Reinforcement Learning on Legged Robots

Jie Tan
Google DeepMind

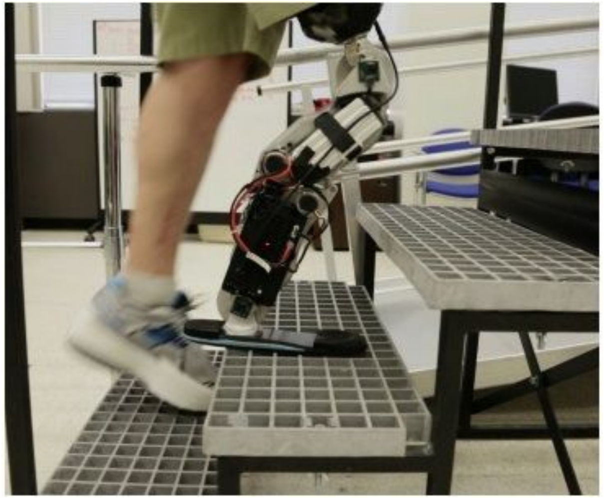
CS 224R Deep Reinforcement Learning
May 10, 2023 @ Stanford University

Today’s Class

1. State-of-the-art of learning legged locomotion via Deep RL?
2. Two approaches to apply Deep RL on real robots
    a. Train in the real world
    b. Sim-to-real transfer

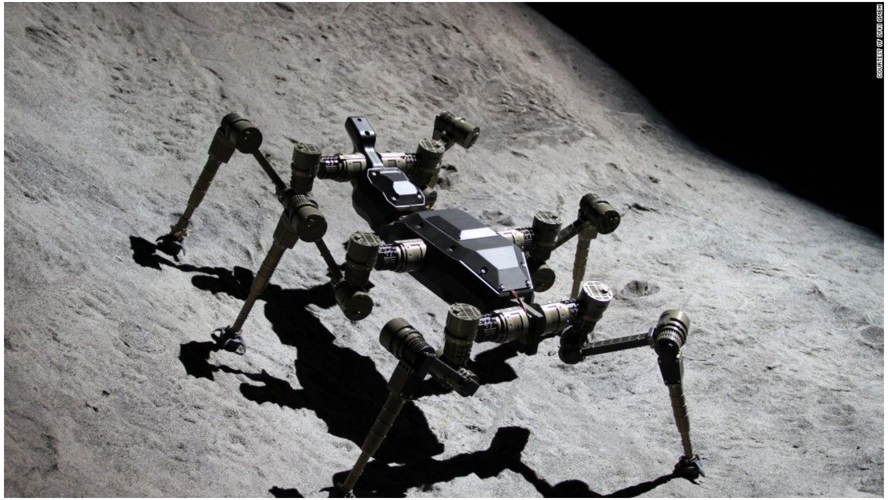
Goal

- Understand challenges of applying RL in the real world
- Understand the root causes of sim-to-real gap
- Learn the most popular sim-to-real transfer methods



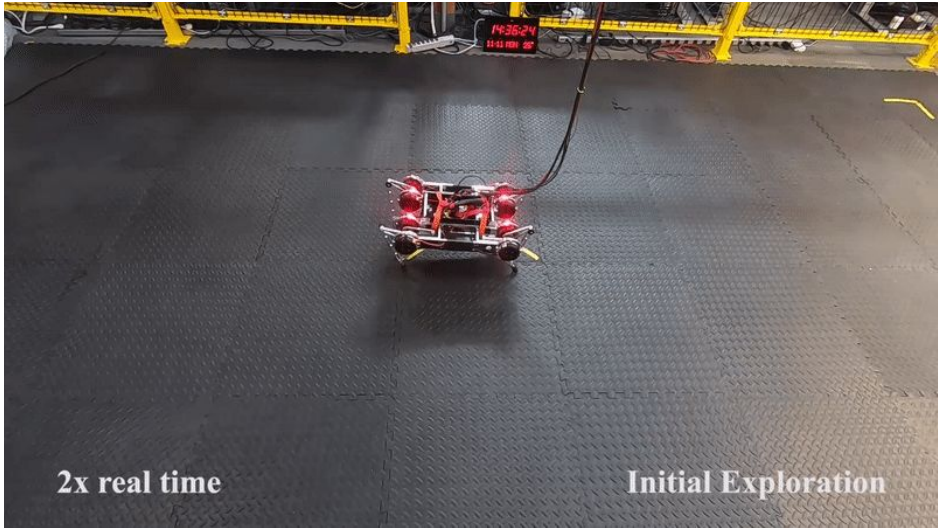

Boston Dynamics
© Boston Dynamics


co-pace
powered by Continental
Continental
Robo-Delivery
Powered by
ANYmal



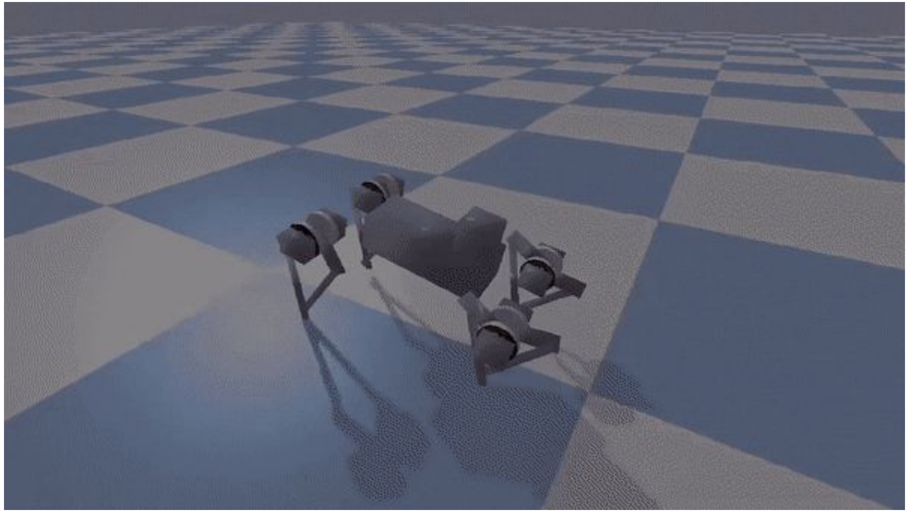
Reinforcement Learning for Legged Robots
State-of-the-Art

RL Training in the Real World



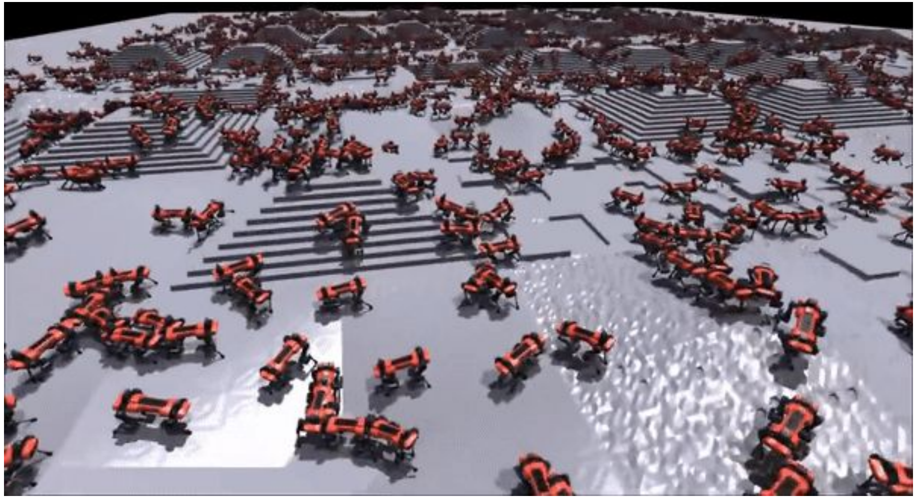
[Learning to Walk via Deep Reinforcement Learning, Haarnoja et al., RSS 2019](#)
[Learning to Walk in the Real World with Minimal Human Effort, Ha et al., CoRL 2020](#)

