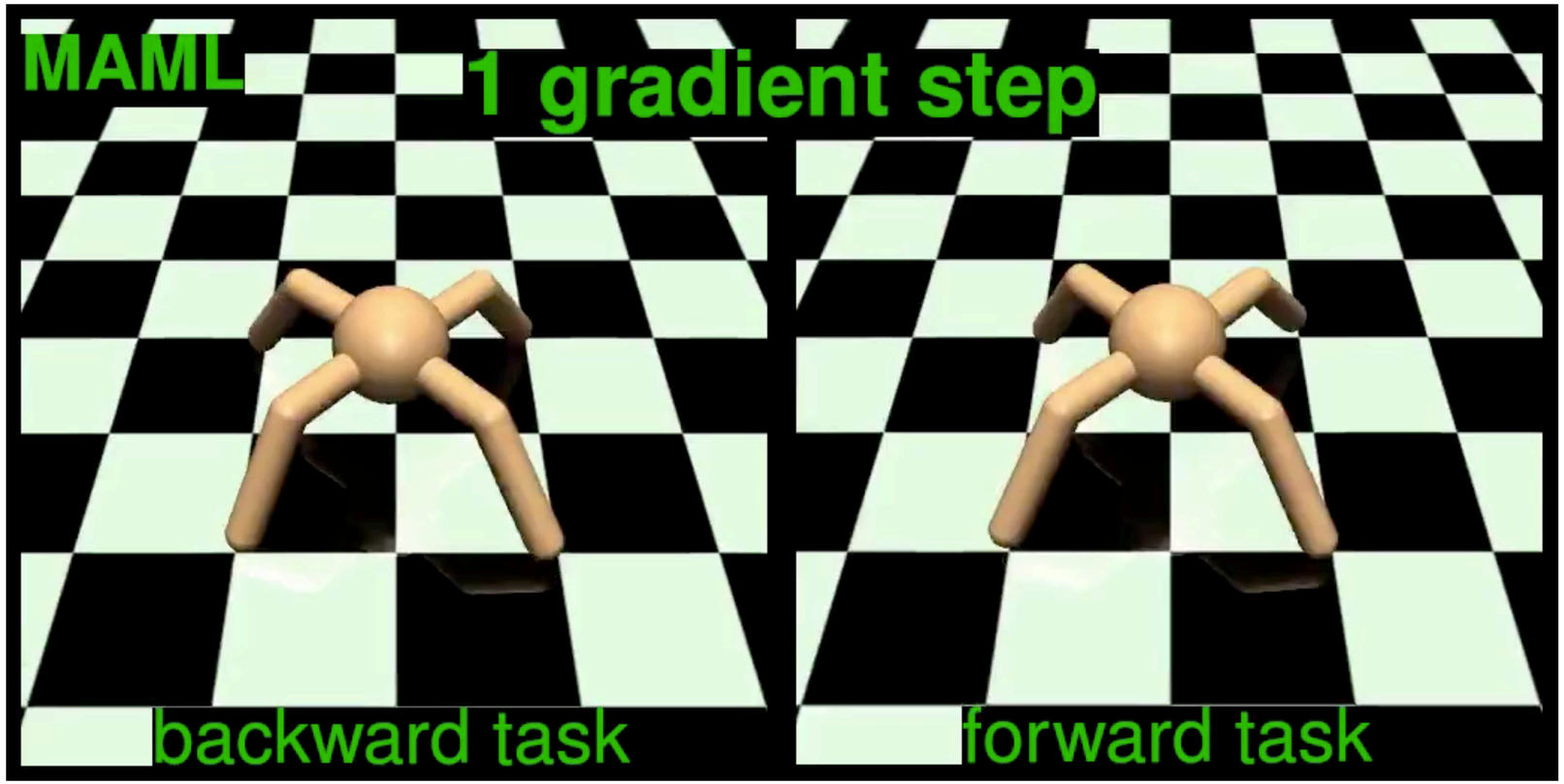




Finn, Abbeel, Levine. *Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks.* ICML 2017

Fast Adaptation in Reinforcement Learning





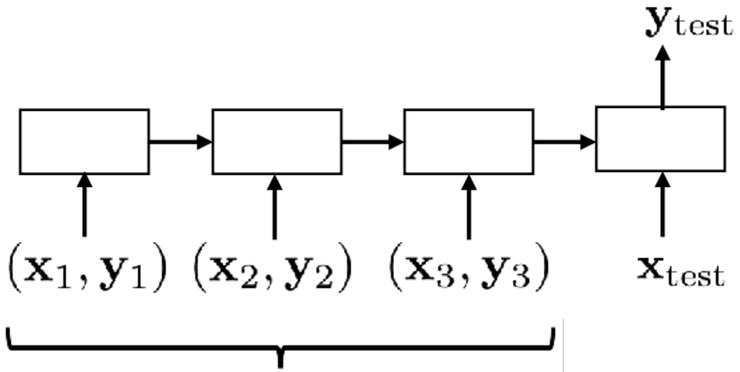
Finn, Abbeel, Levine. *Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks.* ICML 2017

Design of *f* ?

$\mathcal{D}_{\text{train}}$ $\mathbf{x}_{\text{test}}$ ⟶ $\mathbf{y}_{\text{test}}$

Recurrent network

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



network implements the "learned learning procedure"

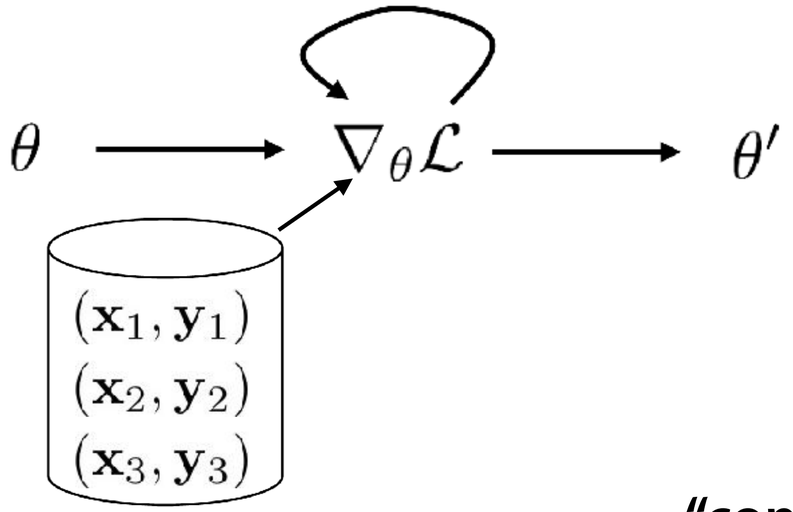
Does it converge?
- Sort of?

What does it converge to?
- Who knows…

What to do if not good enough?
- Nothing

MAML

$$\mathbf{y}_{\text{test}} = f(\mathbf{x}_{\text{test}}; \theta - \alpha \nabla_\theta \mathcal{L}(\mathcal{D}_{\text{train}}))$$



"consistency"

Does it converge?
- Yes (it's gradient descent…)

What does it converge to?
- A local optimum (it's gradient descent…)

What to do if not good enough?
- Keep taking gradient steps (it's gradient descent..)

Does this structure come at a cost?

