

Meta Reinforcement Learning

Adaptable Models & Policies

CS 330

Reminders

Homework 3 out, due **Monday 10/26**.

Project milestone due **Monday 11/2**.

Mid-Quarter Check In

How are you doing?

<https://pollev.com/chelseafinn494>

Mid-Quarter Survey

Thank you!

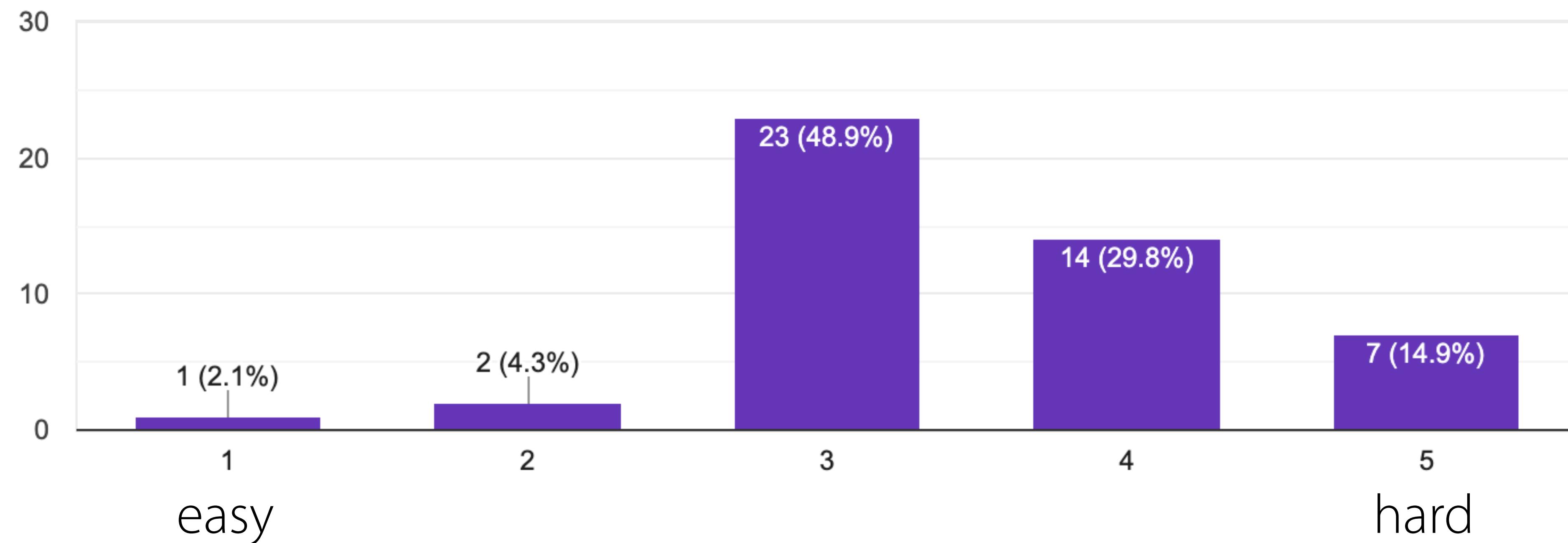
Mid-Quarter Survey

Some of the results

How is the difficulty of the class?



47 responses



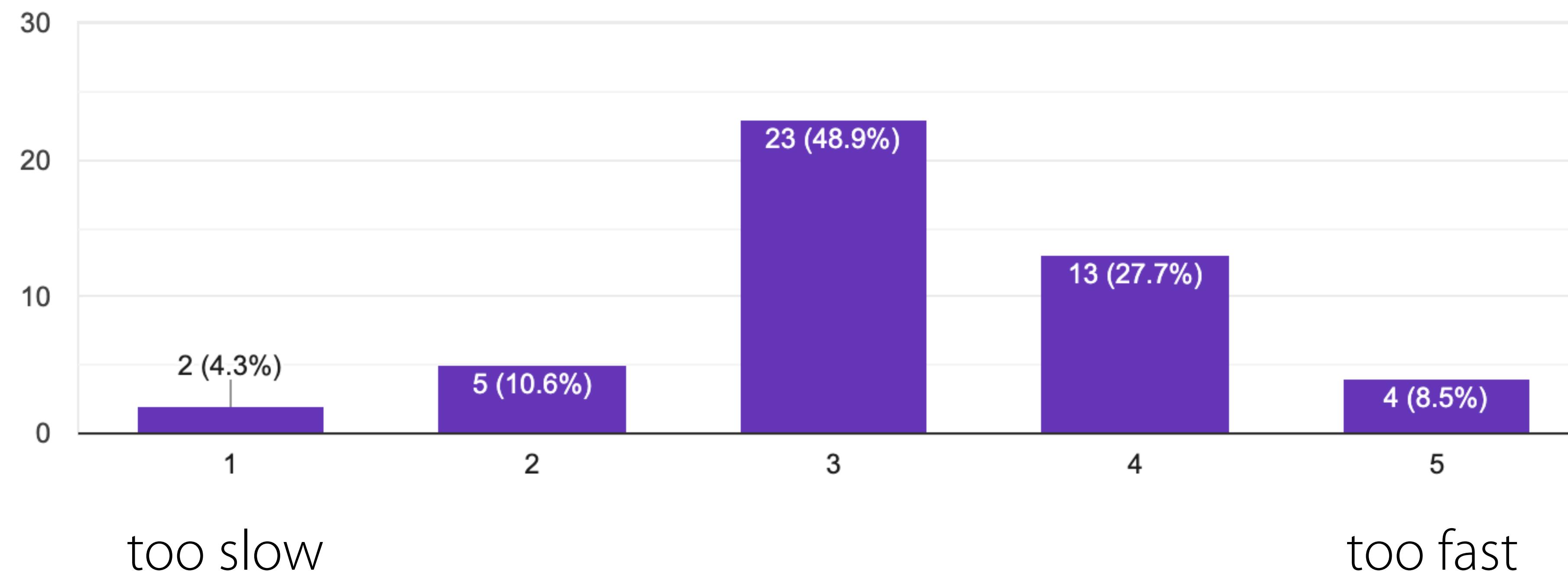
Mid-Quarter Survey

Some of the results

How do you feel about the lecture speed?



47 responses



Mid-Quarter Survey

Some of the comments mentioned multiple times.

Lecture feedback:

- Additional sections on background RL, variational inference concepts
- More concrete examples / instantiations of algorithms
- Reduce some gaps between lectures and assignments

Assignment feedback:

- Provide non-colab option
- Cleaner code with more comments
- PyTorch instead of TF
- Reduce # of iterations the algorithms need to be run

Office hours feedback:

- More office hours

Plan for Today

Meta-RL problem statement

Black-box meta-RL methods

<- comes up in HW4

Optimization-based meta-RL methods

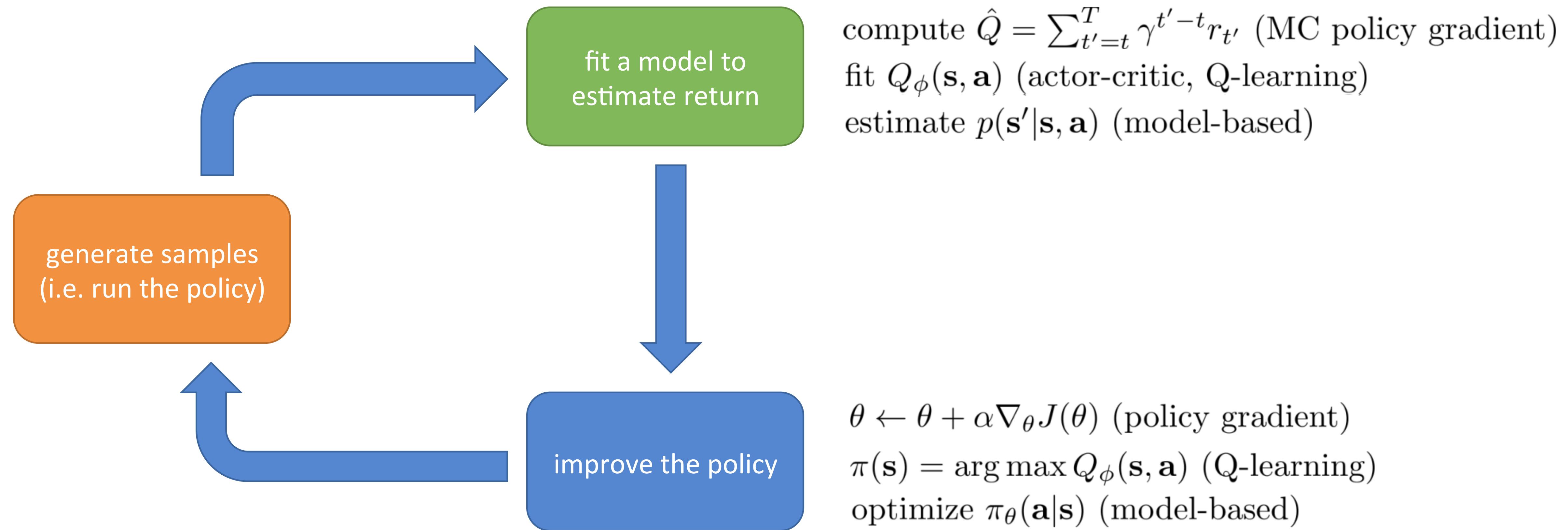
Next time: Learning to explore.

<- focus of HW4

- Lecture goals:
- Understand the **meta-RL problem statement** & set-up
 - Understand the basics of **black-box meta RL algorithms**
 - Understand the basics & challenges of **optimization-based meta RL algorithms**

Recap: The anatomy of a reinforcement learning algorithm

Last lectures: introduced model-free, model-based RL methods



Recall: Problem Settings

Multi-Task Learning

Solve multiple tasks $\mathcal{T}_1, \dots, \mathcal{T}_T$ at once.

$$\min_{\theta} \sum_{i=1}^T \mathcal{L}_i(\theta, \mathcal{D}_i)$$

Transfer Learning

Solve target task \mathcal{T}_b after solving source task \mathcal{T}_a
by *transferring* knowledge learned from \mathcal{T}_a

The Meta-Learning Problem

Given data from $\mathcal{T}_1, \dots, \mathcal{T}_n$, quickly solve new task $\mathcal{T}_{\text{test}}$

In all settings: tasks must share structure.

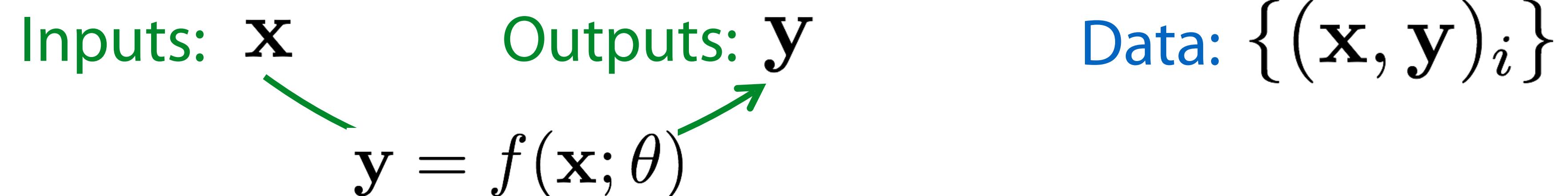
A reinforcement learning task:

The diagram illustrates a reinforcement learning environment model. At the top, two labels are positioned: "action space" on the left and "dynamics" on the right. Two downward-pointing arrows originate from these labels and point to the corresponding components in the equation below. The equation itself is $\mathcal{T}_i \triangleq \{\mathcal{S}_i, \mathcal{A}_i, p_i(s_1), p_i(s' | s, a), r_i(s, a)\}$. Below the equation, four labels are arranged horizontally: "state space" and "initial state distribution" under the first two terms, and "reward" under the last term. Three upward-pointing arrows originate from these labels and point to the corresponding components in the equation.

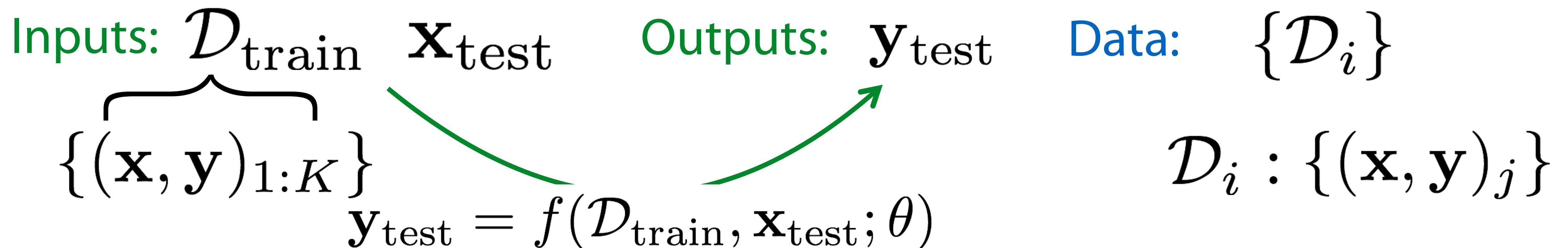
Meta-reinforcement learning
= **meta-learning** with RL tasks

Recall: The Meta-Learning Problem

Supervised Learning:



Meta Supervised Learning:

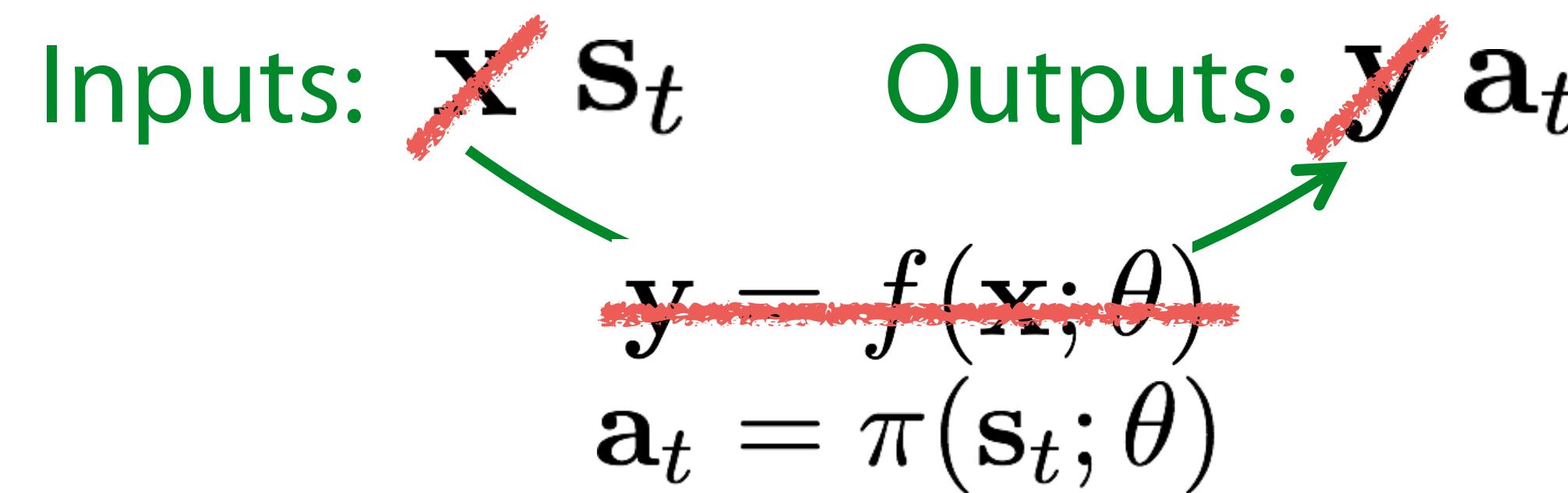


Why is this view useful?

Reduces the problem to the design & optimization of f .