RL in Simulation and Sim-to-Real Transfer



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

RL in Simulation and Sim-to-Real Transfer

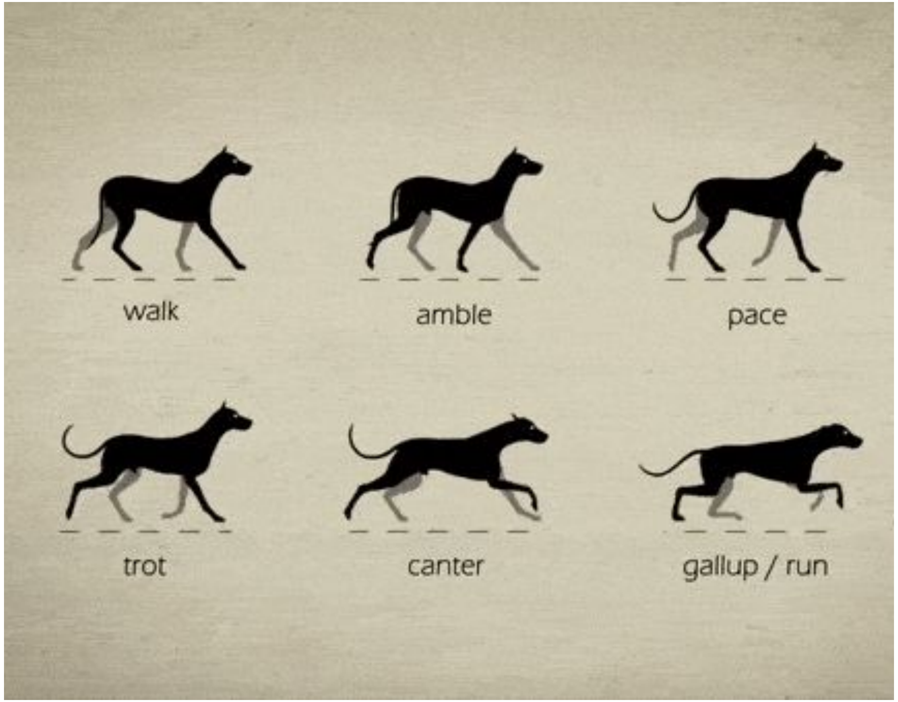


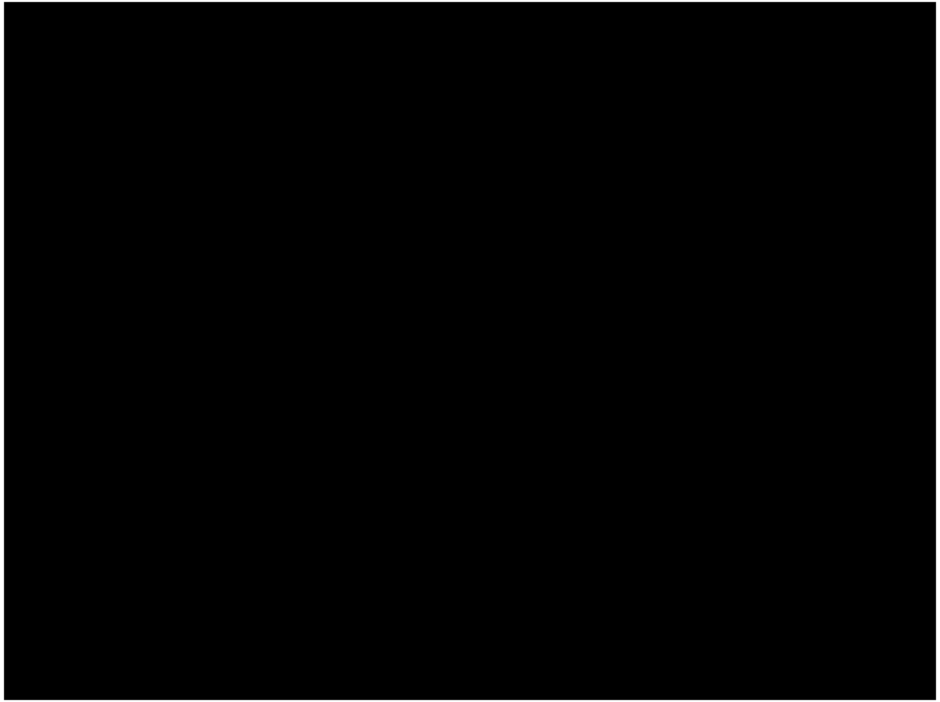
[Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning, Rudin et al., CoRL 2021]

Fast Adaptation to New Environments

RMA: Rapid Motor Adaptation for Legged Robots, Kumar et al., RSS 2021

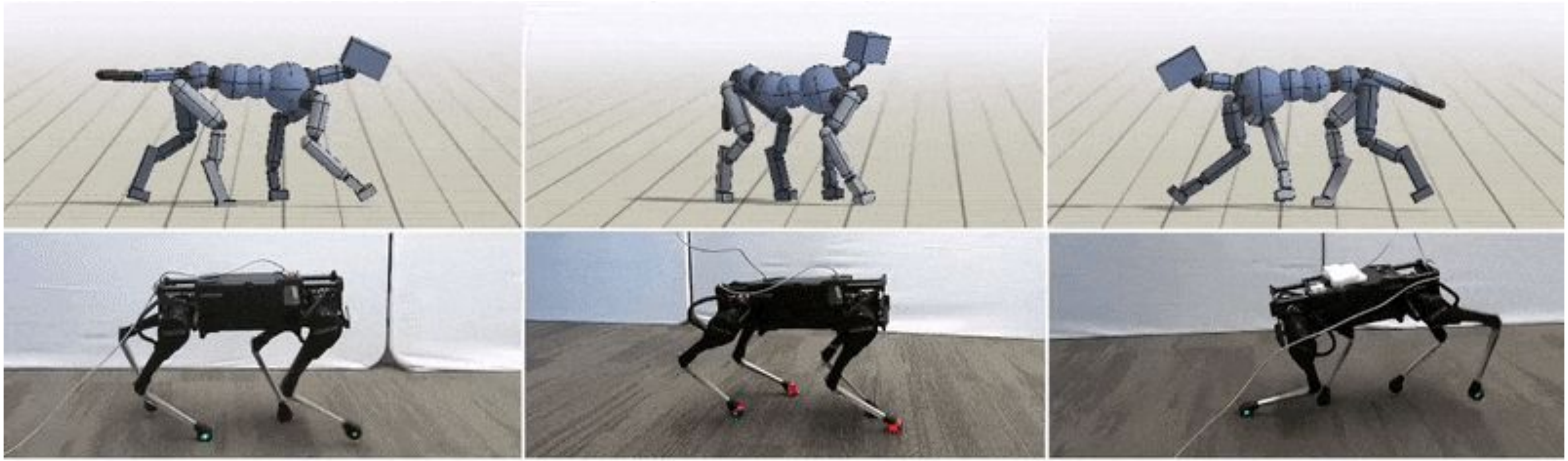
Style





[Fast and Efficient Locomotion via Learned Gait Transitions, Yang et al., CoRL 2021]