Outline

Meta-RL Problem Formulation & Examples

Method Classes: Recurrent Models, Gradient-Based Models

Challenges & Latest Developments

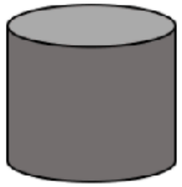
Wishlist in a Meta-RL algorithm

	RL^2, SNAIL	MAML-PG
Consistent	✗	✓
Expressive	✓	✗
Structured Exploration*	~	~
Efficient & off-policy	✗	✗

***Initial work in this direction:** Gupta et al. NIPS '18, Stadie et al. NIPS '18

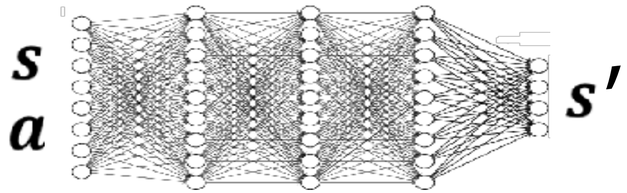
Model-based RL

Collect data



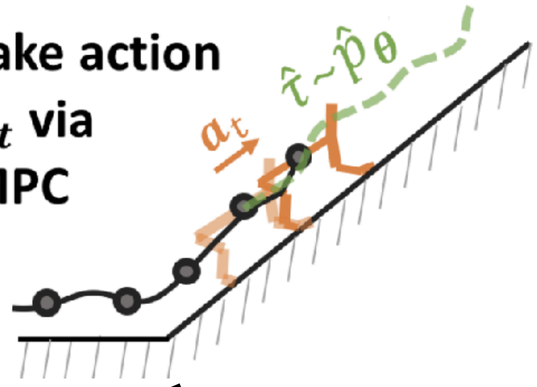
$\{s_t, a_t\}$

Fit model



$\hat{p}_\theta$

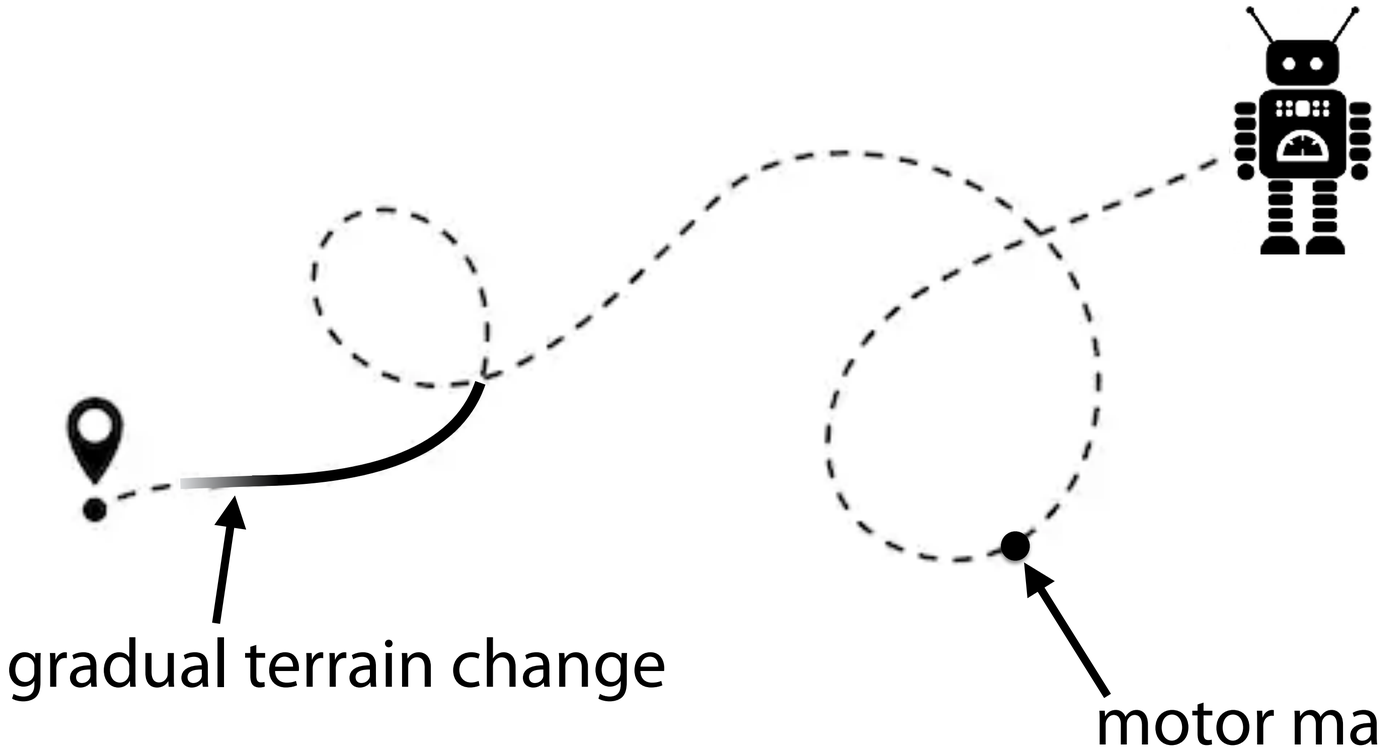
Plan using the model



Model-based RL already achieves **zero-shot adaptation** to **new rewards**.

Goal: learn to adapt model quickly to new environments (online system ID)

