Domain Transfer

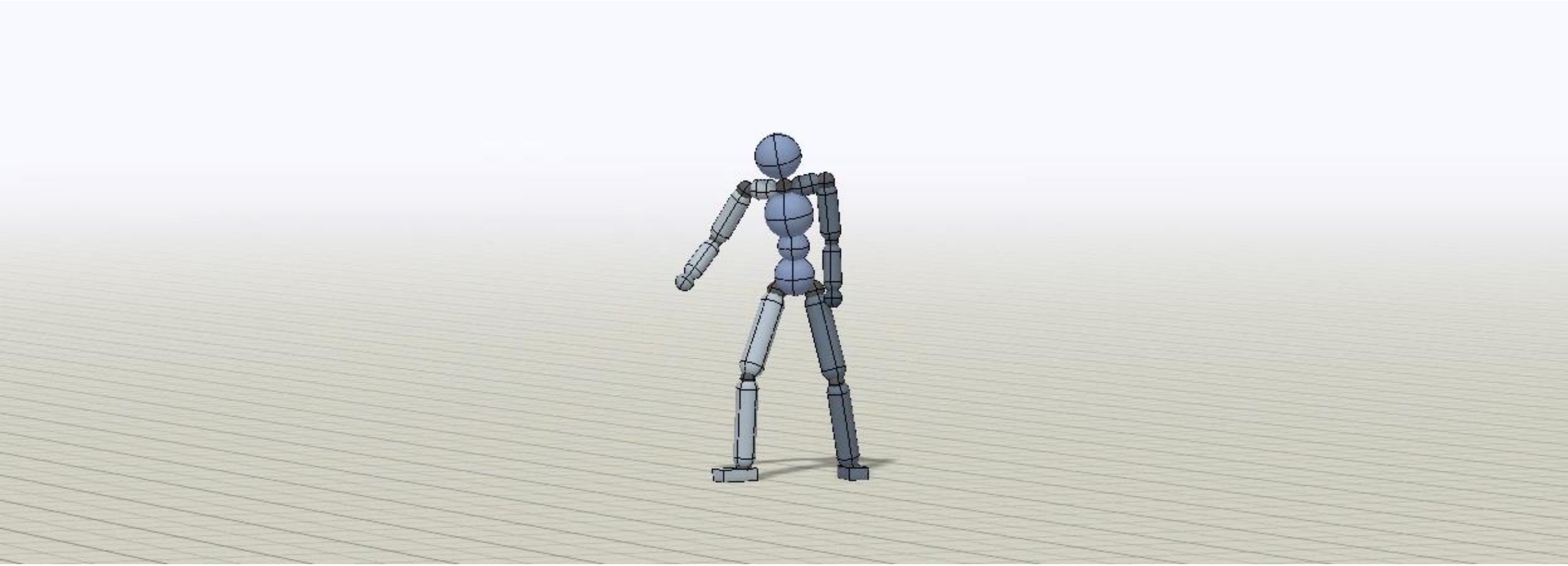
CMPT 729 G100

Jason Peng

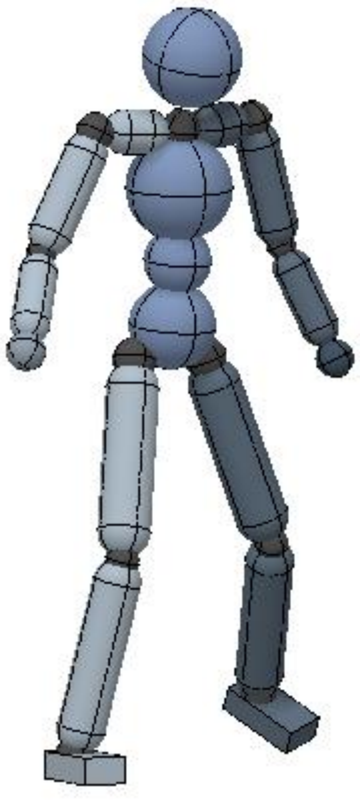
Overview

- Domain Transfer
- System Identification
- Domain Randomization
- Domain Adaptation

Learning in Simulation



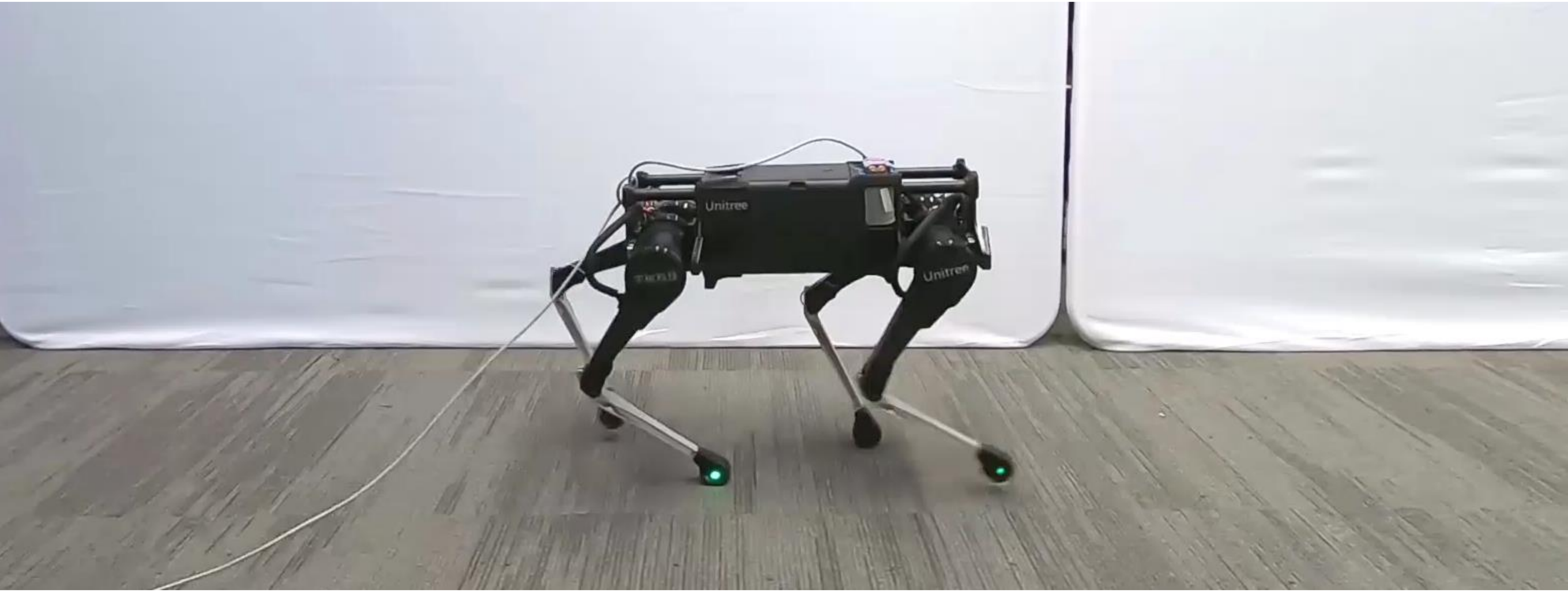
Real Robots



Real Robots

≈

Real Robots



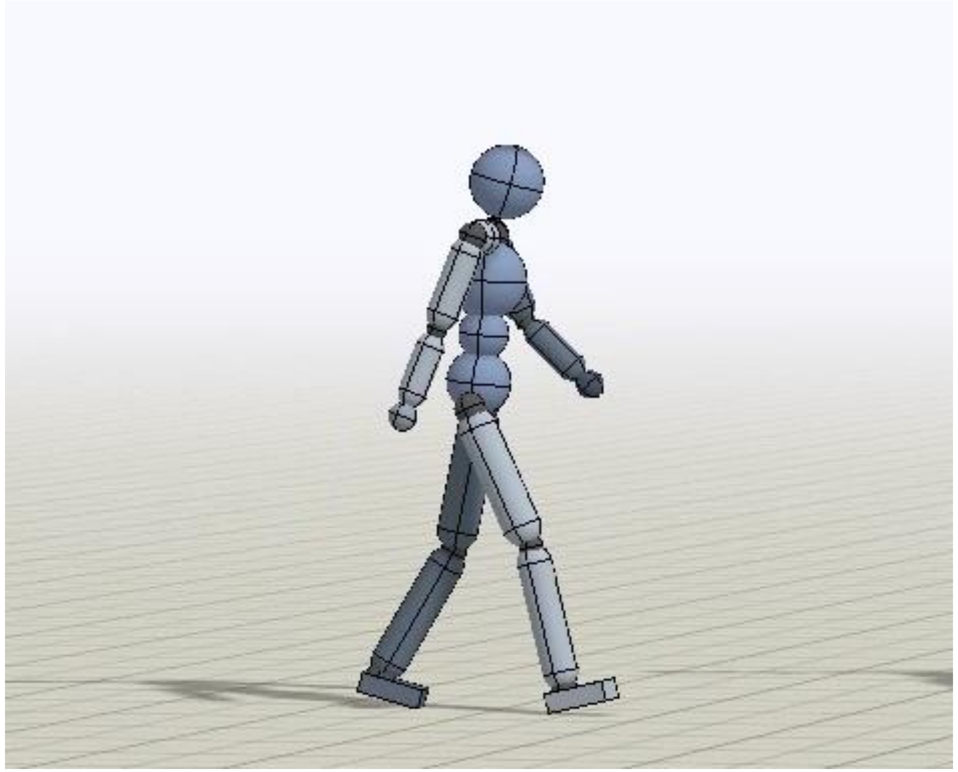
Real-World RL

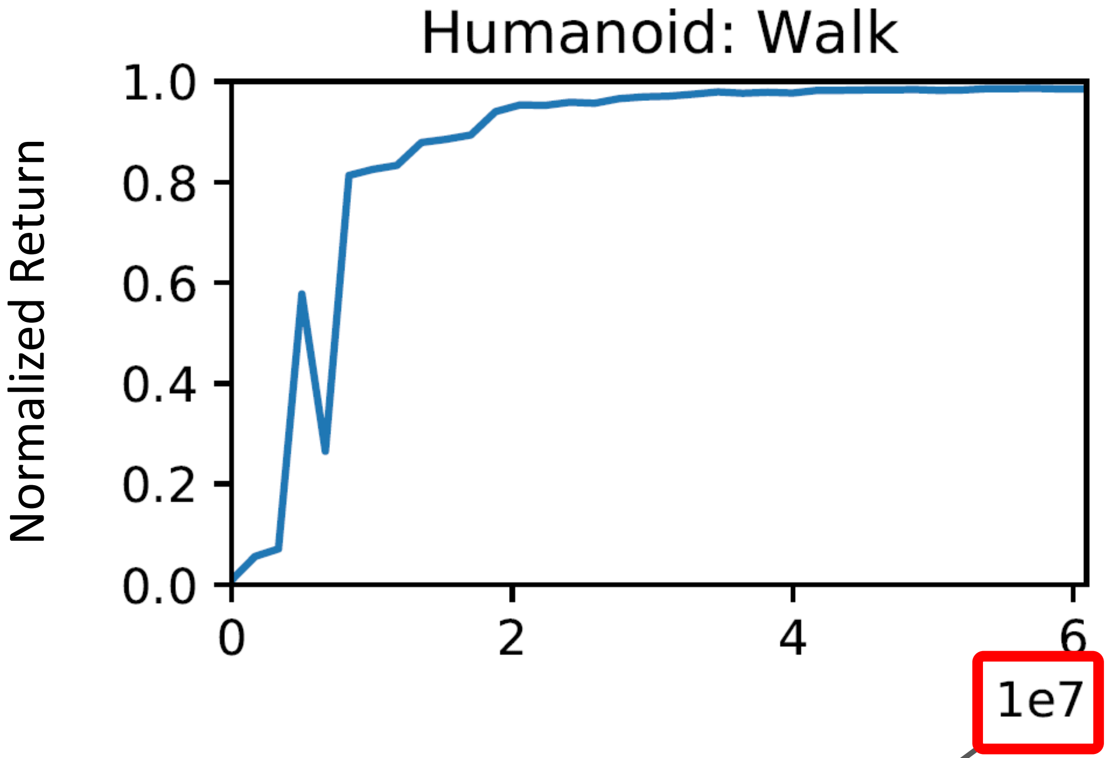
Challenges:

- Sample complexity
- Safety
- State estimation
- Reward calculation
- Episodic resets
- Etc.

The Ingredients of Real-World Robotic Reinforcement Learning
[Zhu et al. 2020]

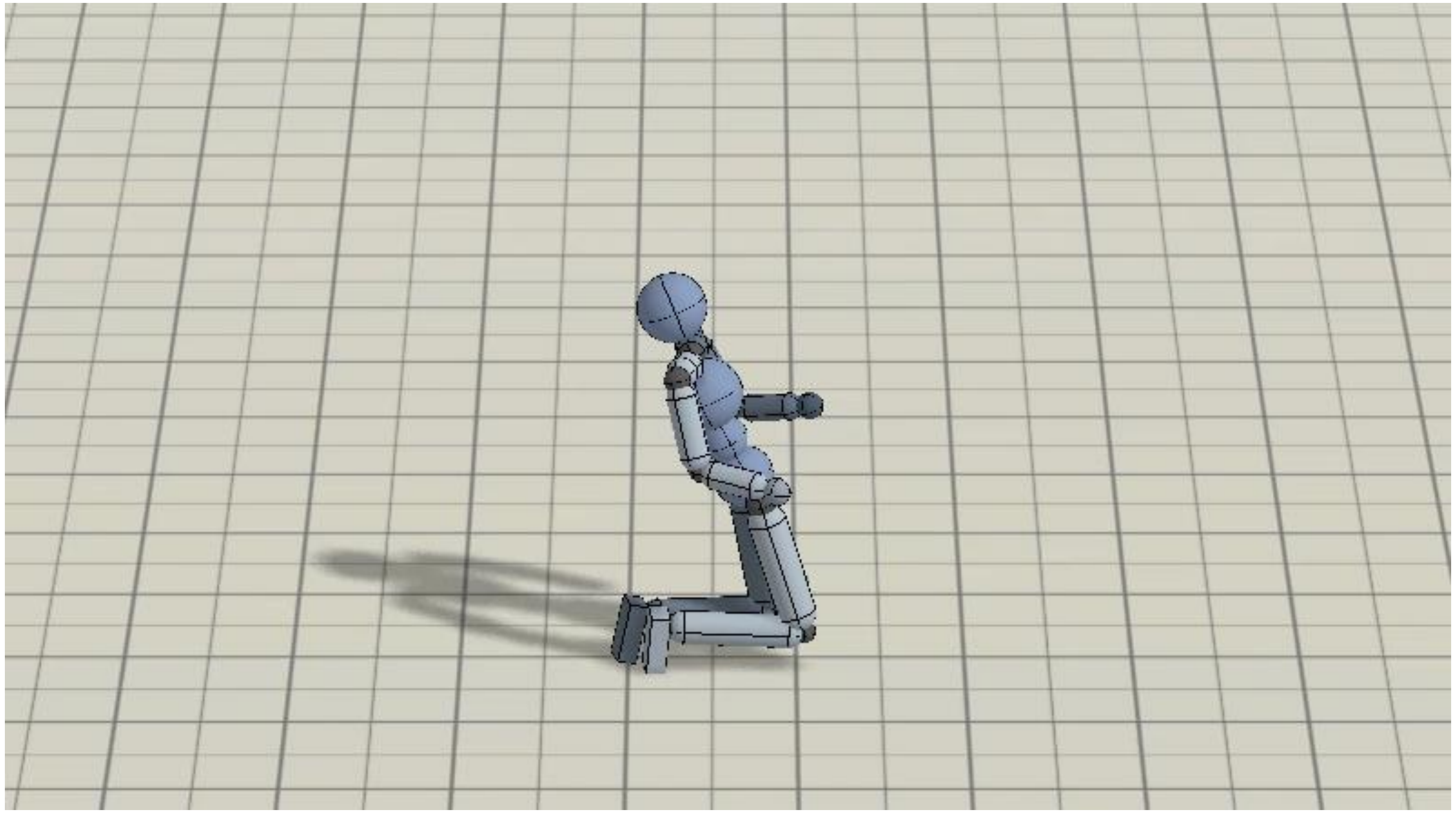
Sample Complexity





~1 month

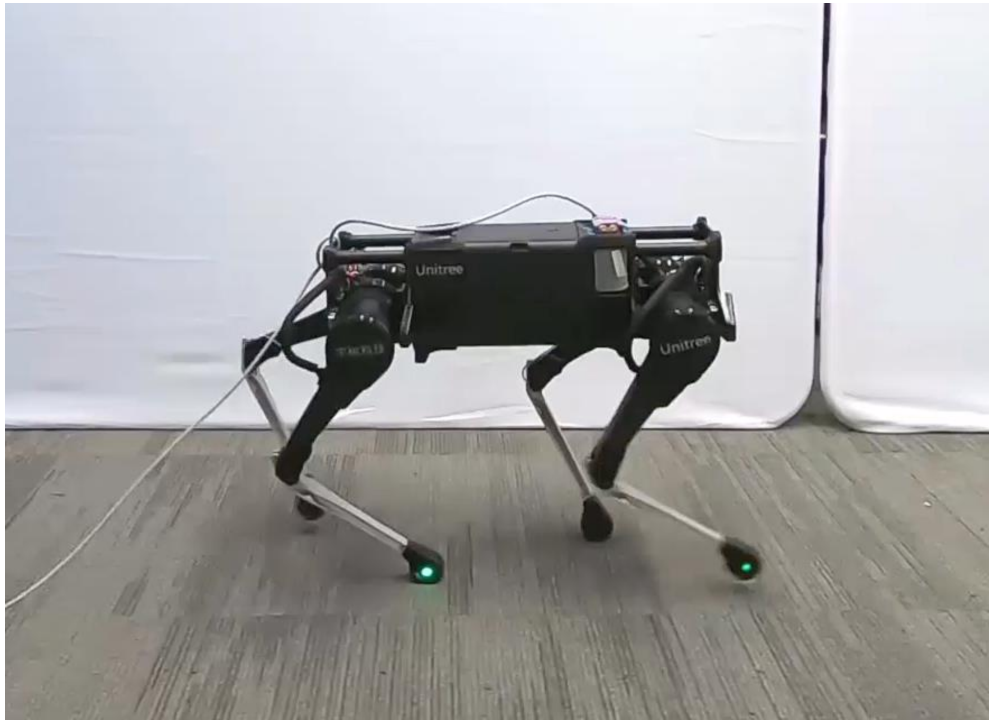
Safety



Random exploration can be dangerous

Sim-to-Real Transfer

Simulation
(Source Domain)

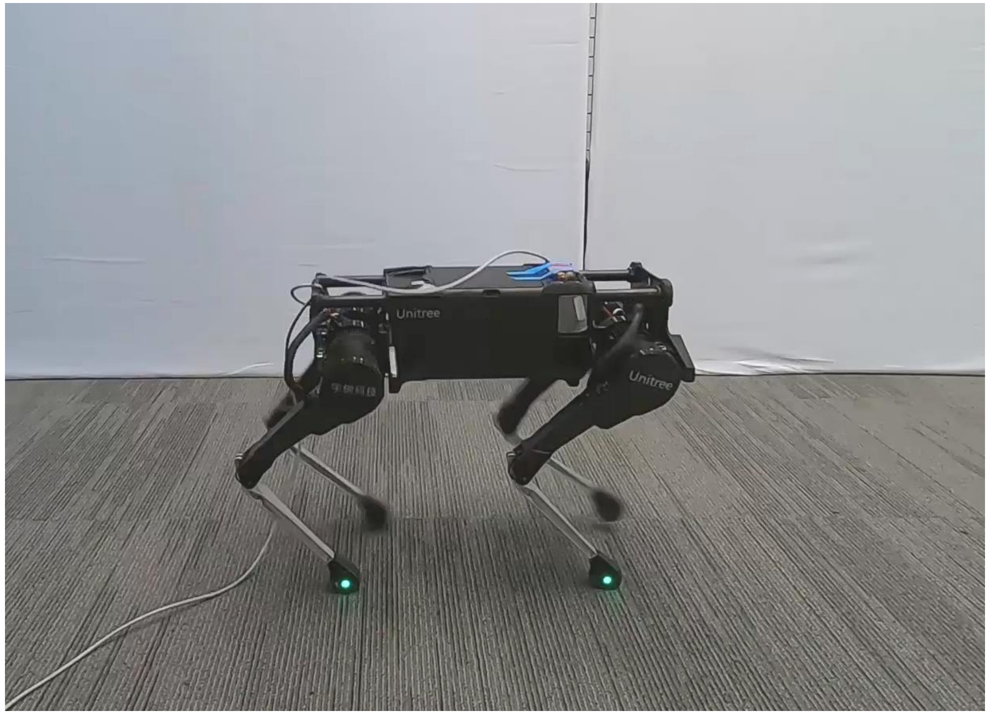


Real World
(Target Domain)

Reality Gap

coarse approximation
of real world

Simulation
(Source Domain)



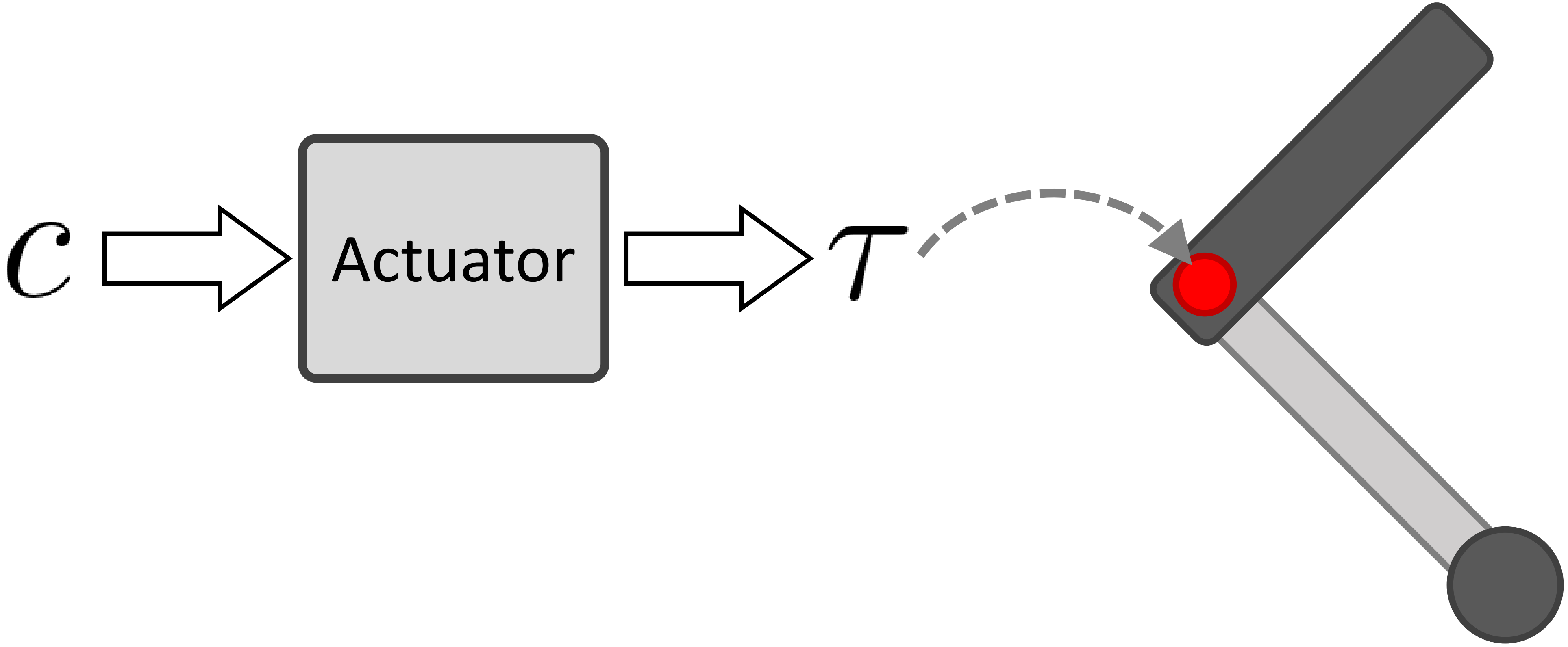
Real World
(Target Domain)