Style

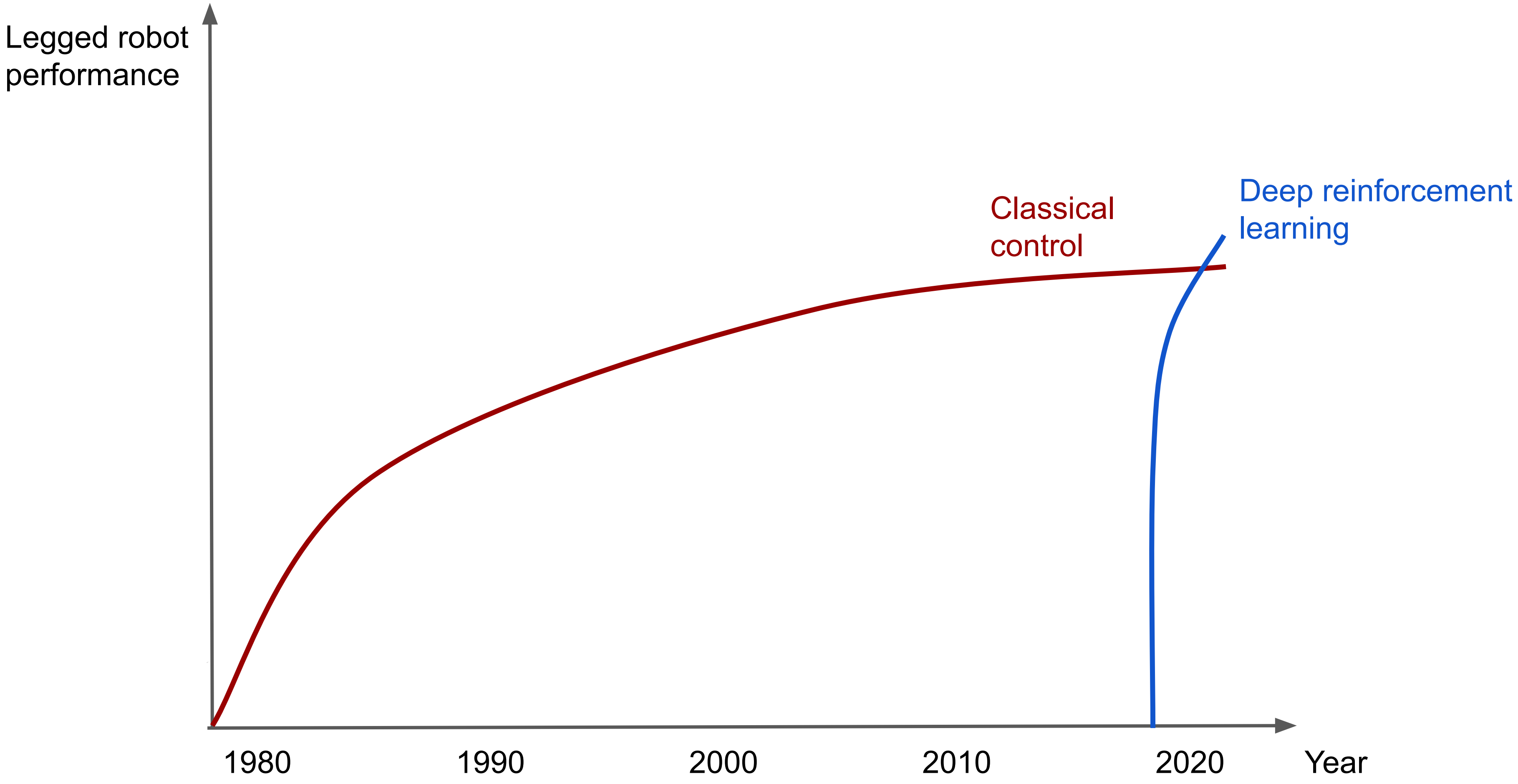


[Learning Agile Robotic Locomotion Skills by Imitating Animals, Peng et al., RSS 2020]

Testing the capabilities on obstacle avoidance along the way

[Learning to Navigate Sidewalks in Outdoor Environments, Sorokin et al., ICRA 2022]

Learning robust perceptive locomotion for quadrupedal robots in the wild, Miki et al., Science Robotics 2022


Legged robot performance
Classical control
Deep reinforcement learning
1980
1990
2000
2010
2020
Year

How to learn locomotion?

Approach 1

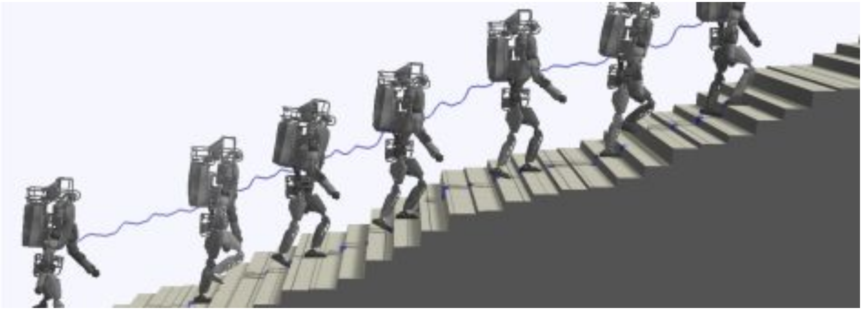
Directly train in the real world

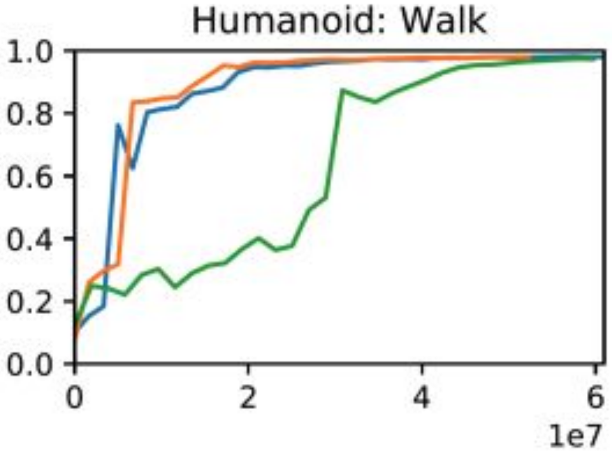
Approach 2

Learn in simulation and sim-to-real transfer

Learning in real world

- ✗ Data efficiency





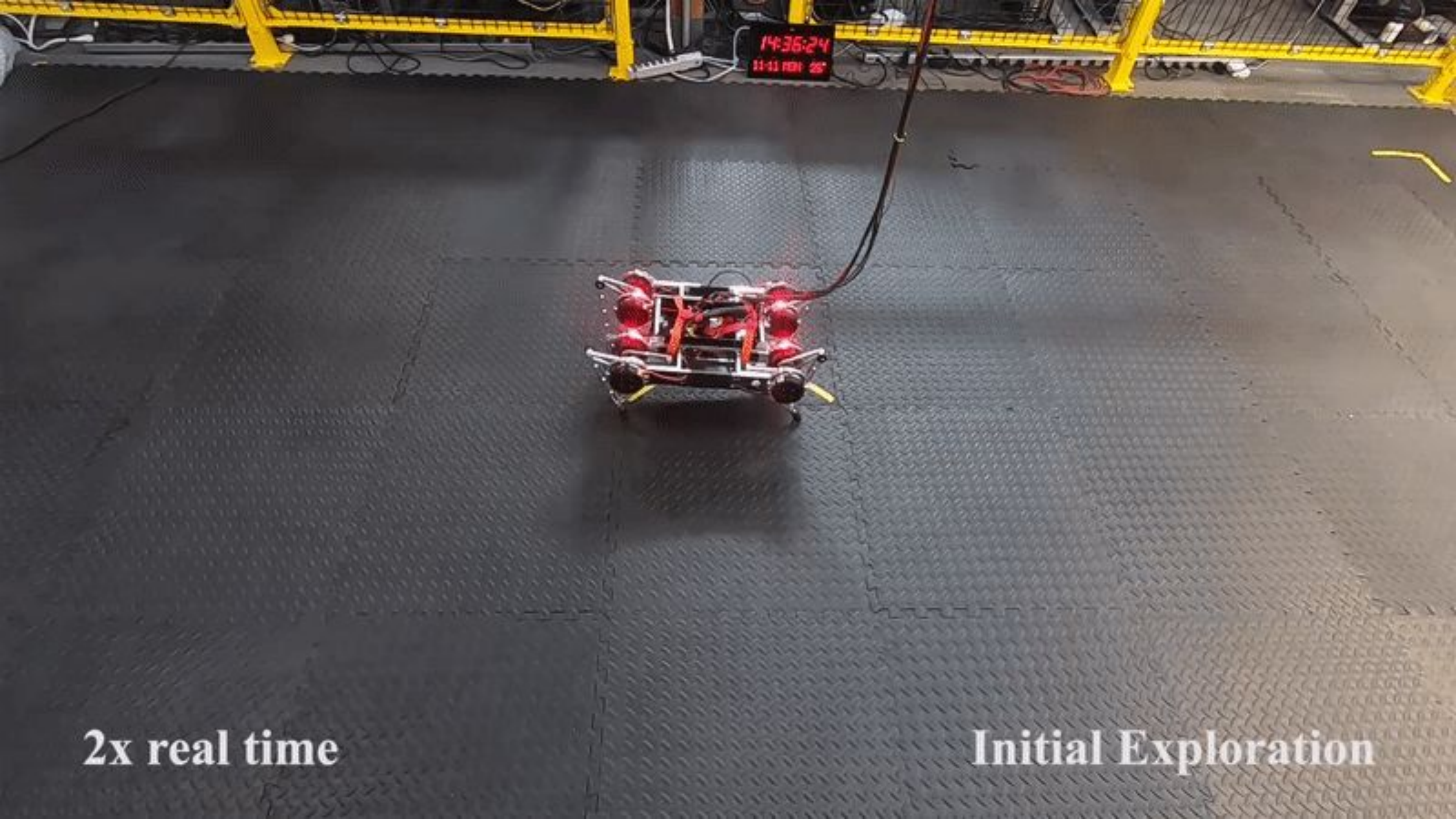
[DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills, Peng et al., SIGGRAPH 2018]