Goal: learn to adapt model quickly to new environments



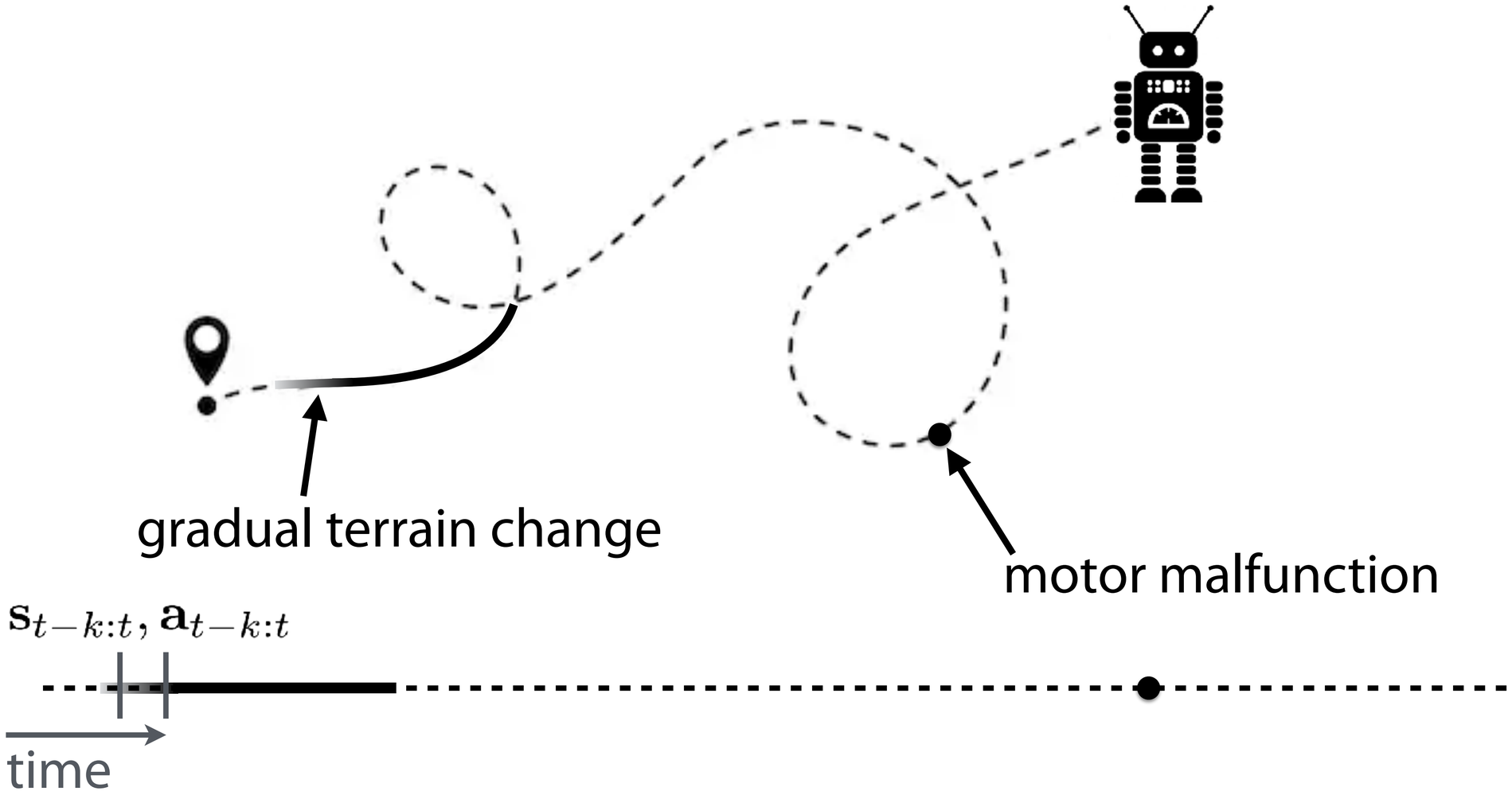

Anusha Nagabandi


Ignasi Clavera


Simin Liu

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

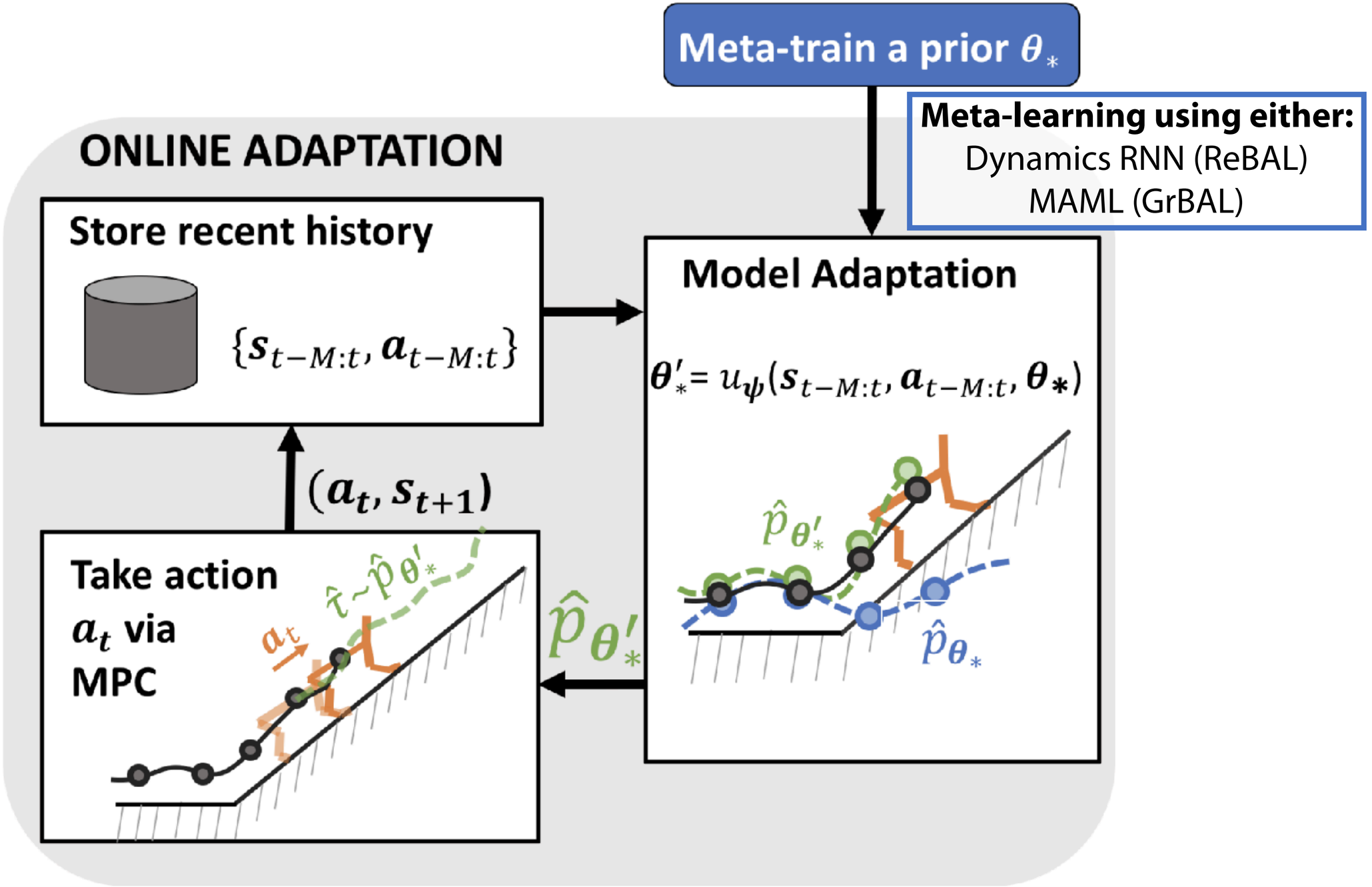
Goal: learn to adapt model quickly to new environments



online adaptation = few-shot learning **tasks** are **temporal slices** of experience

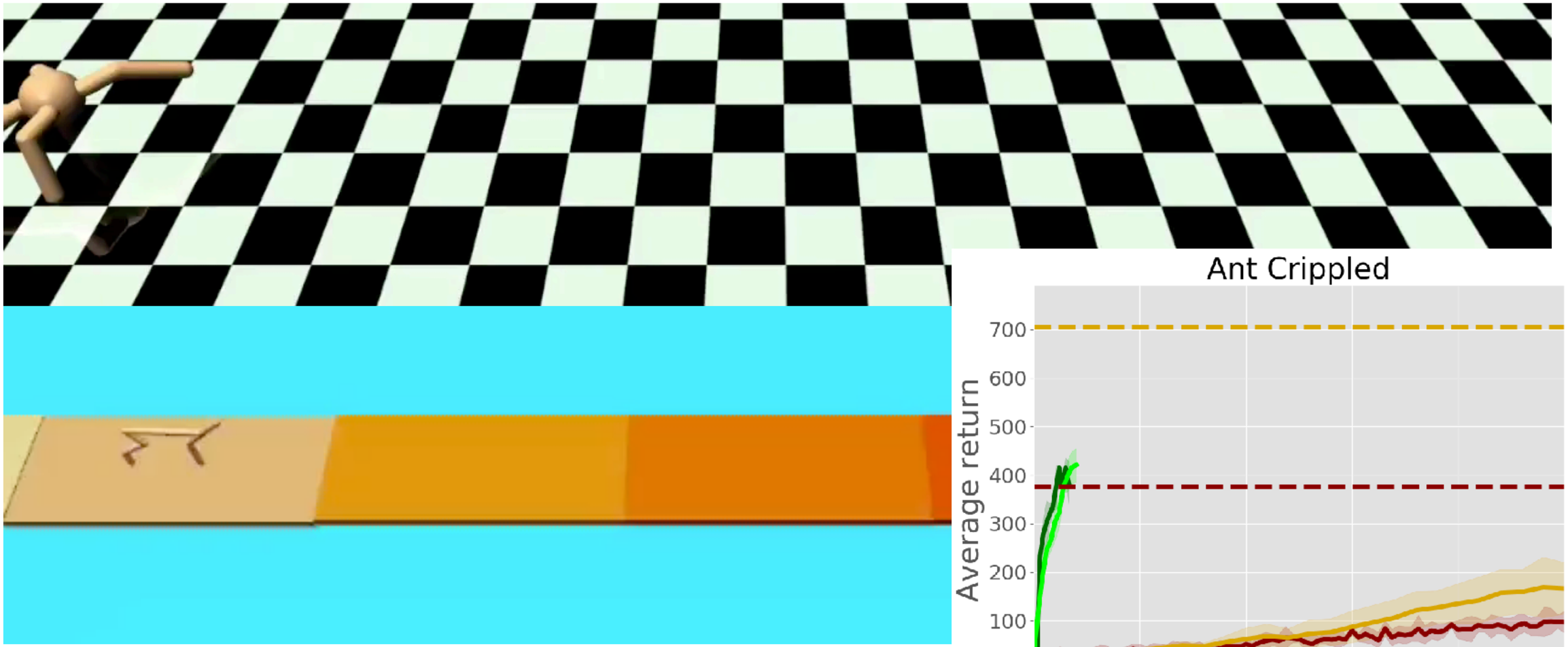
The task distribution is more-or-less free!