The Meta Reinforcement Learning Problem

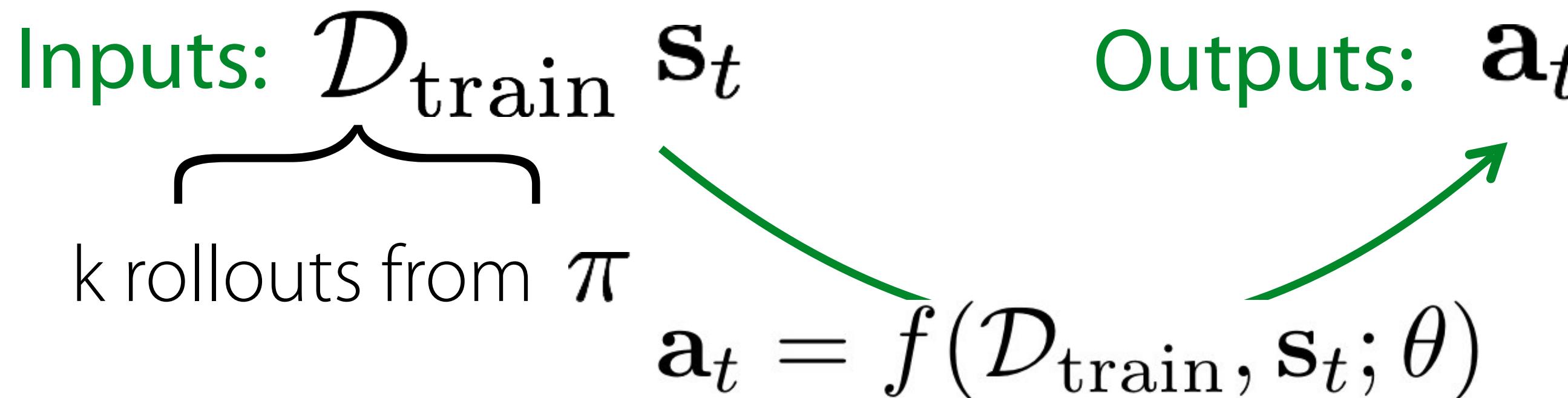
Reinforcement Learning:



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

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

Meta Reinforcement Learning:



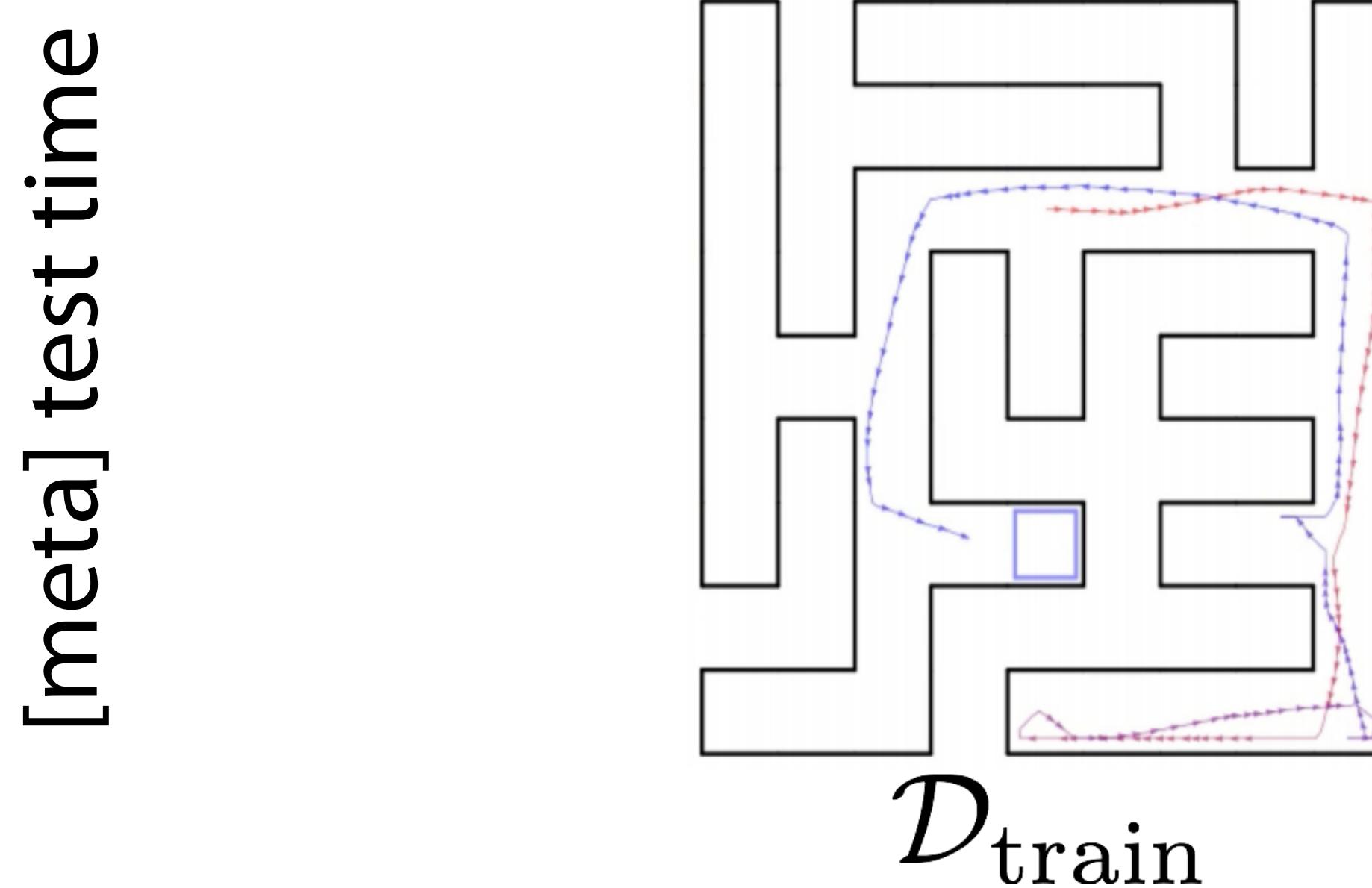
Data: $\{\mathcal{D}_i\}$

dataset of datasets
collected for each task

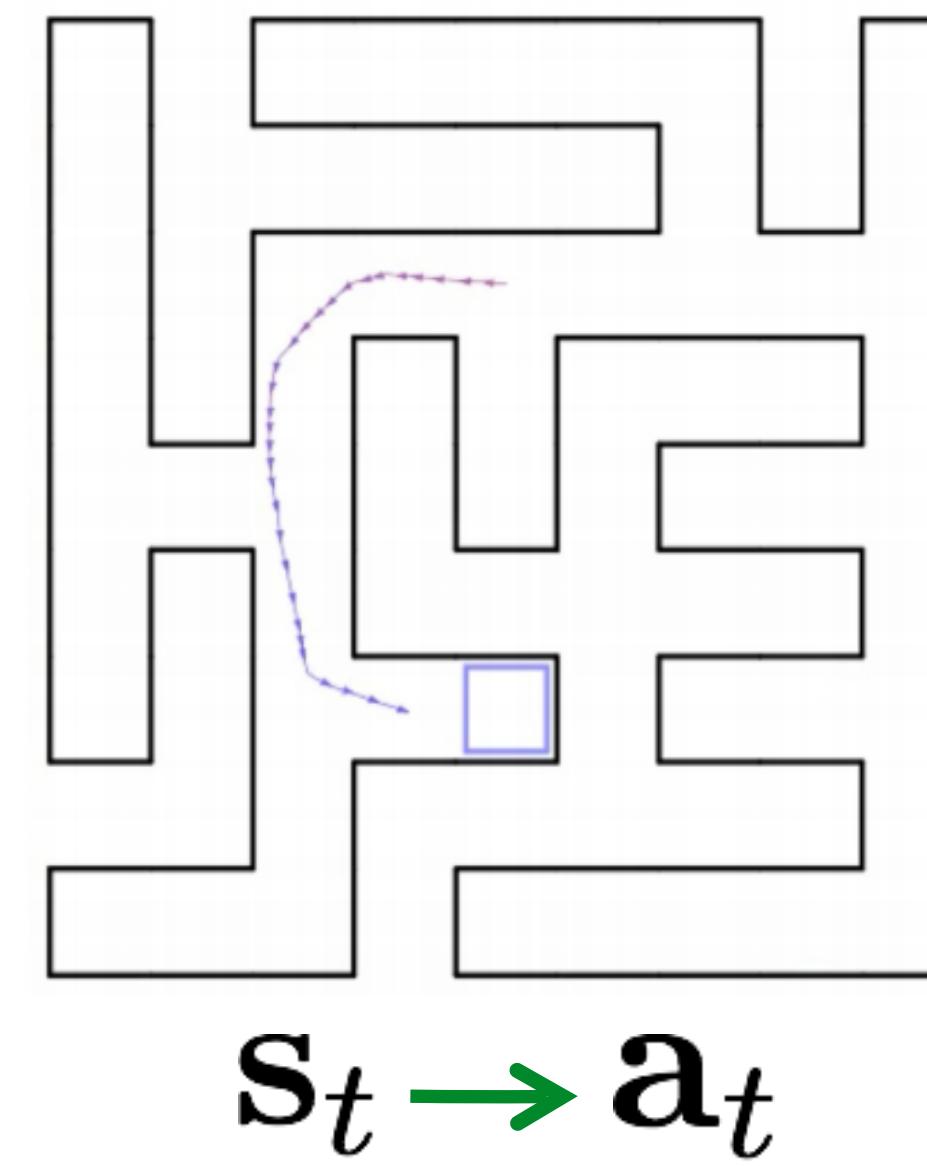
Design & optimization of f *and* collecting appropriate data
(learning to explore)

Meta-RL Example: Maze Navigation

Given a small amount of experience

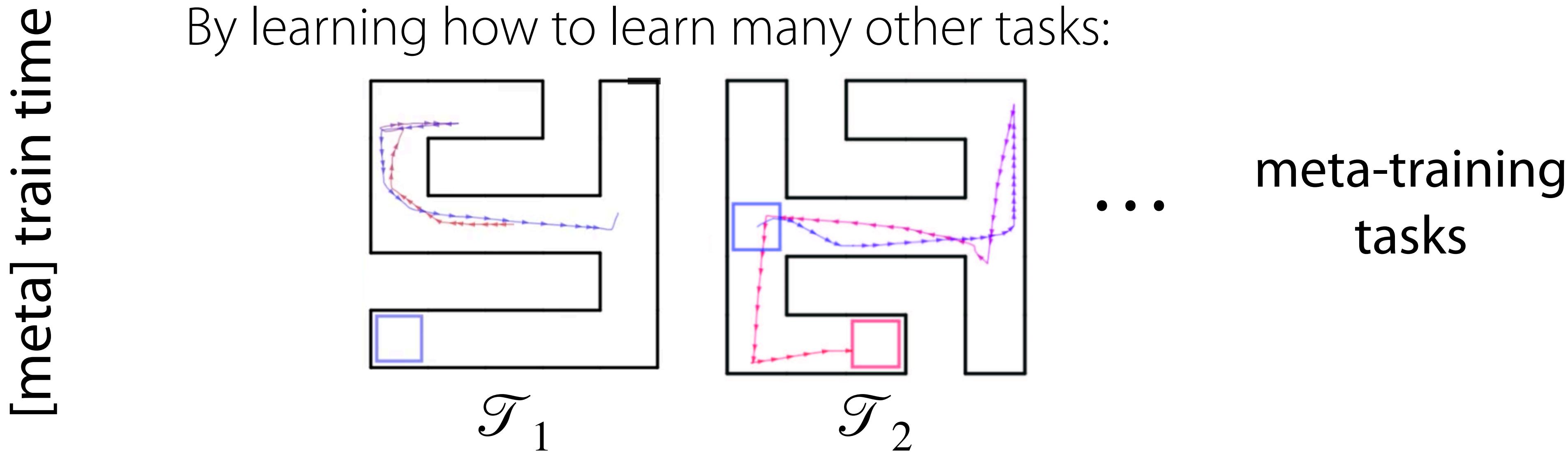


Learn to solve the task

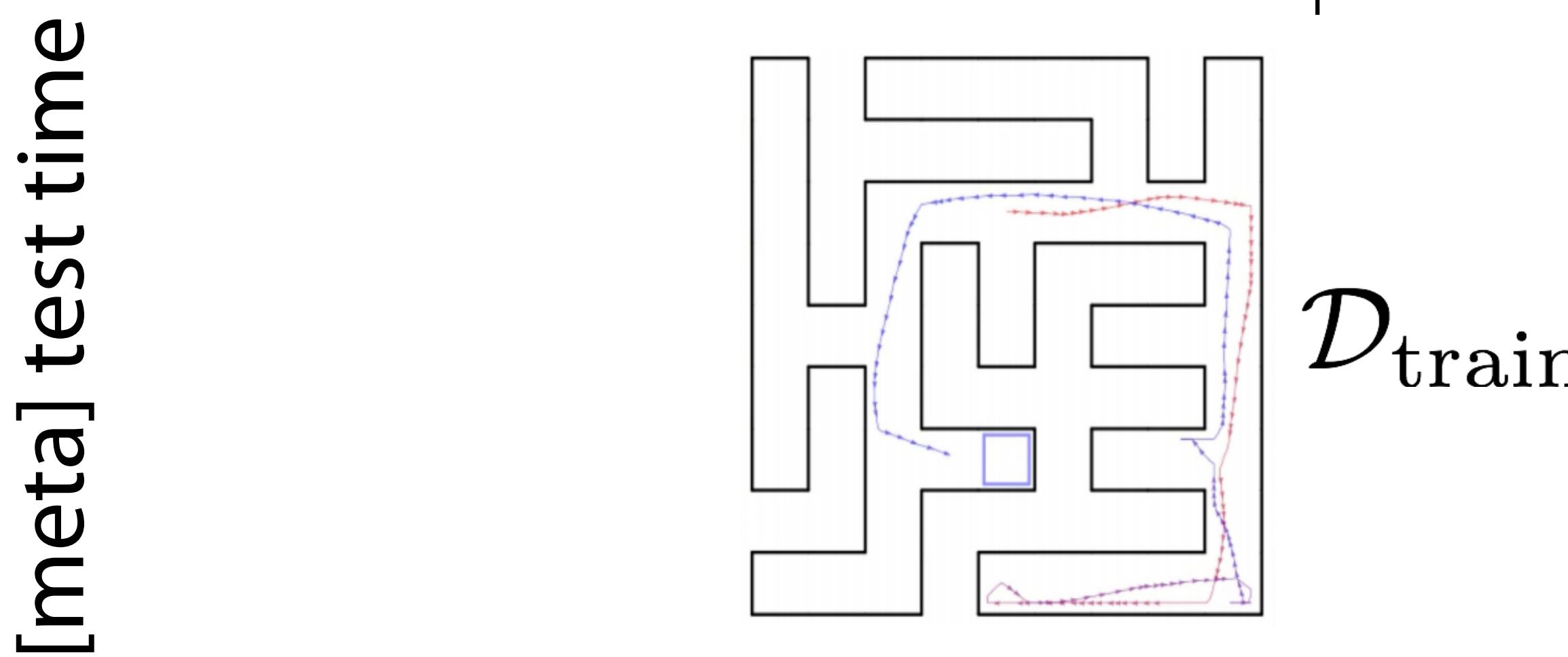


Meta-RL Example: Maze Navigation

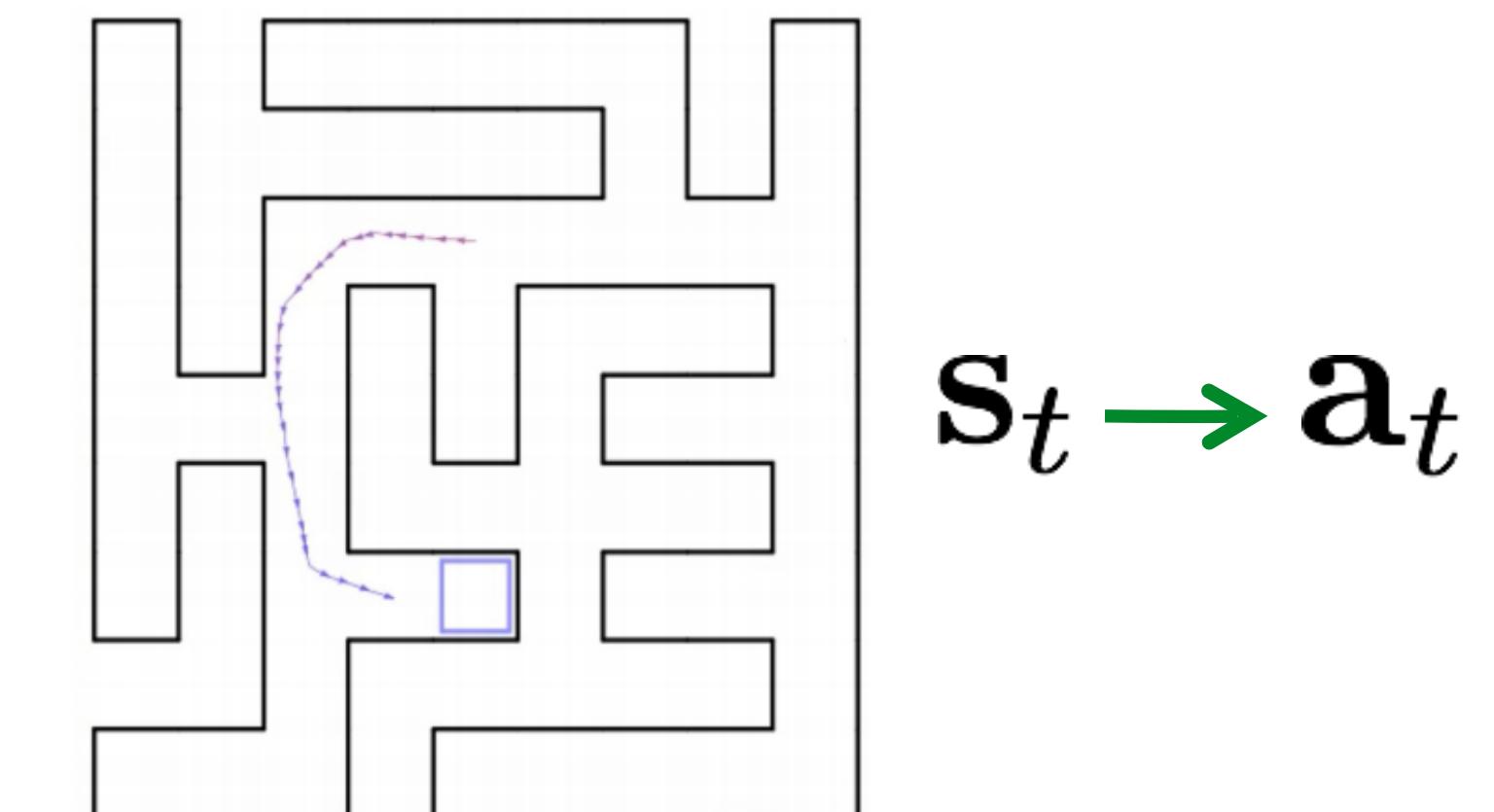
By learning how to learn many other tasks:



Given a small amount of experience

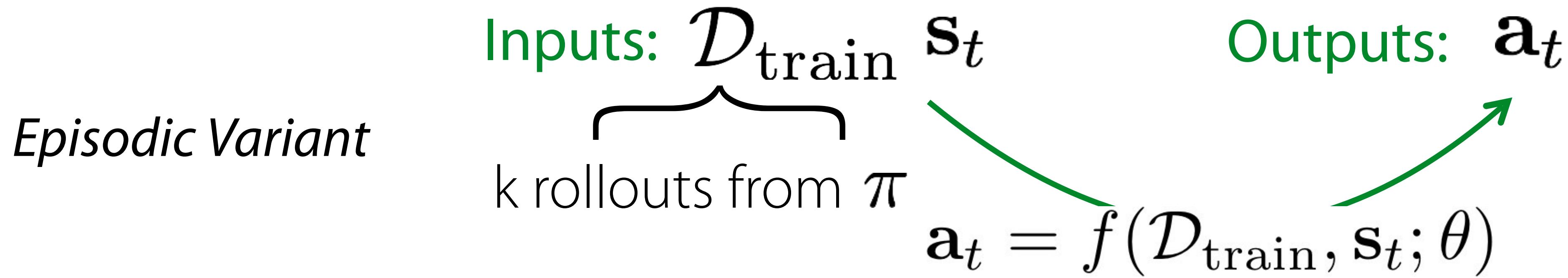


Learn to solve the task



The Meta Reinforcement Learning Problem

Meta Reinforcement Learning:



Plan for Today

Meta-RL problem statement

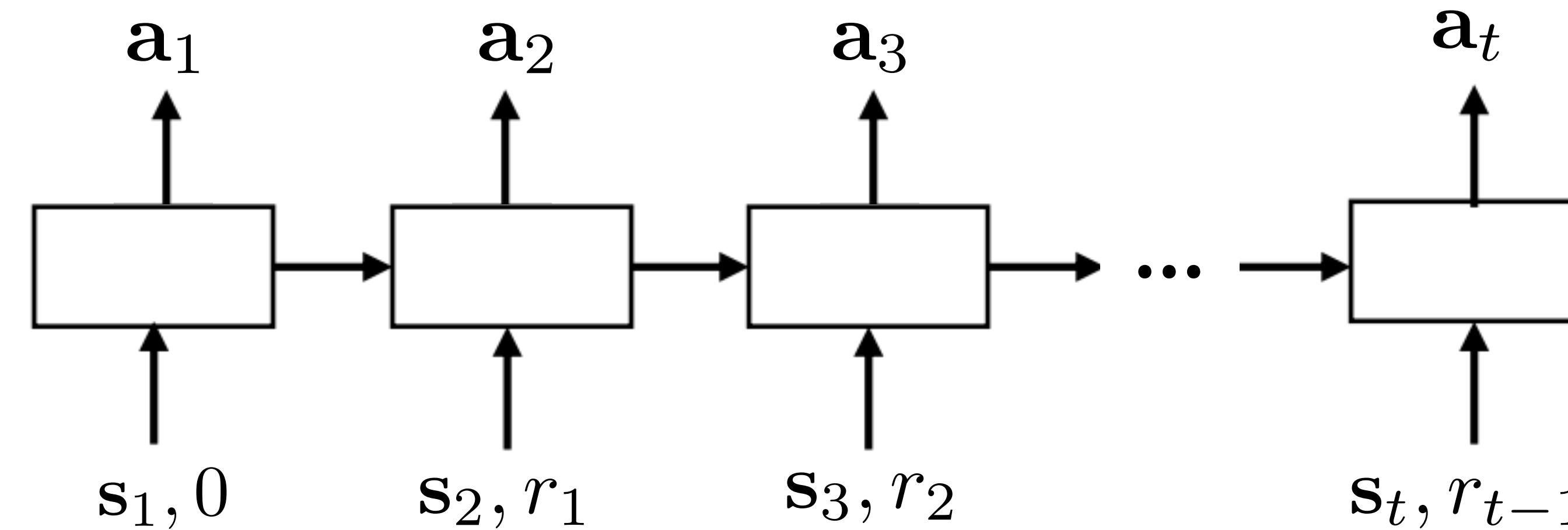
Black-box meta-RL methods

Optimization-based meta-RL methods

Black-box meta-RL

Black-box network
(LSTM, NTM, Conv, ...)

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

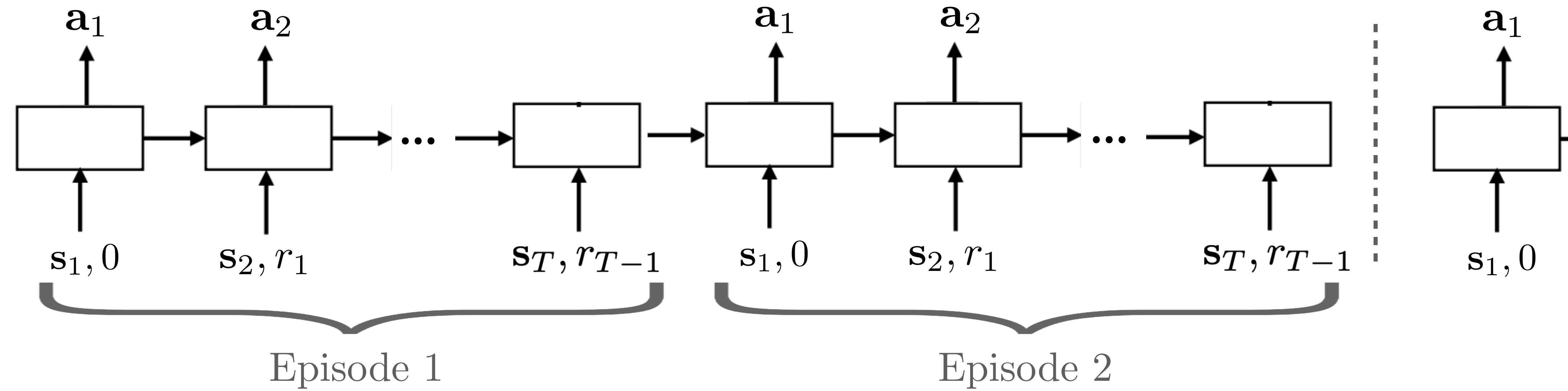


Question: How is this different from simply using a recurrent policy?
(answer in chat or by raising your hand)

Reward is passed as input
(& trained across multiple MDPs)

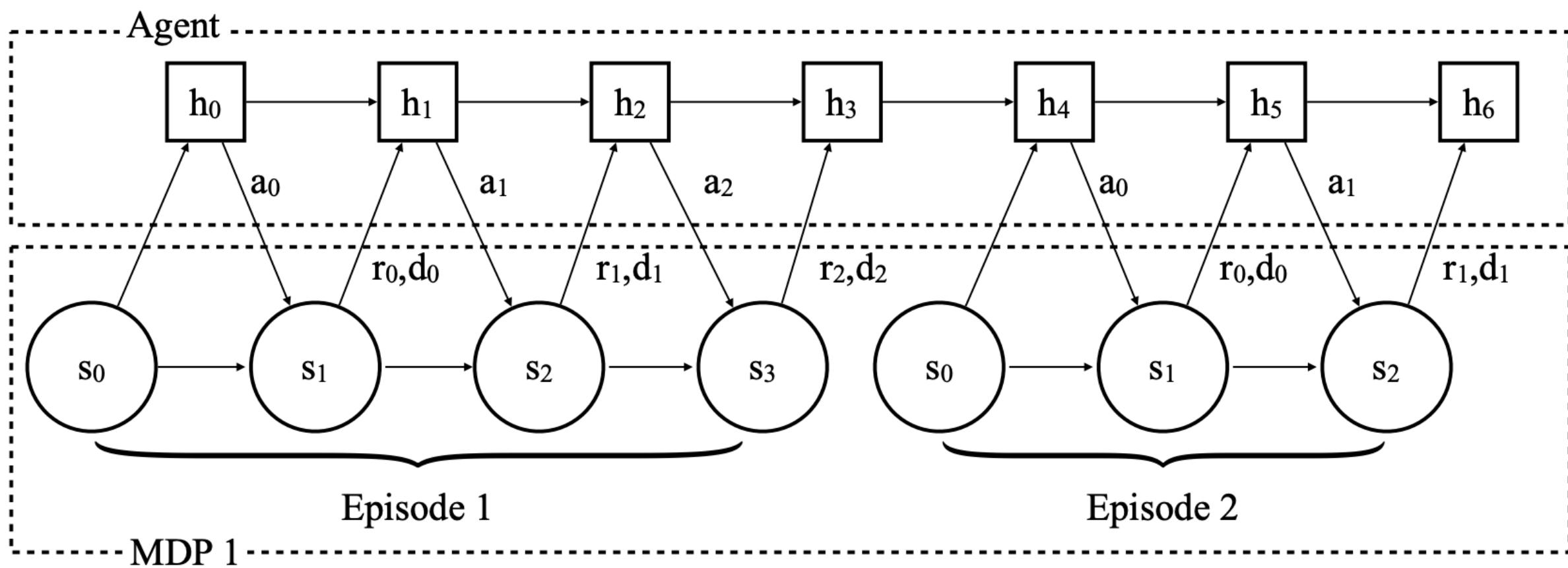
Hidden state maintained
across episodes within a task!

Black-box meta-RL



1. Sample task \mathcal{T}_i
2. Roll-out policy $\pi(a | s, \mathcal{D}_{\text{tr}})$ for N episodes (under dynamics $p_i(s' | s, a)$ and reward $r_i(s, a)$)
3. Store sequence in replay buffer for task \mathcal{T}_i .
4. Update policy to maximize discounted return for all tasks.

RNN architecture



Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel. *RL²: Fast Reinforcement Learning via Slow Reinforcement Learning*. 2017

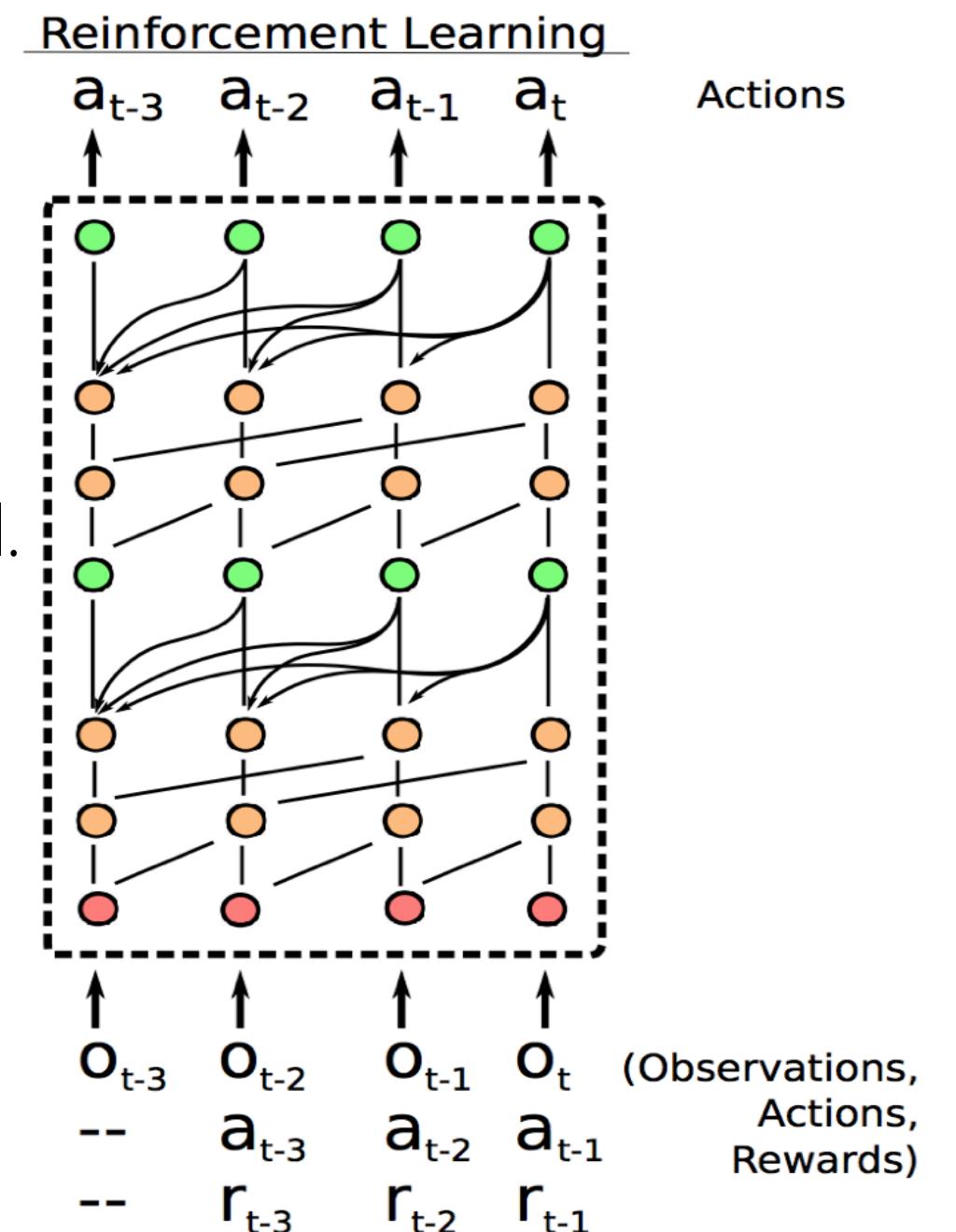
Wang, Kurth-Nelson, Tirumala, Soyer, Leibo, Munos, Blundell, Kumaran, Botvinick. *Learning to Reinforcement Learn*. CogSci 2017

TRPO/A3C (on-policy)

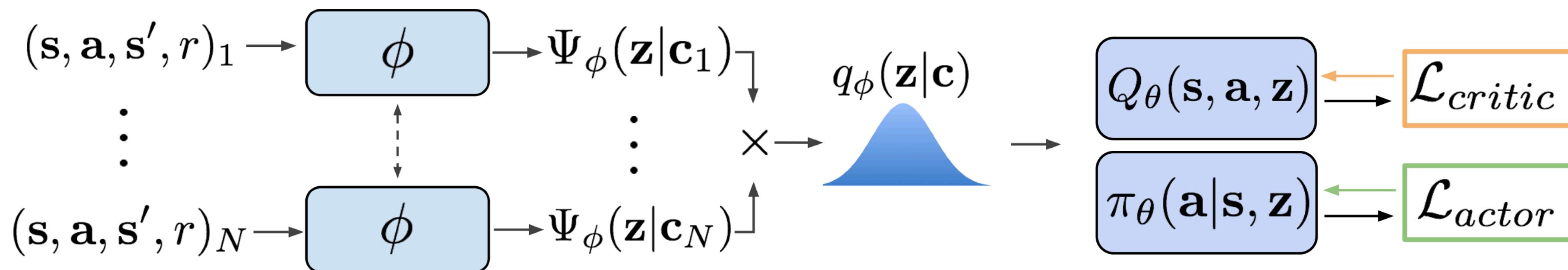
Attention + 1D conv

TRPO (on-policy)