Learning in real world

- ✗ Data efficiency
- ✗ Human supervision

Learning in real world

- ✗ Data efficiency
- ✗ Human supervision
- ✗ Safety


14:36:24
2x real time
Initial Exploration

Problem Setup

Observations:

[
  8 motor angles,
  roll,
  pitch,
  prev_action
] * last 6 timesteps

Action:

8 desired motor angles

Reward function: $r_{\mathbf{w}}(\mathbf{s}, \mathbf{a}) = \boxed{[w_1, w_2]^T \cdot \mathbf{R}_0^{-1}(\mathbf{x}_t - \mathbf{x}_{t-1})} + \boxed{w_3(\theta_t - \theta_{t-1})} - \boxed{0.001|\ddot{\mathbf{a}}|^2}$

Learning to Walk in Real World with Minimal Human Efforts, Ha et al., CoRL, 2020

Soft Actor Critic

$$\max_{\pi \in \Pi} \mathbb{E}_{\tau \sim \rho_\pi} \left[\sum_{t=0}^{T} r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

$$\text{s.t. } \mathbb{E}_{\rho_\pi} \left[-\log \left(\pi_t(\cdot | \mathbf{s}_t) \right) \right] \geq \mathcal{H}$$

Learning to Walk in Real World with Minimal Human Efforts, Ha et al., CoRL, 2020

Safety-Constrained SAC: Formulation

Safety Constraints

$$\max_{\pi \in \Pi} \mathbb{E}_{\tau \sim \rho_\pi} \left[\sum_{t=0}^{T} r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

$$\text{s.t. } \boxed{\mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_\pi} \left[f_s(\mathbf{s}_t, \mathbf{a}_t) \right] \geq 0, \ \forall t}$$

$$\mathbb{E}_{\rho_\pi} \left[-\log \left(\pi_t(\cdot | \mathbf{s}_t) \right) \right] \geq \mathcal{H}$$

where

$$f_s(\mathbf{s}_t, \mathbf{a}_t) = \min(\hat{p} - |p_t|, \hat{r} - |r_t|)$$

Learning to Walk in Real World with Minimal Human Efforts, Ha et al., CoRL, 2020

Safety-Constrained SAC: Formulation

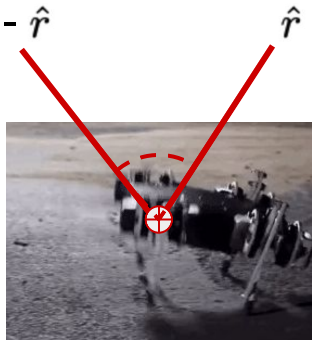
$$\max_{\pi \in \Pi} \mathbb{E}_{\tau \sim \rho_\pi} \left[\sum_{t=0}^{T} r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

$$\text{s.t.} \quad \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_\pi} \left[f_s(\mathbf{s}_t, \mathbf{a}_t) \right] \geq 0, \ \forall t.$$

$$\mathbb{E}_{\rho_\pi} \left[-\log \left(\pi_t(\cdot | \mathbf{s}_t) \right) \right] \geq \mathcal{H}$$

where

$$f_s(\mathbf{s}_t, \mathbf{a}_t) = \min(\hat{p} - |p_t|, \boxed{\hat{r}} - |r_t|)$$



Learning to Walk in Real World with Minimal Human Efforts, Ha et al., CoRL, 2020

Solving CMDP: Lagrangian Method

$$\max_{\pi \in \Pi} \mathbb{E}_{\tau \sim \rho_\pi} \left[\sum_{t=0}^{T} r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

$$\text{s.t.} \quad \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_\pi} \left[\sum f_s(\mathbf{s}_t, \mathbf{a}_t) \right] \geq 0$$

$$= \max_{\pi} \min_{\lambda \geq 0} \mathbb{E}_{\pi \sim \rho_\pi} \left[\sum_{t=0}^{T} r(s_t, a_t) + \lambda f_s(s_t, a_t) \right]$$

Lagrangian

$$\mathcal{L}(\pi, \lambda) = \mathbb{E}_{\tau \sim \rho_\pi} \left[\sum_{t=0}^{T} r(\mathbf{s}_t, \mathbf{a}_t) + \lambda f_s(\mathbf{s}_t, \mathbf{a}_t) \right]$$

$$\max_{\pi} \min_{\lambda \geq 0} \mathcal{L}(\pi, \lambda)$$

$$= 0 \quad \geq 0$$

$$\max_{\pi} \boxed{\min_{\lambda \geq 0}} \mathbb{E}_{\pi \sim \rho_\pi} \left[\sum_{t=0}^{T} r(s_t, a_t) + \lambda f_s(s_t, a_t) \right]$$

$$\to \infty \quad \leq 0$$

$$\max_{\pi} \boxed{\min_{\lambda \geq 0}} \mathbb{E}_{\pi \sim \rho_\pi} \left[\sum_{t=0}^{T} r(s_t, a_t) + \lambda f_s(s_t, a_t) \right]$$

Learning to Walk in Real World with Minimal Human Efforts, Ha et al., CoRL, 2020

$$\max_{\pi} \min_{\lambda \geq 0} \mathbb{E}_{\pi \sim \rho_\pi} \left[\sum_{t=0}^{T} r(s_t, a_t) + \lambda f_s(s_t, a_t) \right]$$

Pseudocode

1. Randomly initialize π, set $\lambda = 0$
2. Roll out policy π
3. Calculate policy gradient $\frac{\partial \mathcal{L}}{\partial \pi}$
4. $\pi = \pi + \alpha \frac{\partial \mathcal{L}}{\partial \pi}$
5. Calculate gradient $\frac{\partial \mathcal{L}}{\partial \lambda}$
6. $\lambda = \max(0, \lambda - \beta \frac{\partial \mathcal{L}}{\partial \lambda})$
7. Go to 2

Gradient ascend for policy

Learning to Walk in Real World with Minimal Human Efforts, Ha et al., CoRL, 2020

$$\max_{\pi} \min_{\lambda \geq 0} \mathbb{E}_{\pi \sim \rho_\pi} \left[\sum_{t=0}^{T} r(s_t, a_t) + \lambda f_s(s_t, a_t) \right]$$