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



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

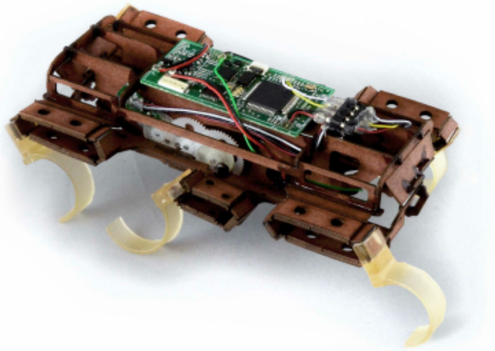
Dynamic Environments with Online Adaptation



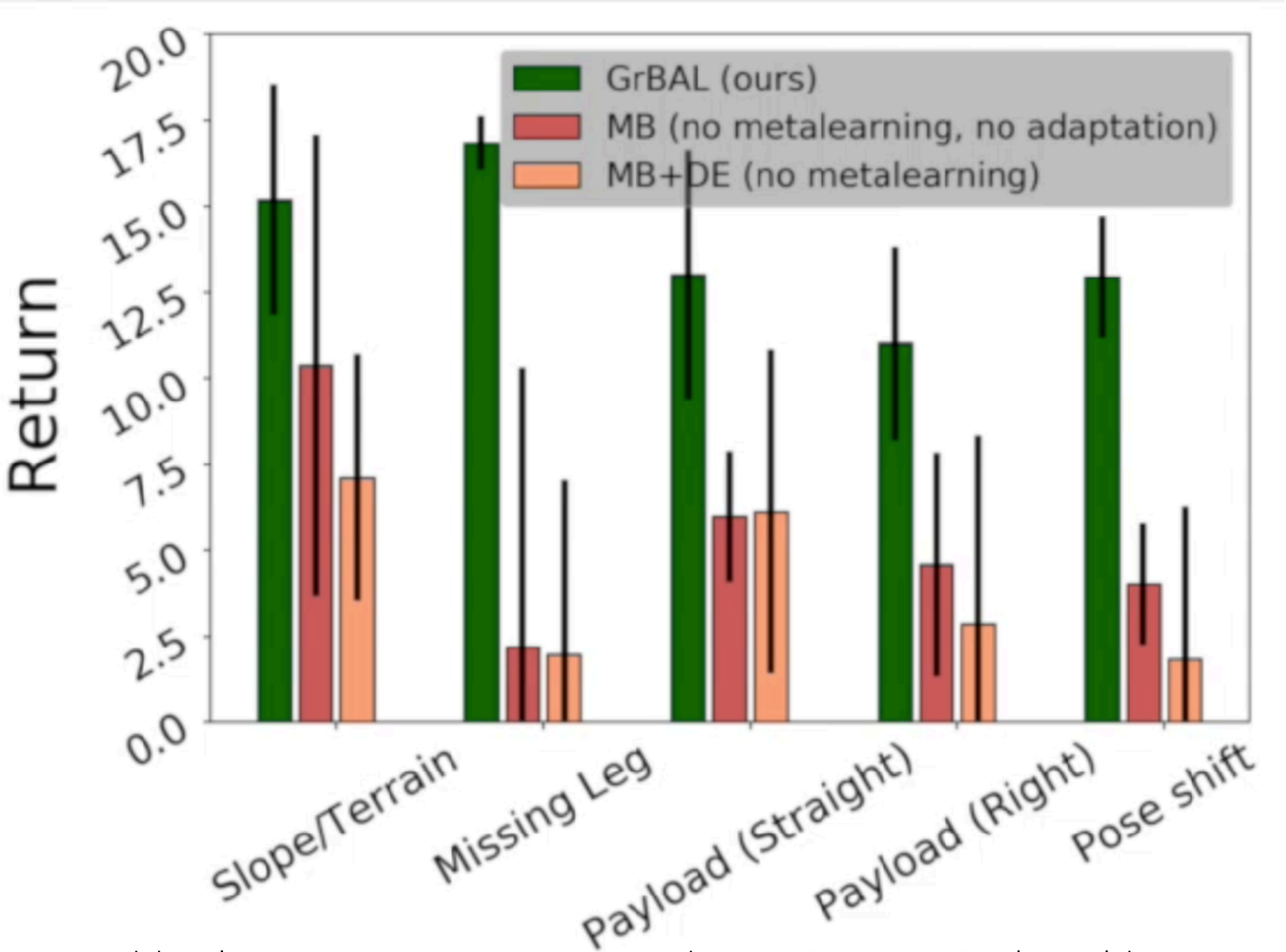
Nagabandi*, Clavera*, Liu, Fearing, Abbeel, Levine, Finn. Learning to Adapt in Dynamic Real-World Environments through Meta-RL

Roach Robot

Meta-train on **variable terrains**



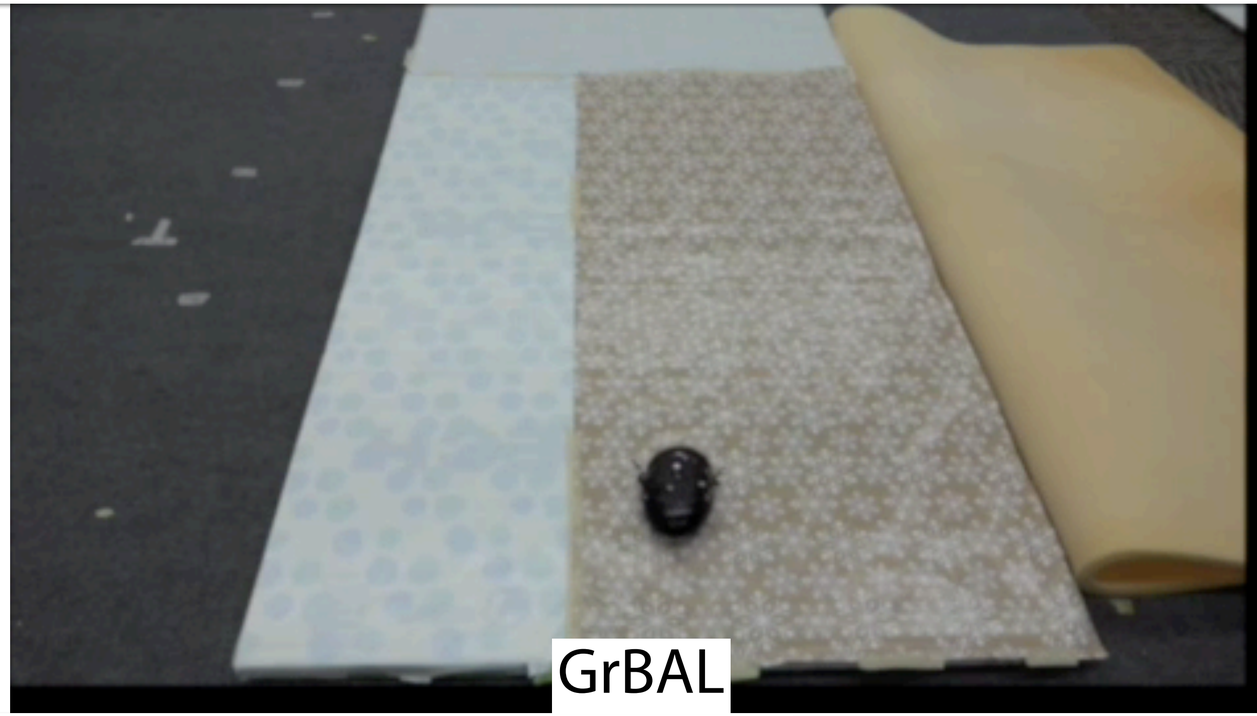
Meta-test with **slope, missing leg, payload, calibration errors**