System Identification

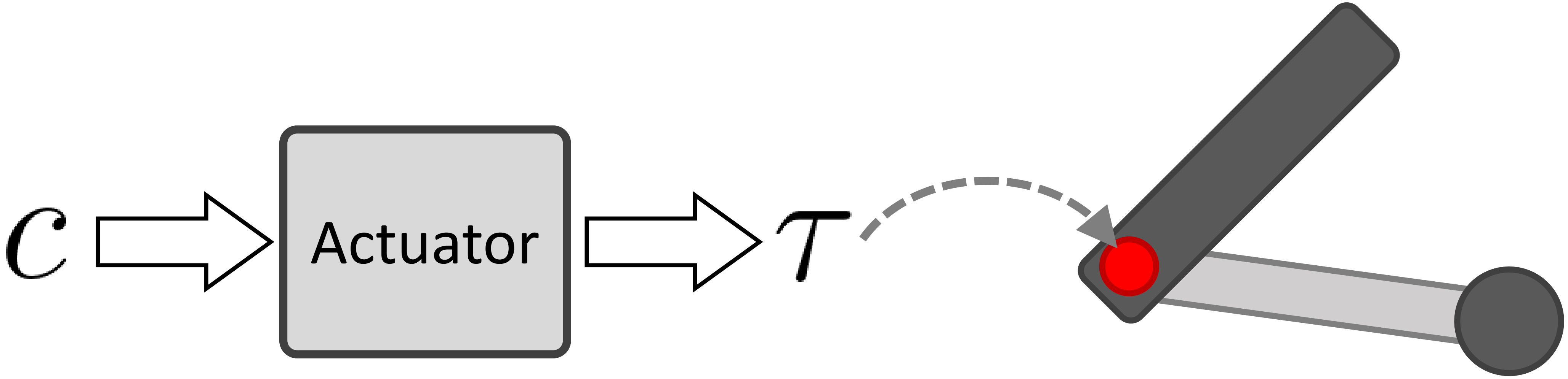
System Identification

Idea: Build a more accurate simulator

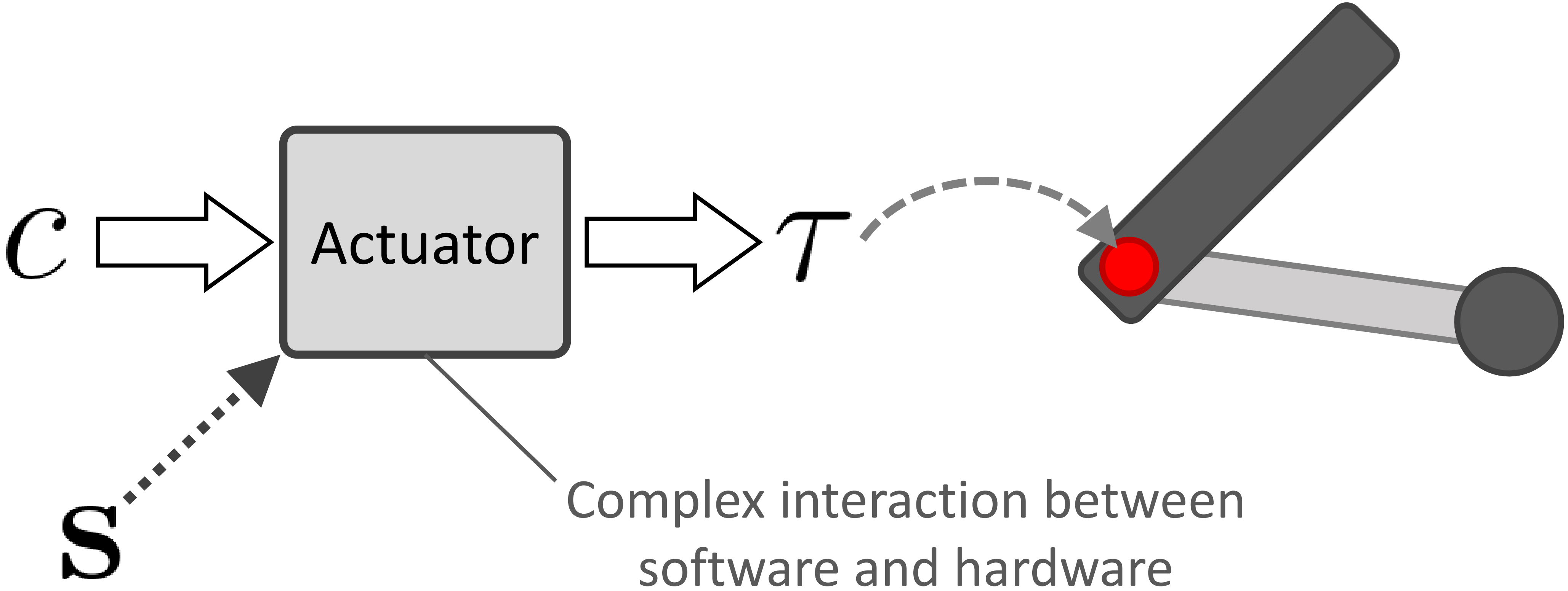
Actuators



Actuators



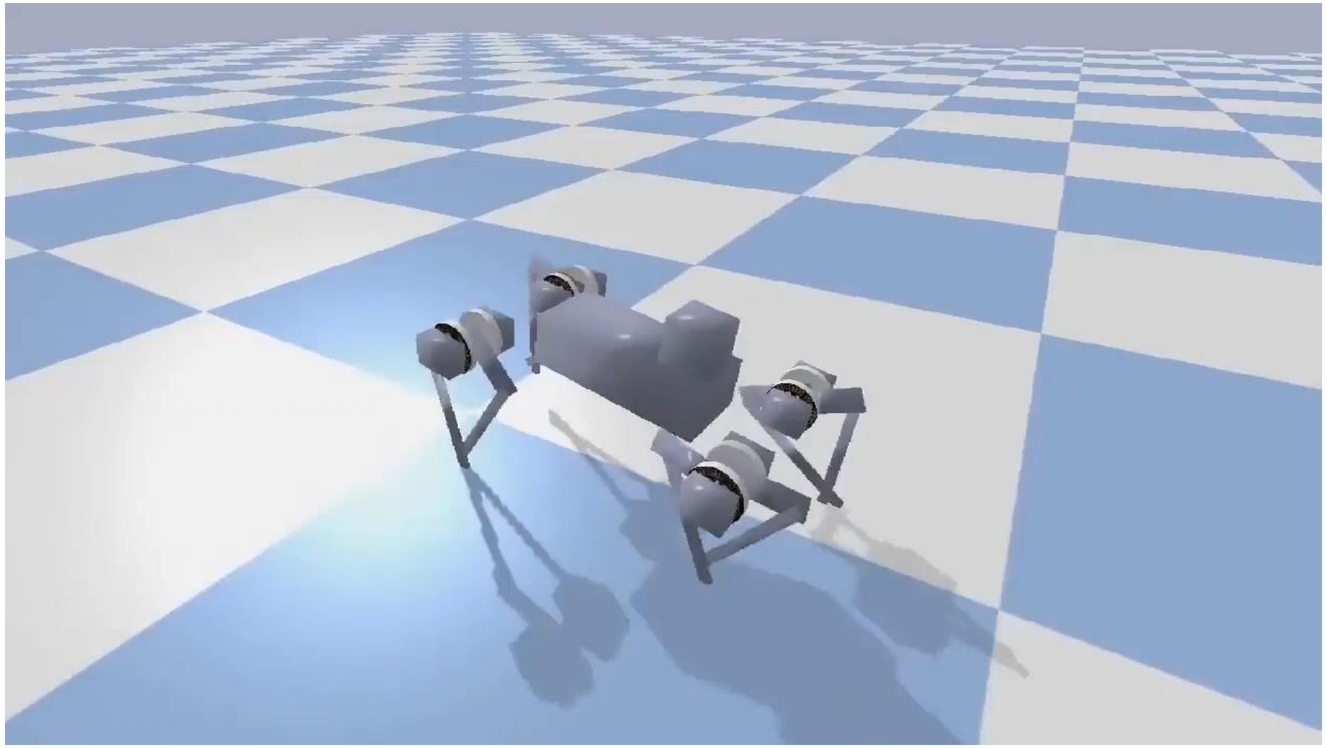
Actuators



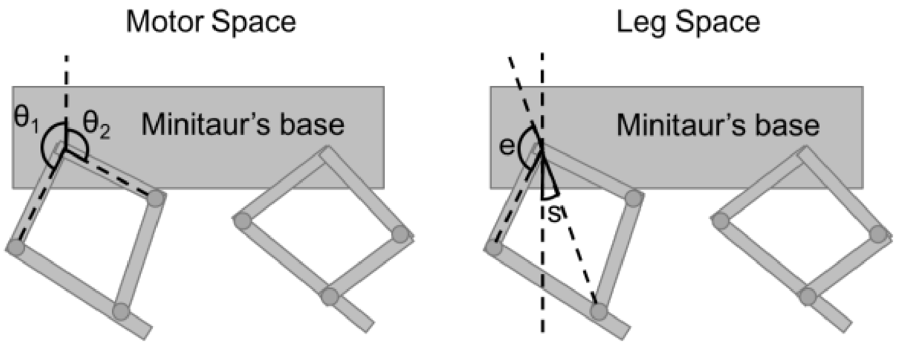
c
Actuator
τ
$\mathbf{s}$
Complex interaction between
software and hardware

System Identification

Idea: Build a more accurate simulator



Actuator Model



$$\tau = K_t I$$

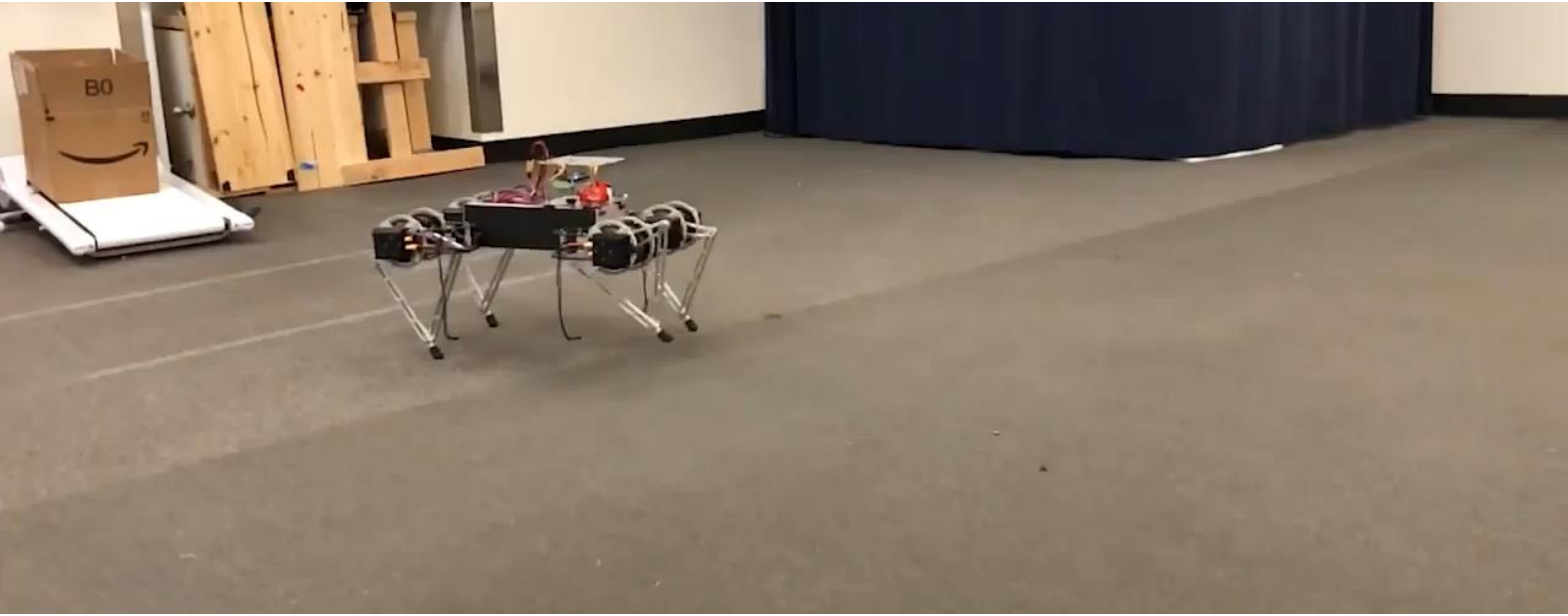
$$I = \frac{V_{\text{pwm}} - V_{\text{emf}}}{R}$$

$$V_{\text{emf}} = K_t \dot{q}$$

$$V_{\text{pwm}} = V(k_p(\bar{q} - q_n) + k_d(\bar{\dot{q}} - \dot{q}_n))$$

Sim-to-Real: Learning Agile Locomotion For Quadruped Robots
[Tan et al. 2018]

System Identification



Sim-to-Real: Learning Agile Locomotion For Quadruped Robots
[Tan et al. 2018]

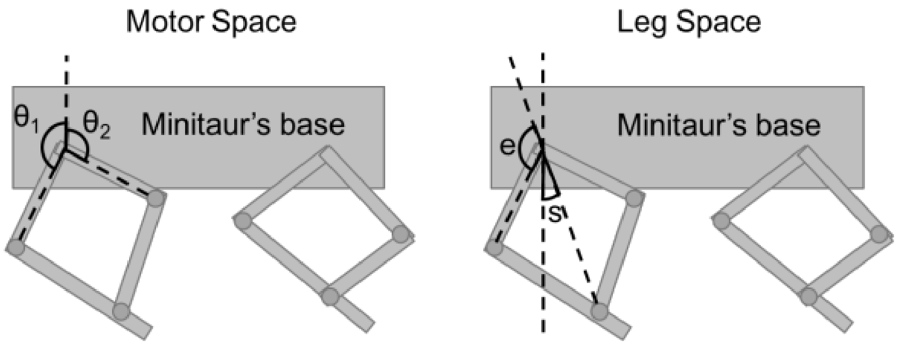
System Identification

Idea: Build a more accurate simulator

- High-fidelity simulators can be hard to build and computationally expensive.

- Can we improve simulator with data?

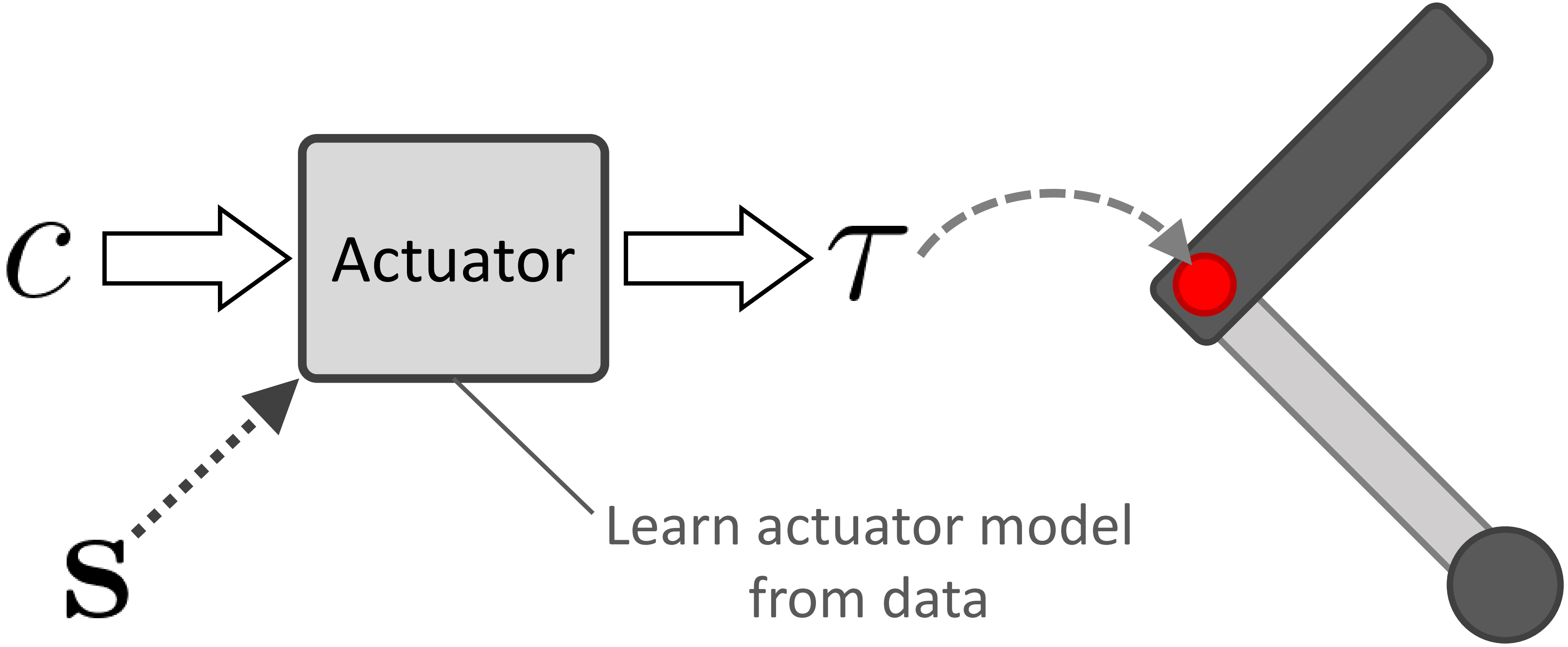
Actuator Model



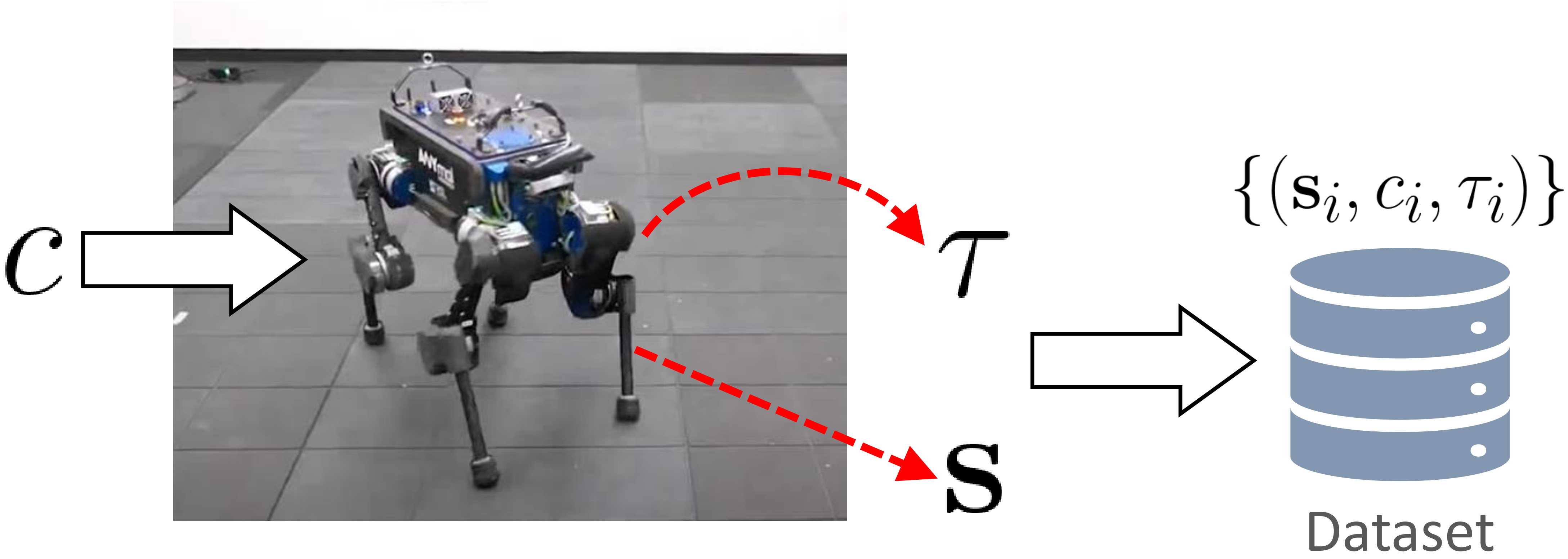
$$
\begin{aligned}
\tau &= K_t I \\
I &= \frac{V_{\text{pwm}} - V_{\text{emf}}}{R} \\
V_{\text{emf}} &= K_t \dot{q}
\end{aligned}
$$

$$V_{\text{pwm}} = V(k_p(\bar{q} - q_n) + k_d(\bar{\dot{q}} - \dot{q}_n))$$

Actuators



Actuator Model

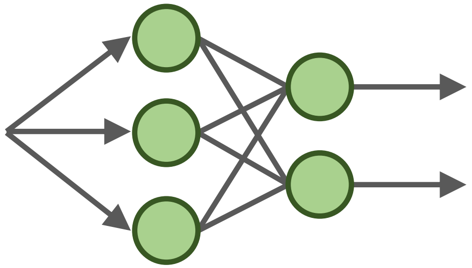


Learning Agile and Dynamic Motor Skills for Legged Robots
[Hwangbo et al. 2019]