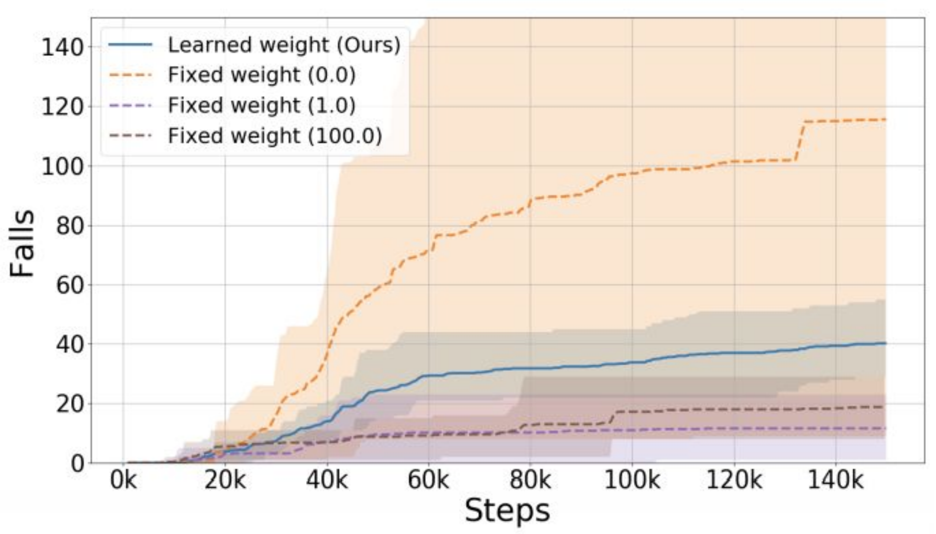
Pseudocode

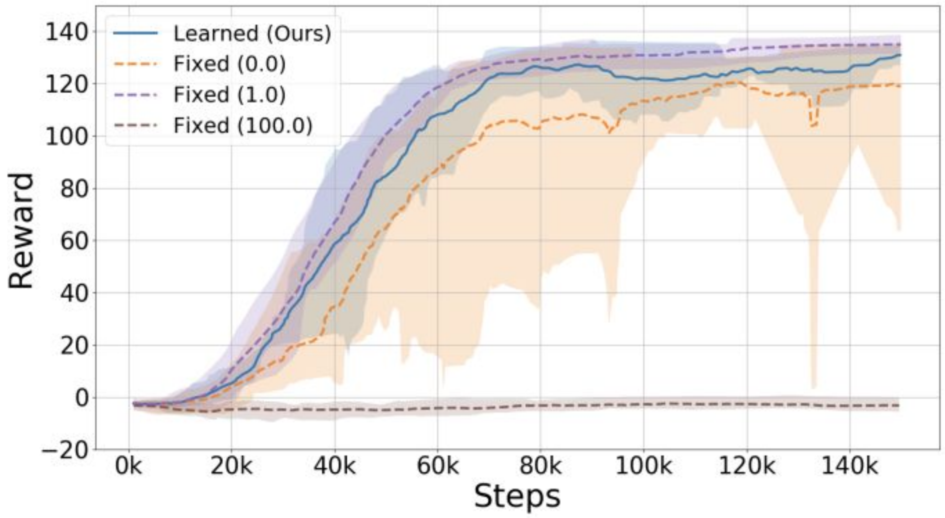
1. Randomly initialize π, set $\lambda = 0$
2. Roll out policy π
3. Calculate policy gradient $\frac{\partial \mathcal{L}}{\partial \pi}$
4. $\pi = \pi + \alpha \frac{\partial \mathcal{L}}{\partial \pi}$
5. Calculate gradient $\frac{\partial \mathcal{L}}{\partial \lambda}$
6. $\lambda = \max(0, \lambda - \beta \frac{\partial \mathcal{L}}{\partial \lambda})$
7. Go to 2

Gradient descent for Lagrangian multiplier

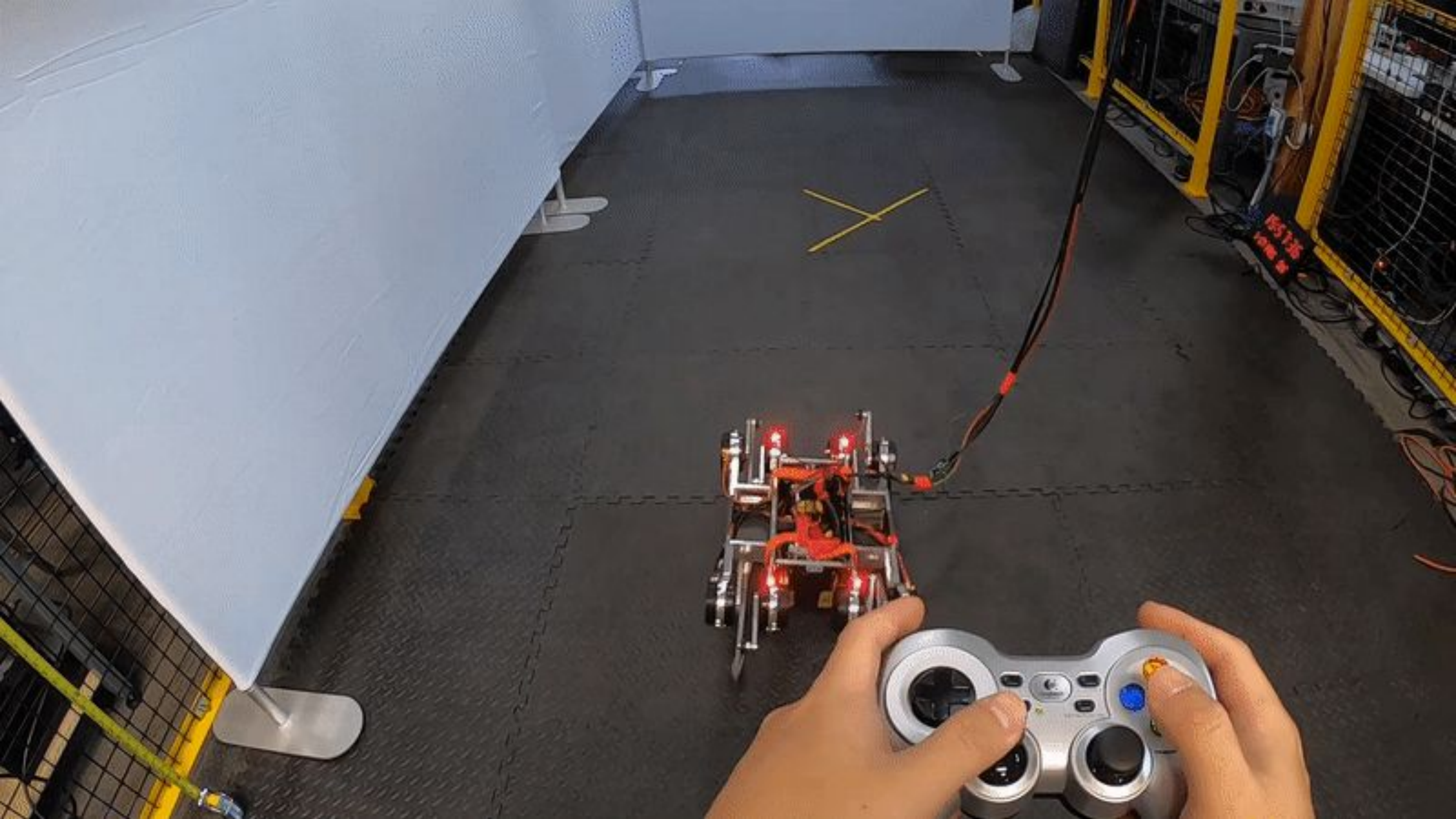
Learning to Walk in Real World with Minimal Human Efforts, Ha et al., CoRL, 2020