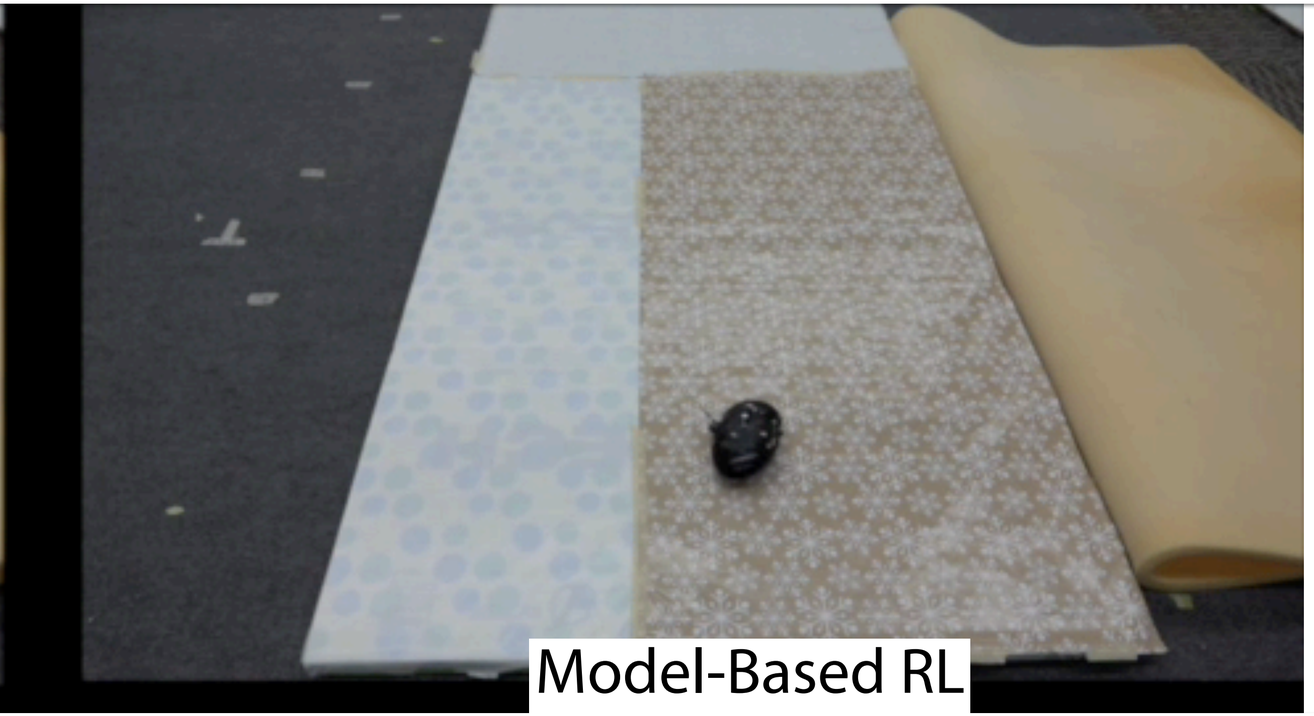
Nagabandi*, Clavera*, Liu, Fearing, Abbeel, Levine, Finn. Learning to Adapt in Dynamic Real-World Environments through Meta-RL

Roach Robot

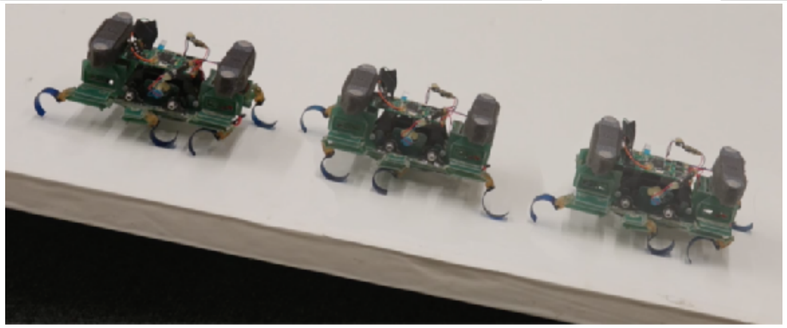
Meta-train on **variable terrains** **Meta-test** with **slope**, **missing leg, payload, calibration errors**



GrBAL



Model-Based RL
(no adaptation)

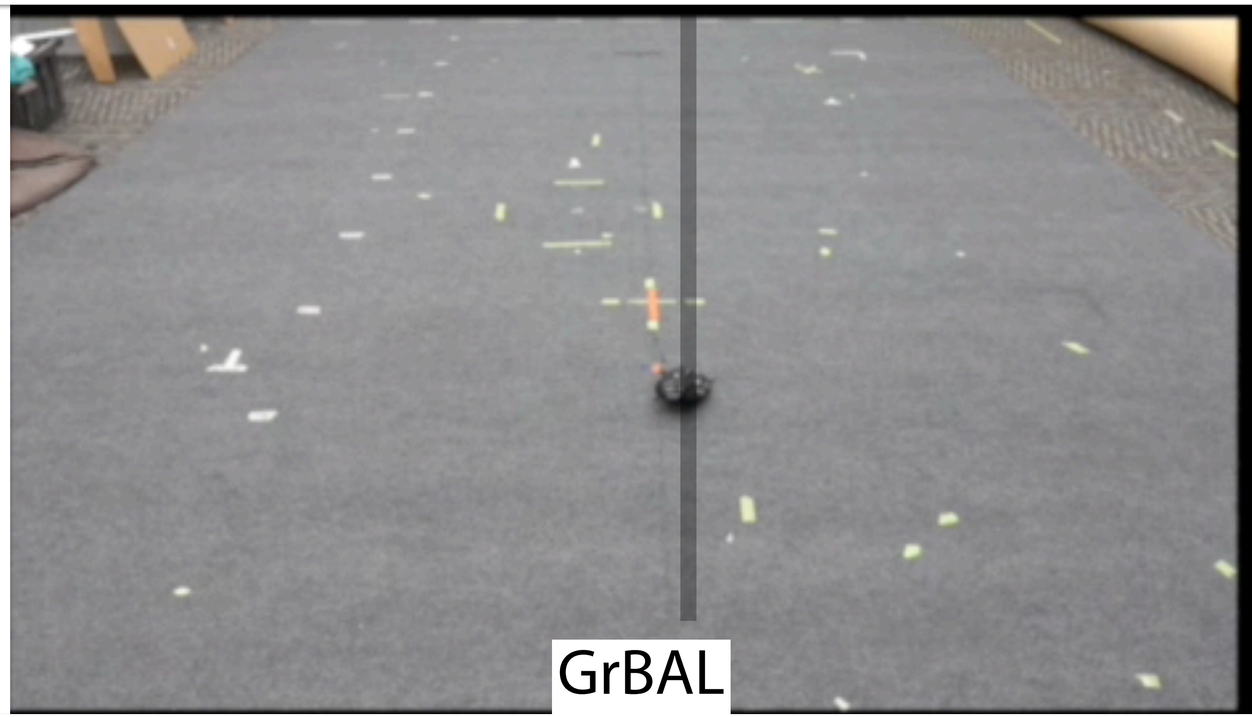


Nagabandi*, Clavera*, Liu, Fearing, Abbeel, Levine, Finn. Learning to Adapt in Dynamic Real-World Environments through Meta-RL