Actuator Model

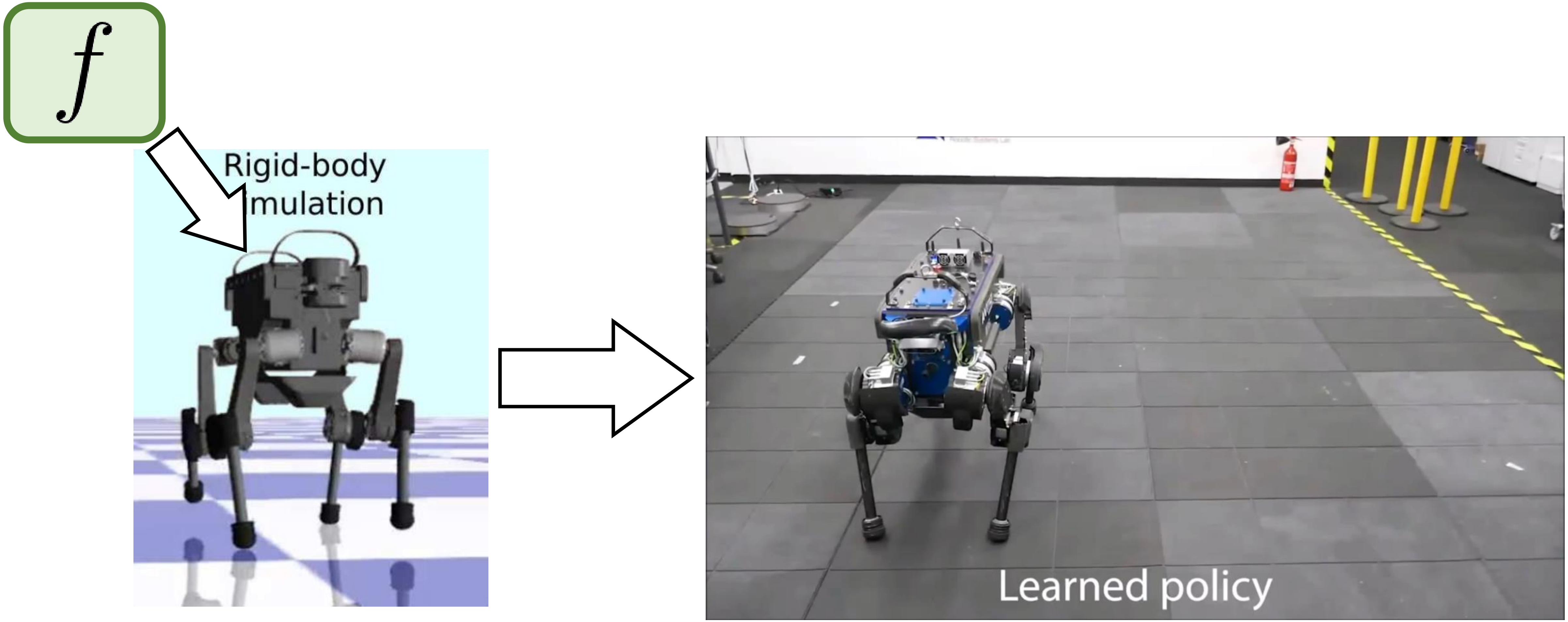
$$\arg\max_{f} \; \mathbb{E}_{(\mathbf{s}_i, c_i, \tau_i) \sim \mathcal{D}} \left[\log \underline{f(\tau_i | \mathbf{s}_i, c_i)} \right]$$

actuator model



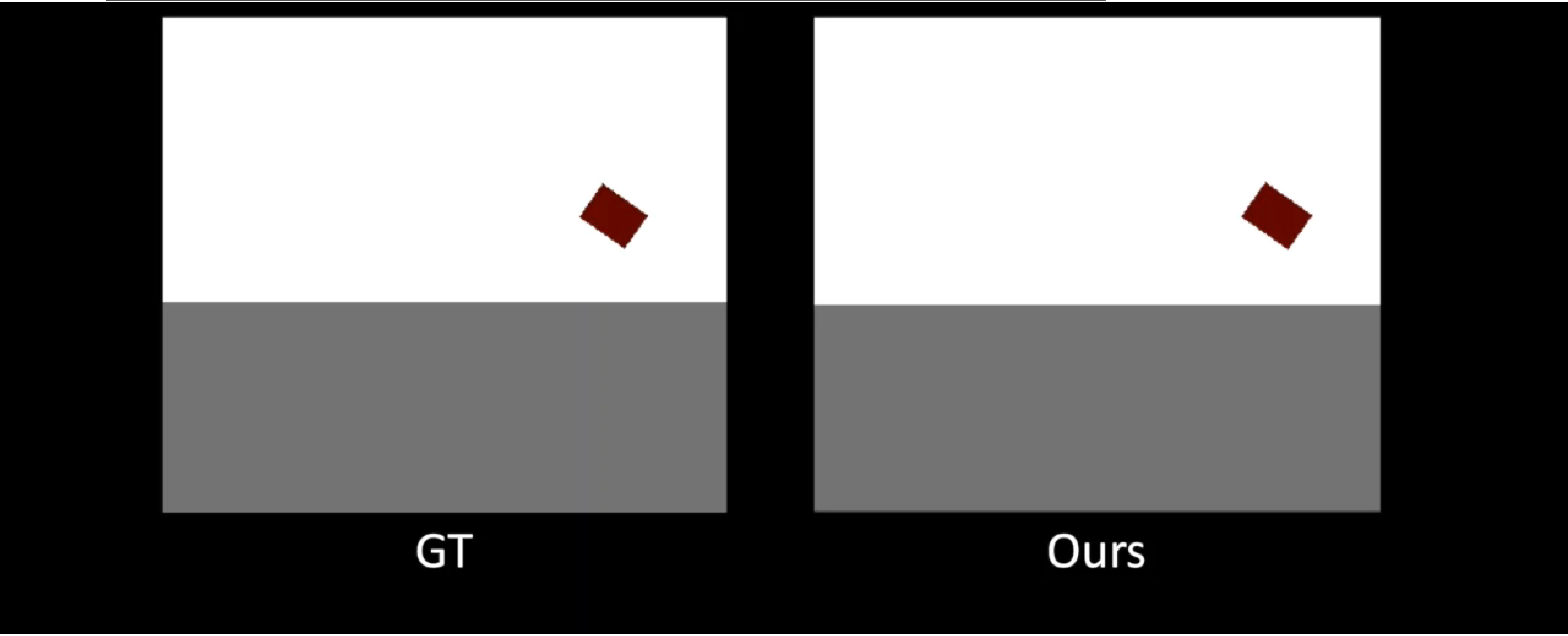
Learning Agile and Dynamic Motor Skills for Legged Robots
[Hwangbo et al. 2019]

Actuator Model



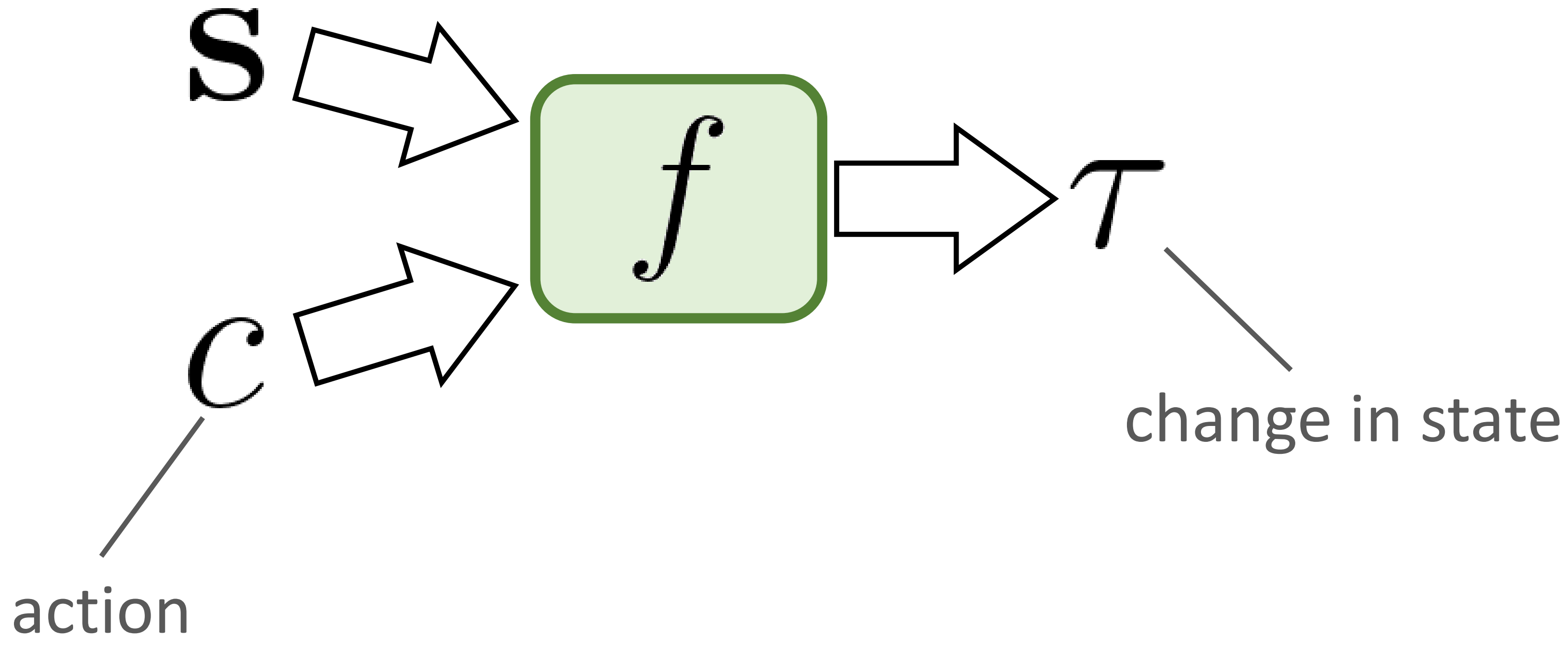
Learning Agile and Dynamic Motor Skills for Legged Robots
[Hwangbo et al. 2019]

System Identification

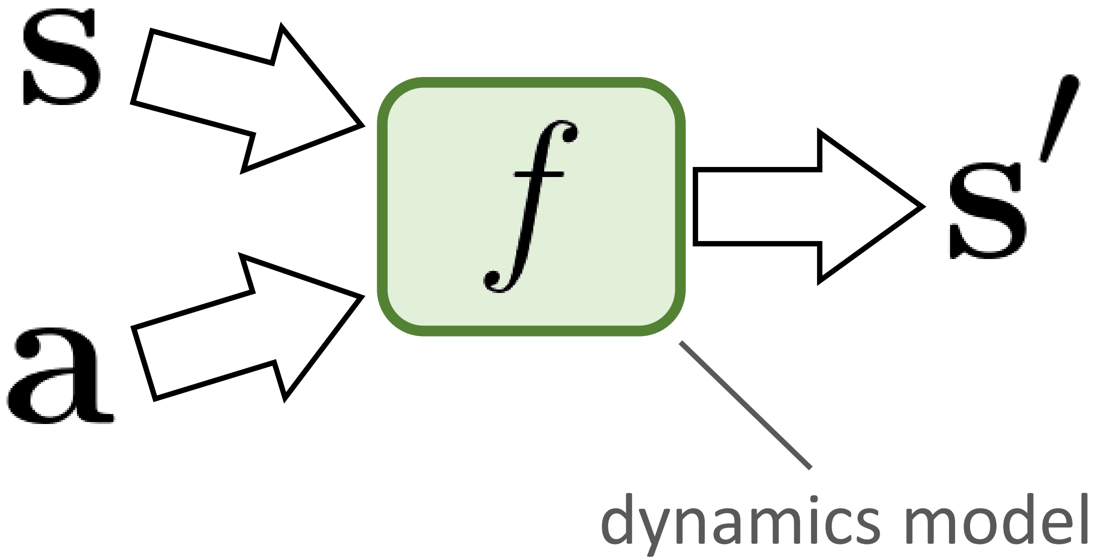


Data-Augmented Contact Model for Rigid Body Simulation
[Jiang et al. 2022]

Actuator Model

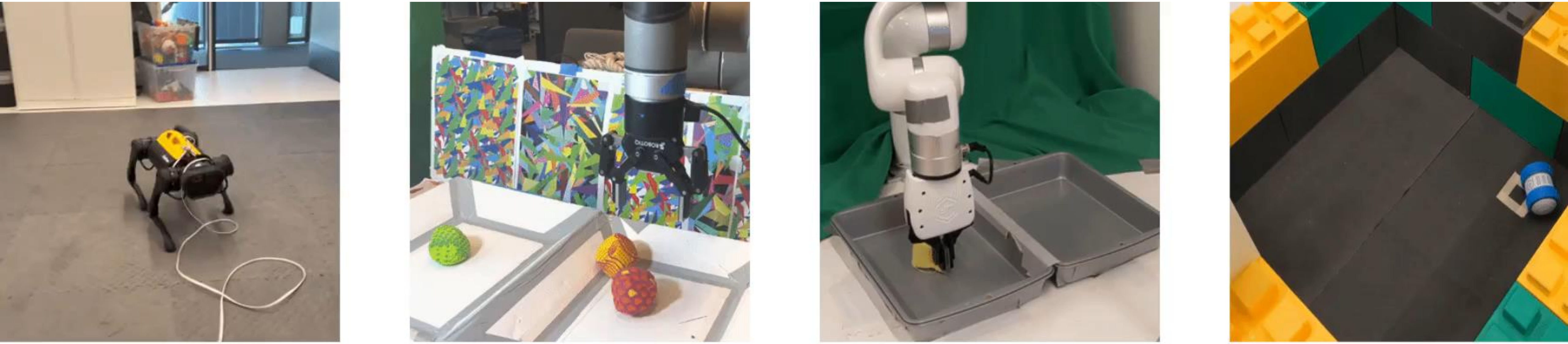


Actuator Model



Why not learn the whole simulator?

Model-Based RL



A1 Quadruped Walking

UR5 Multi-Object Visual Pick Place

XArm Visual Pick and Place

Sphero Ollie Visual Navigation

DayDreamer: World Models for Physical Robot Learning
[Wu et al. 2022]

Dynamics Models

System ID

- Learn subset of the dynamics
- Fewer parameters
- More domain knowledge
- Better generalization

Model-Based RL

- Learn full dynamics
- More parameters
- Less domain knowledge
- More prone to OOD errors

Domain Randomization

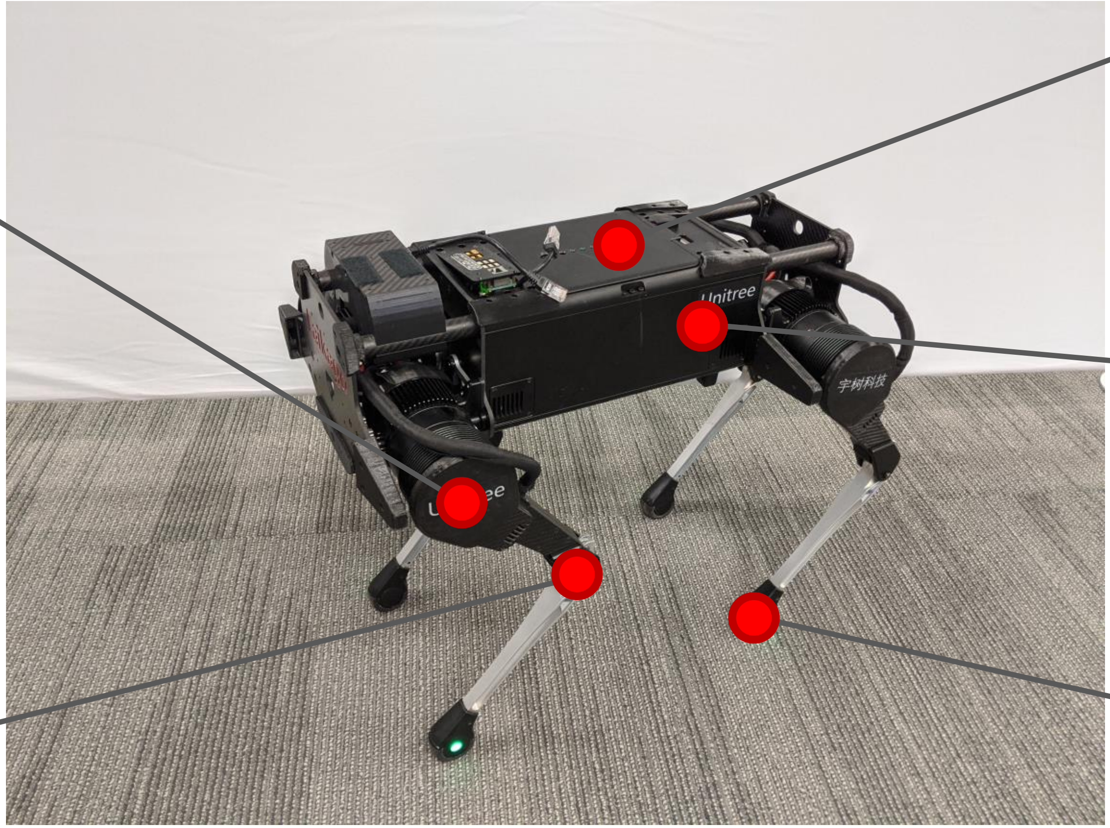
Domain Randomization

- Developing accurate simulators can be very difficult
  - Real world has a lot of unmodeled effects

Domain Randomization:

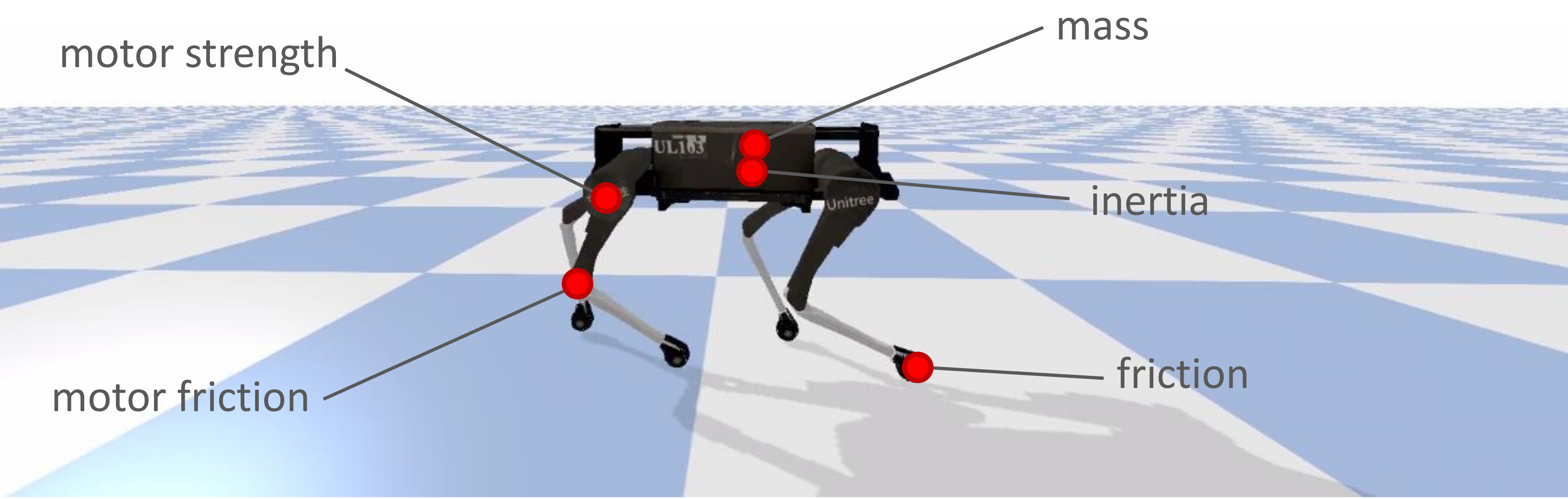
- instead of developing more **accurate** simulators, develop more **robust** policies

Domain Randomization



Domain Randomization

- Simulate potential variations in the dynamics
- Train policy to be robust to these variations



Domain Randomization

$$p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$$

Domain Randomization

$$p(\mathbf{s}'|\mathbf{s}, \mathbf{a}, \underline{\mu})$$

dynamics parameters
(e.g. mass, inertia, friction, etc.)

Domain Randomization

$$p(\mathbf{s}'|\mathbf{s}, \mathbf{a}, \underline{\mu}^*)$$

ground-truth
dynamics parameters

Domain Randomization

$$p(\mathbf{s}'|\mathbf{s},\mathbf{a},\underline{\mu^*})$$

ground-truth
dynamics parameters

Mass
Ground-truth: m^*
Estimate: $[m^l, m^h]$

Domain Randomization

$$p(\mathbf{s}'|\mathbf{s}, \mathbf{a}, \mu)$$

$$\mu \sim p(\mu)$$ randomization distribution

Mass
Ground-truth: m^*
Estimate: $[m^l, m^h]$

Domain Randomization

$$J(\pi) = \mathbb{E}_{\tau \sim p(\tau | \pi, \mu^*)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right]$$

real-world dynamics

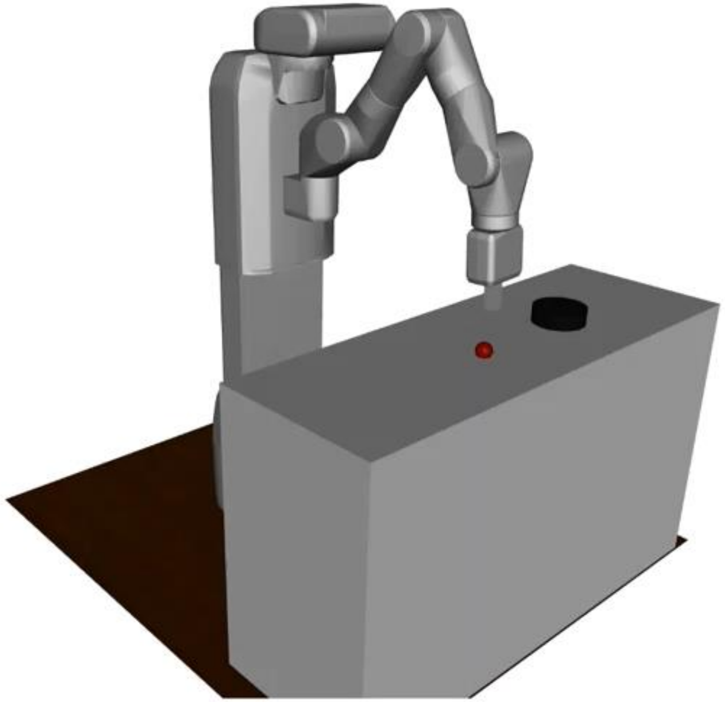
Domain Randomization

$$J(\pi) = \mathbb{E}_{\tau \sim p(\tau|\pi,\mu^*)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right]$$

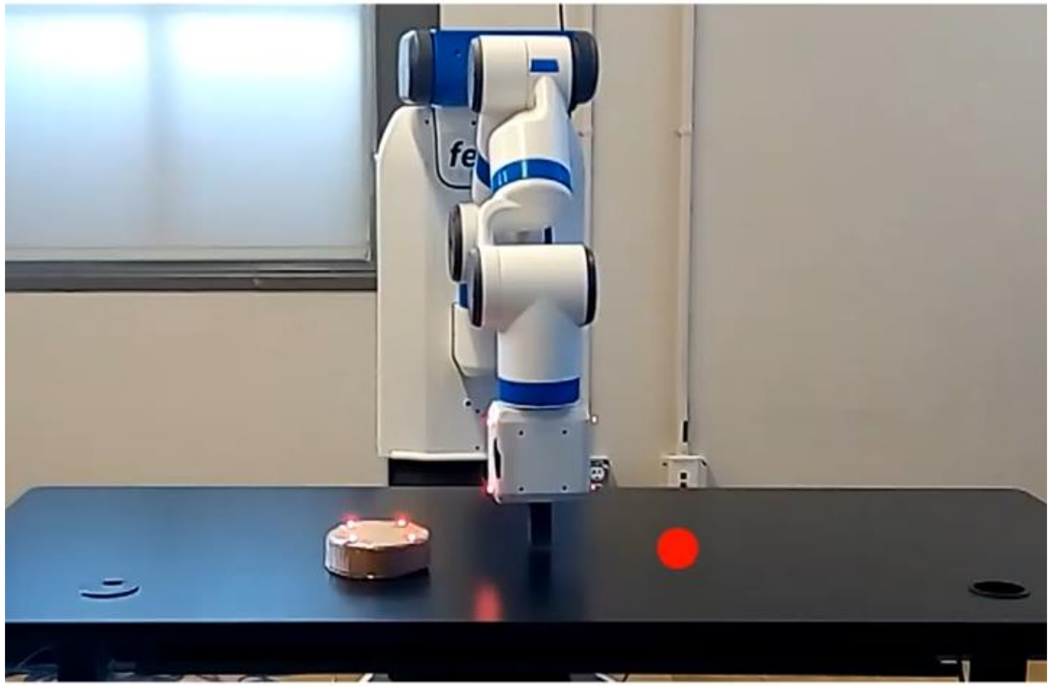
$$\hat{J}(\pi) = \mathbb{E}_{\underline{\mu \sim p(\mu)}} \mathbb{E}_{\tau \sim p(\tau|\pi,\underline{\mu})} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right]$$

Optimize performance across uncertain dynamics

Domain Randomization



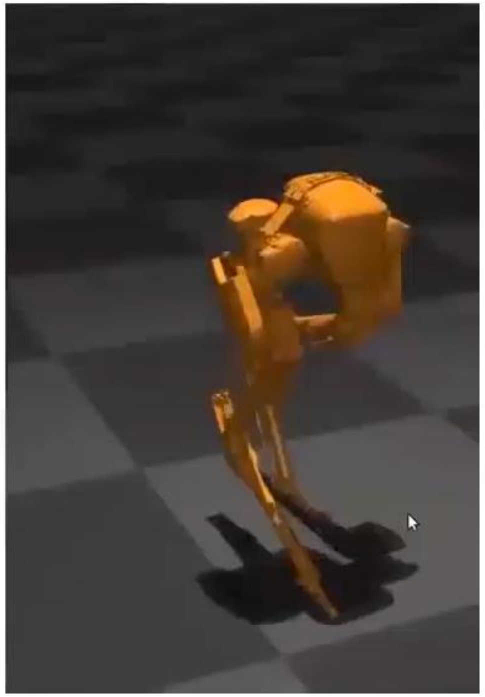
Simulation



Real World

Sim-to-Real Transfer of Robotic Control with Dynamics Randomization
[Peng et al. 2018]