Mishra, Rohaninejad, Chen, Abbeel.
A Simple Neural Attentive Meta-Learner. ICLR 2018



Feedforward + average SAC (off-policy)



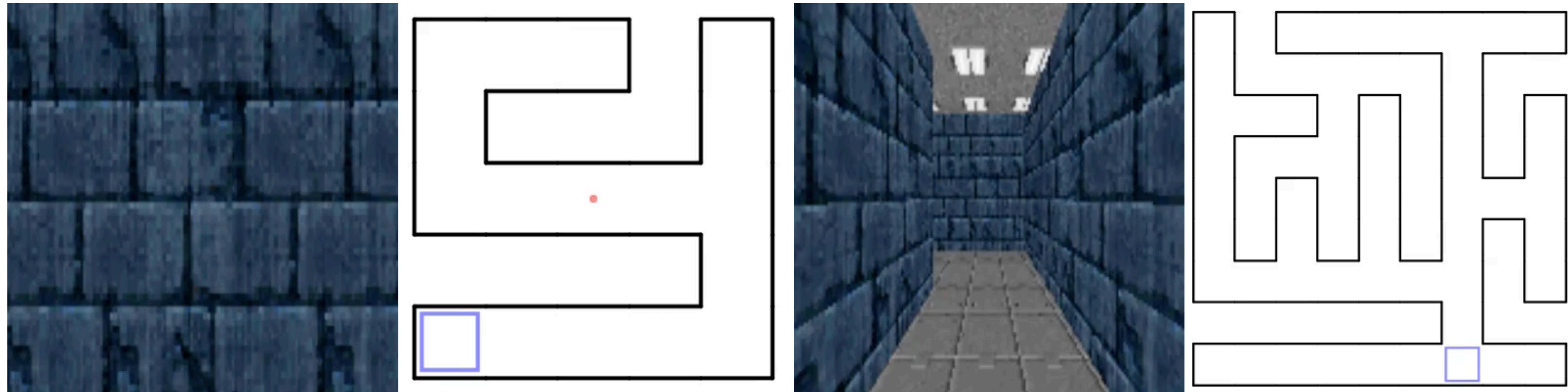
Rakelly, Zhou, Quillen, Finn, Levine. *Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables*. ICML 2019.

Meta-RL Example #1

From: Mishra, Rohaninejad, Chen, Abbeel. *A Simple Neural Attentive Meta-Learner*. ICLR 2018

Experiment: Learning to visually navigate a maze

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



Meta-RL Example #1

From: Mishra, Rohaninejad, Chen, Abbeel. *A Simple Neural Attentive Meta-Learner*. ICLR 2018

Experiment: Learning to visually navigate a maze

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

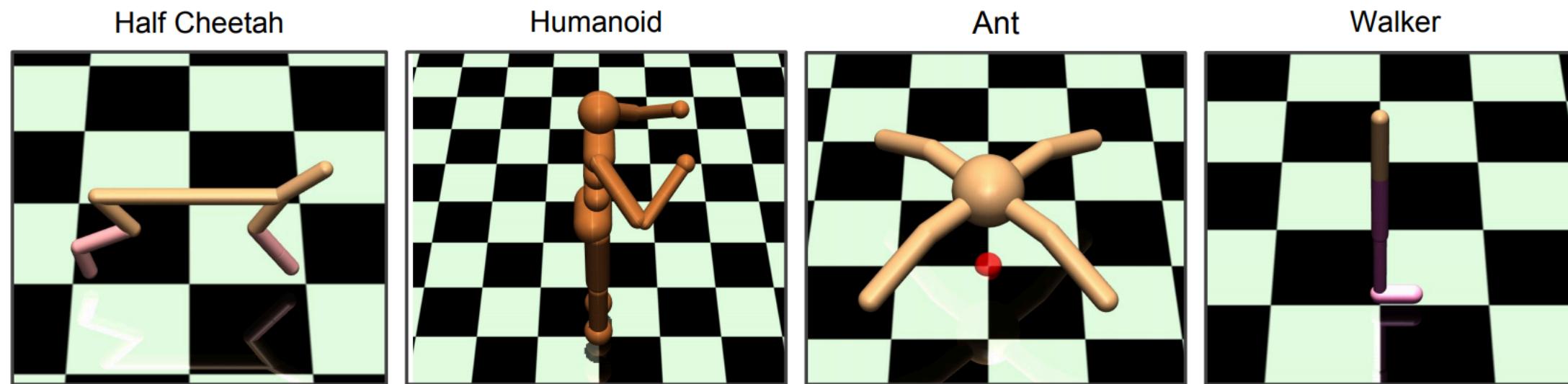
Method	Small Maze		Large Maze	
	Episode 1	Episode 2	Episode 1	Episode 2
Random	188.6 ± 3.5	187.7 ± 3.5	420.2 ± 1.2	420.8 ± 1.2
LSTM	52.4 ± 1.3	39.1 ± 0.9	180.1 ± 6.0	150.6 ± 5.9
SNAIL (ours)	50.3 ± 0.3	34.8 ± 0.2	140.5 ± 4.2	105.9 ± 2.4

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

Meta-RL Example #2

Rakelly, Zhou, Quillen, Finn, Levine. *Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables.* ICML 2019.

Experiment: Continuous control problems

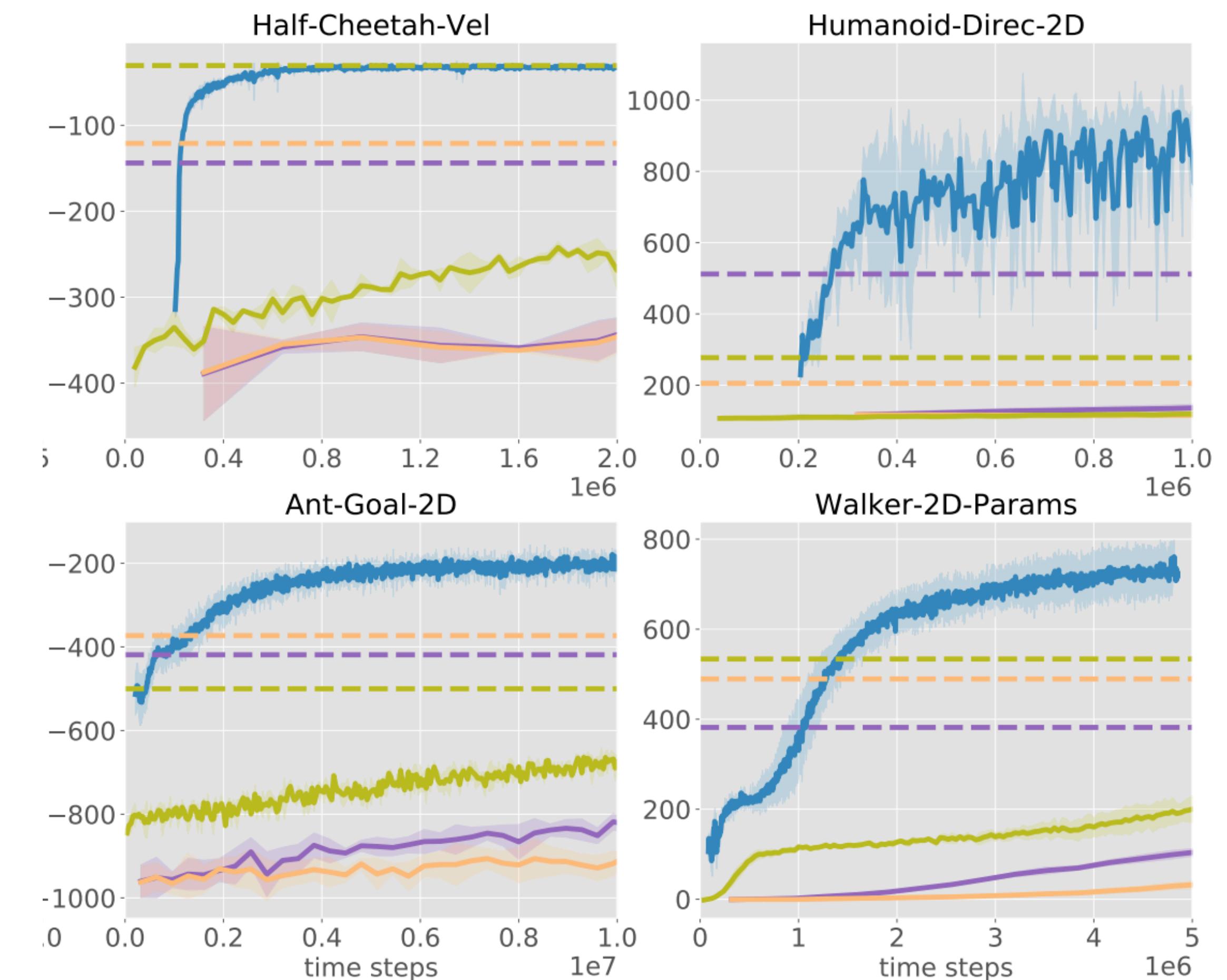


- different directions, velocities
- different physical dynamics

Meta-RL algos are very efficient at new tasks.

What about meta-training efficiency?

Question: Do you expect off-policy meta-RL to be more or less efficient than on-policy meta-RL?



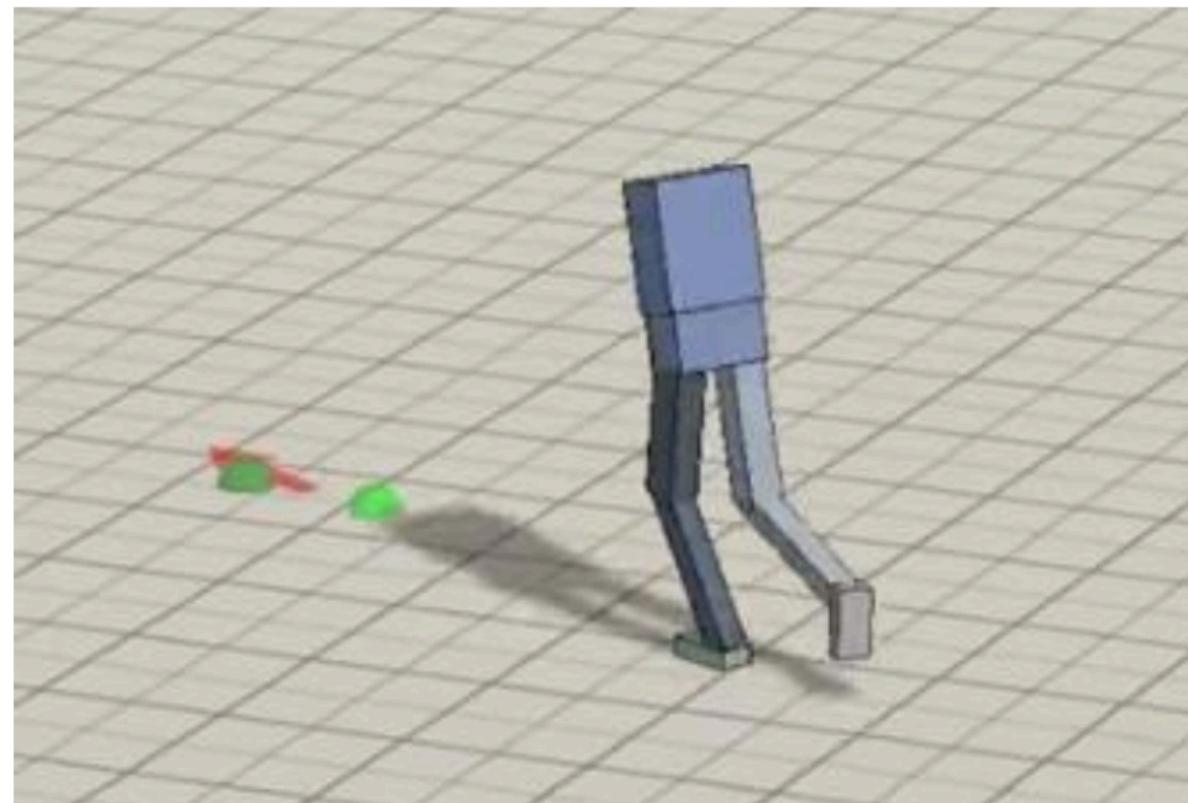
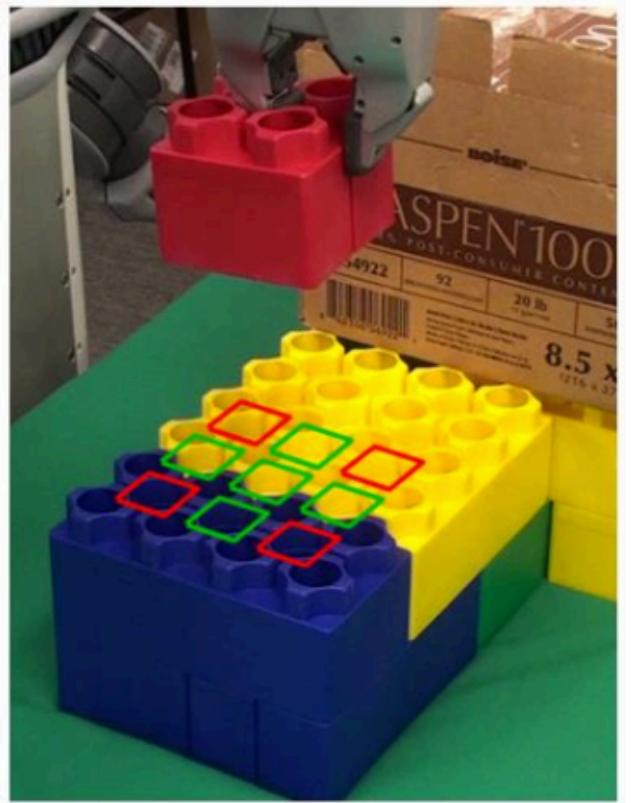
Black-box:

	PEARL		RL2
	ProMP		MAML

Opt-based:

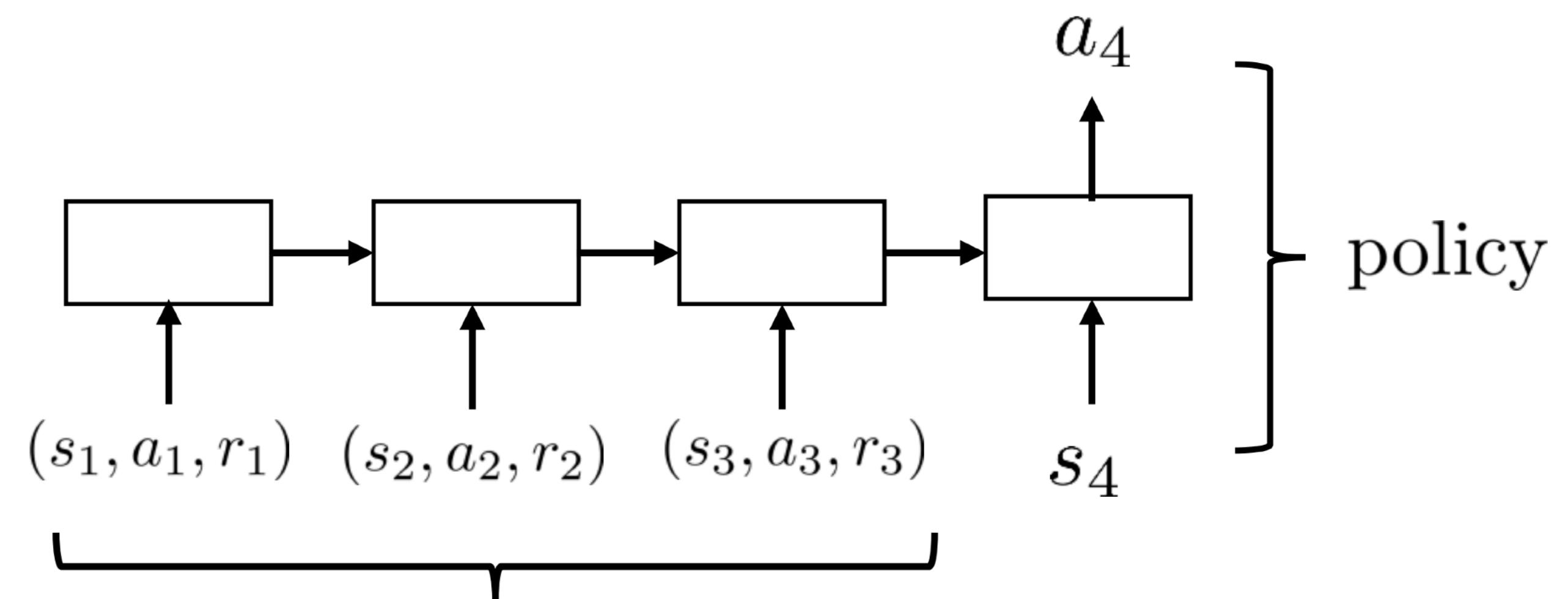
Digression: Connection to Multi-Task Policies

multi-task policy: $\pi_\theta(\mathbf{a} \mid \mathbf{s}, \mathbf{z}_i)$



\mathbf{z}_i : stack location

\mathbf{z}_i : walking direction



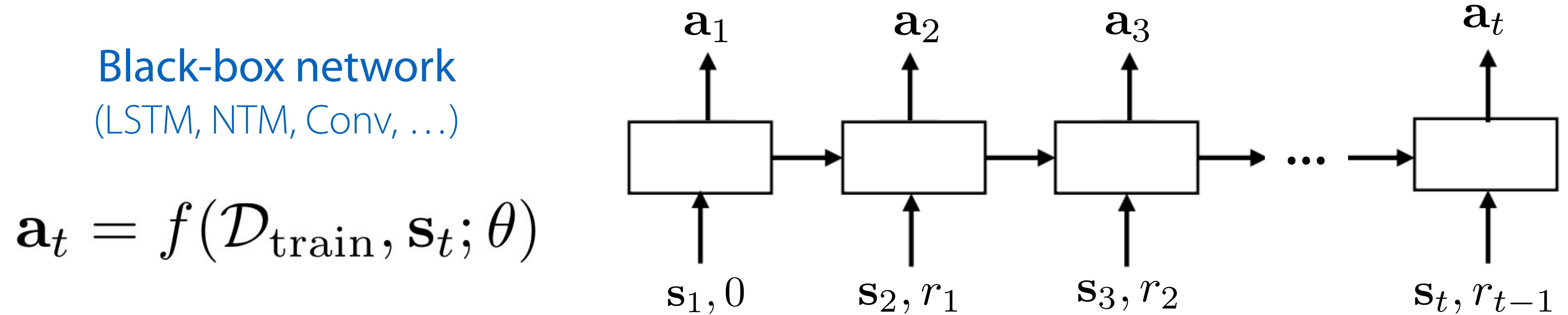
task identifier \mathbf{z}_i

Multi-task policy with experience as task identifier.

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**)

Black-box meta-RL



- + general & expressive
- + a variety of design choices in architecture
- hard to optimize
- ~ inherits sample efficiency from outer RL optimizer

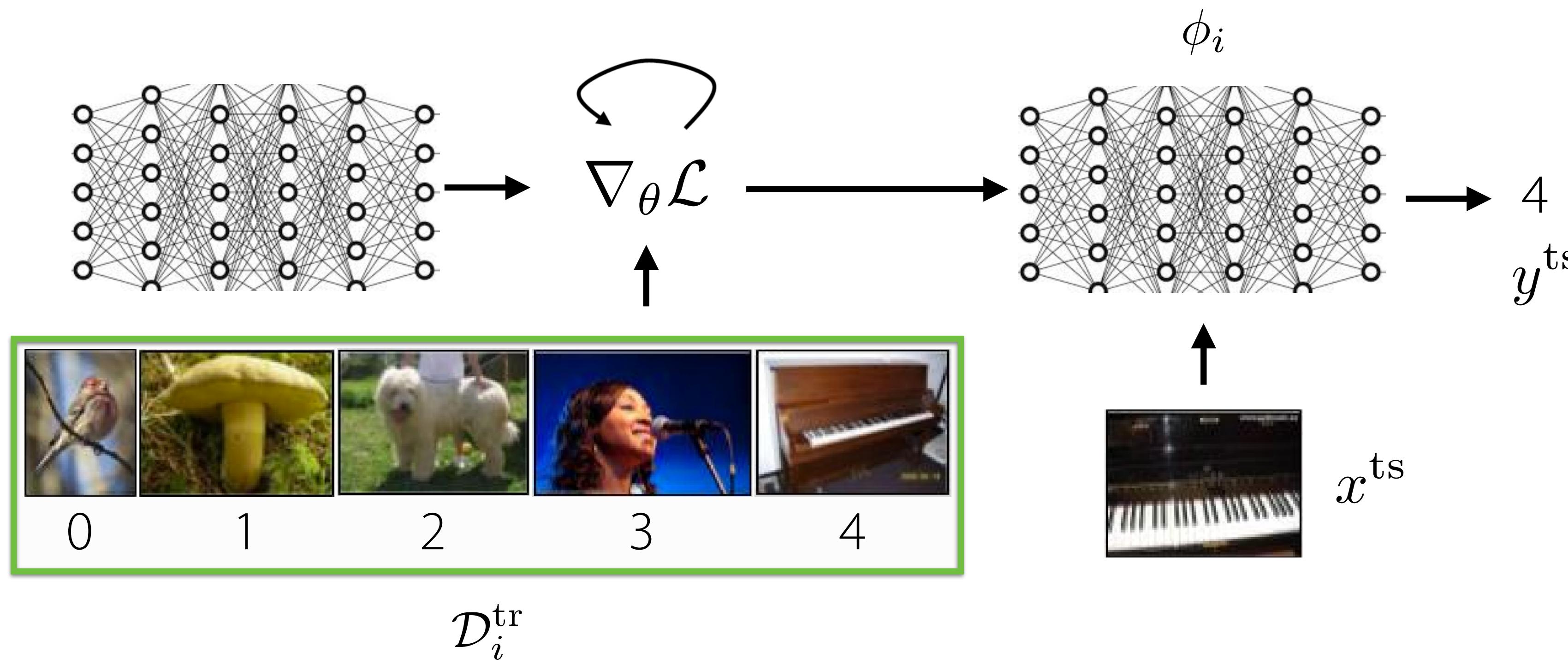
Plan for Today

Meta-RL problem statement

Black-box meta-RL methods

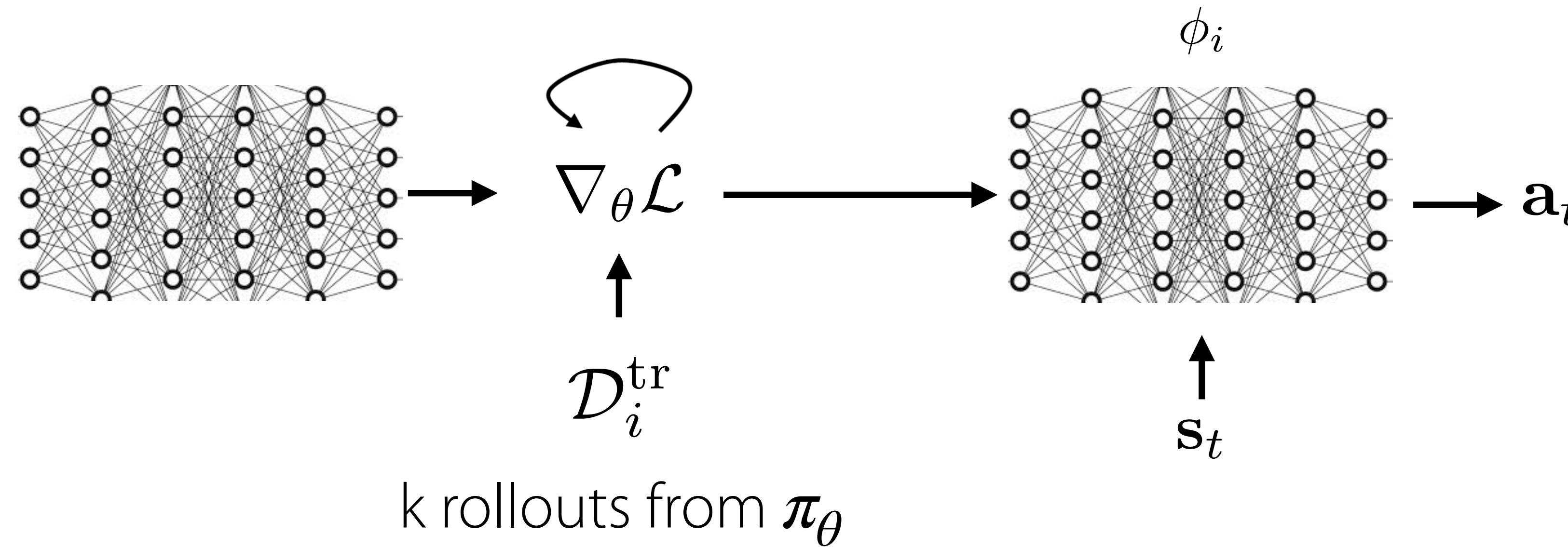
Optimization-based meta-RL methods

Recap: Optimization-Based Meta-Learning



Key idea: embed optimization inside the inner learning process

Recap: Optimization-Based Meta-Learning Meta-RL



Key idea: embed optimization inside the inner learning process

Question: What should we use for the inner optimization and why? (in chat or by raising hand)

Policy gradients?

- + gradient-based!
- on-policy (inefficient)

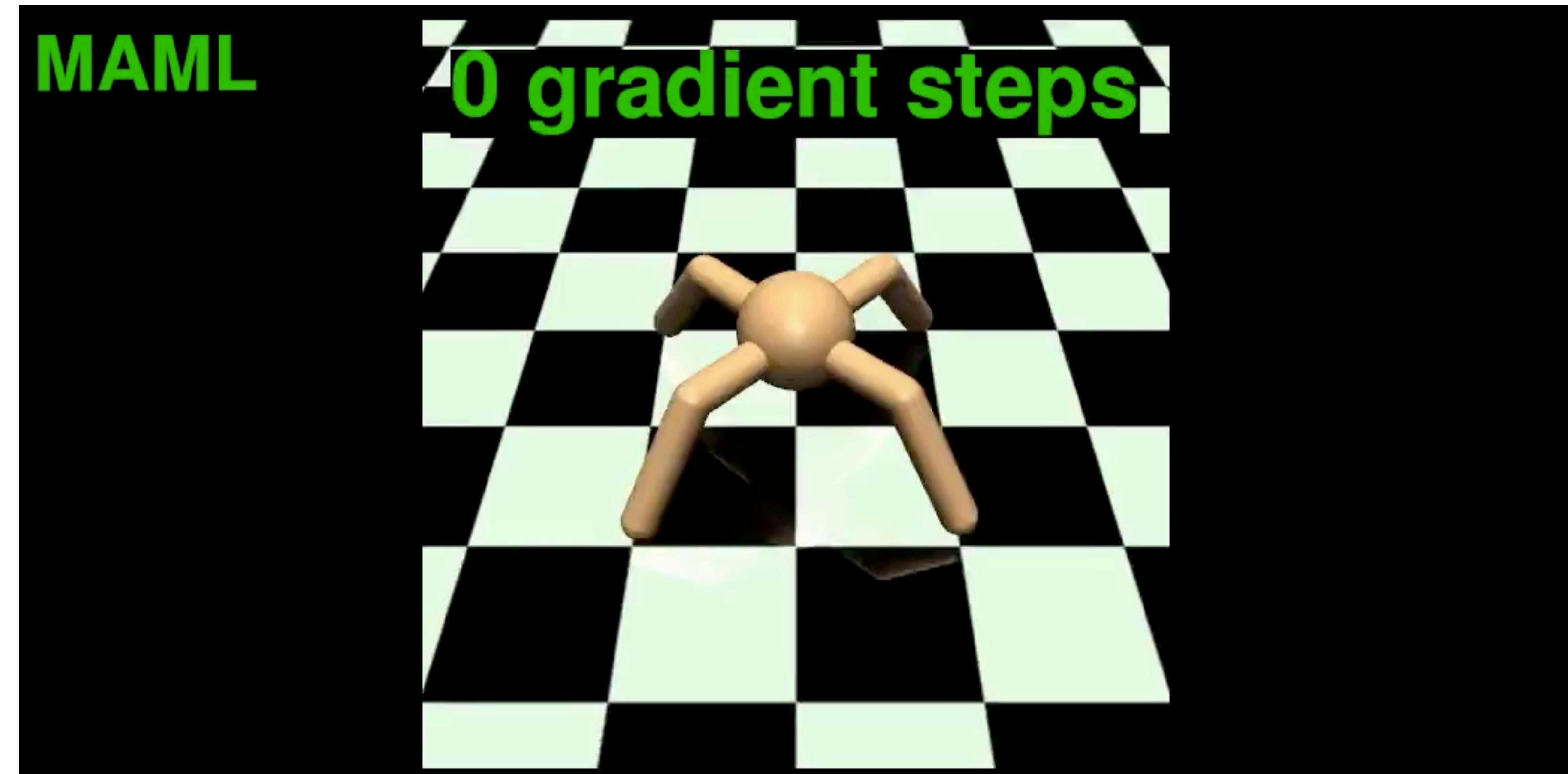
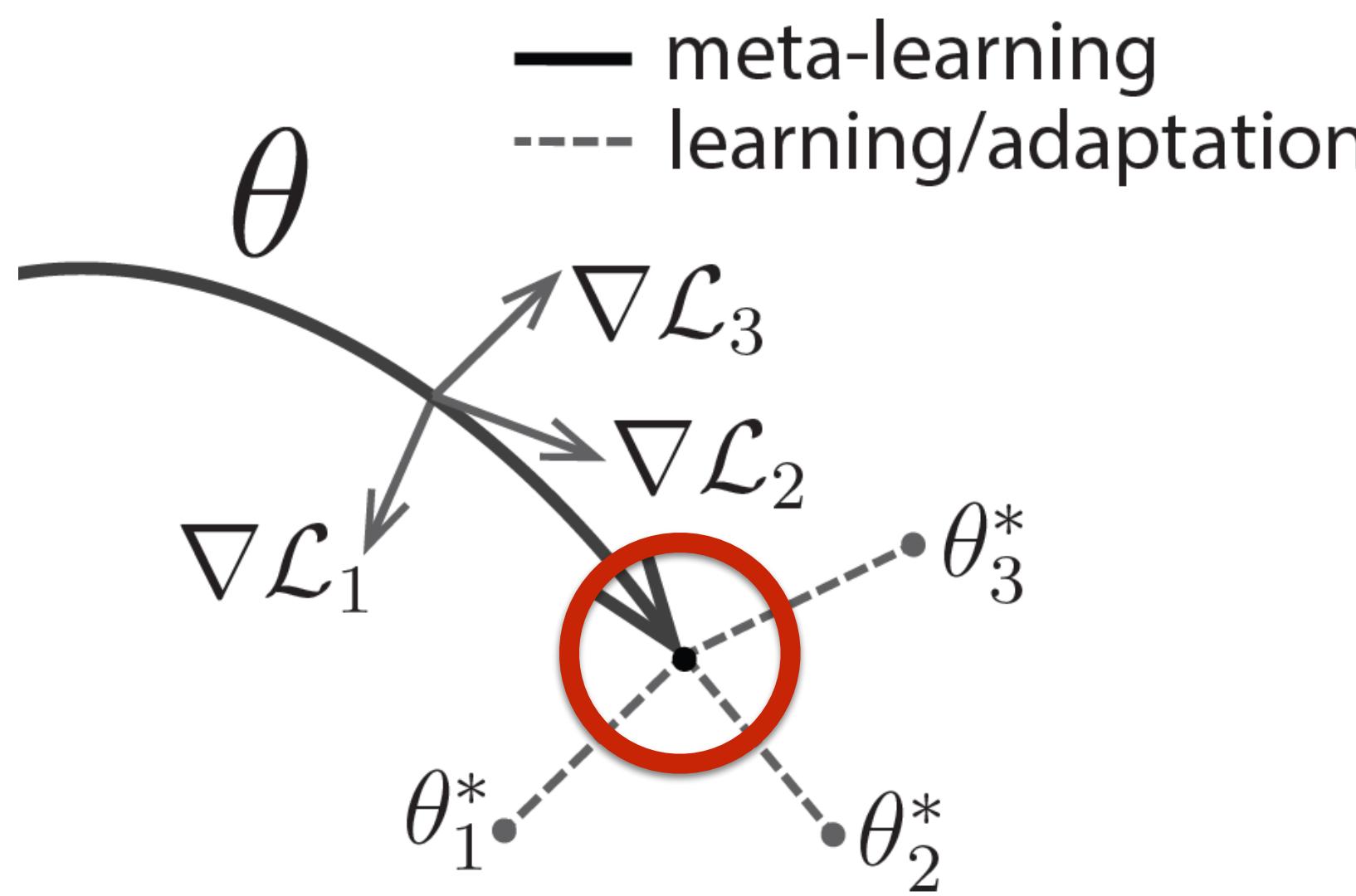
Q-learning?