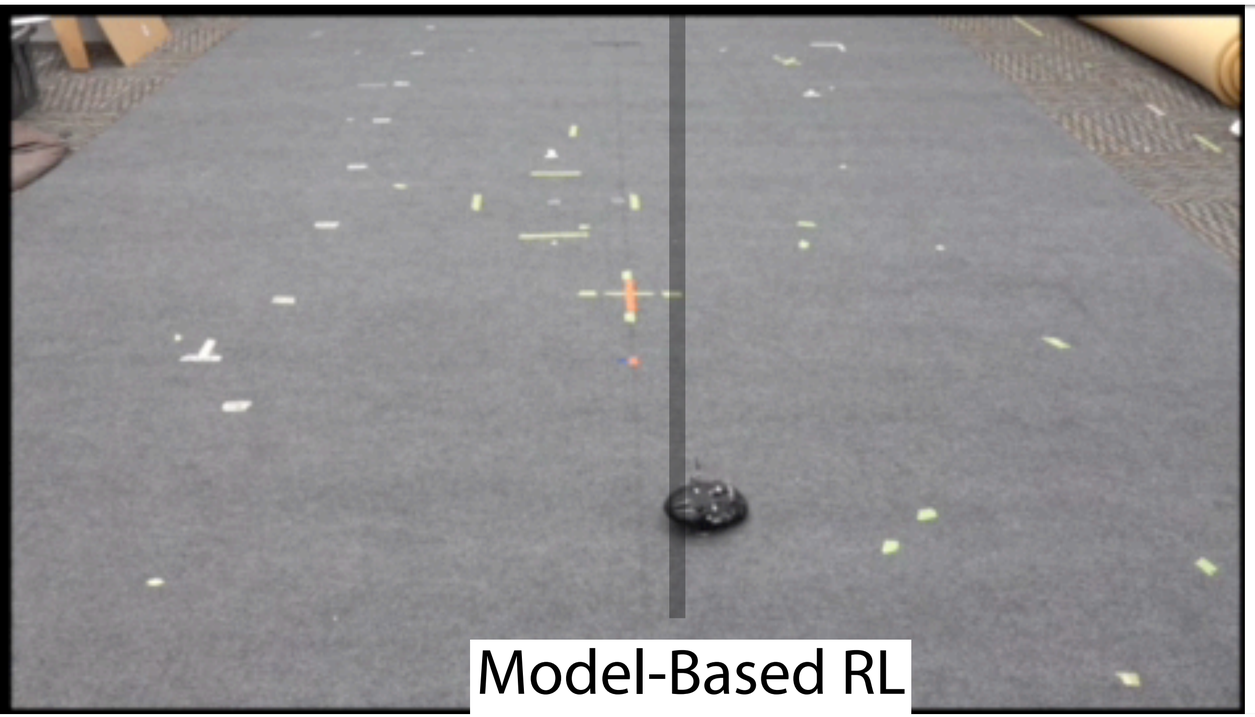
Roach Robot

Meta-train on **variable terrains**

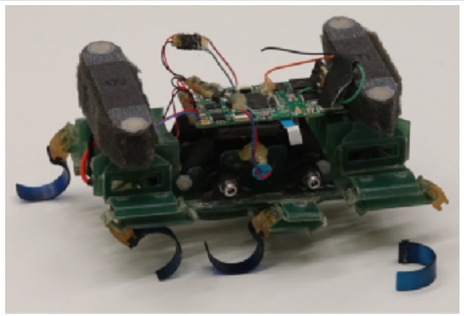
Meta-test with **slope, missing leg, payload, calibration errors**



GrBAL



Model-Based RL
(no adaptation)



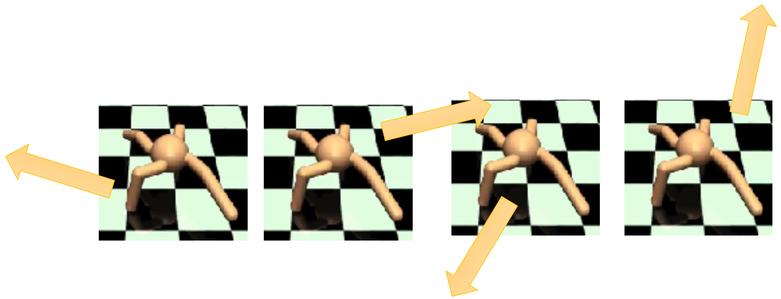
Nagabandi*, Clavera*, Liu, Fearing, Abbeel, Levine, Finn. Learning to Adapt in Dynamic Real-World Environments through Meta-RL

Wishlist in a Meta-RL algorithm

	RL^2, SNAIL	MAML-PG	ReBAL, GrBAL
Consistent	✗	✓	✓
Expressive	✓	✗	~
Structured Exploration	~	~	✗
Efficient & off-policy	✗	✗	✓

Still ***lots*** of room for exploration & improvement!

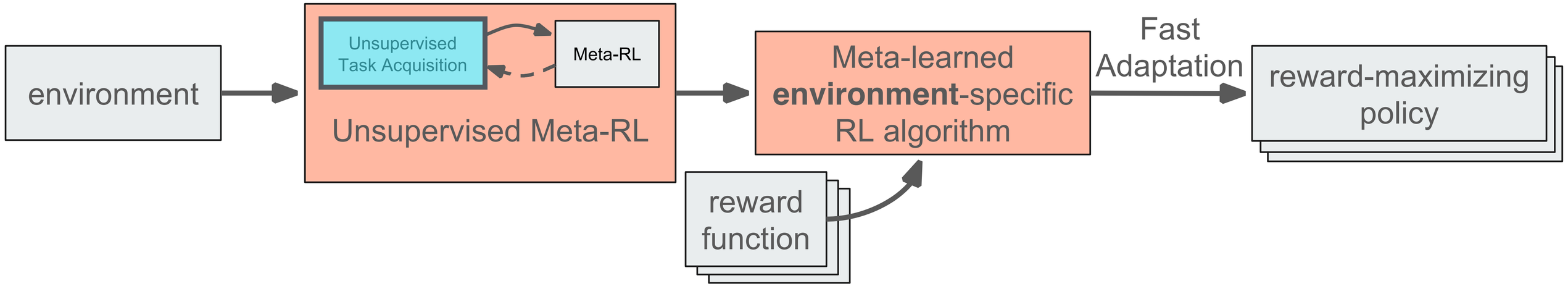
One more important challenge: **where do the tasks come from?**



So far: manual defined distribution of tasks.
(and corresponding rewards)

Unsupervised Meta-RL

General Recipe




Abhishek Gupta


Ben Eysenbach


Sergey Levine

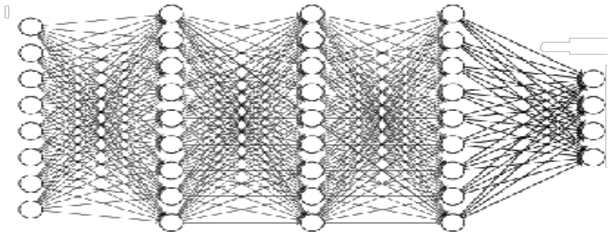
Gupta, Eysenbach, Finn, Levine. Unsupervised Meta-Learning for Reinforcement Learning.