Domain Randomization

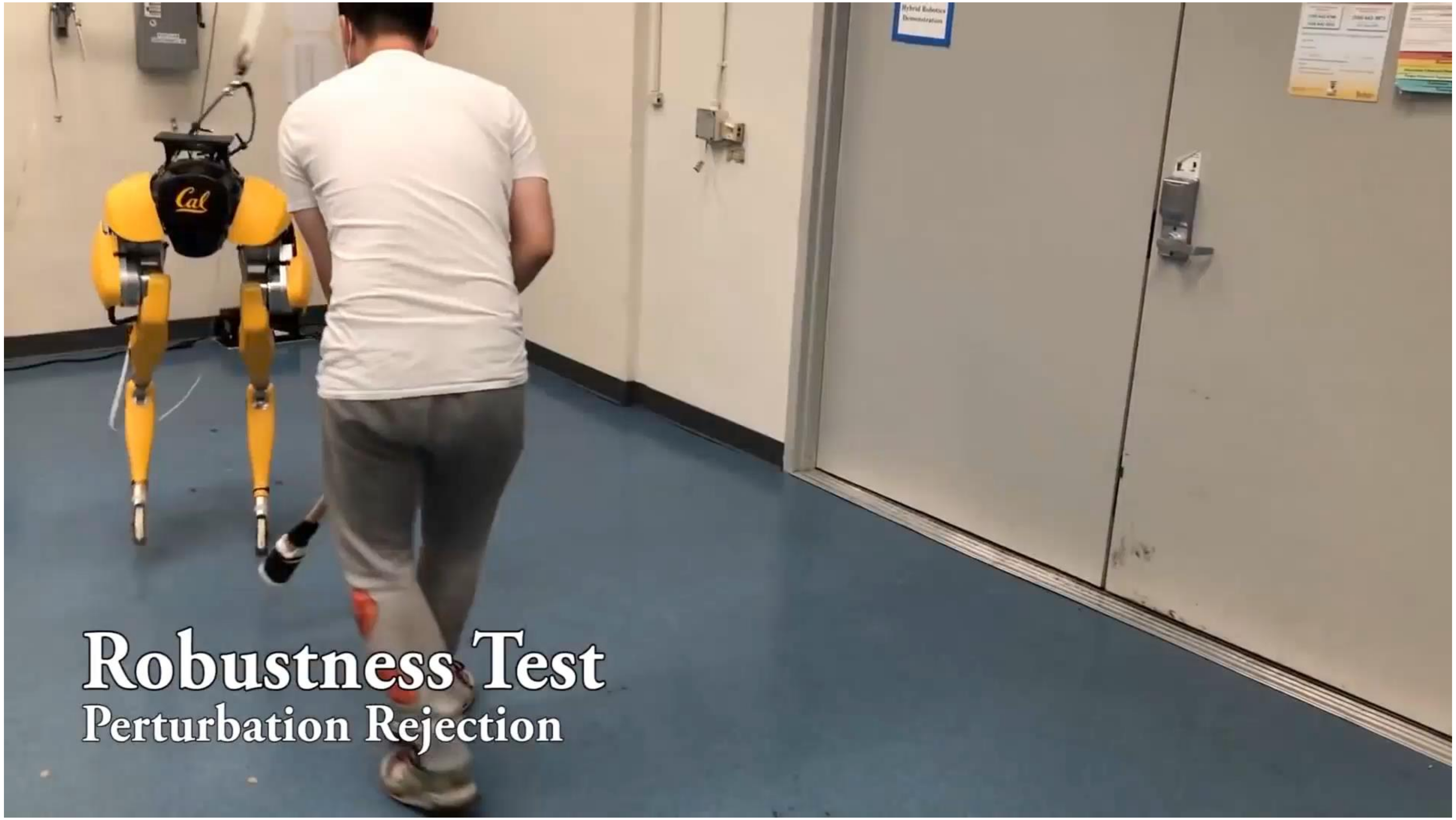


Simulation

Real World

Reinforcement Learning for Robust Parameterized Locomotion Control of Bipedal Robots
[Li et al. 2021]

Robust Policies



Reinforcement Learning for Robust Parameterized Locomotion Control of Bipedal Robots
[Li et al. 2021]

Randomization Distribution

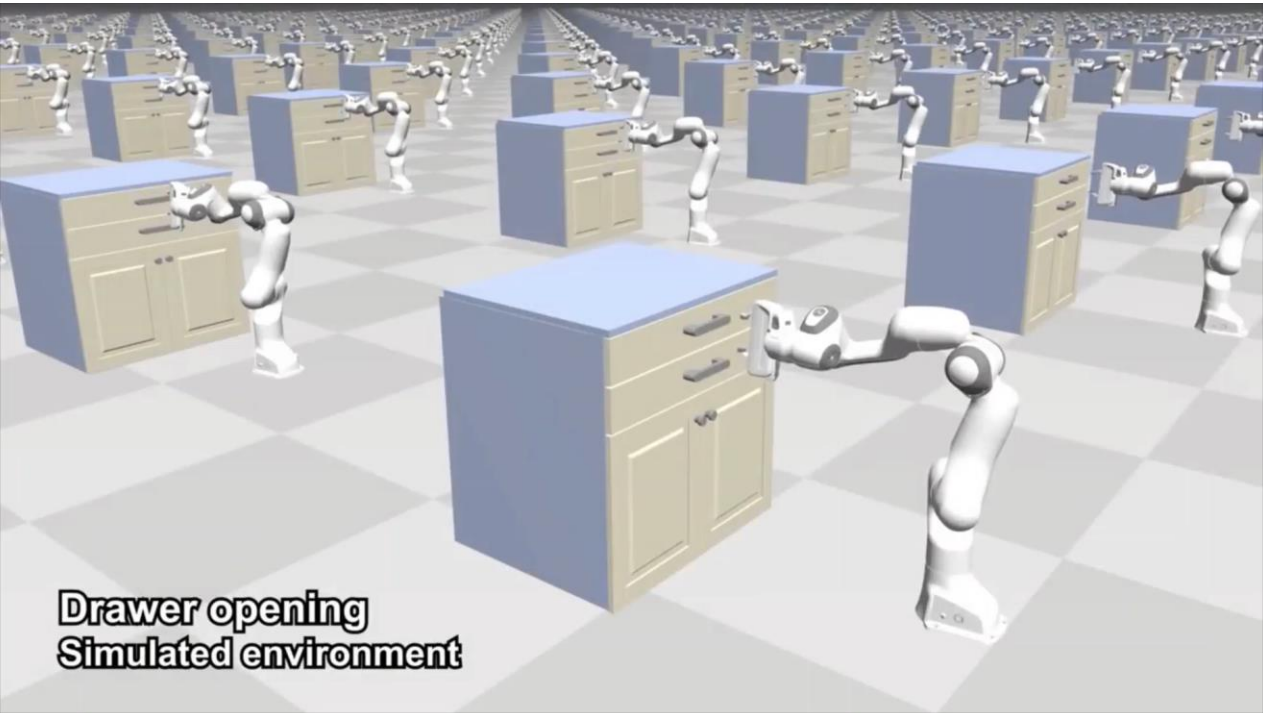
$$\hat{J}(\pi) = \mathbb{E}_{\mu \sim \underline{p(\mu)}} \mathbb{E}_{\tau \sim p(\tau|\pi,\mu)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right]$$

How to pick randomization distribution?

Diversity > Accuracy:

- Use a sufficiently large randomization range, such that the policy can cope with variations in real-world dynamics (even unmodeled effects)
- Adapt randomization distribution with real-world data

Adaptive Domain Randomization

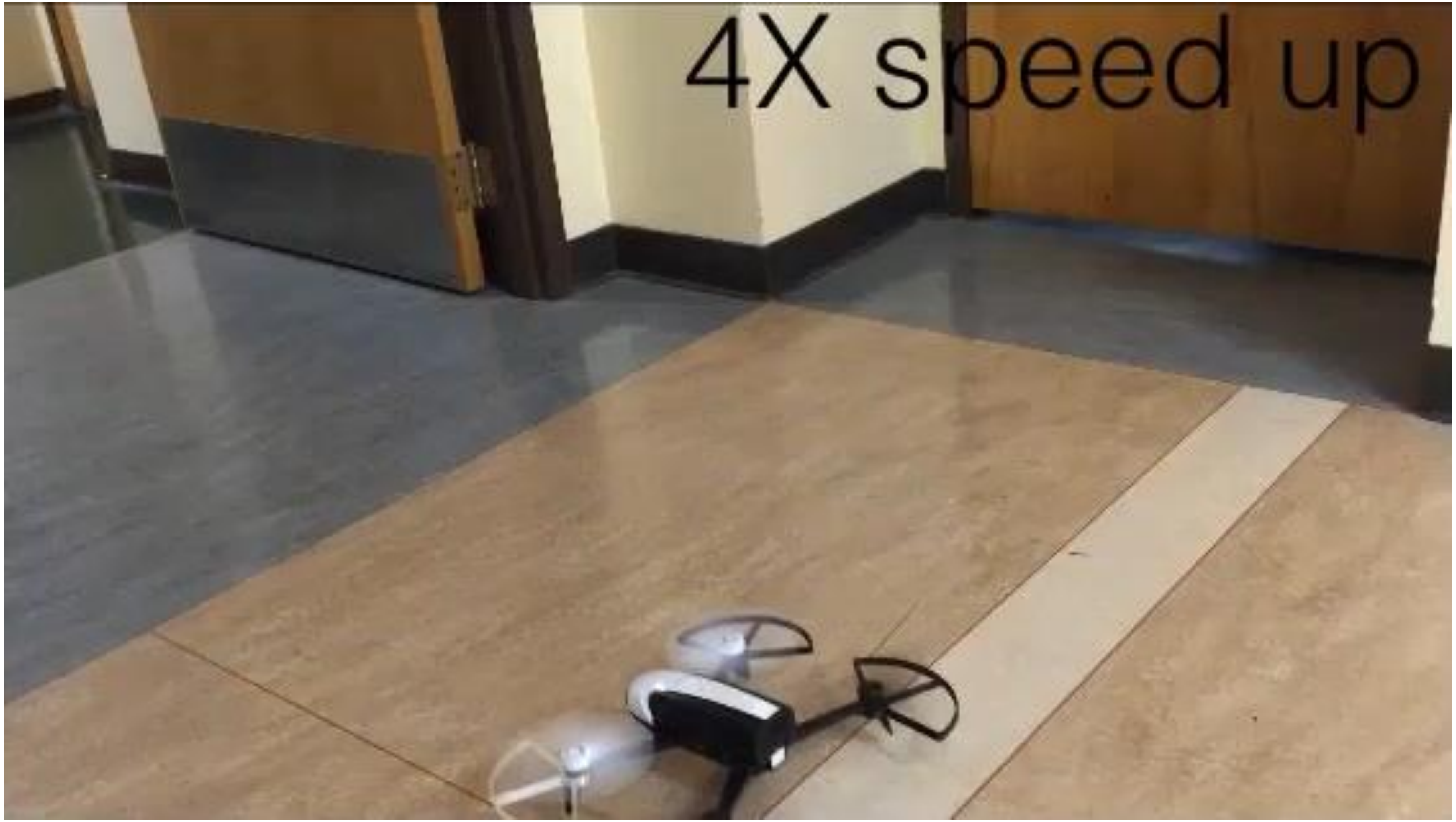


Simulation

Real World

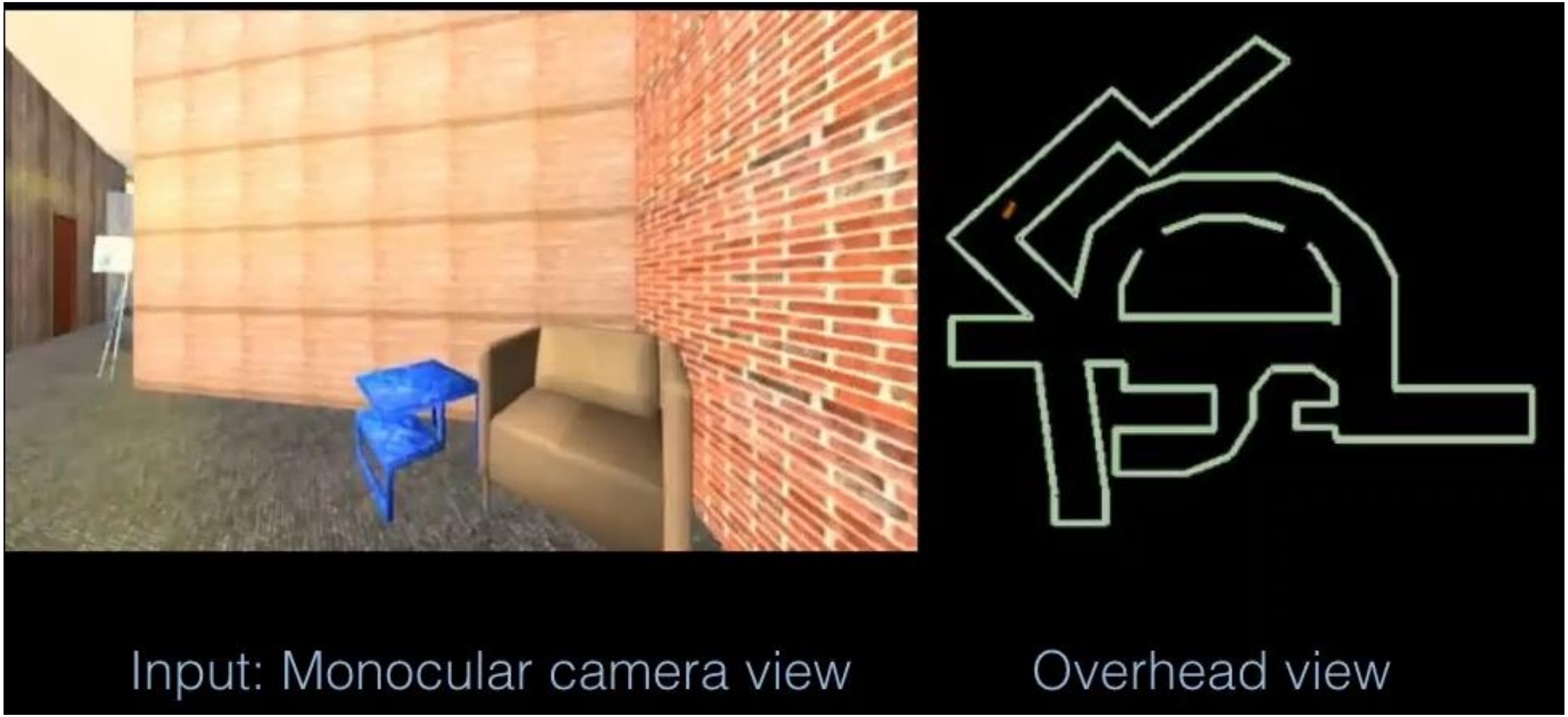
Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience [Chebotar et al. 2019]

Visual Navigation



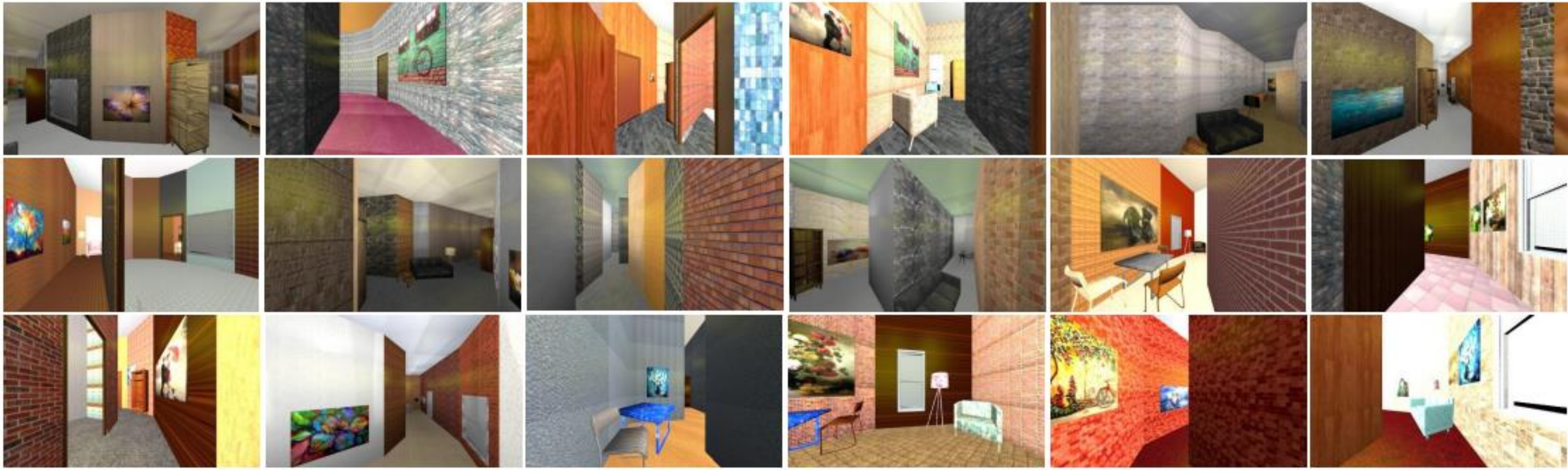
CAD2RL: Real Single-Image Flight Without a Single Real Image
[Sadeghi et al. 2016]

Visual Navigation



CAD2RL: Real Single-Image Flight Without a Single Real Image
[Sadeghi et al. 2016]

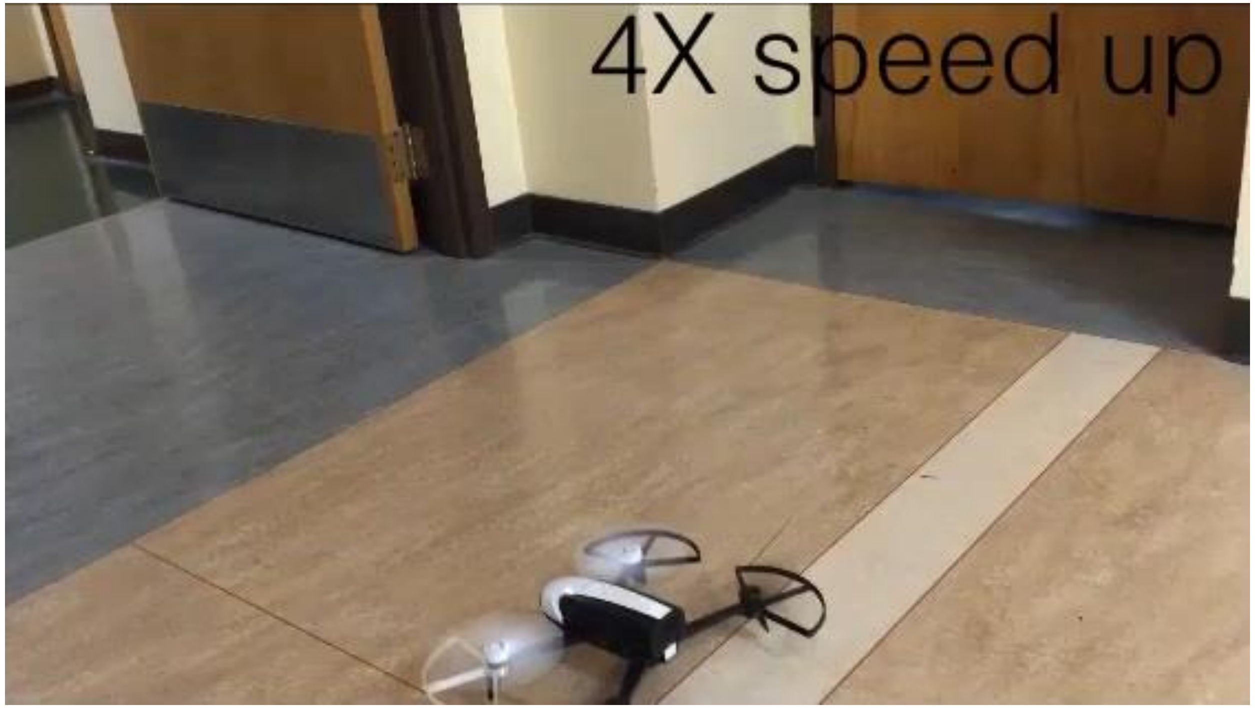
Visual Randomization



CAD2RL: Real Single-Image Flight Without a Single Real Image
[Sadeghi et al. 2016]

Visual Navigation

Camera View



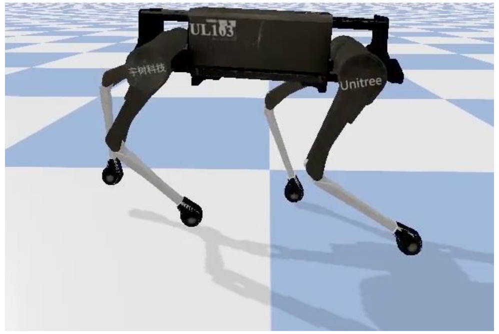
Third-Person View

CAD2RL: Real Single-Image Flight Without a Single Real Image
[Sadeghi et al. 2016]

Over-Conservatism

$$\hat{J}(\pi) = \mathbb{E}_{\mu \sim \underline{p(\mu)}} \mathbb{E}_{\tau \sim p(\tau|\pi,\mu)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right]$$

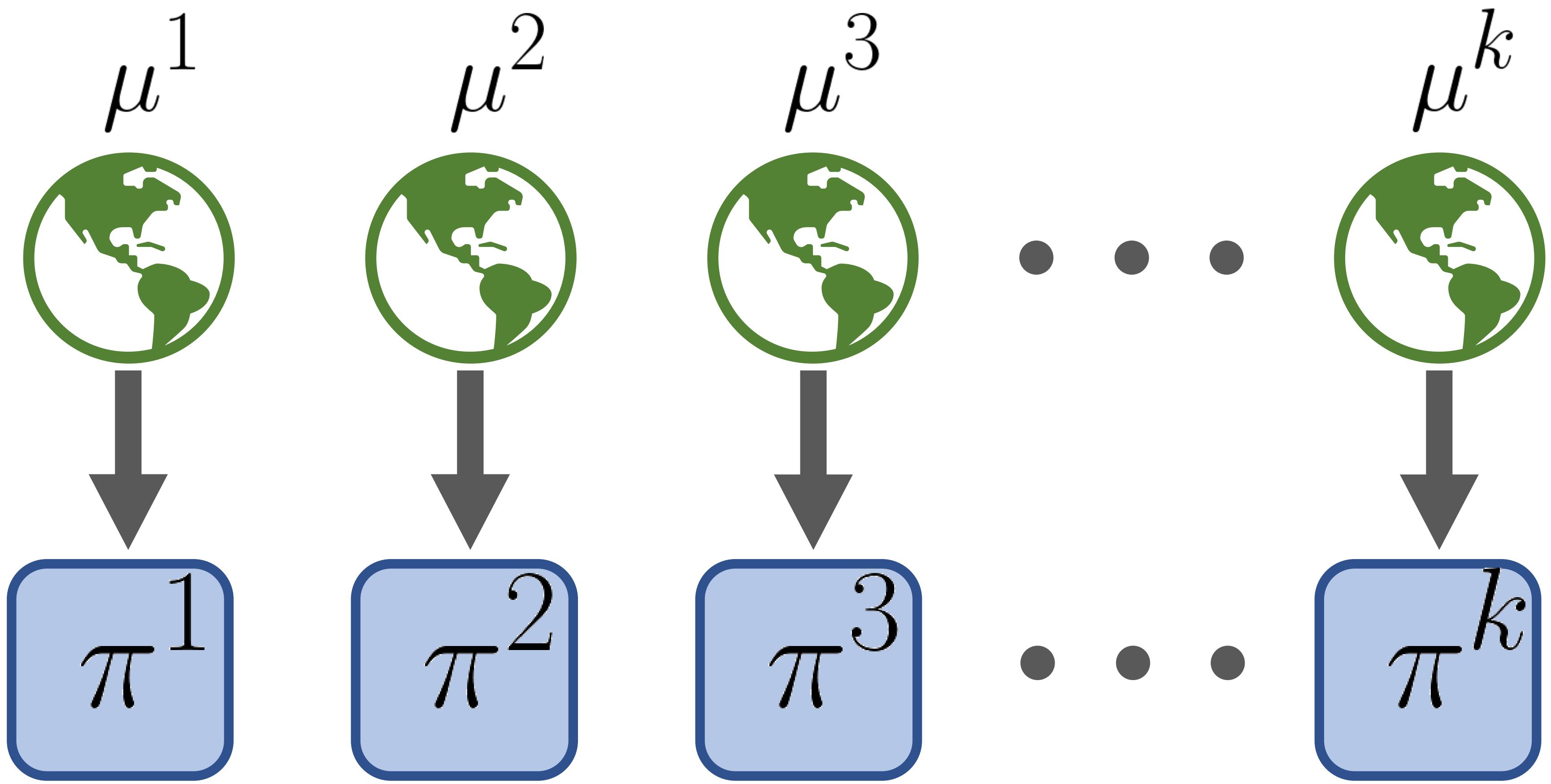
Randomization can lead to
overly conservative behaviors