- dynamic programming
(requires many steps)
- + off-policy (efficient)

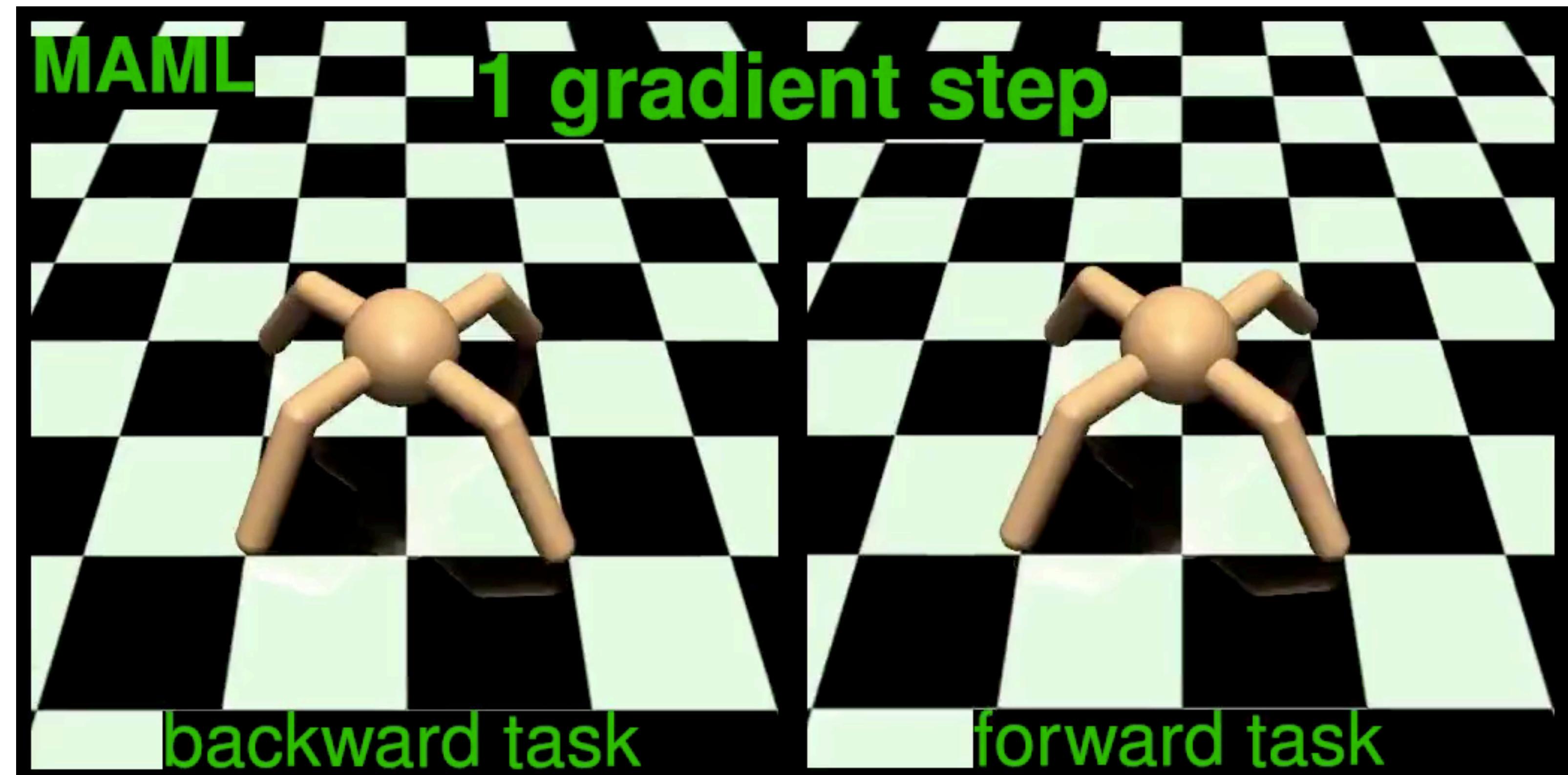
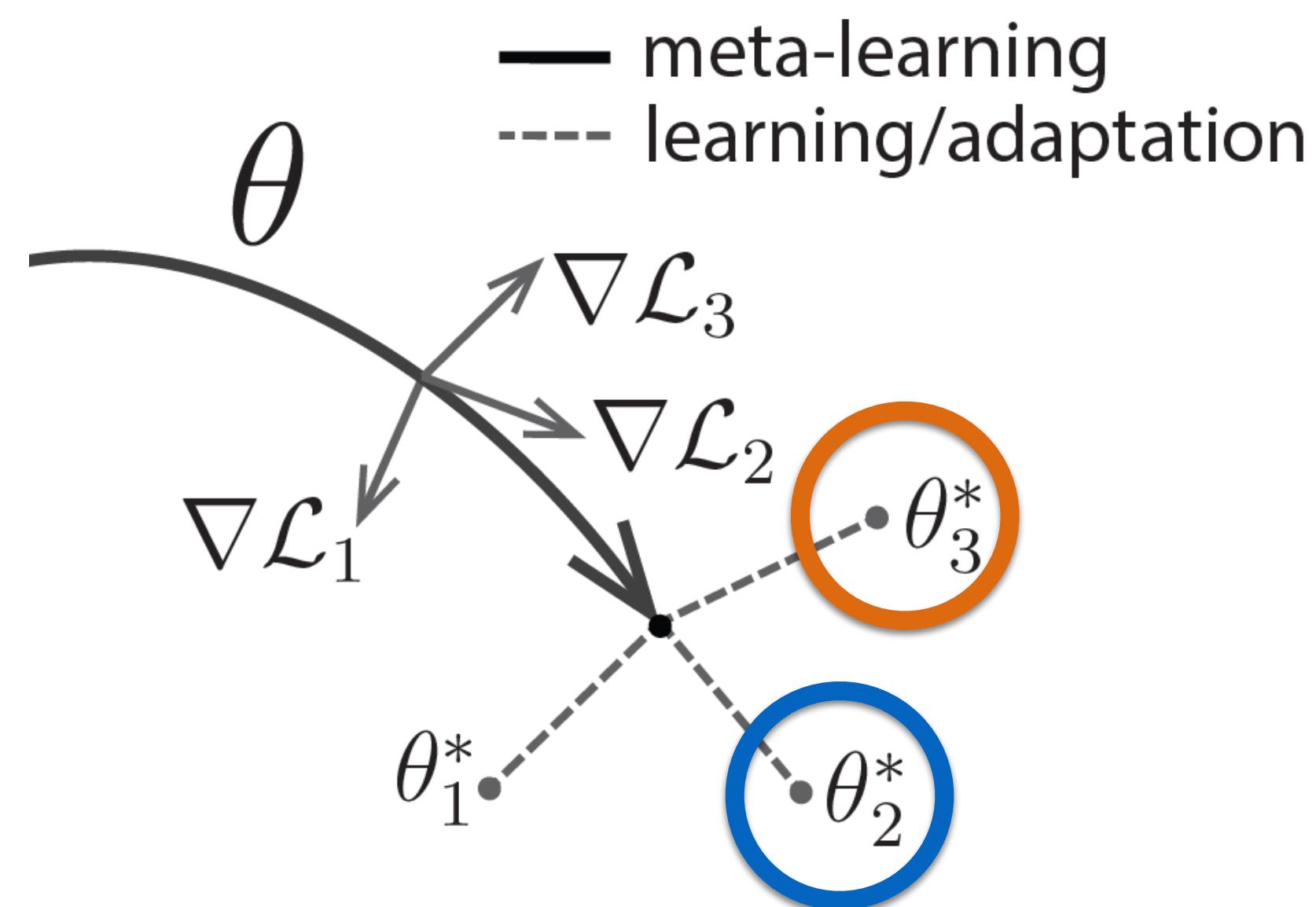
Model-based RL?

- + gradient-based
(model learning=supervised)
- + off-policy (efficient)

MAML + Policy Gradients



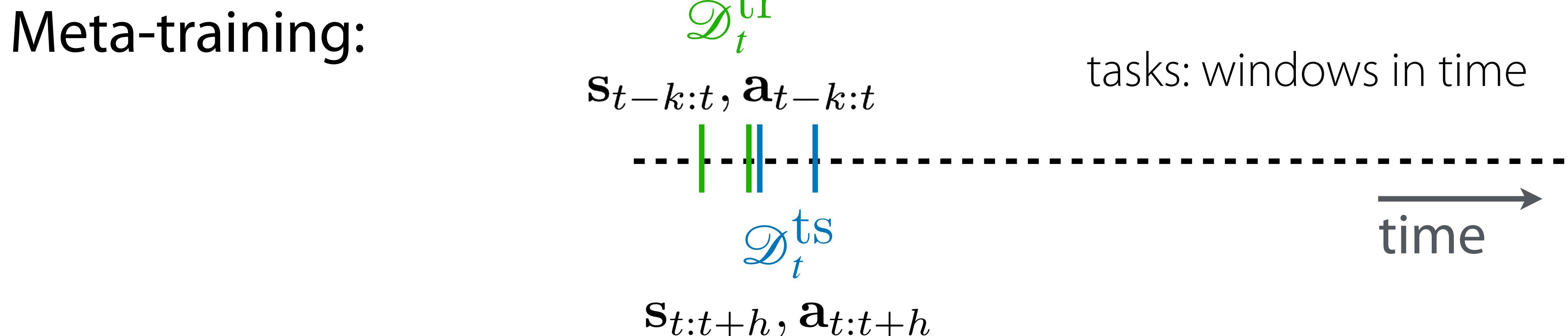
MAML + Policy Gradients



MAML + Model-Based RL



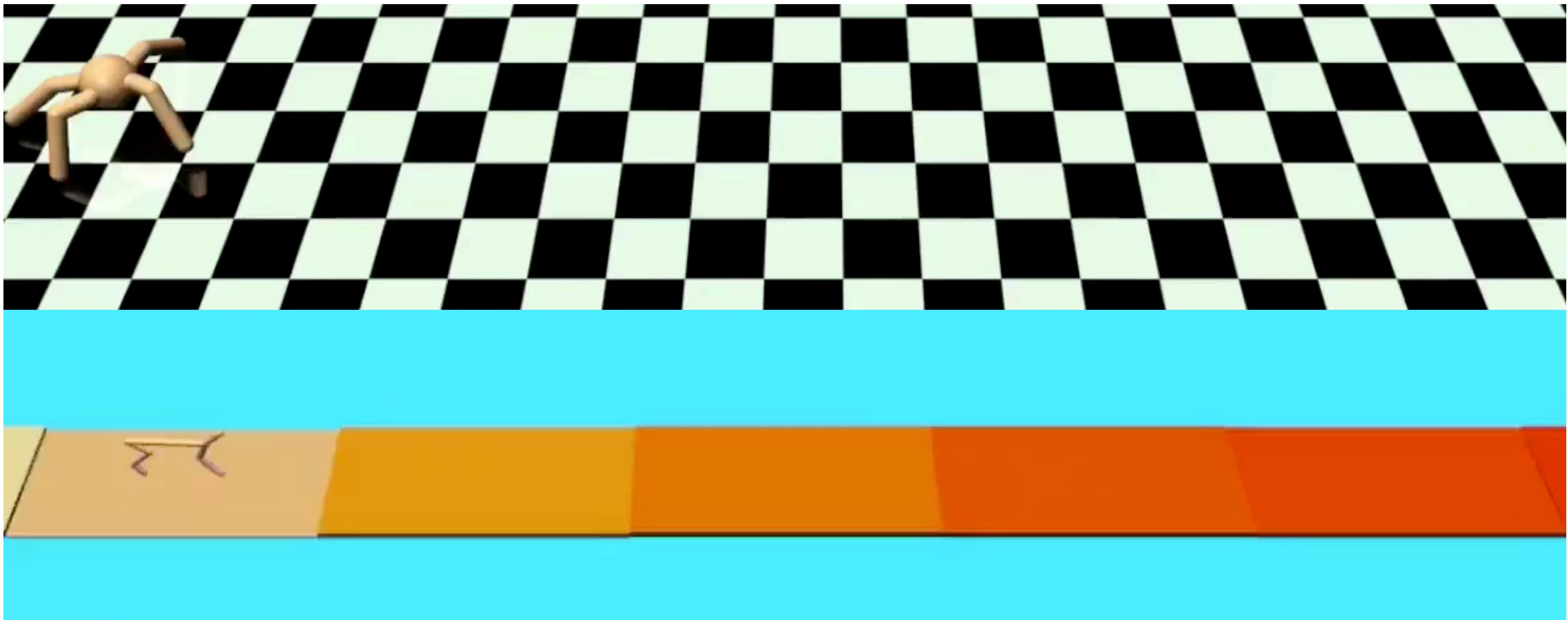
- Meta-test time:**
1. Adapt model $f_\theta \rightarrow f_{\phi_t}$ to last k time steps
 2. Plan a_t, \dots, a_{t+h} using adapted model f_{ϕ_t}



Dynamic Environments without Adaptation

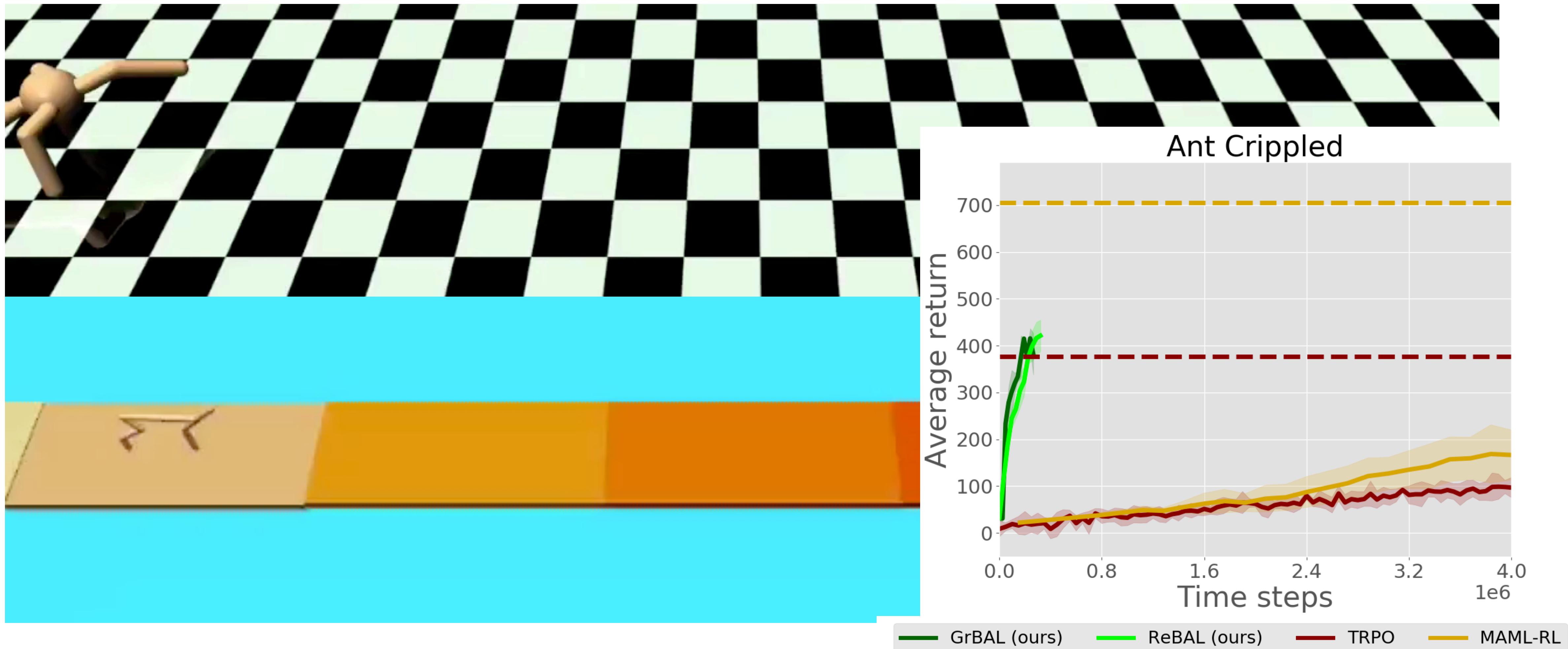
Model-Based RL Only

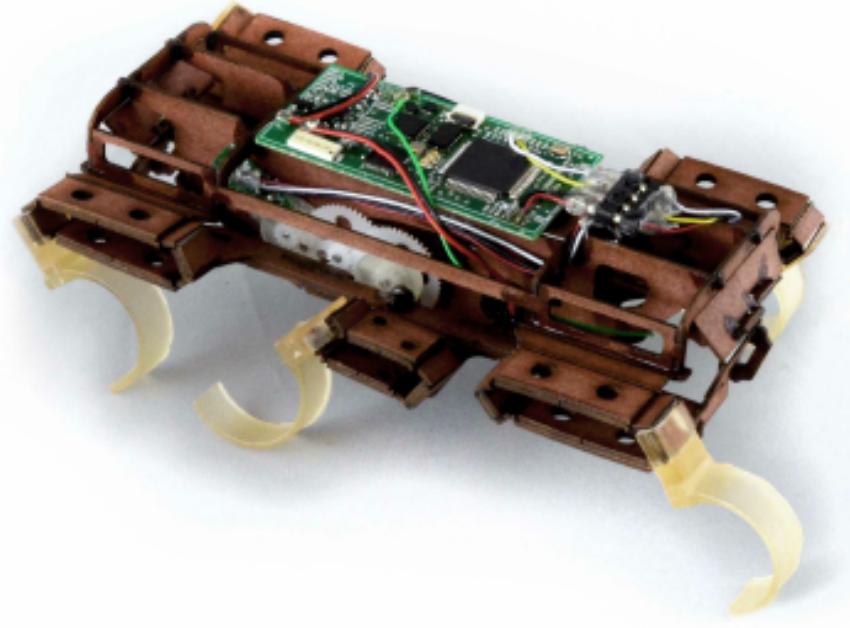
Tries to fit single model $f(s' | s, a)$ to varying $p_t(s' | s, a)$.