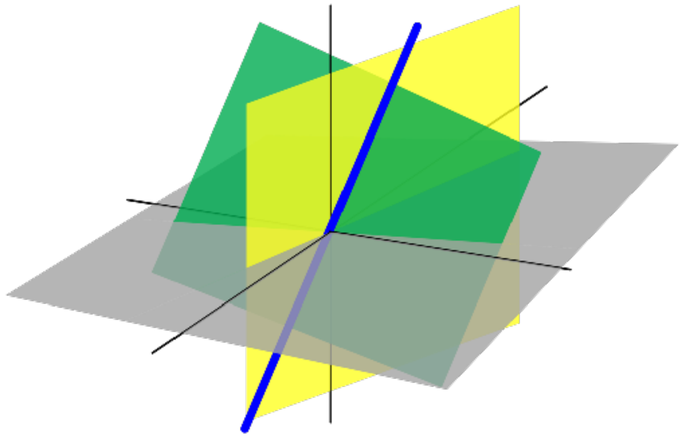
Random Task Proposals

- Use randomly initialize discriminators for reward functions

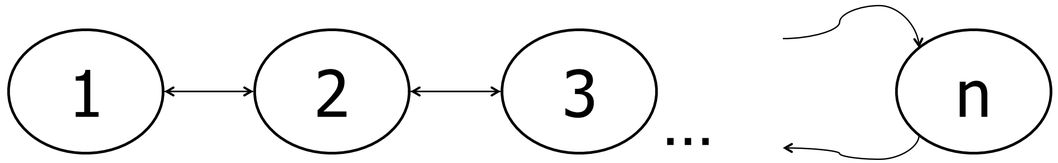
$$R(s, z) = \log p_D(z|s)$$

D → **randomly initialized network**



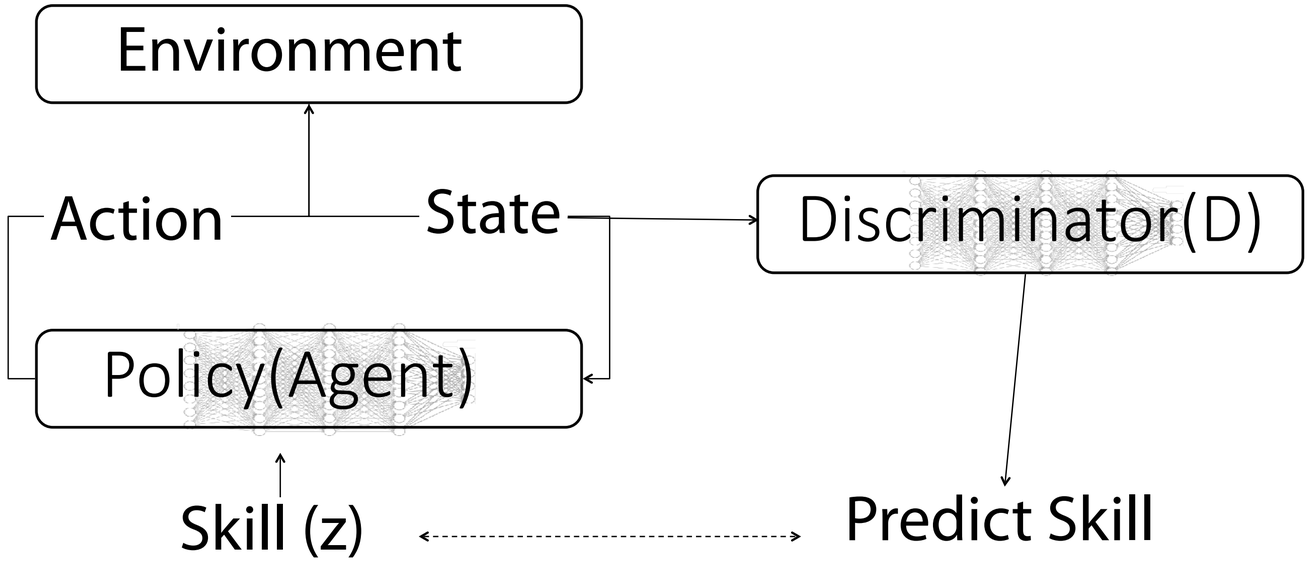


- Important: Random functions over state space, not random policies



Random policy – exponential
Random reward – polynomial

Diversity-Driven Proposals

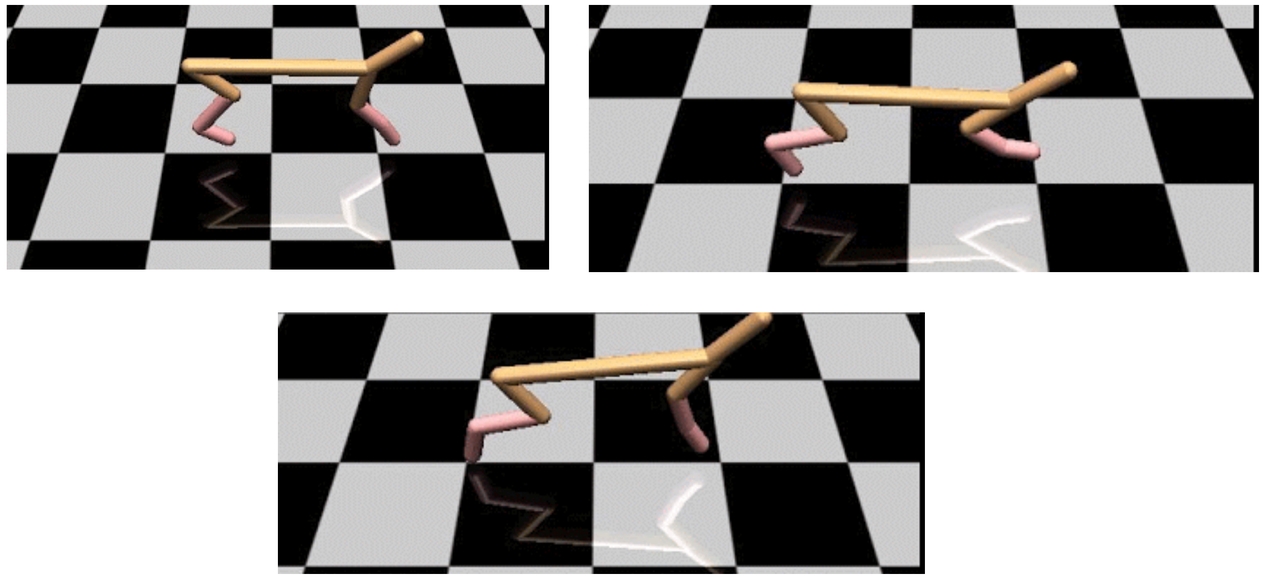


- Policy → visit states which are discriminable
- Discriminator → predict skill from state

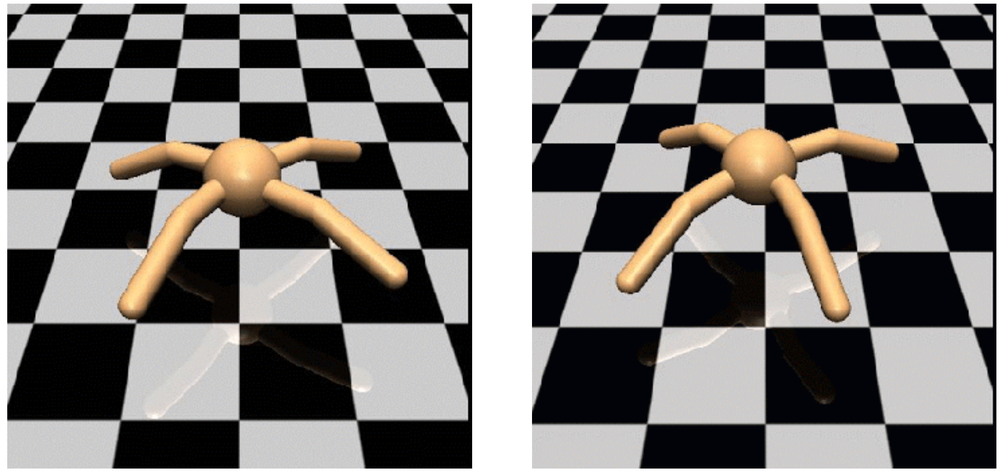
Task Reward for UML: $R(s, z) = \log p_D(z|s)$

Eysenbach, Gupta, Ibarz, Levine. Diversity is All You Need.