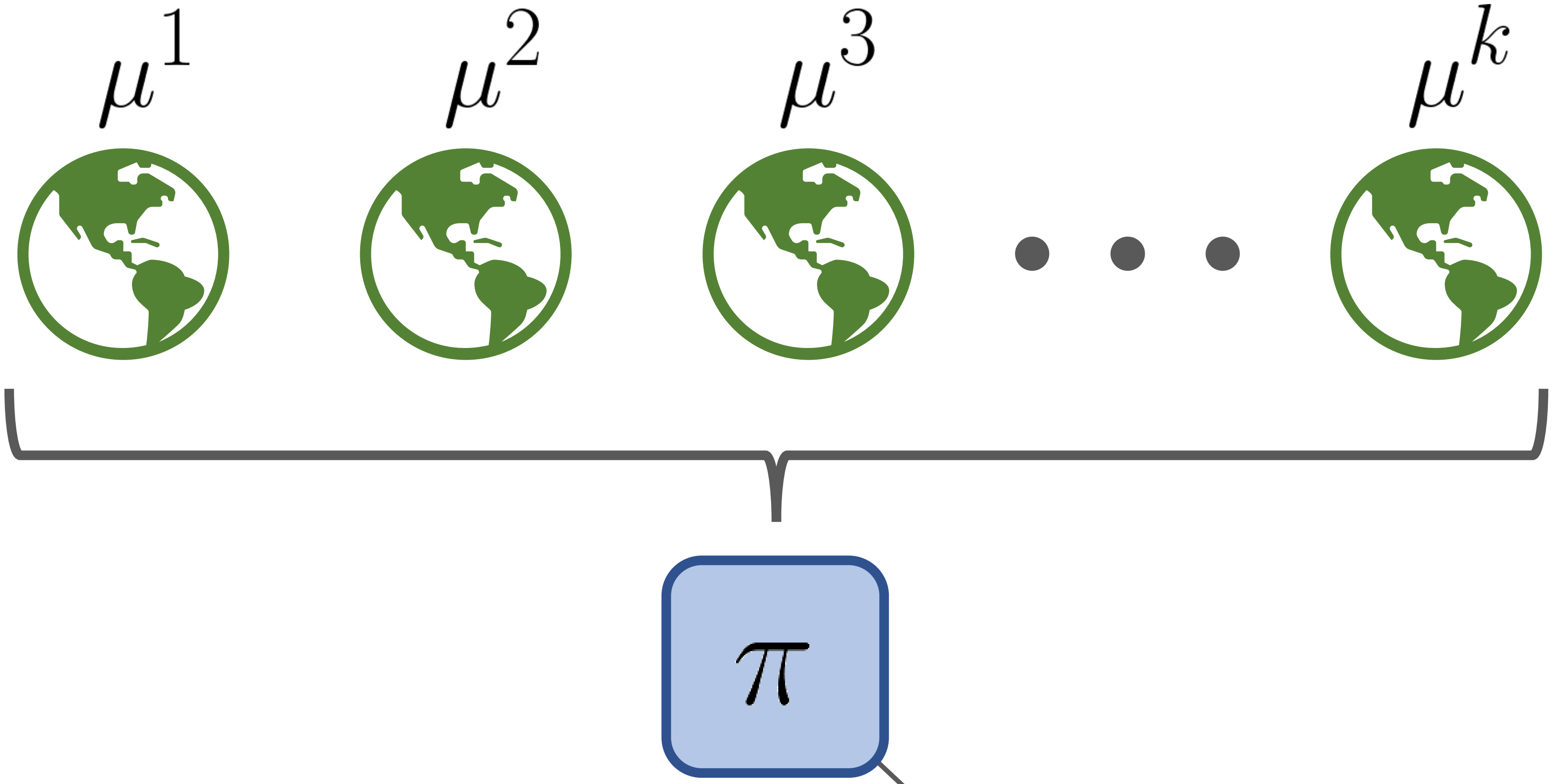
Robustness
(more randomization)

Optimality
(less randomization)

Over-Conservatism



Over-Conservatism



There may be no single policy
that is optimal for all environments

Domain Adaptation

Domain Adaptation

- Adjust behavior of the policy according to environment
  - Online system identification
  - Adaptive strategy
  - Finetuning

Amortized Models

$$\hat{J}(\underline{\pi}) = \mathbb{E}_{\mu \sim p(\mu)} \mathbb{E}_{\tau \sim p(\tau|\pi,\mu)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right]$$

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

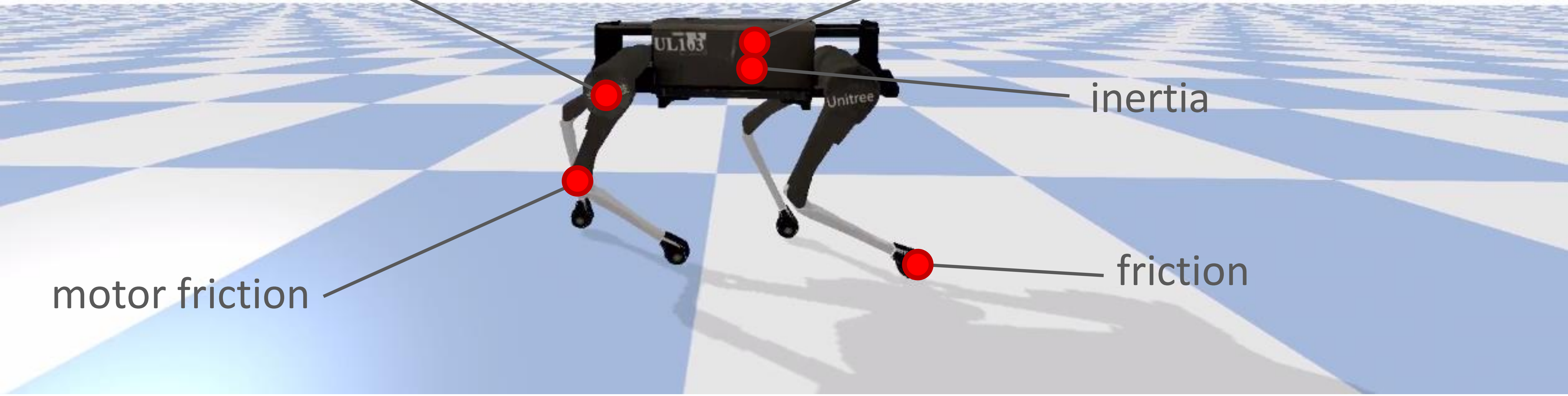
Amortized Models

$$\hat{J}(\underline{\pi}) = \mathbb{E}_{\mu \sim p(\mu)} \mathbb{E}_{\tau \sim p(\tau|\pi,\mu)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right]$$

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

- Directly condition policy on dynamics parameters
- Transfer to new environment:
  - Identify dynamics parameters that best characterizes new environment

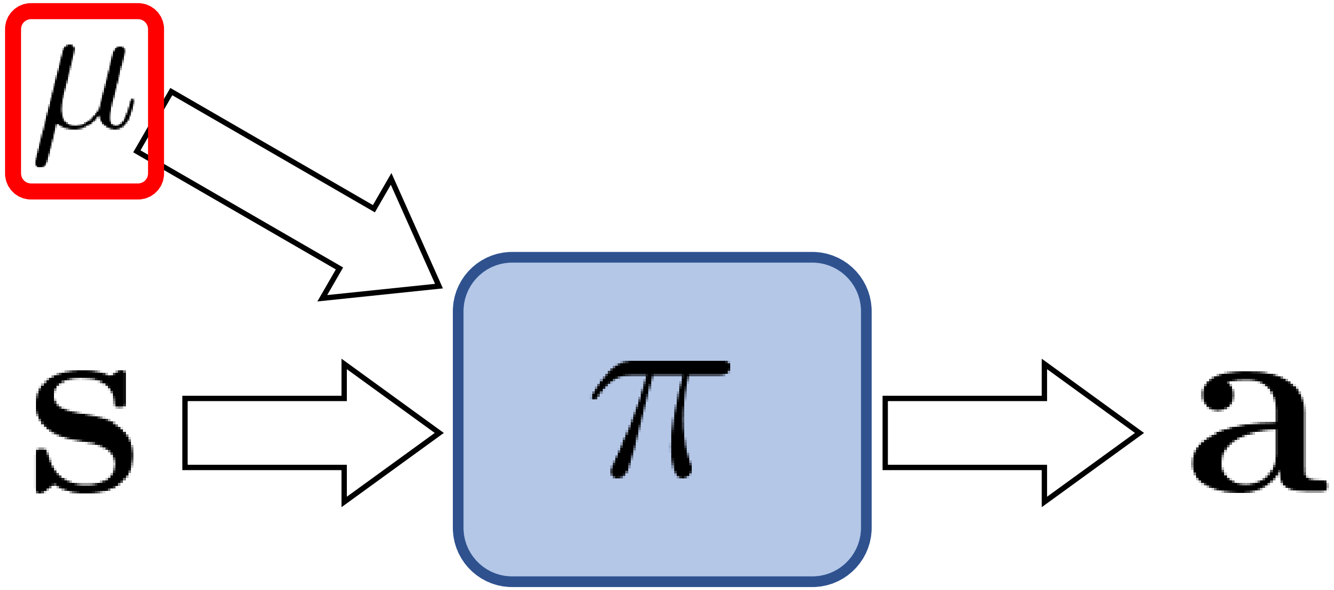
Universal Policies



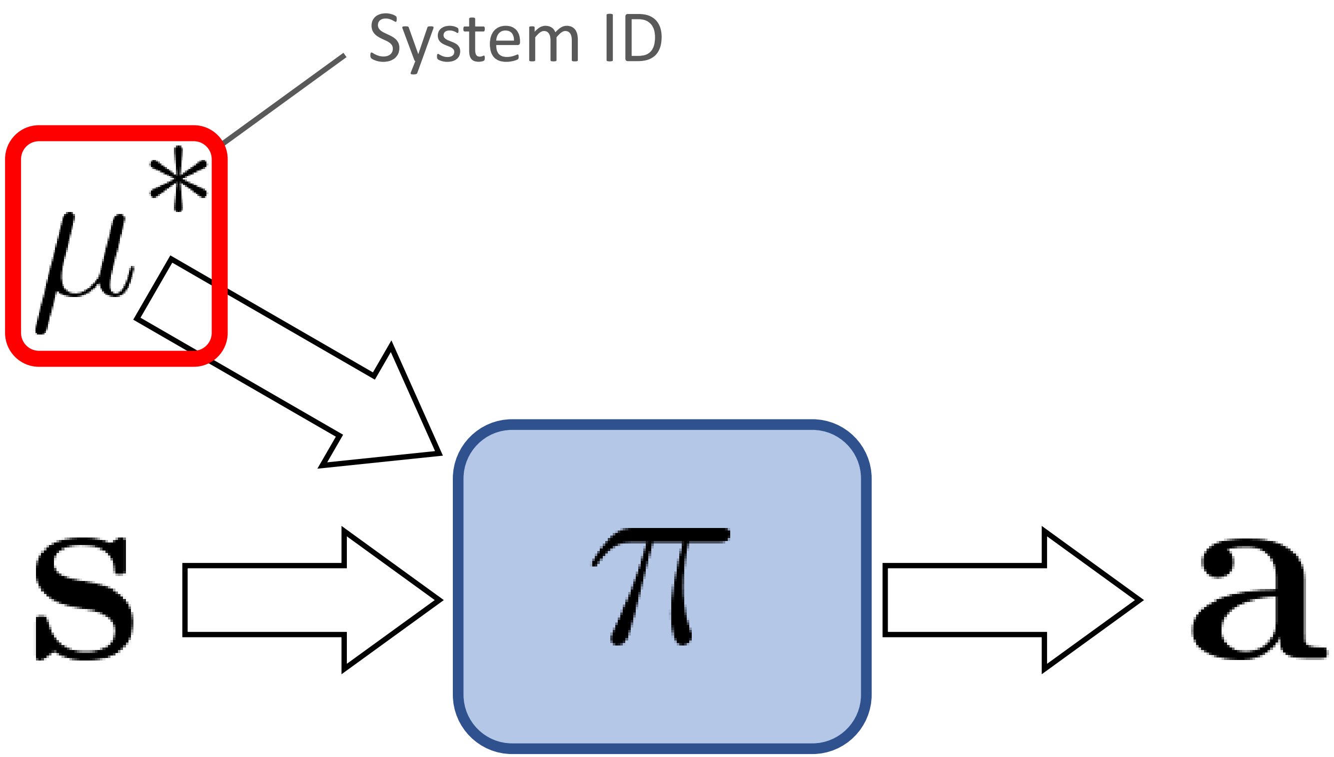
Universal Policies

$$\mu = \begin{bmatrix} \text{mass} \\ \text{inertia} \\ \text{friction} \\ \text{motor strength} \\ \text{motor friction} \\ \text{Etc.} \end{bmatrix}$$

Universal Policies



Universal Policies



Learning SystemID

Forward-dynamics:

$$p(\mathbf{s}'|\mathbf{s}, \mathbf{a}, \mu)$$

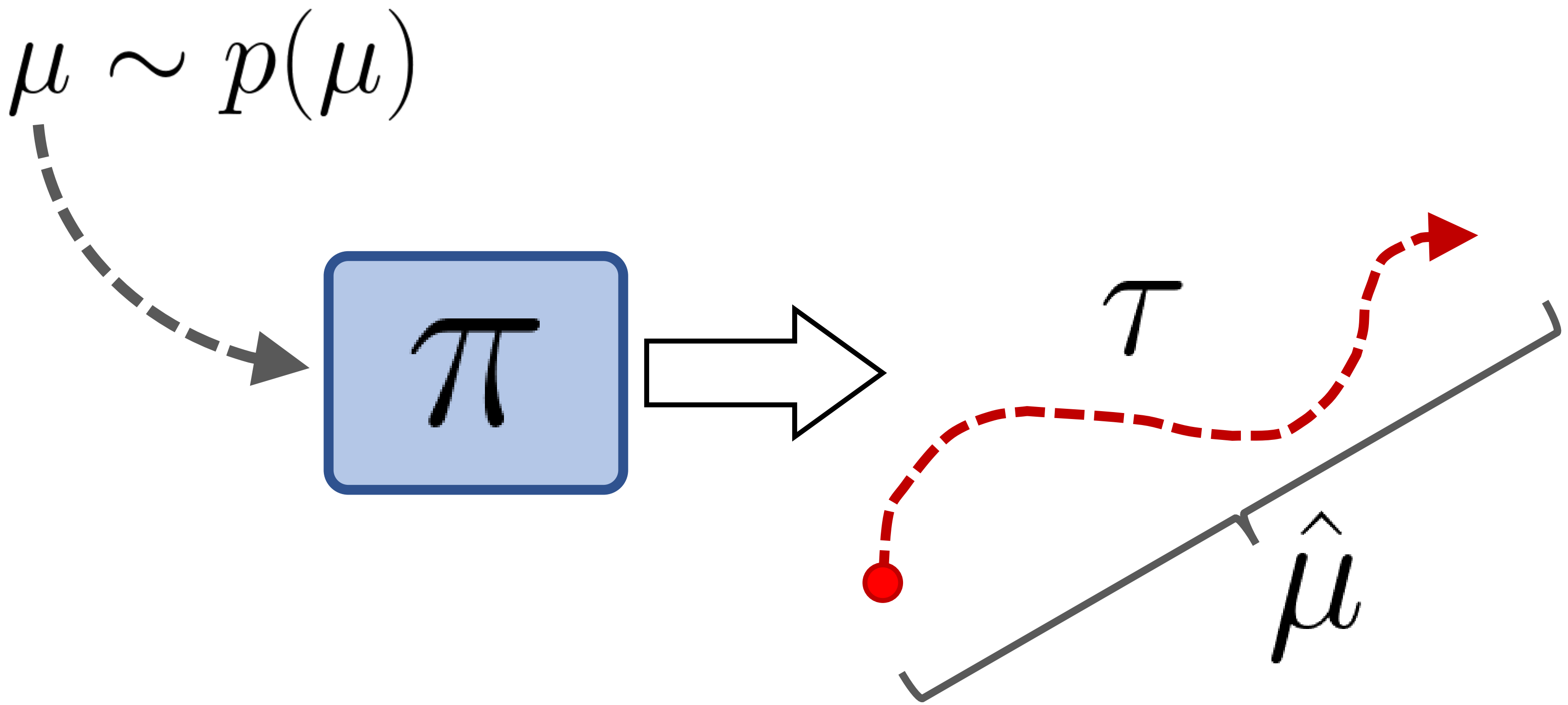
Inverse-dynamics:

$$p(\mathbf{a}|\mathbf{s}, \mathbf{s}', \mu)$$

System identification:

$$\boxed{p(\mu|\mathbf{s}, \mathbf{a}, \mathbf{s}')}$$

Learning SystemID



Online System Identification

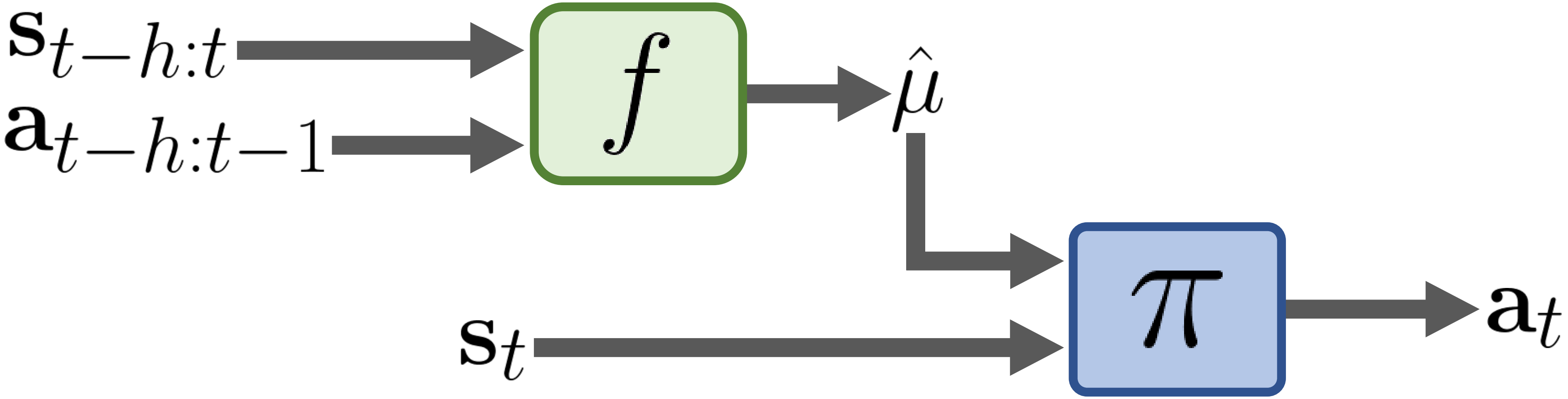
$$f(\mu | \mathbf{s}_{t-h:t}, \mathbf{a}_{t-h:t-1})$$

$$\begin{pmatrix} \mathbf{s}_t \\ \mathbf{a}_{t-1} \\ \mathbf{s}_{t-1} \\ \vdots \\ \mathbf{s}_{t-h} \\ \mathbf{a}_{t-h} \end{pmatrix} \Rightarrow \boxed{f} \Rightarrow \mu$$

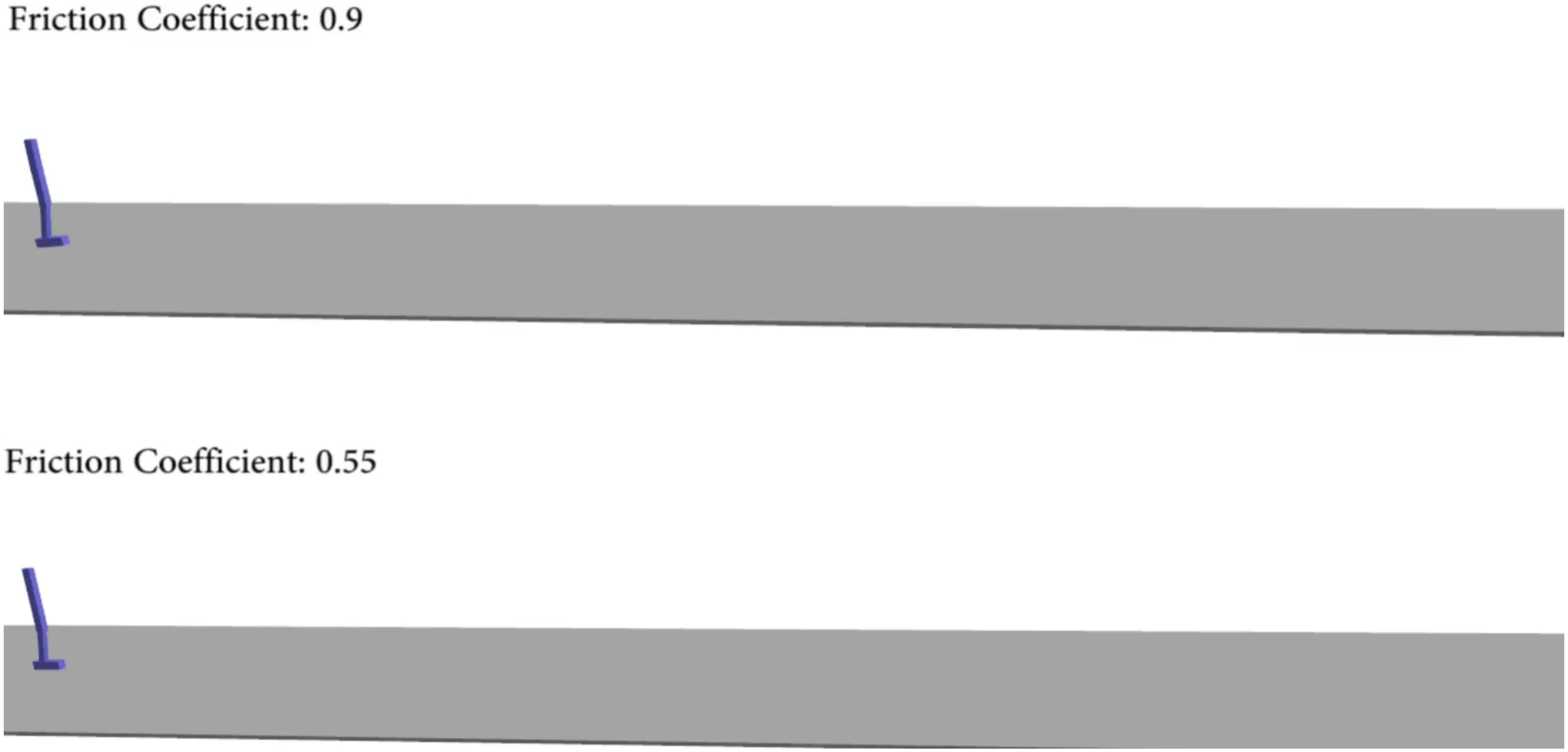
Online System Identification

$$\arg\max_f \mathbb{E}_{\mu\sim p(\mu)}\mathbb{E}_{\tau\sim p(\tau|\underline{\pi},\mu)}\left[\log f(\mu|\tau)\right]$$