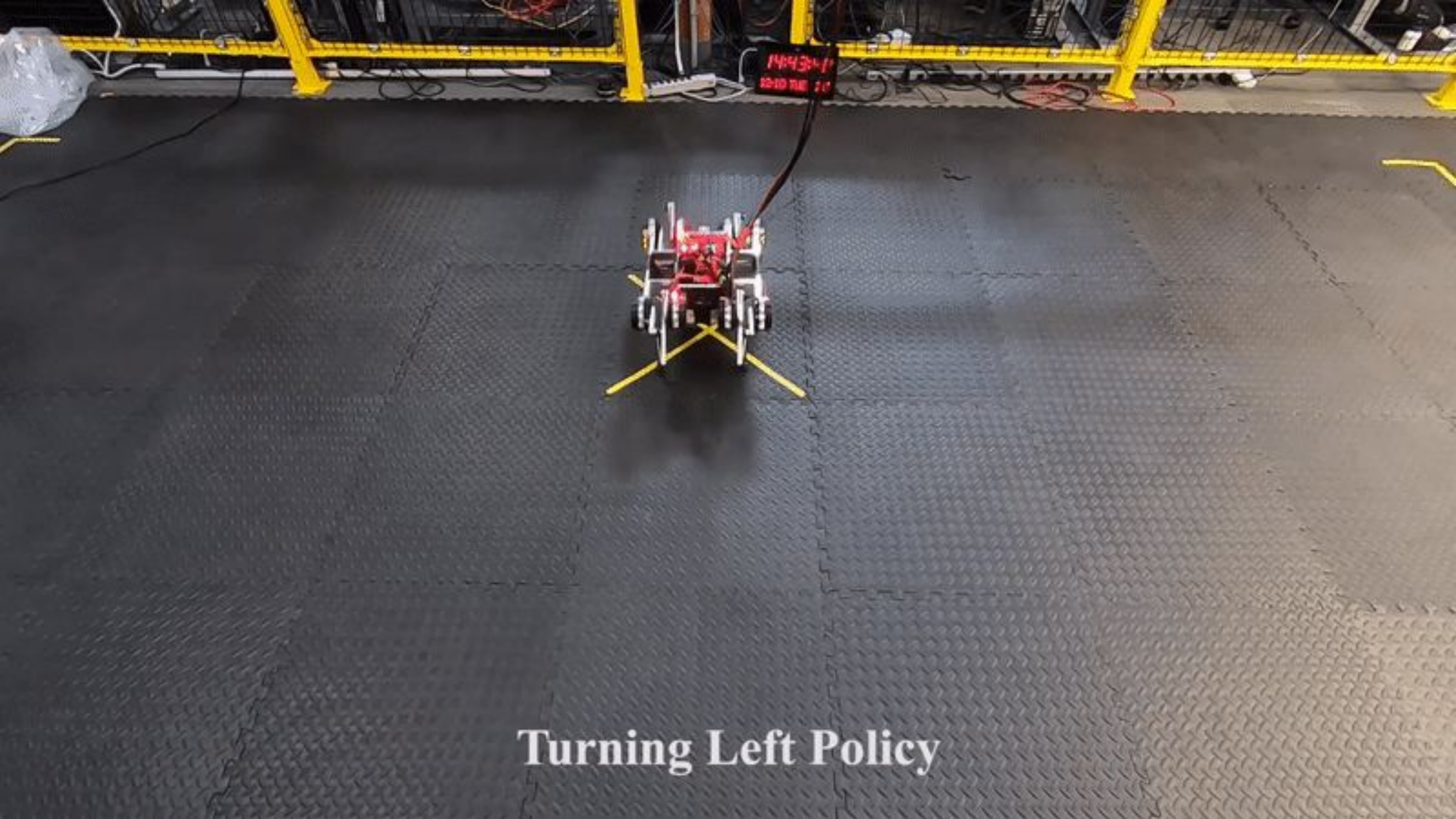
Safety-Constrained SAC: Evaluation



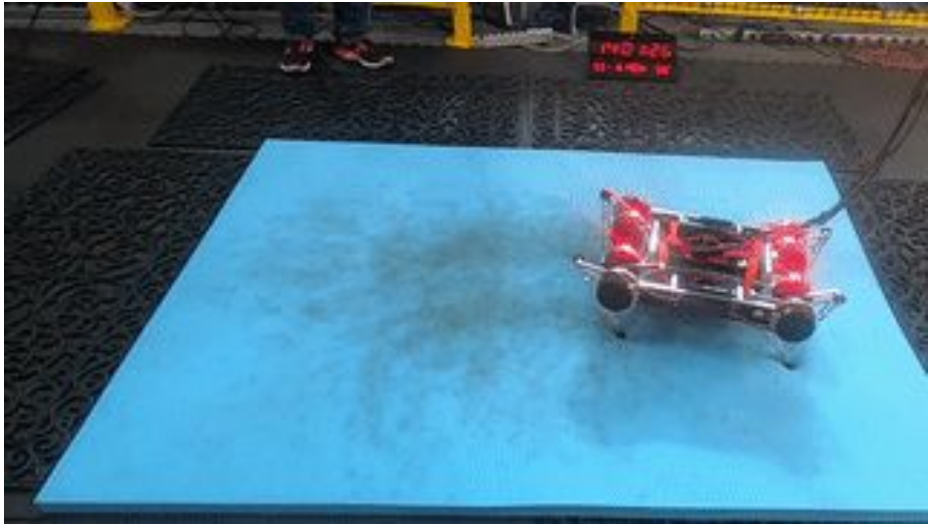


Learning to Walk in Real World with Minimal Human Efforts, Ha et al., CoRL, 2020

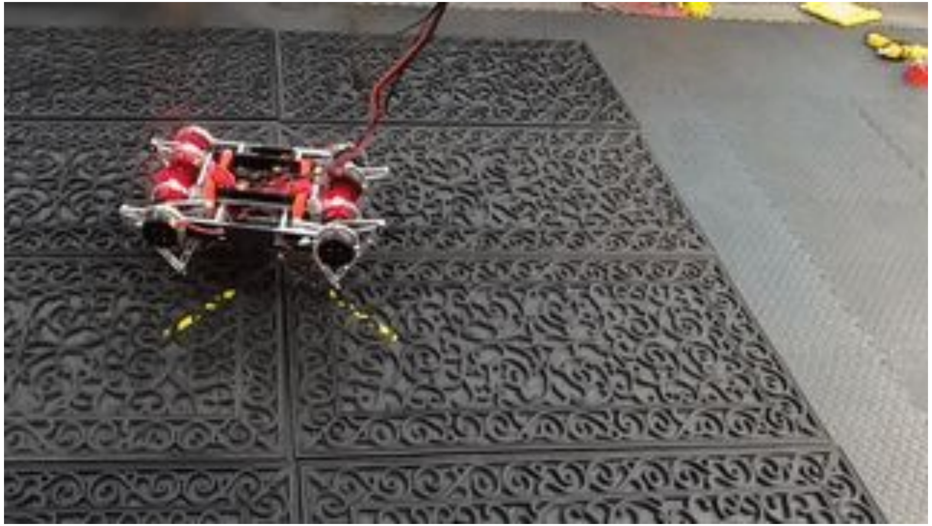



Turning Left Policy

Learning on challenging terrains



Memory foam



Rubber mat with crevices

Learning to Walk in Real World with Minimal Human Efforts, Ha et al., CoRL, 2020

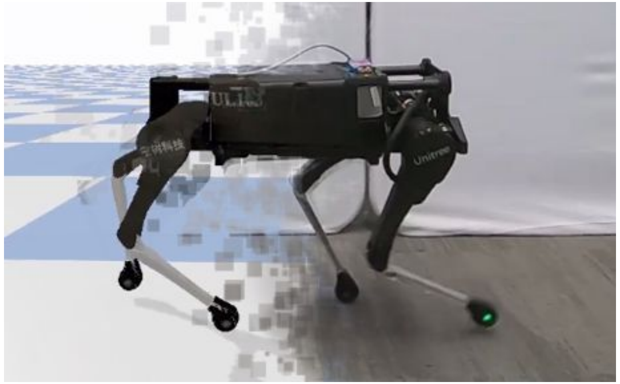
How to learn locomotion?

Approach 1

Directly train in the real world

Approach 2

Learn in simulation and sim-to-real transfer



Why Sim-to-Real?

Real world

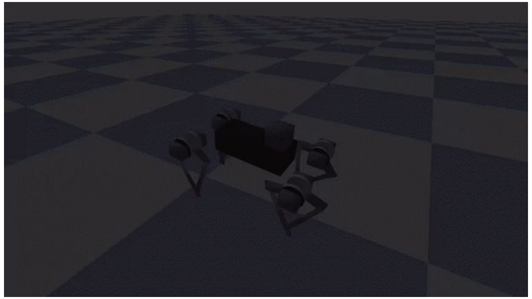
- Slow
- Unsafe
- Expensive
- Human supervision

Simulation

- Fast
- Safe
- Cheap
- Scalable

What’s the sim-to-real gap?

Dynamics:

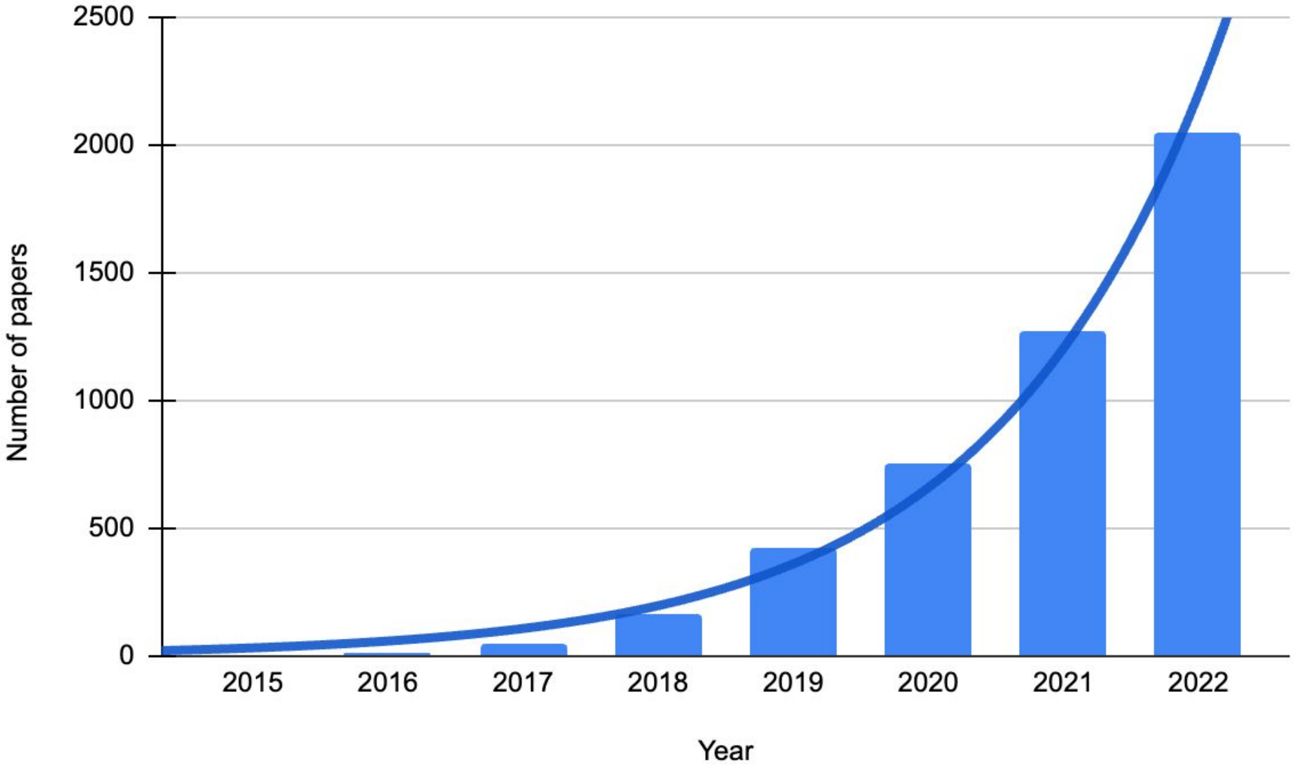


Perception:

What are the causes of sim-to-real gap?

- Unmodeled dynamics
- Wrong simulation parameters
- Inaccurate contact models
- Latency
- Actuator dynamics
- Noise
- Stochastic real environment
- Numerical accuracy
- ...

Trend on Sim-to-Real

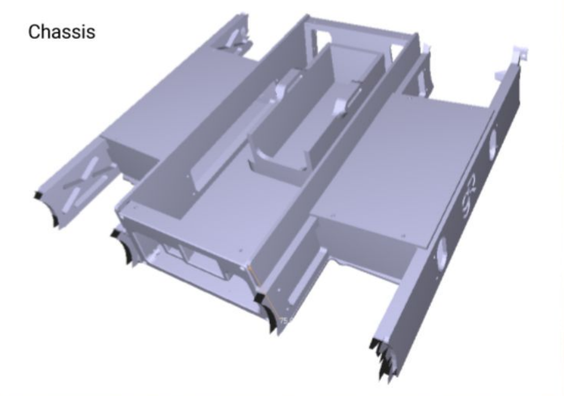


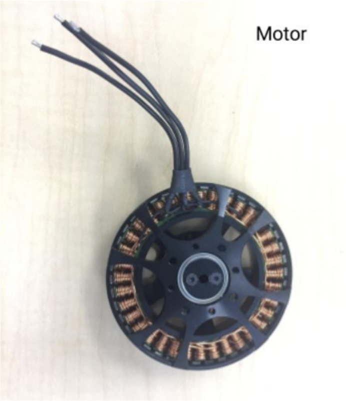
How to overcome sim-to-real gap?

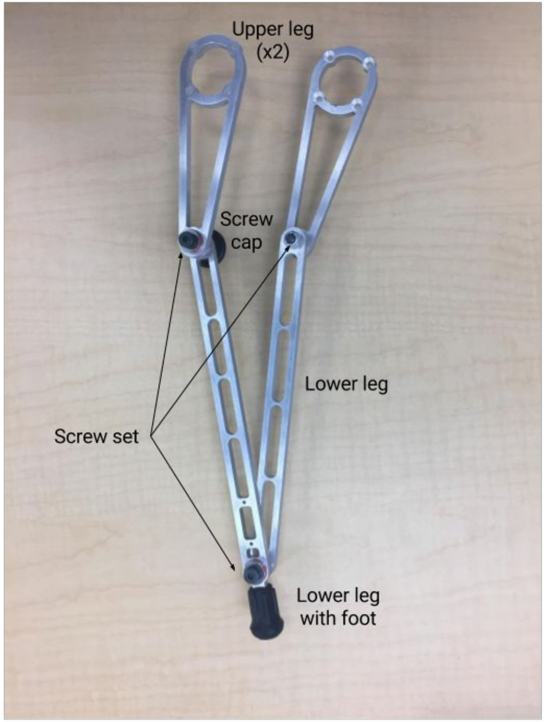
- Improve simulation
  - System identification
    - Sim-to-Real: Learning Agile Locomotion For Quadruped Robots
    - Simulation-Based Design of Dynamic Controllers for Humanoid Balancing
- Improve policy
  - Domain randomization
    - Sim-to-Real Transfer of Robotic Control with Dynamics Randomization
    - Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience
  - Domain adaptation
    - Learning Agile Robotic Locomotion Skills by Imitating Animals
    - Rapid Motor Adaptation for Legged Robots

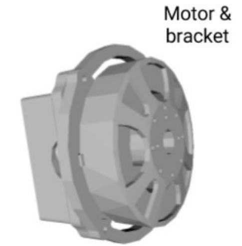
System Identification

System Identification





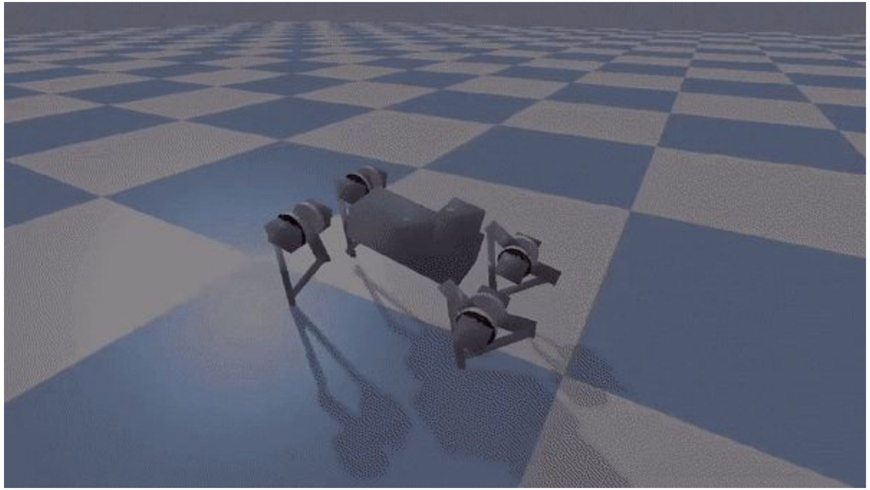




System Identification

- How to measure Mass?
- How to measure Center of Mass?
- How to measure Motor Damping (viscous friction)?
  - Spin the motor to a specific speed
  - Remove power
  - Record the data: motor speed vs. time
  - Fit the data based on physical equation about motor damping: $\tau_d = k\omega$
  - Find out motor damping coefficient k

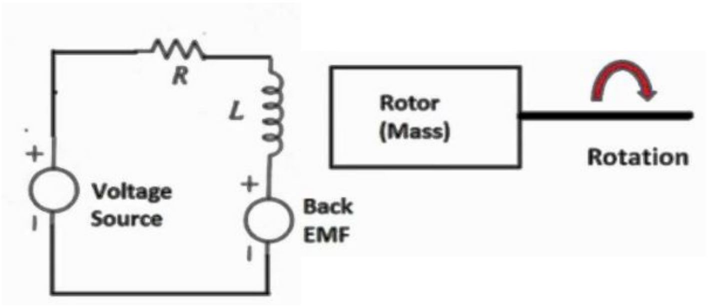
Actuator dynamics and **latency** are two important causes of sim-to-real gap.