Examples of Acquired Tasks



Cheetah

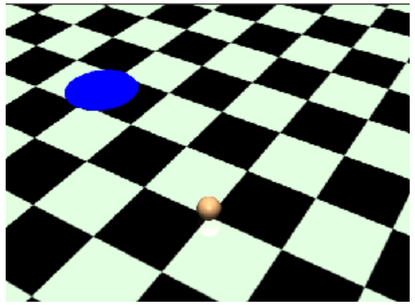


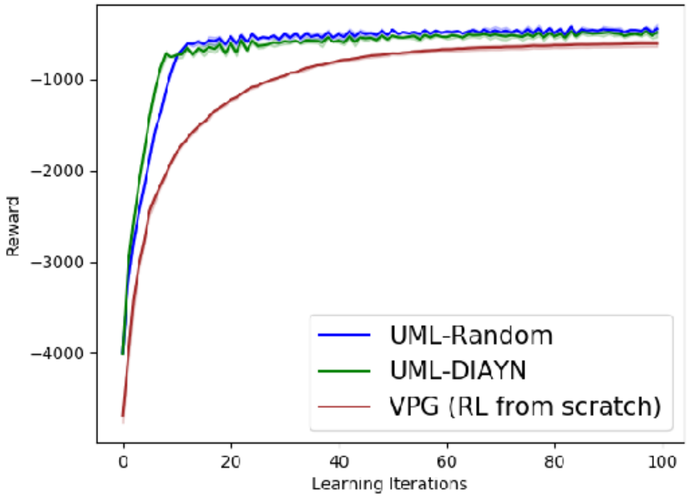
Ant

Eysenbach, Gupta, Ibarz, Levine. Diversity is All You Need.

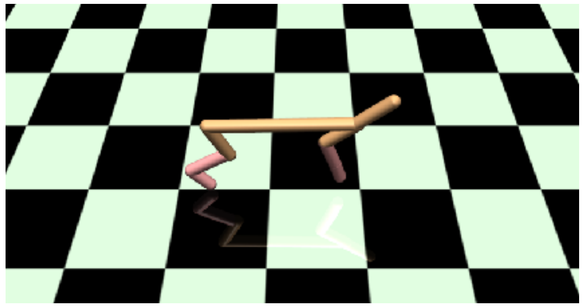
Does it work?

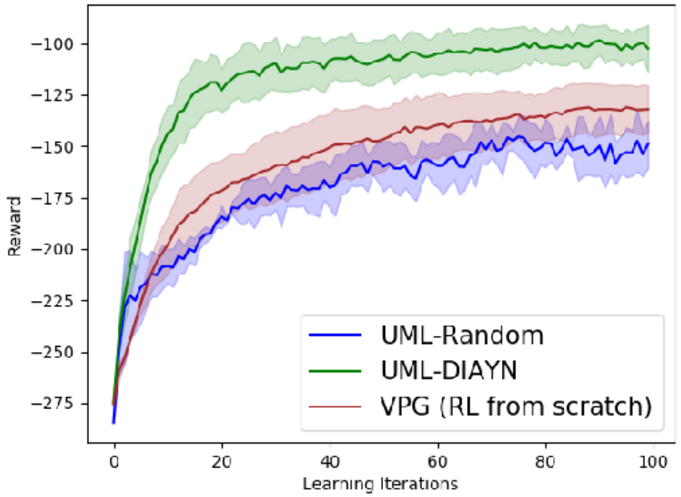
2D Navigation



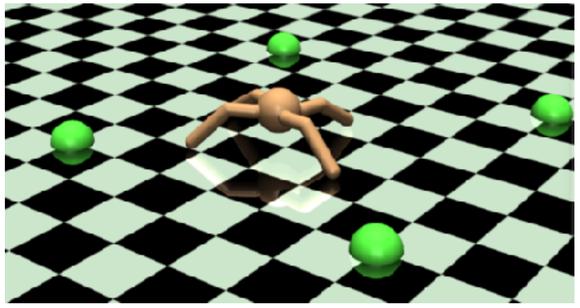


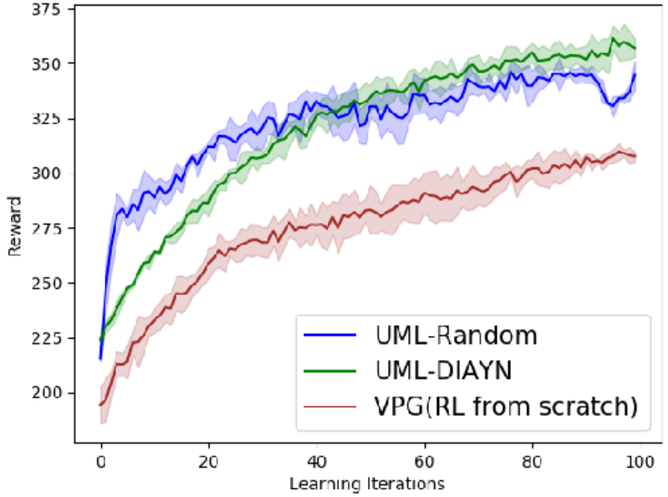
Cheetah





Ant





Meta-test performance with rewards

Takeaway: Relatively **simple** mechanisms for proposing tasks work surprisingly well.

Gupta, Eysenbach, Finn, Levine. Unsupervised Meta-Learning for Reinforcement Learning.

Key References

Recurrent Models: Yan Duan et al. '17 (RL^2), Jane Wang et al. CogSci '17 (Learning to Reinforcement Learn), Nikhil Mishra et al. ICLR '18 (SNAIL)
Model-Agnostic Meta-Learning: Finn et al. ICML '17

Further Reading

Value-based: Sung et al. arXiv '17 *(Meta-critic networks*)
Exploration: Gupta et al. NIPS '18, Stadie et al. NIPS '18
Unsupervised: Gupta et al. arXiv '18 (*Unsupervised Meta-Learning for Reinforcement Learning*)
Model-based: Nagabandi*, Clavera* et al. arXiv '18
Evolutionary Strategies: Houthooft et al. NIPS '18
Cognitive Science: Wang et al. Nature Neuroscience '18 (*PFC as a Meta-RL System*)

Takeaway

Don't reinforcement learn from scratch.

Meta-RL provides an approach to do this.

Open Problems

- a meta-RL algorithm that is both **consistent** and **expressive**
- meta-RL algorithms that learn to **explore in a structured way**
- constructing **task distributions** for meta-RL in an automated way
- meta-RL algorithms that can **incorporate off-policy data** effectively

Collaborators

Sergey Levine

Pieter Abbeel

Anusha Nagabandi

Ignasi Clavera

Simin Liu

Ron Fearing

Abhishek Gupta

Ben Eysenbach