Dynamic Environments without Adaptation

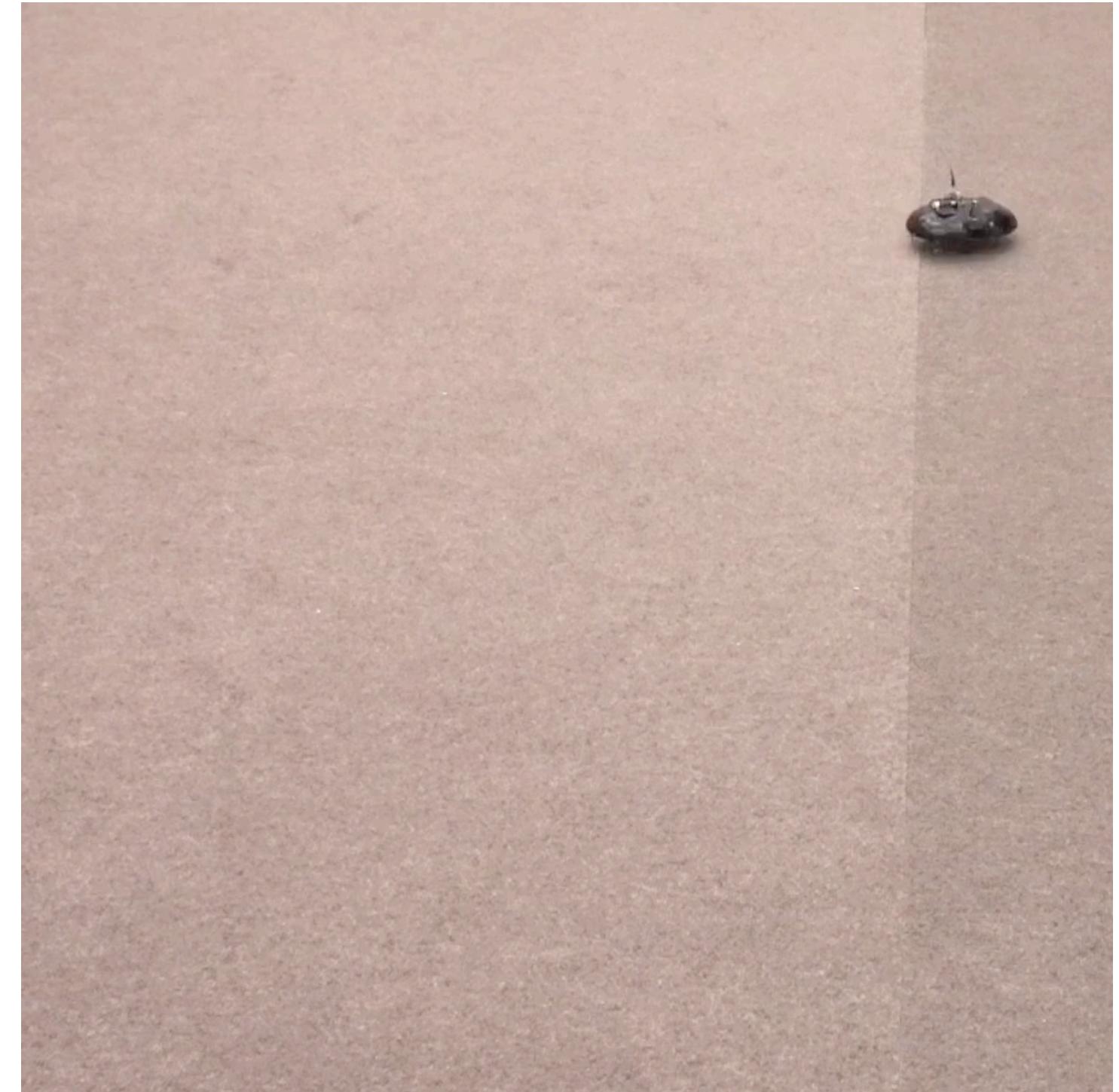
MAML+Model-based RL





VelociRoACH Robot

Meta-train on variable terrains

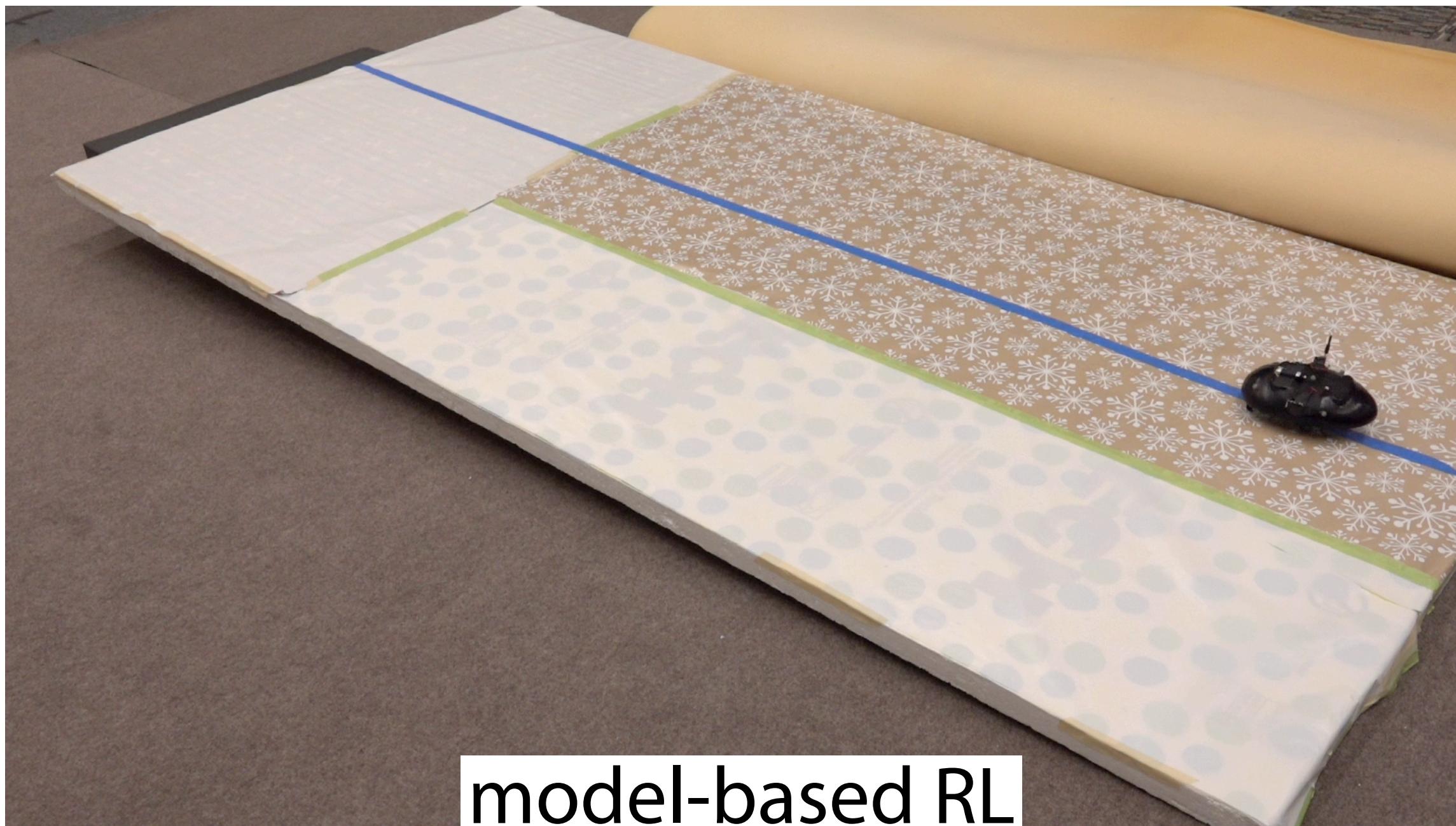


Meta-test with slope, missing leg, payload, calibration errors

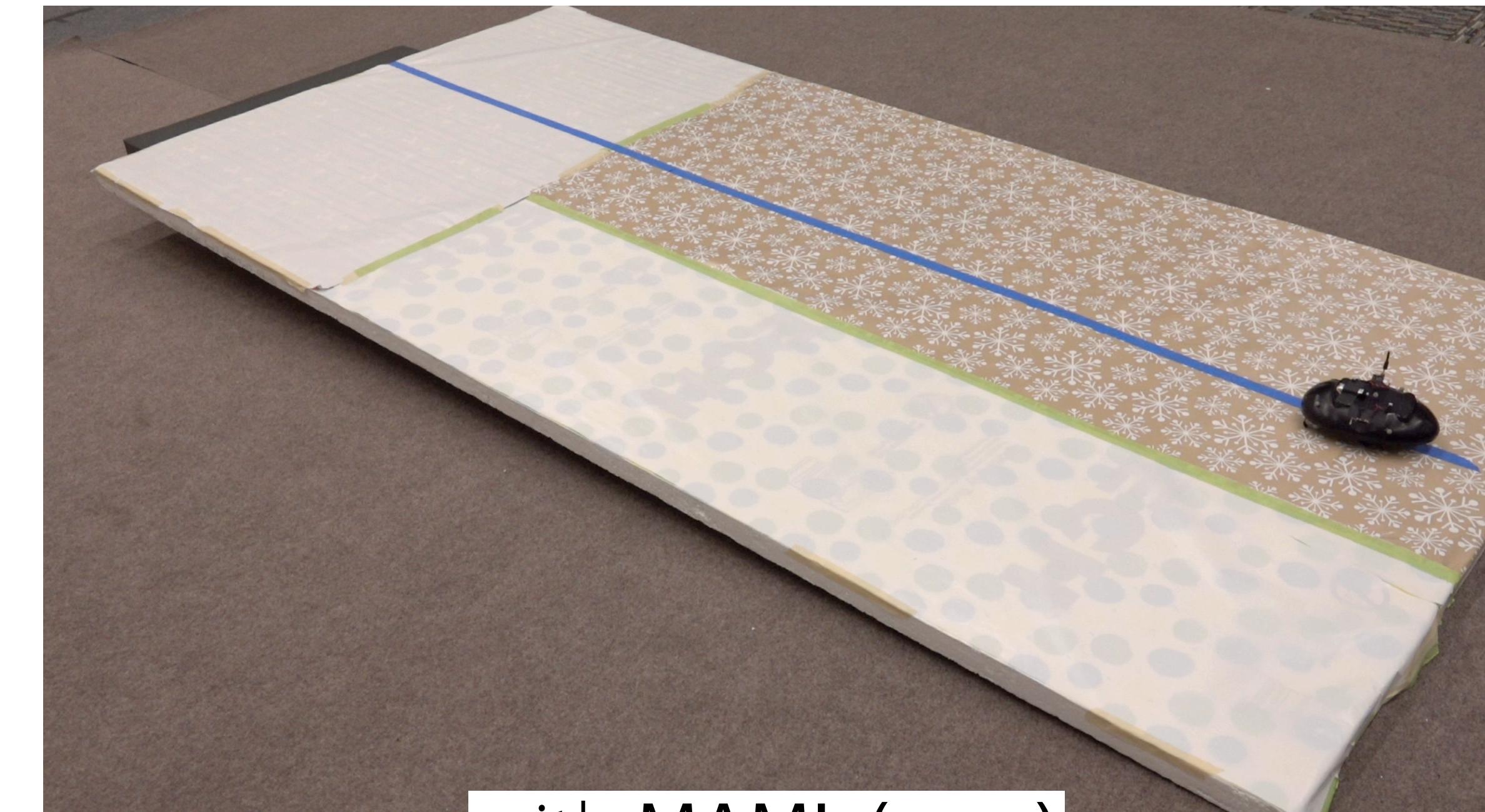
VelociRoACH Robot

Meta-train on variable terrains

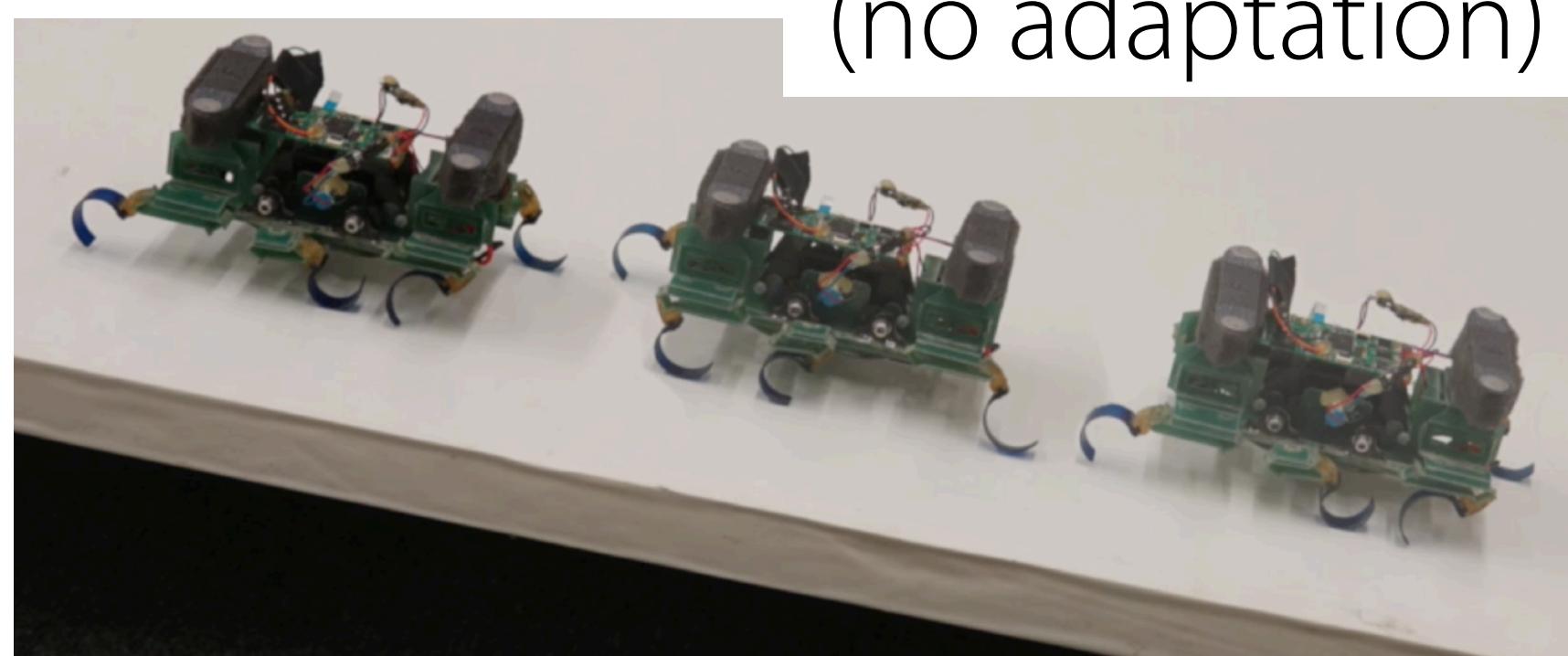
Meta-test with slope, missing leg, payload, calibration errors



model-based RL
(no adaptation)