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



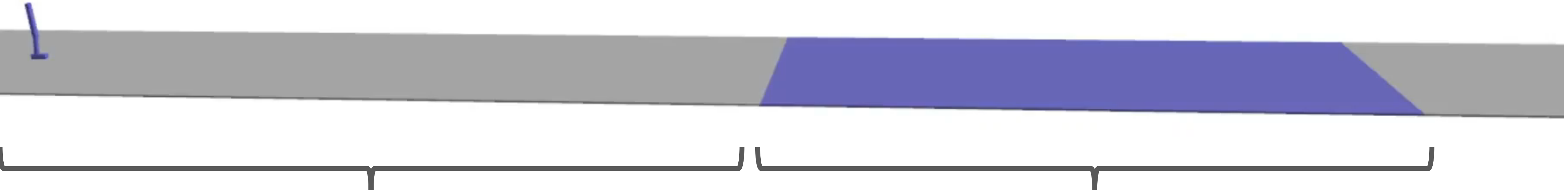
Online System Identification



Preparing for the Unknown: Learning a Universal Policy with Online System Identification
[Yu et al. 2017]

Online System Identification

$$f(\mu|\mathbf{s}_{t-h:t}, \mathbf{a}_{t-h:t-1})$$



Friction: 0.9

Friction: 0.55

Preparing for the Unknown: Learning a Universal Policy with Online System Identification
[Yu et al. 2017]

Online System Identification

$$f(\mu|\mathbf{s}_{t-h:t}, \mathbf{a}_{t-h:t-1})$$

Friction: 0.9

Friction: 0.55

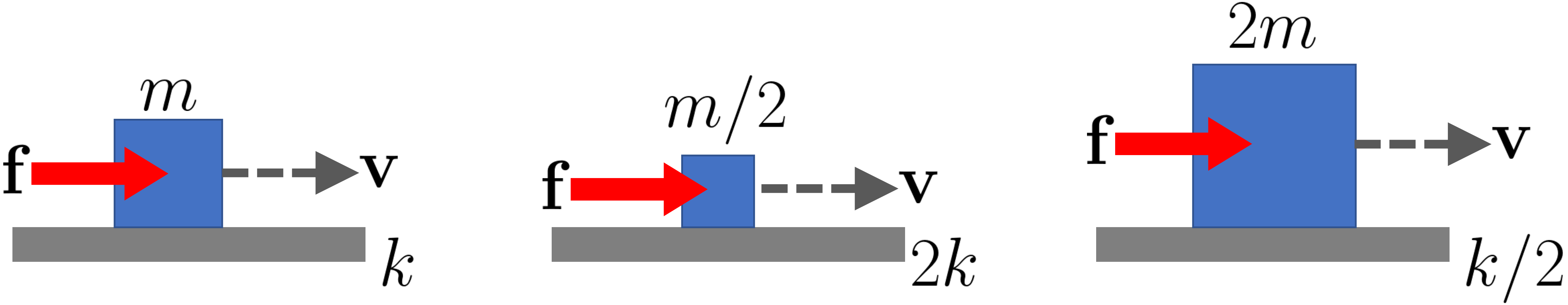
Preparing for the Unknown: Learning a Universal Policy with Online System Identification
[Yu et al. 2017]

Aliasing

$$f(\underline{\mu}|\mathbf{s}_{t-h:t}, \mathbf{a}_{t-h:t-1})$$

- μ can be very high dimensional (100s of parameters)
- Different settings of the parameters can have similar effects (i.e. aliasing)

$$\mu = (m, k)$$



Strategies

$$\mu^1 \approx \mu^2$$

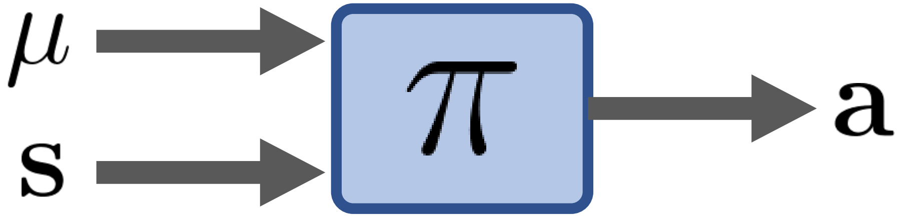
$$\downarrow \qquad \downarrow$$

$$\pi(\mathbf{a}|\mathbf{s}, \mu^1) = \pi(\mathbf{a}|\mathbf{s}, \mu^2)$$

Parameters that lead to the same dynamics
entails the same optimal strategy

Strategies

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

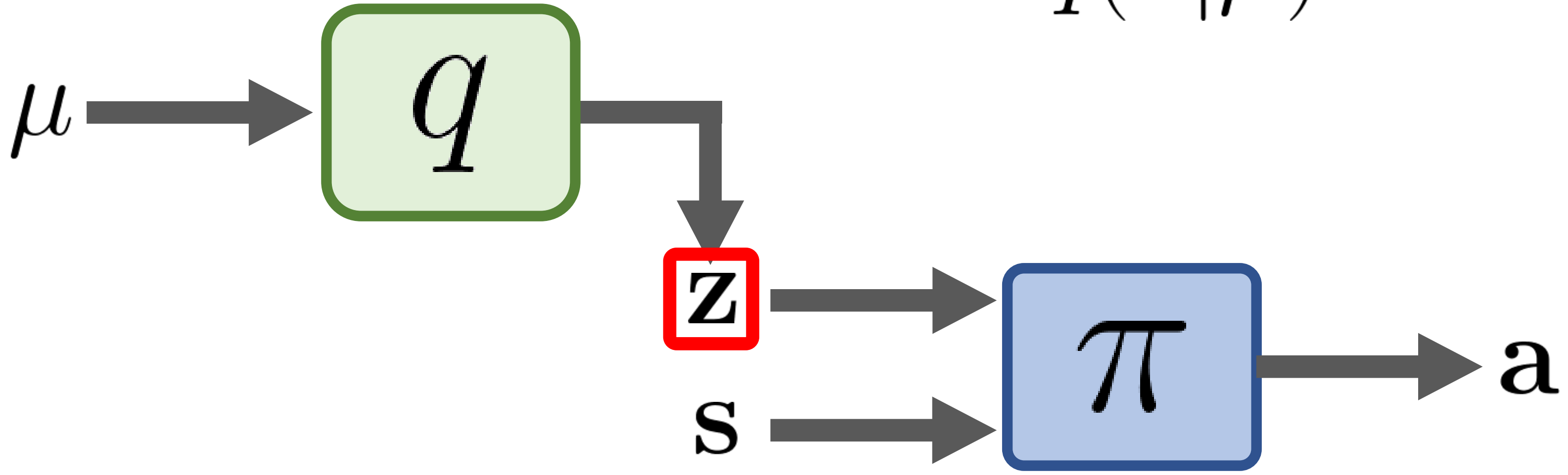


Strategies

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

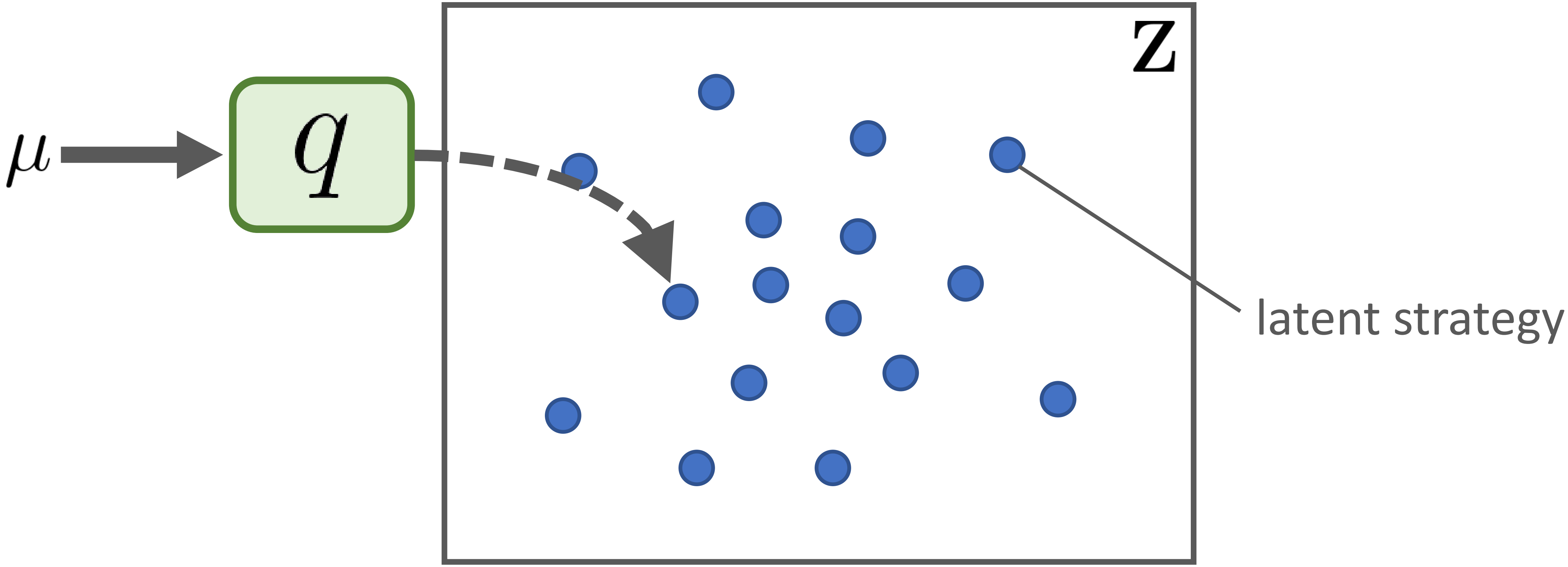
strategy

$$q(\mathbf{z}|\mu)$$

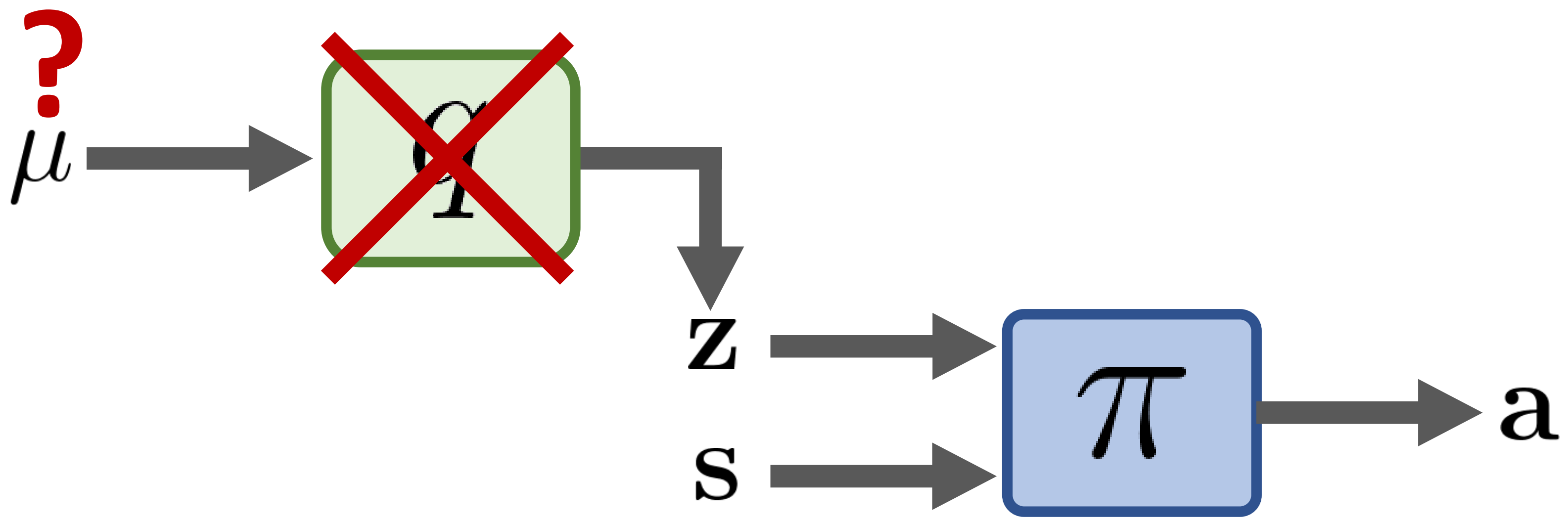


train end-to-end

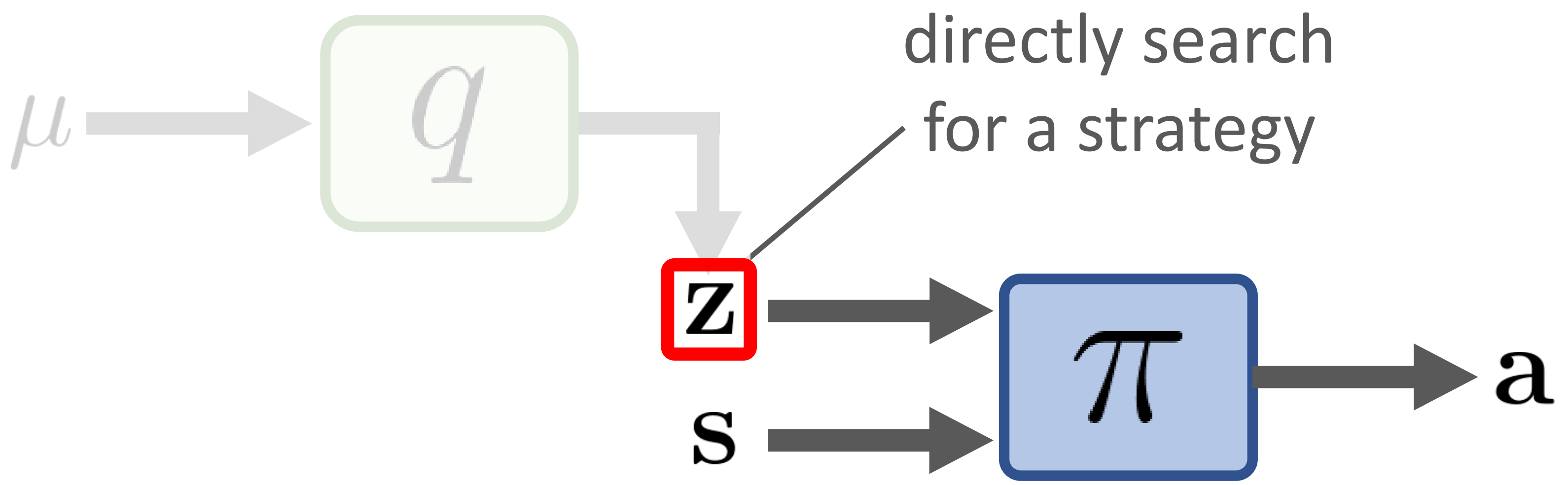
Latent Strategies



Transfer



Transfer

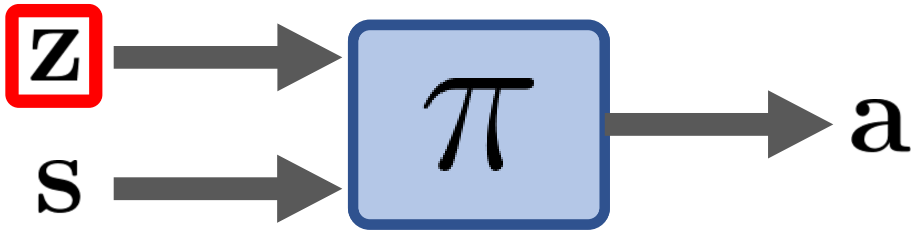


μ
q
directly search
for a strategy
$\mathbf{z}$
$\mathbf{s}$
π
$\mathbf{a}$

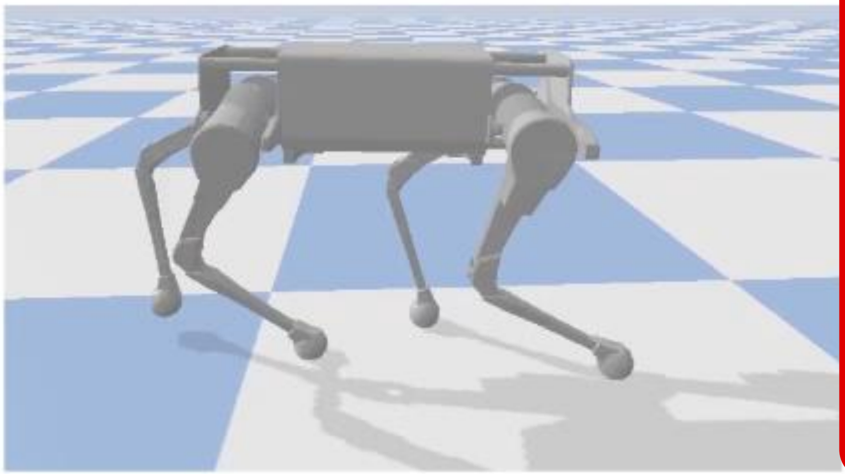
Transfer

$$\arg\max_{\mathbf{z}} \; \mathbb{E}_{\tau\sim p(\tau|\pi,\mathbf{z})}\left[\sum_{t=0}^{T-1} r_t\right]$$

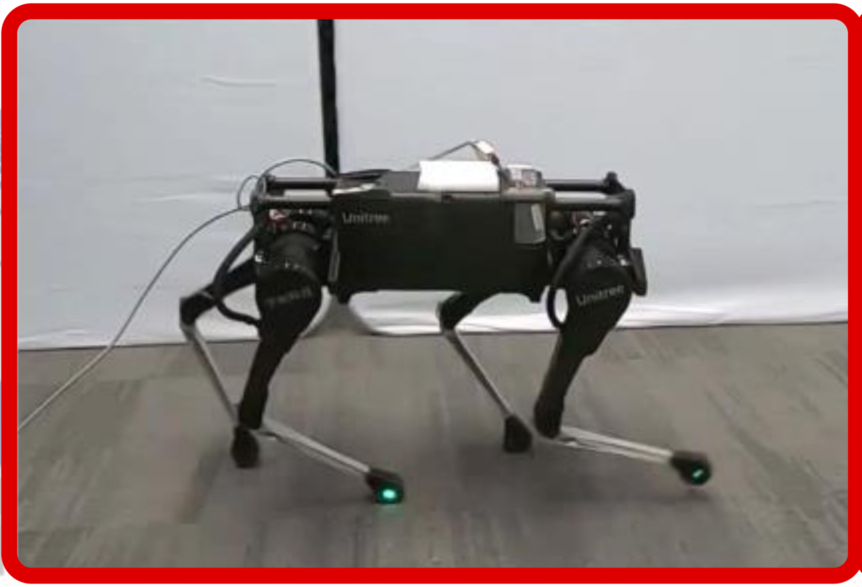
use derivative-free optimizer
(e.g. random search, CEM, PG, etc)



Domain Adaptation



Reference

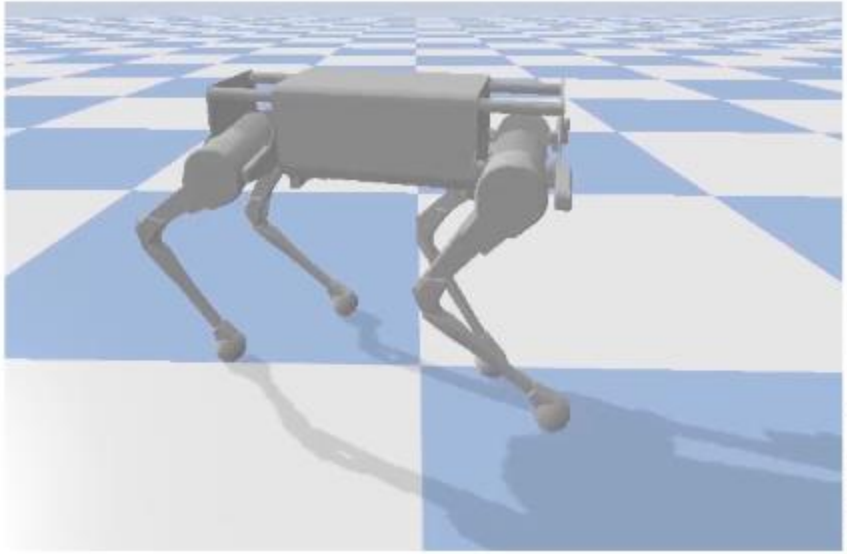


Real Robot
(Before Adaptation)

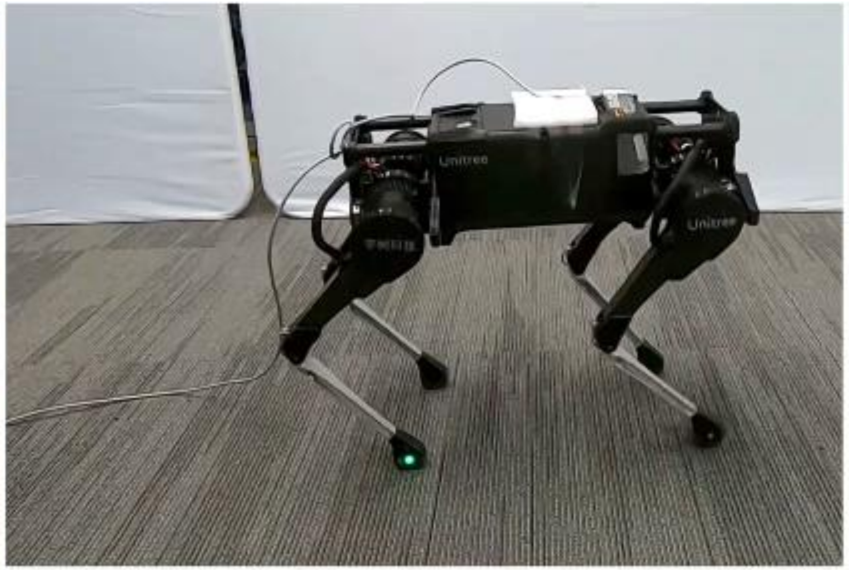
Real Robot
(After Adaptation)

Learning Agile Robotic Locomotion Skills by Imitating Animals
[Peng et al. 2020]

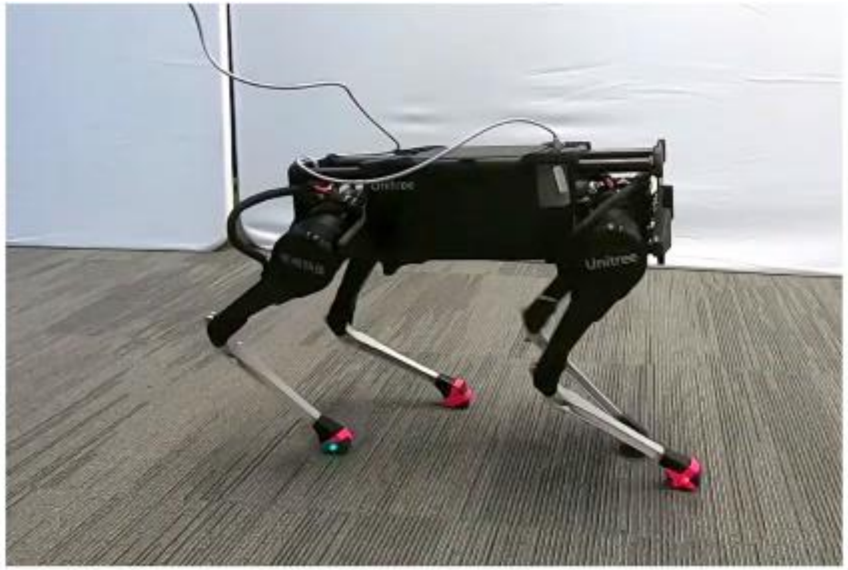
Domain Adaptation



Reference



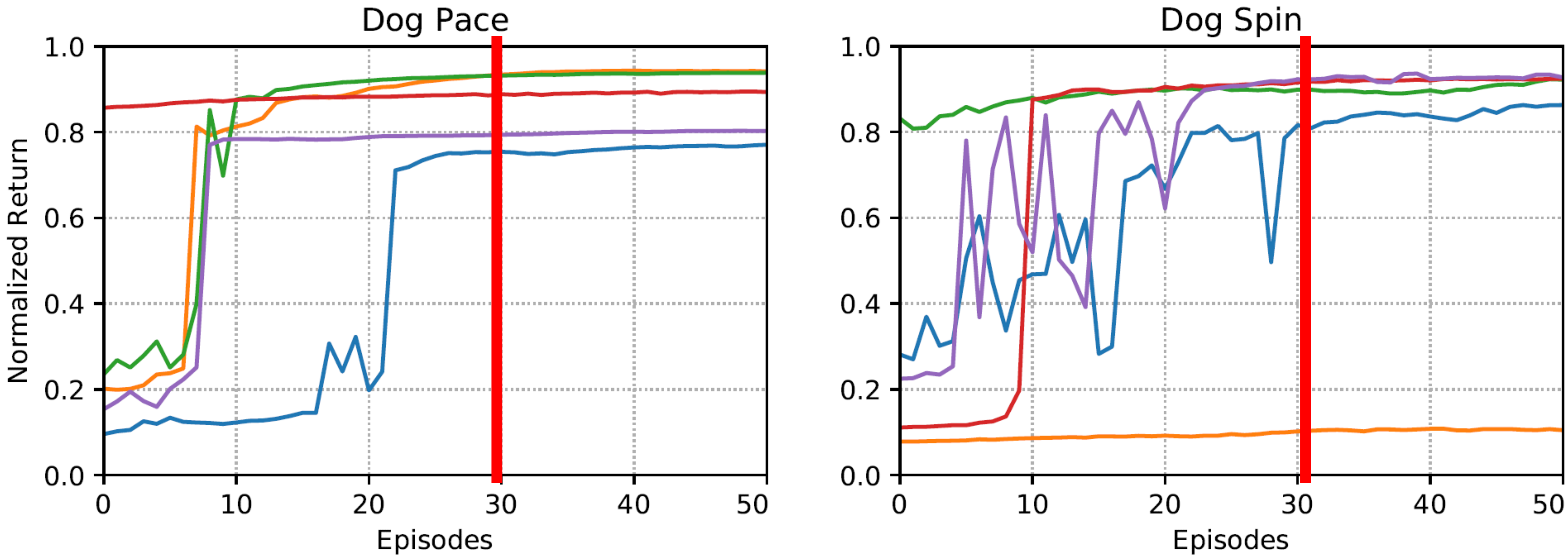
Real Robot
(Before Adaptation)



Real Robot
(After Adaptation)

Learning Agile Robotic Locomotion Skills by Imitating Animals
[Peng et al. 2020]

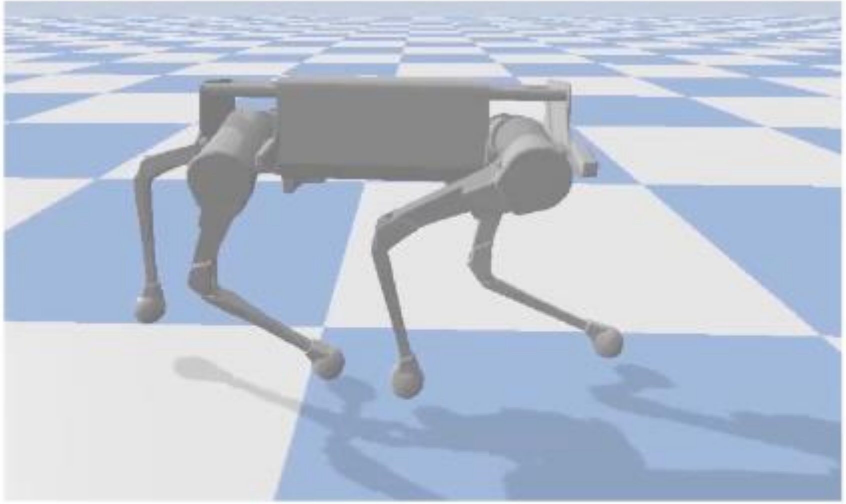
Domain Adaptation

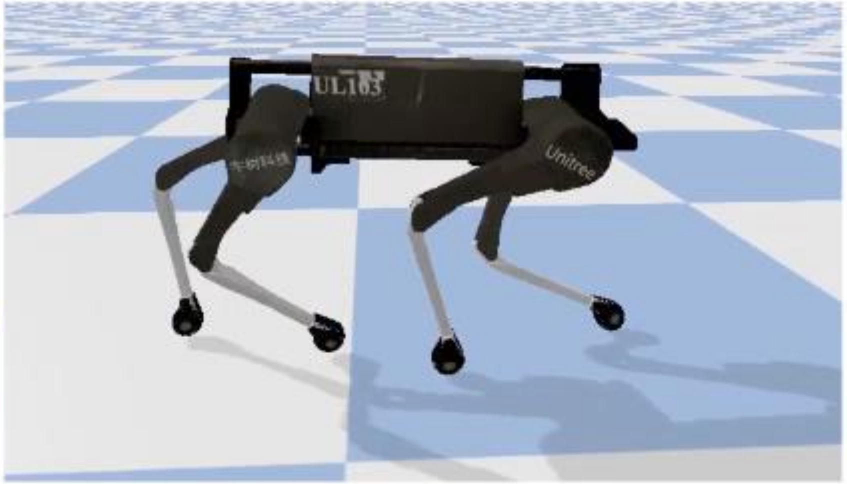


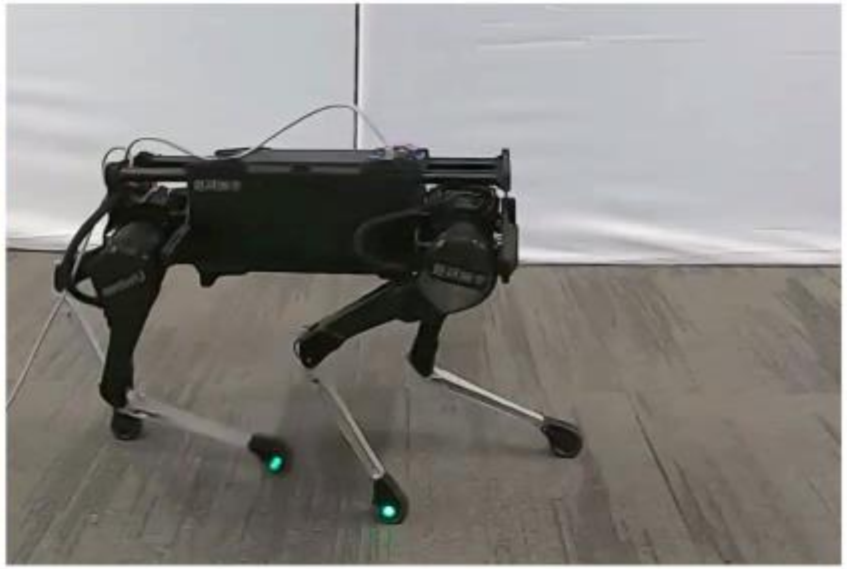
~5 mins

Learning Agile Robotic Locomotion Skills by Imitating Animals
[Peng et al. 2020]

Domain Adaptation







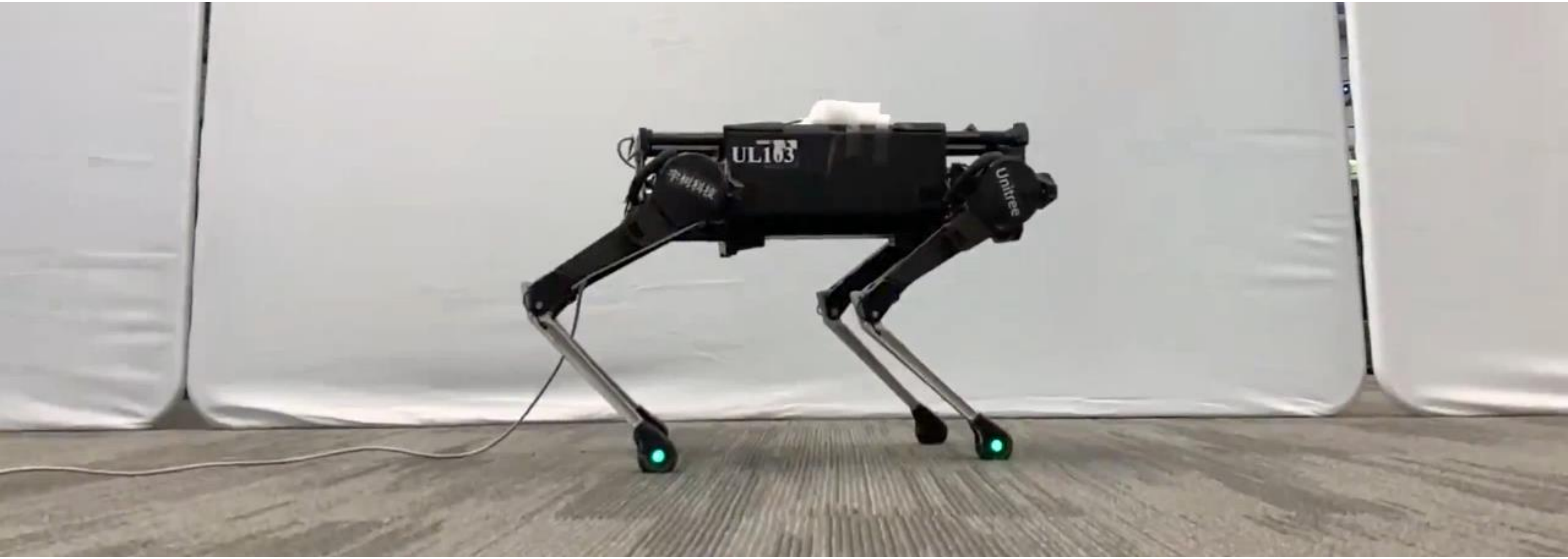
Reference

Real Robot
(Before Adaptation)

Real Robot
(After Adaptation)

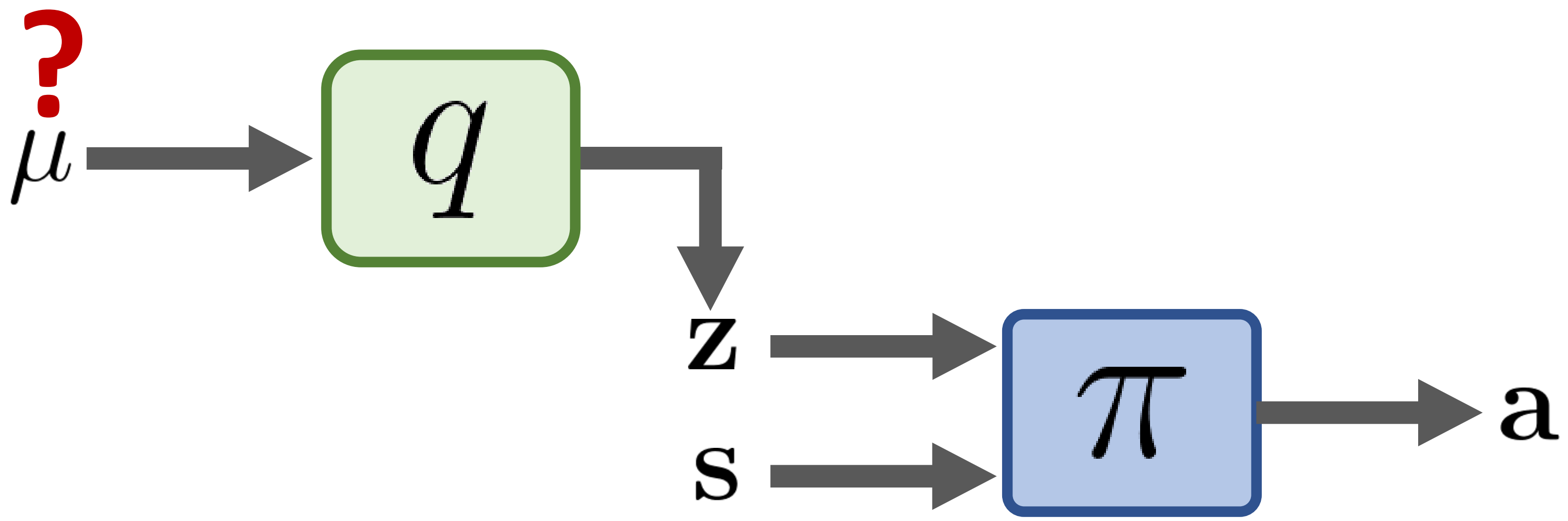
Learning Agile Robotic Locomotion Skills by Imitating Animals
[Peng et al. 2020]

Domain Adaptation

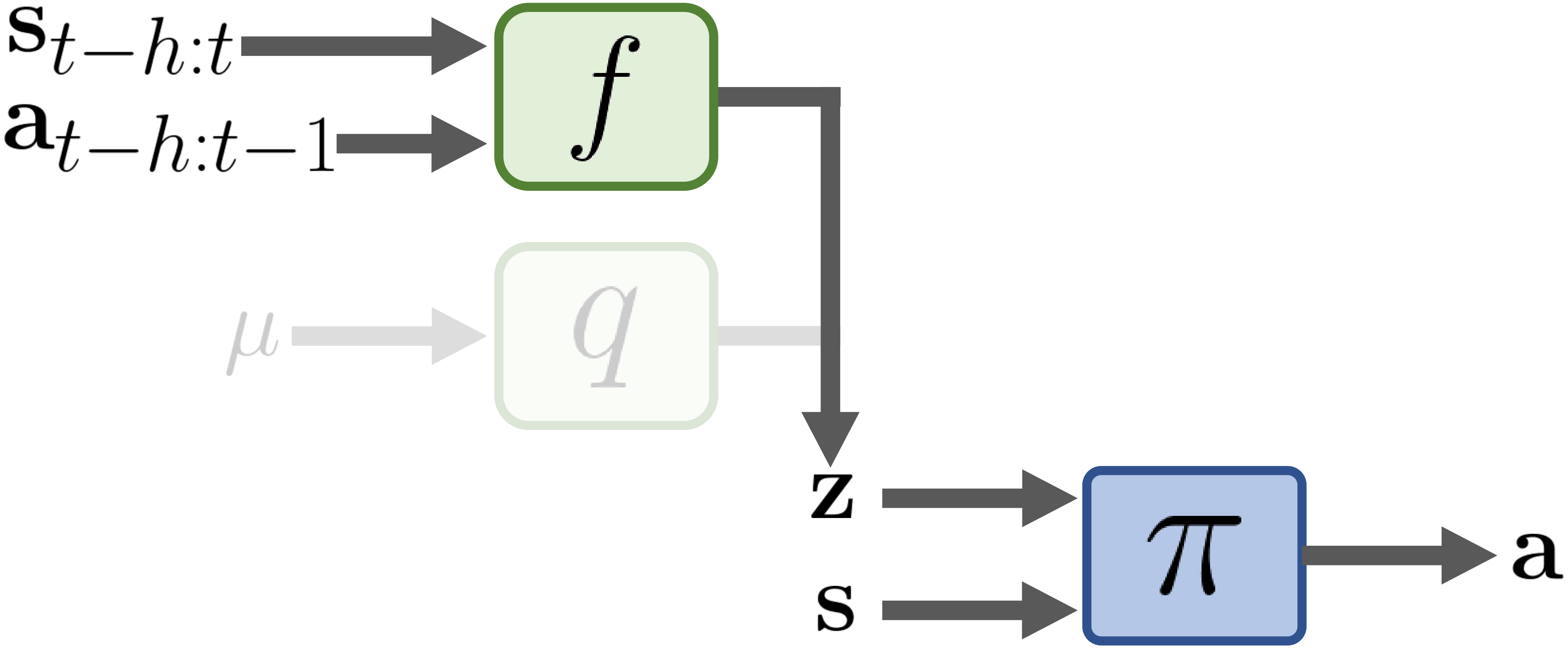


Learning Agile Robotic Locomotion Skills by Imitating Animals
[Peng et al. 2020]

Online Strategy Identification

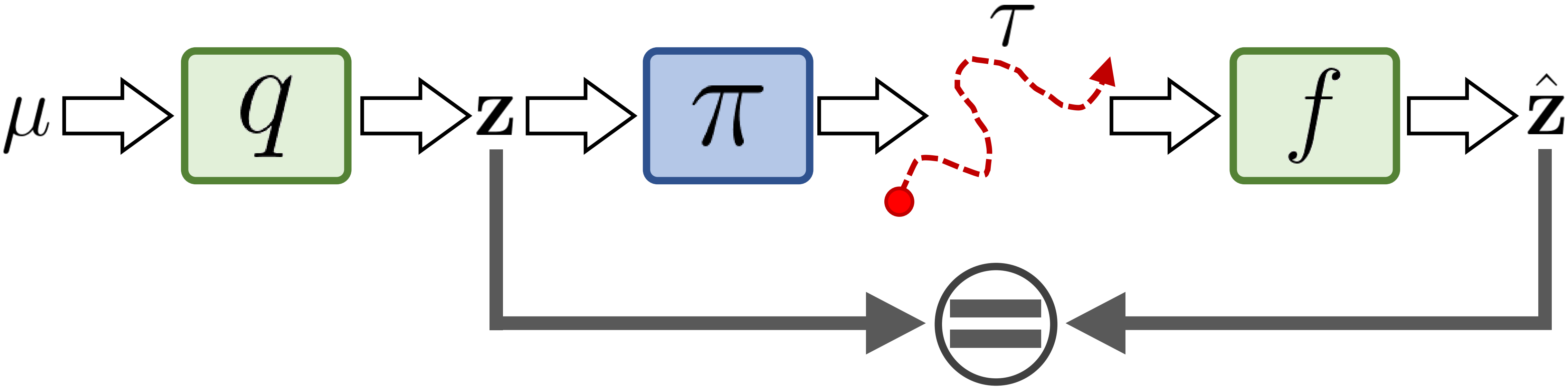


Online Strategy Identification

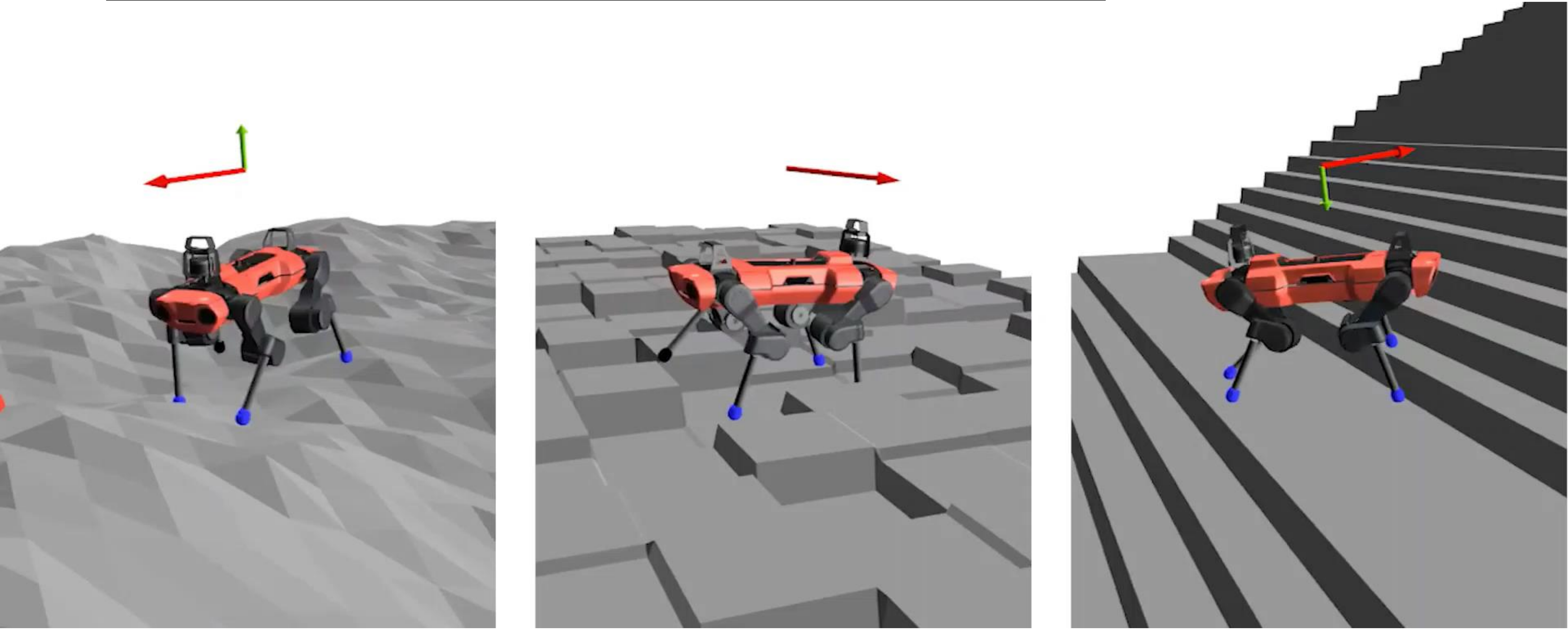


Online Strategy Identification

$$\arg\max_{f} \; \mathbb{E}_{\underline{\mu\sim p(\mu)}}\mathbb{E}_{\underline{\mathbf{z}\sim q(\mathbf{z}|\mu)}}\mathbb{E}_{\underline{\tau\sim p(\tau|\pi,\mathbf{z})}}\left[\log \underline{f(\mathbf{z}|\tau)}\right]$$



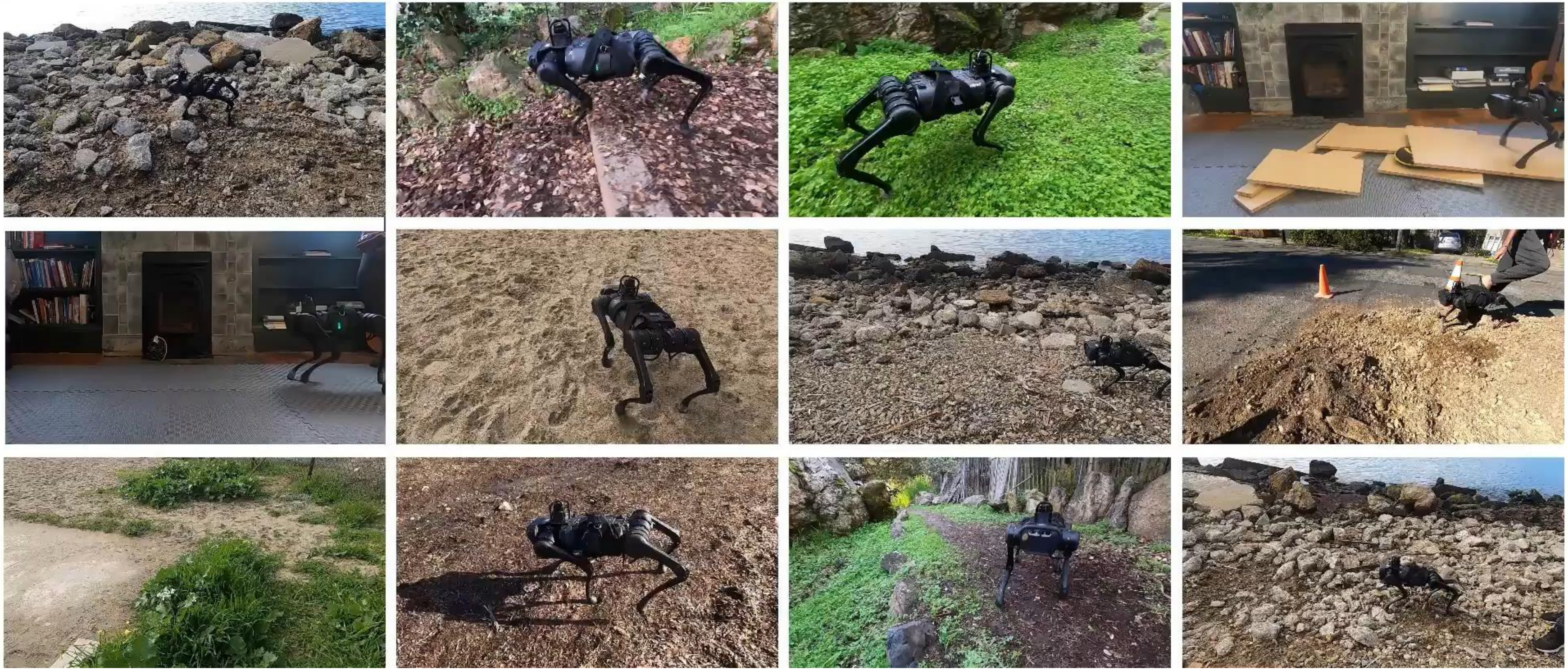
Terrain Adaptation



Learning Quadrupedal Locomotion Over Challenging Terrain
[Lee et al. 2020]

Learning Quadrupedal Locomotion Over Challenging Terrain
[Lee et al. 2020]

Legged Locomotion

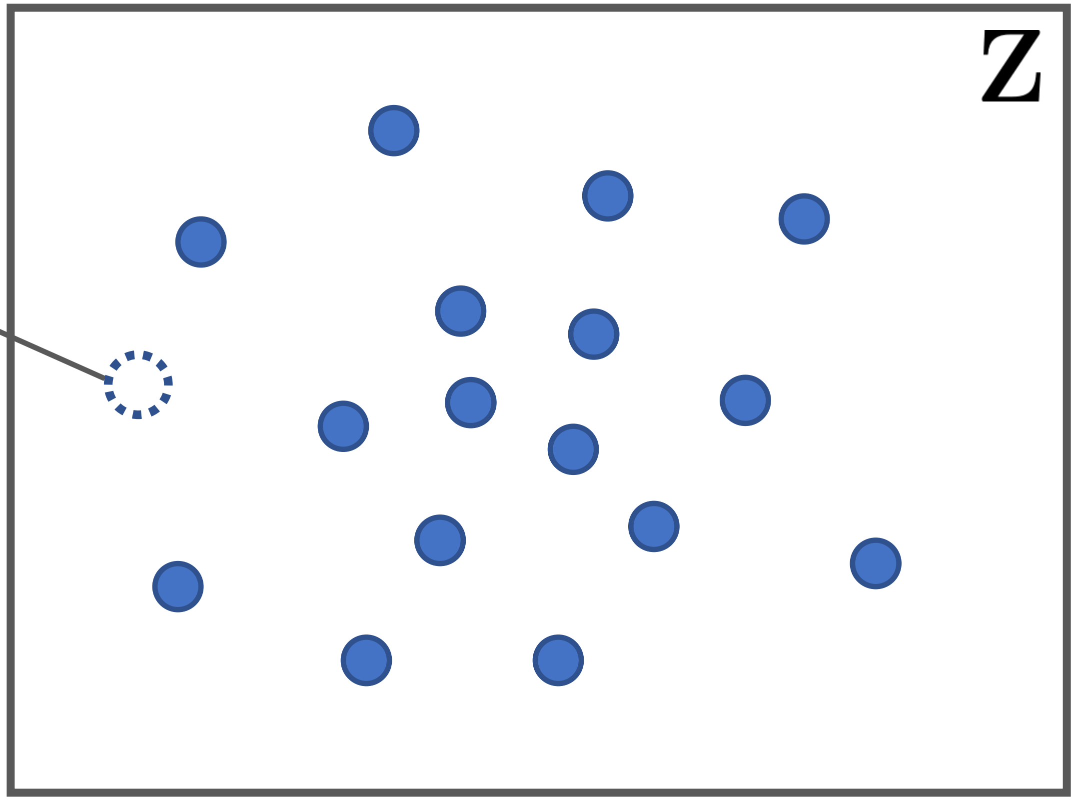


RMA: Rapid Motor Adaptation for Legged Robots
[Kumar et al. 2022]

Adaptive Strategies

- Fast adaptation (online methods: few seconds)
- Need to design rich training environment to learn versatile strategies

What if none of the strategies work?