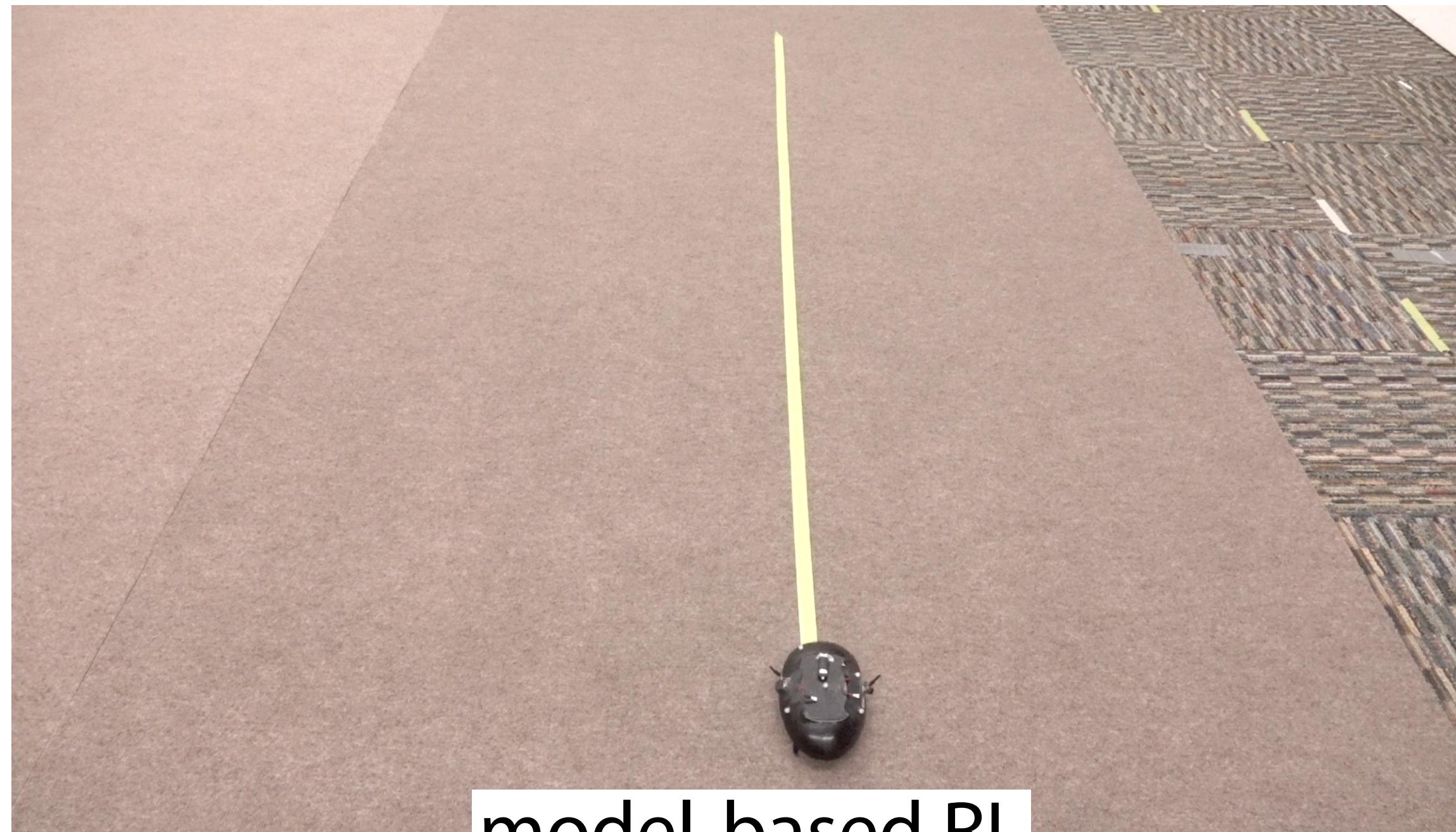
with MAML (ours)



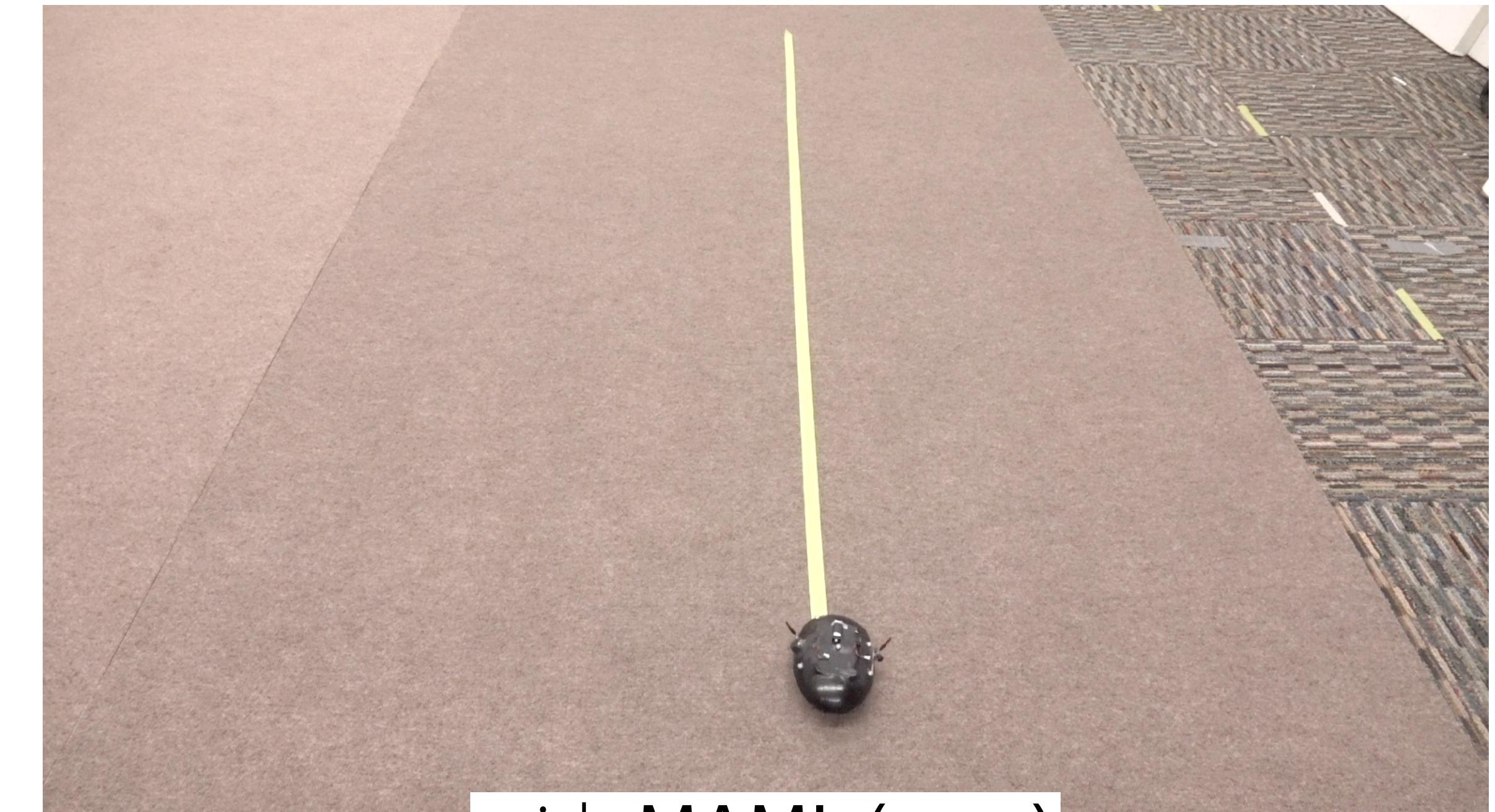
VelociRoACH Robot

Meta-train on variable terrains

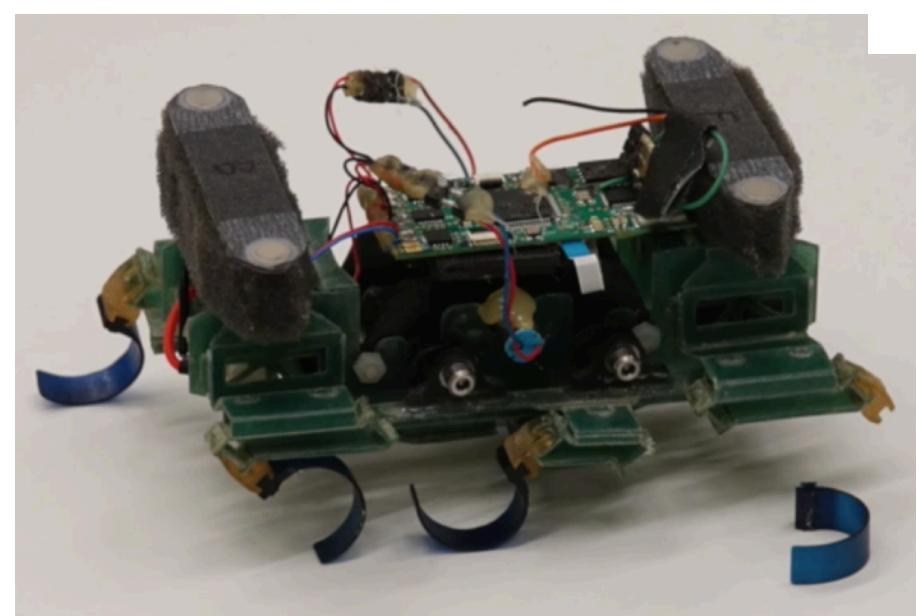
Meta-test with slope, missing leg, payload, calibration errors



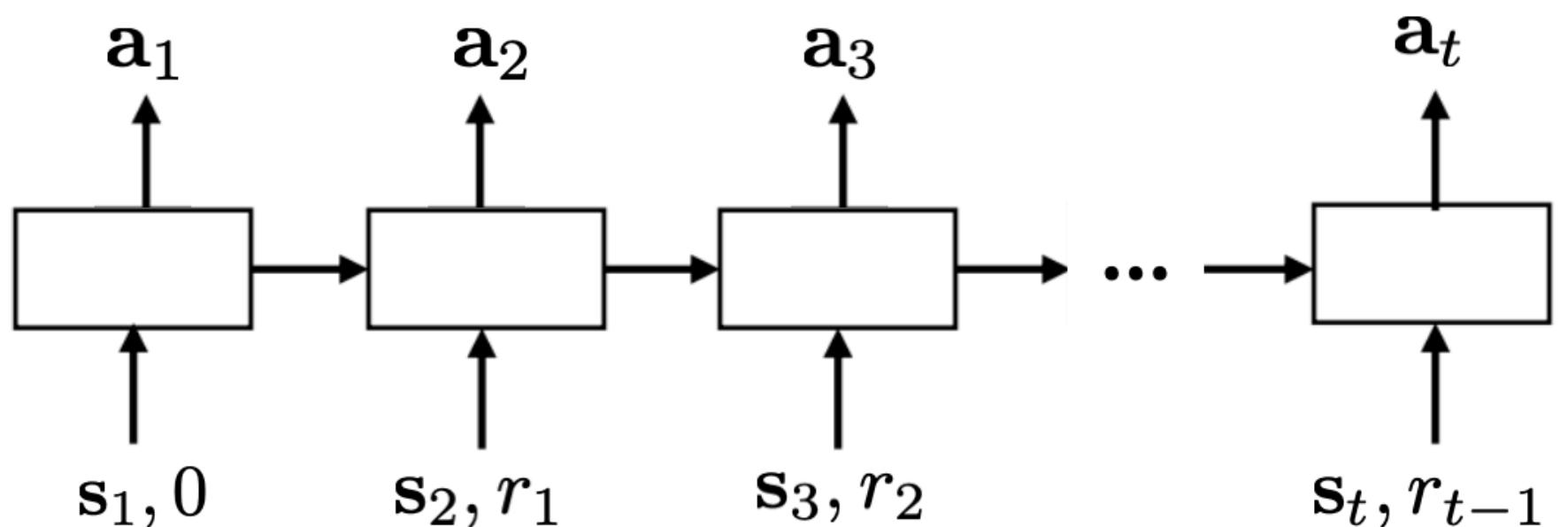
model-based RL
(no adaptation)



with MAML (ours)

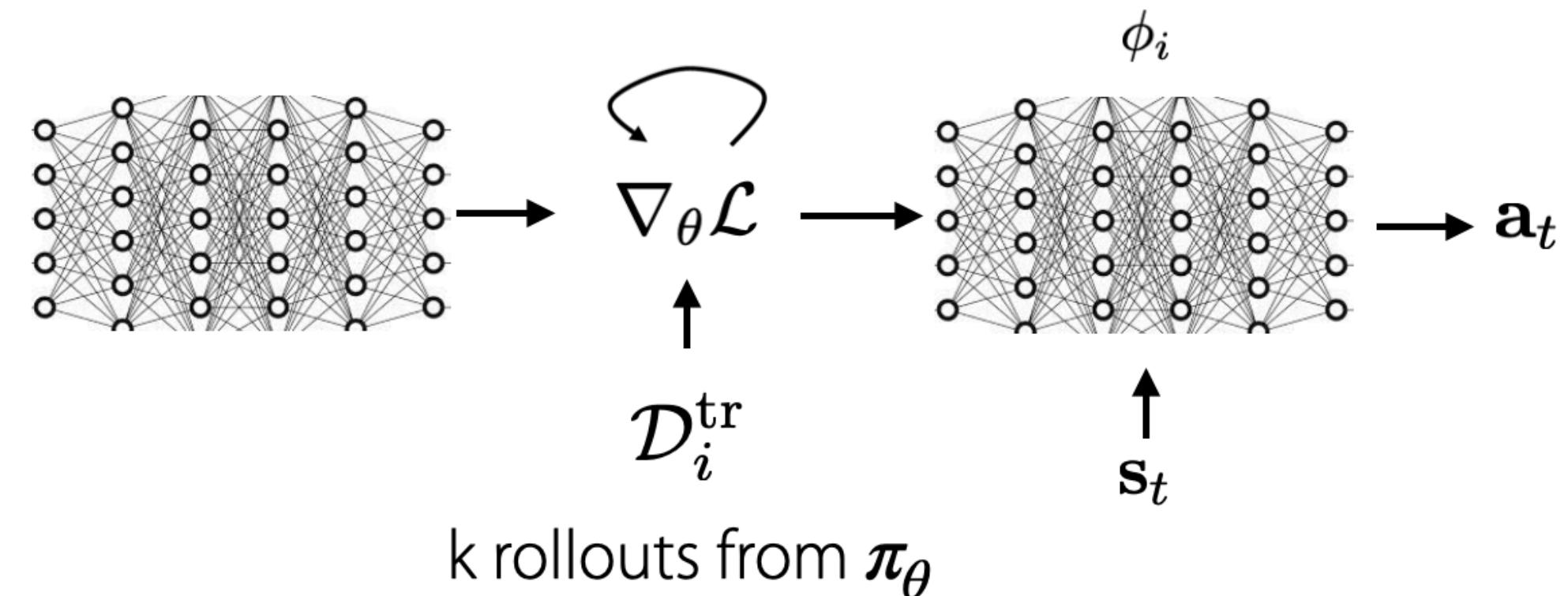


Black-Box Meta-RL



- + general & expressive
- + a variety of design choices in architecture & objective
- hard to optimize

Optimization-Based Meta-RL



- + inductive bias of optimization built in
- + easy to combine with policy gradients, model-based methods
- policy gradients very noisy
- hard to combine with value-based RL methods

Both: inherit sample efficiency from outer RL optimizer

Plan for Today

Meta-RL problem statement

Black-box meta-RL methods

Optimization-based meta-RL methods

Next time: Learning to explore.

- Understand the **meta-RL problem statement** & set-up
- Understand the basics of **black-box meta RL algorithms**
- Understand the basics & challenges of **optimization-based meta RL algorithms**

Lecture goals:

Next time

Today: meta-RL basics

Next Monday: learning to explore via meta-RL

Next Wednesday: Bayesian perspective on meta-RL

Reminders

Homework 3 due **Monday 10/26.**

Project milestone due **Monday 11/2.**