Domain Adaptation

$$\arg\max_{\mathbf{z}} \ \mathbb{E}_{\tau \sim p(\tau|\pi,\mathbf{z})} \left[\sum_{t=0}^{T-1} r_t \right]$$

low-dimensional
+ restrictive

Domain Adaptation

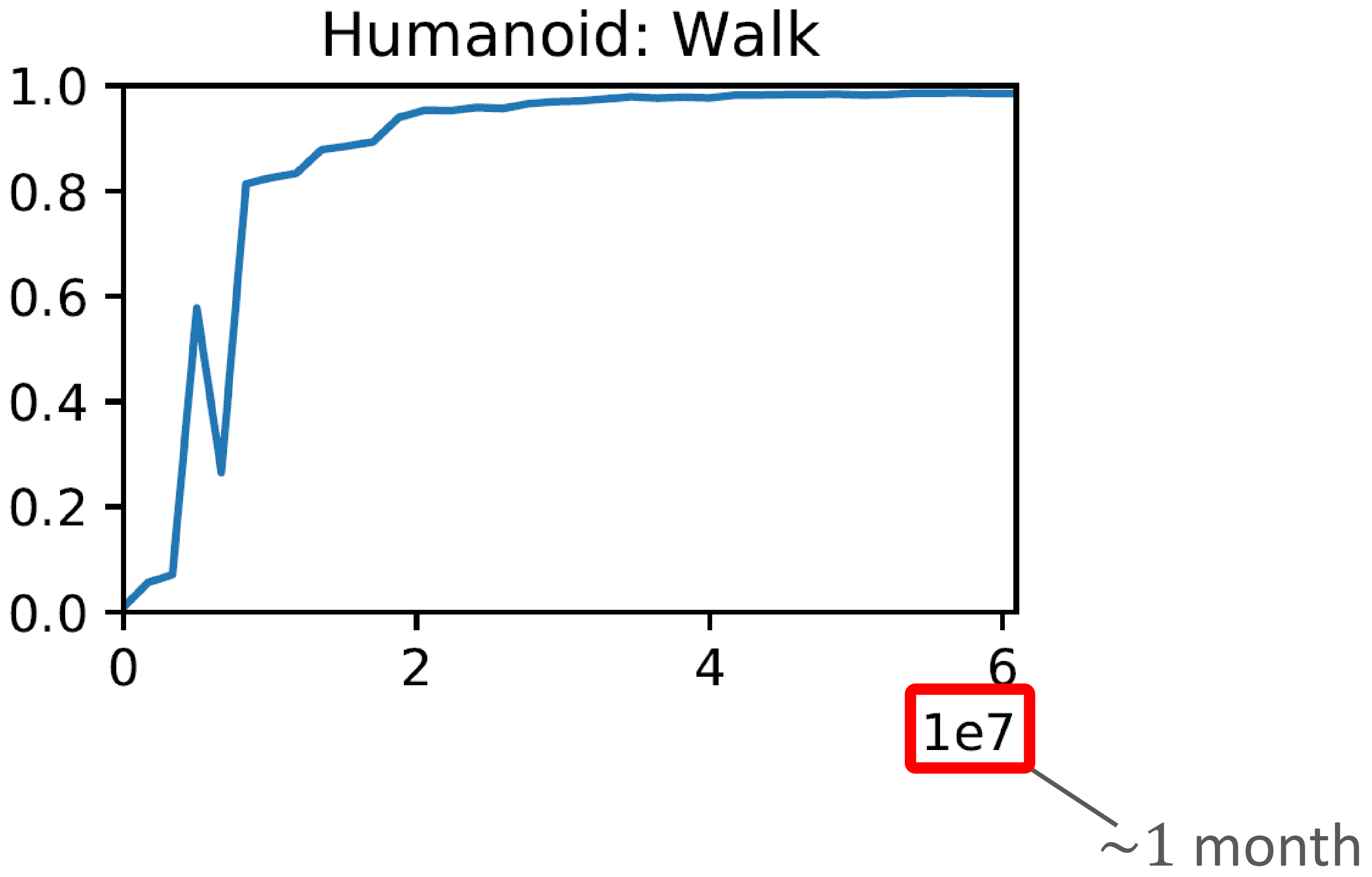
$$\arg\max_{\mathbf{z}} \; \mathbb{E}_{\tau \sim p(\tau|\pi,\mathbf{z})} \left[\sum_{t=0}^{T-1} r_t \right]$$

$$\Big\Downarrow \text{ RL finetuning}$$

$$\arg\max_{\pi} \; \mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\sum_{t=0}^{T-1} r_t \right]$$

high dimensional
+ flexible

Finetuning

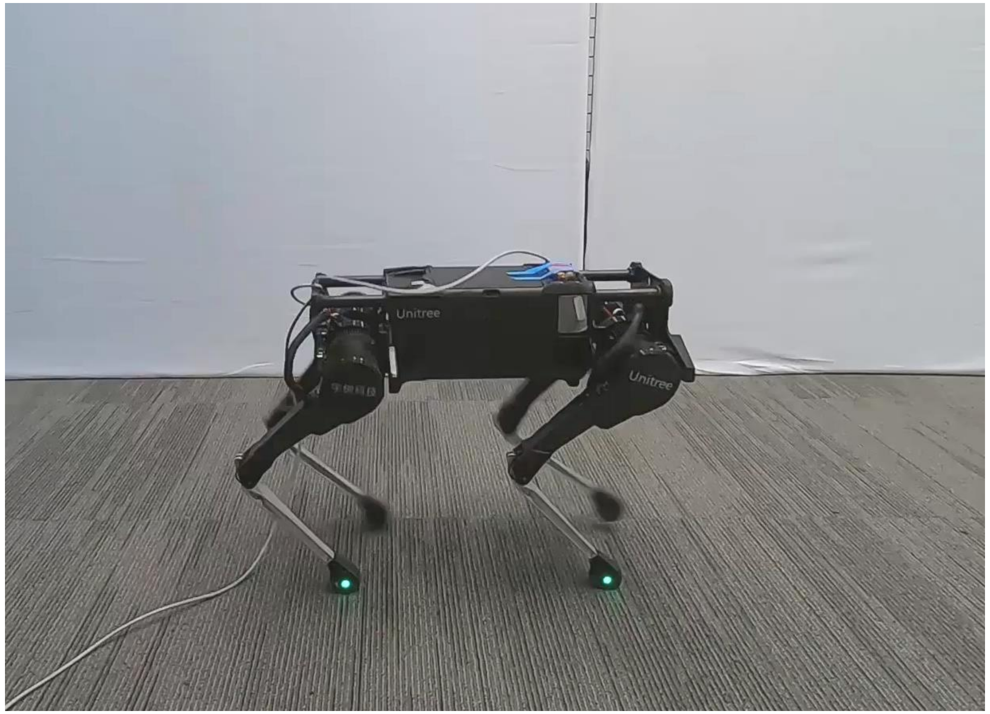


Finetuning

SAC, REDQ, DroQ

Finetuning

Simulation
(Source Domain)



Real World
(Target Domain)

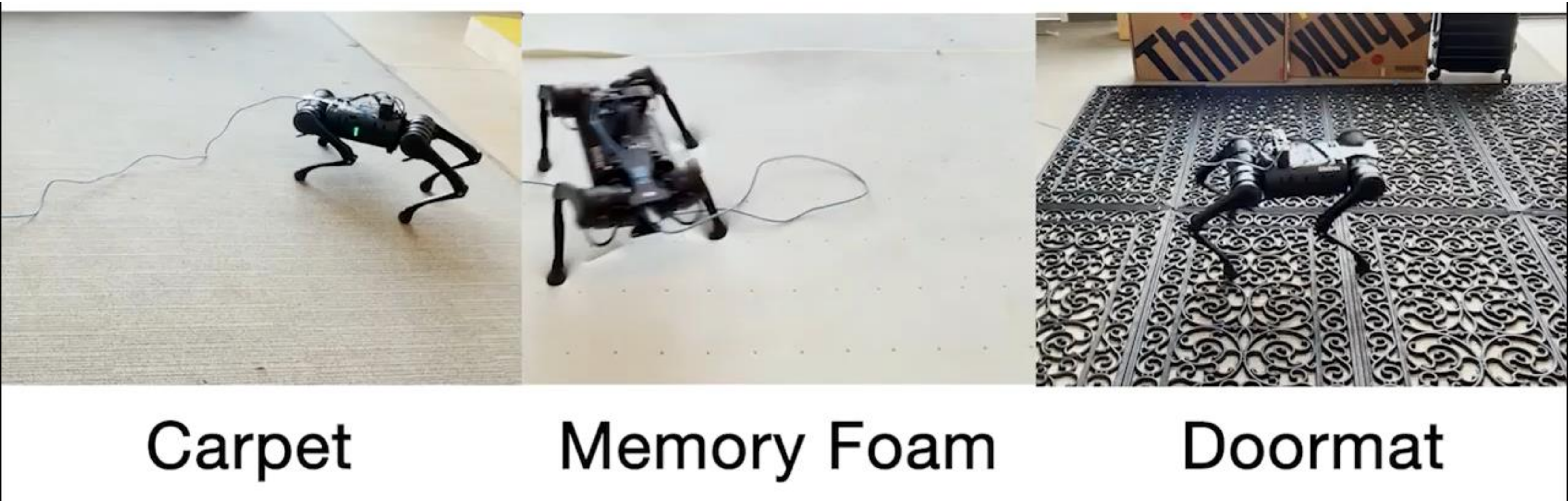
Real-World Finetuning

Legged Robots that Keep on Learning: Fine-Tuning Locomotion Policies in the Real World
[Smith et al. 2022]

Real-World Finetuning

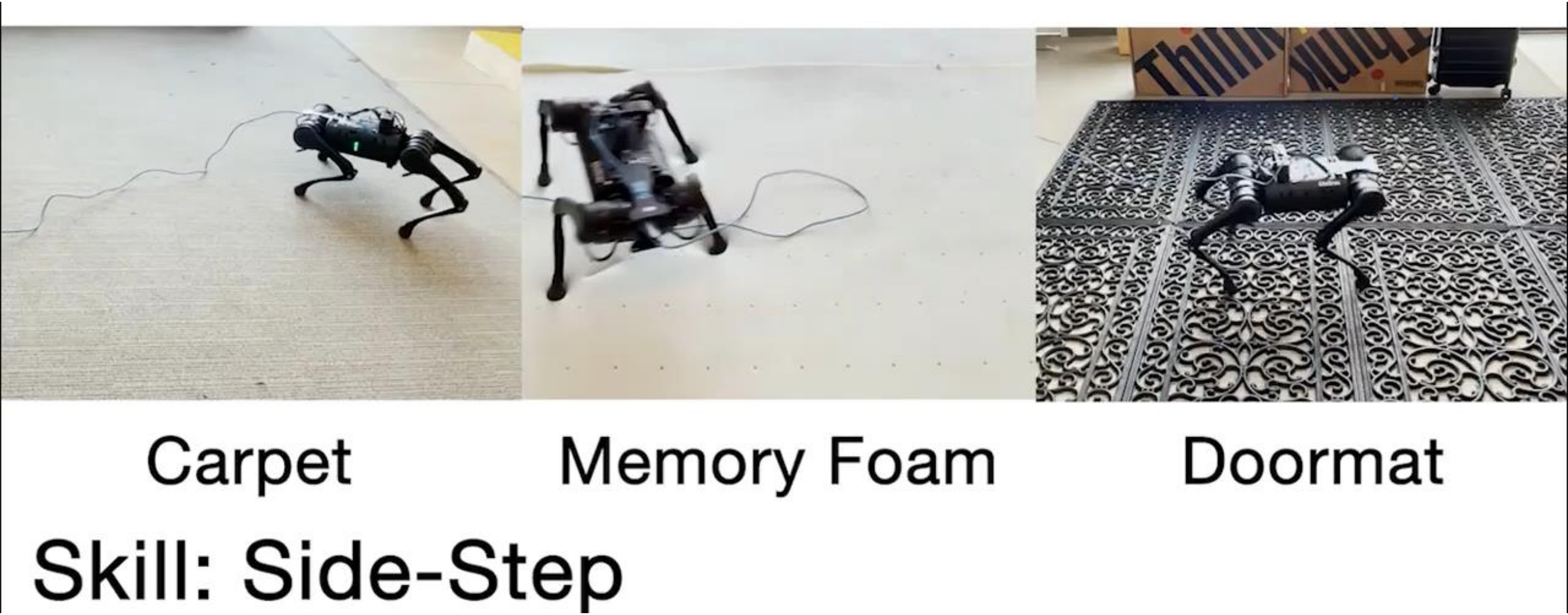
Legged Robots that Keep on Learning: Fine-Tuning Locomotion Policies in the Real World
[Smith et al. 2022]

Real-World Finetuning



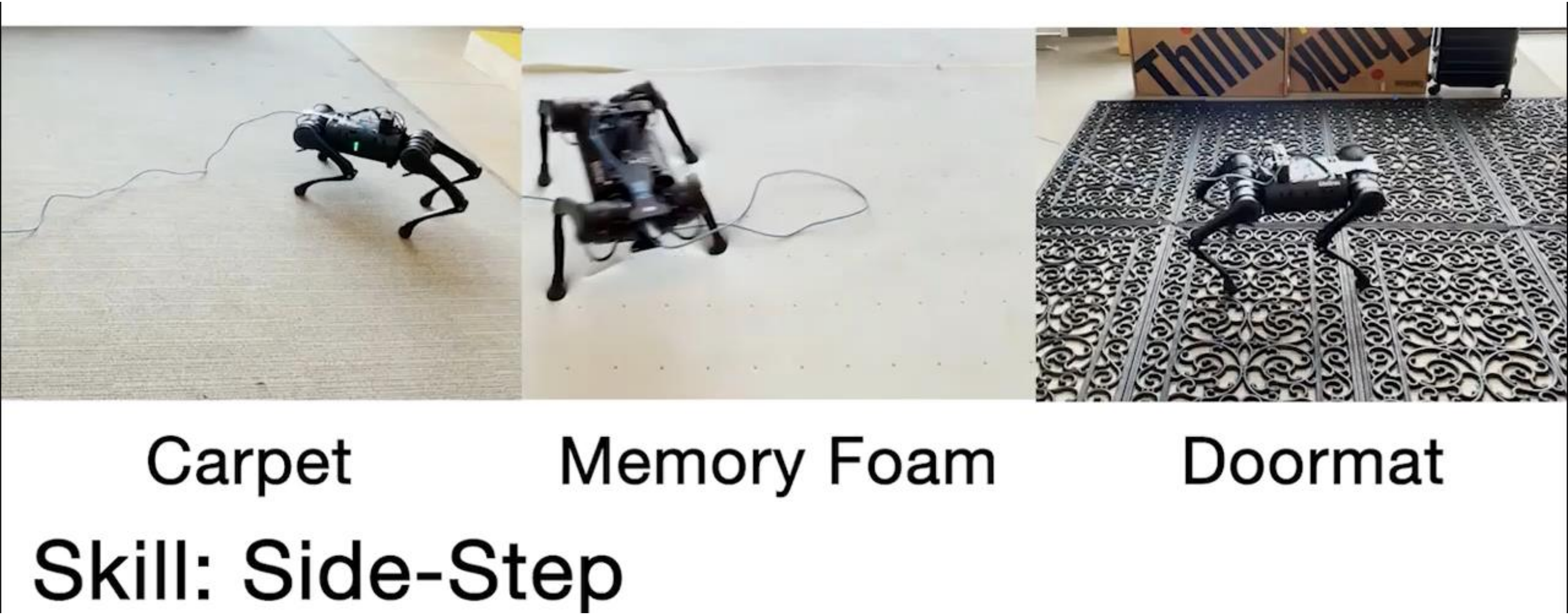
Legged Robots that Keep on Learning: Fine-Tuning Locomotion Policies in the Real World
[Smith et al. 2022]

Real-World Finetuning



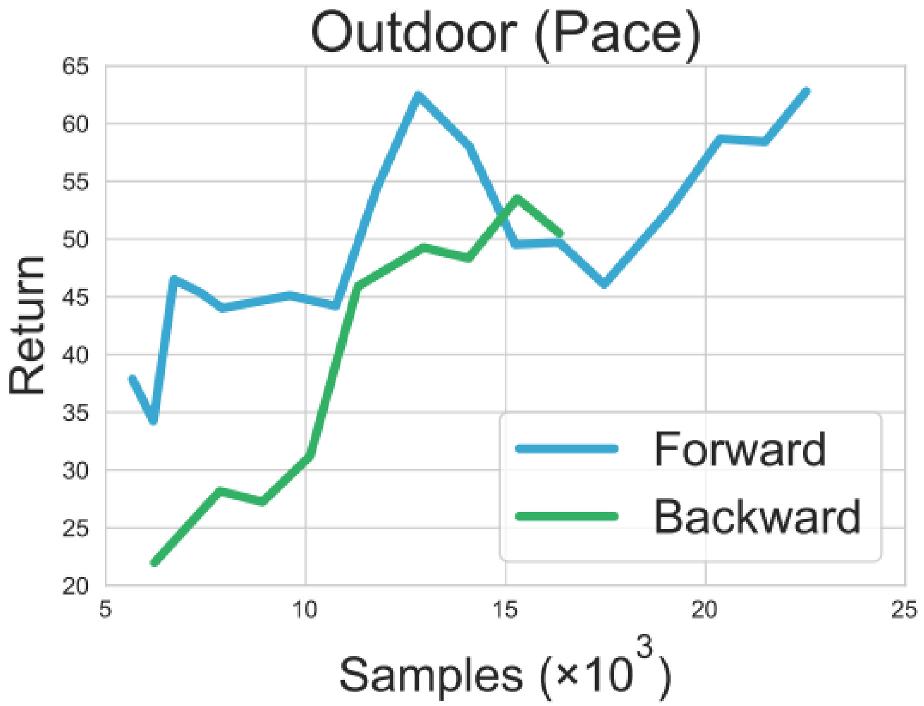
Legged Robots that Keep on Learning: Fine-Tuning Locomotion Policies in the Real World
[Smith et al. 2022]

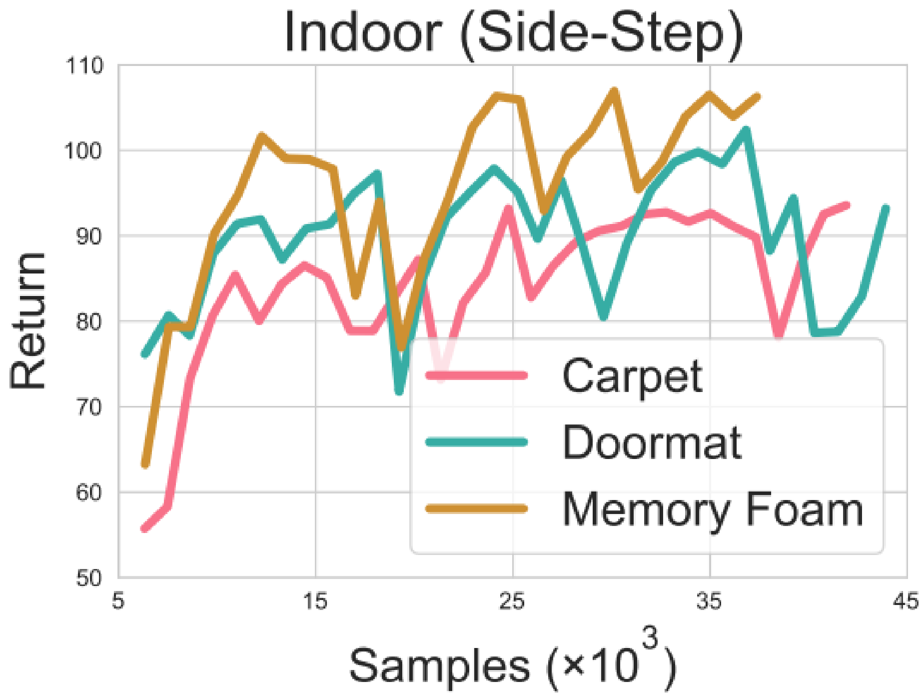
Real-World Finetuning



Legged Robots that Keep on Learning: Fine-Tuning Locomotion Policies in the Real World
[Smith et al. 2022]

Real-World Finetuning





(~1 hour)

Legged Robots that Keep on Learning: Fine-Tuning Locomotion Policies in the Real World
[Smith et al. 2022]

Summary

- Domain Transfer
- System Identification
- Domain Randomization
- Domain Adaptation

In practice: There is no silver bullet. Often need to combine multiple techniques for successful transfer.