[Sim-to-Real: Learning Agile Locomotion For Quadruped Robots, RSS 2018]

Actuator Model

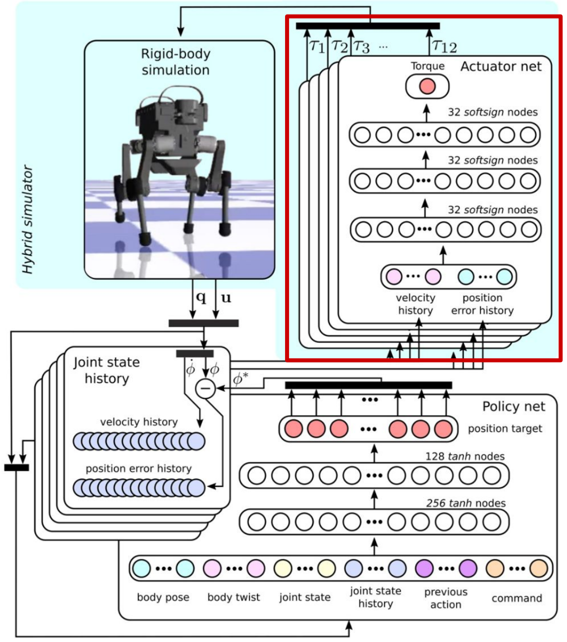
Analytical models



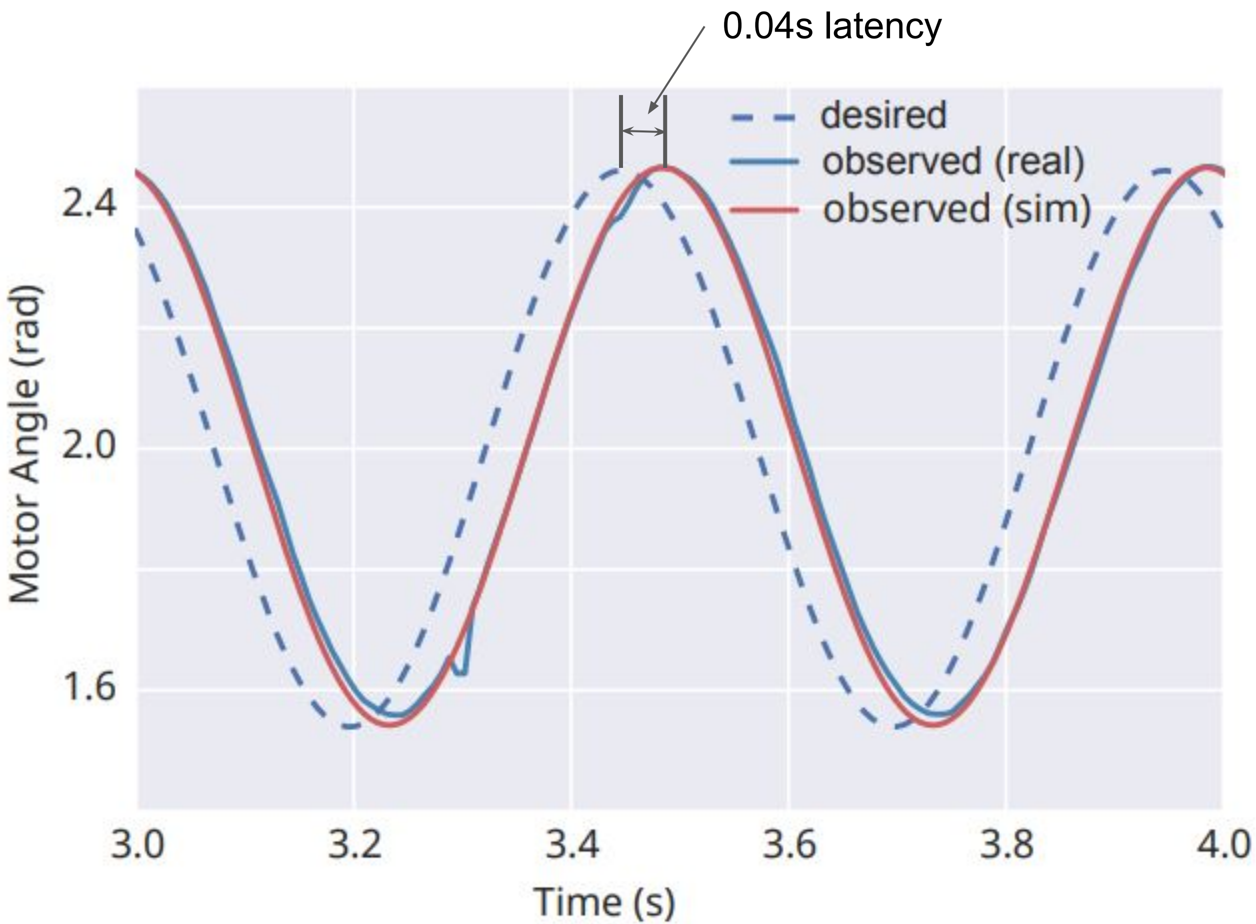
$$\tau = f(I)$$
$$I = \frac{V * \mathrm{PWM} - V_{\mathrm{emf}}}{R}$$
$$V_{\mathrm{emf}} = K_t \dot{q}$$

[Sim-to-Real: Learning Agile Locomotion For Quadruped Robots, RSS 2018]

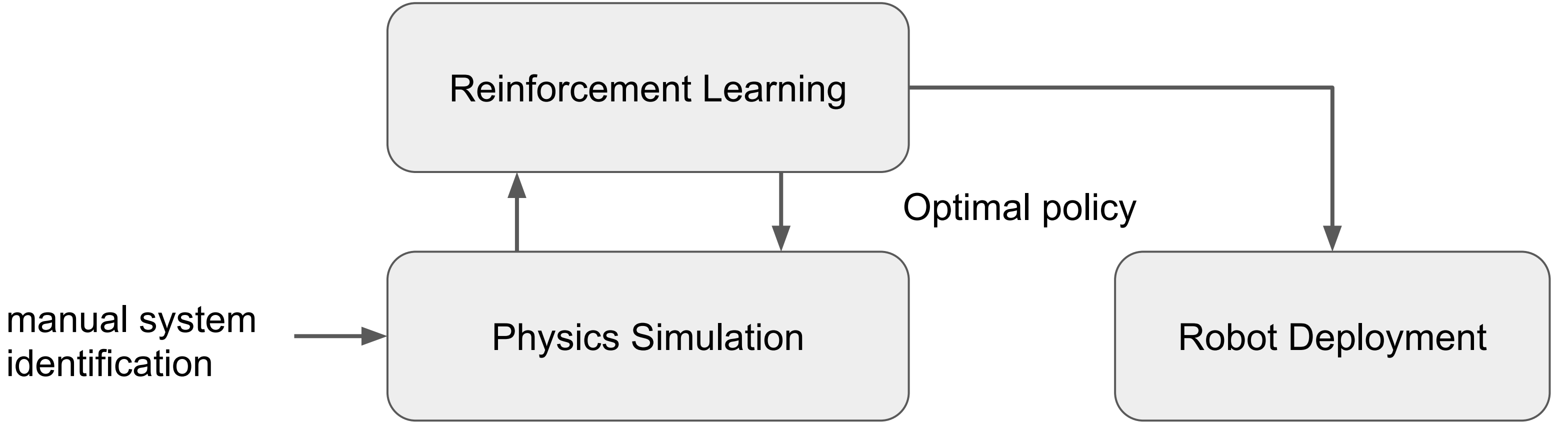
Neural network models



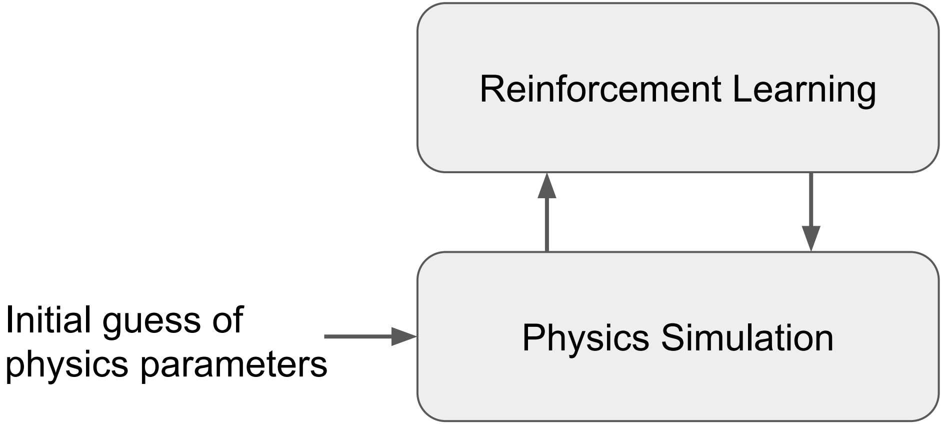
[Learning agile and dynamic motor skills for legged robots, Science Robotics 2019]



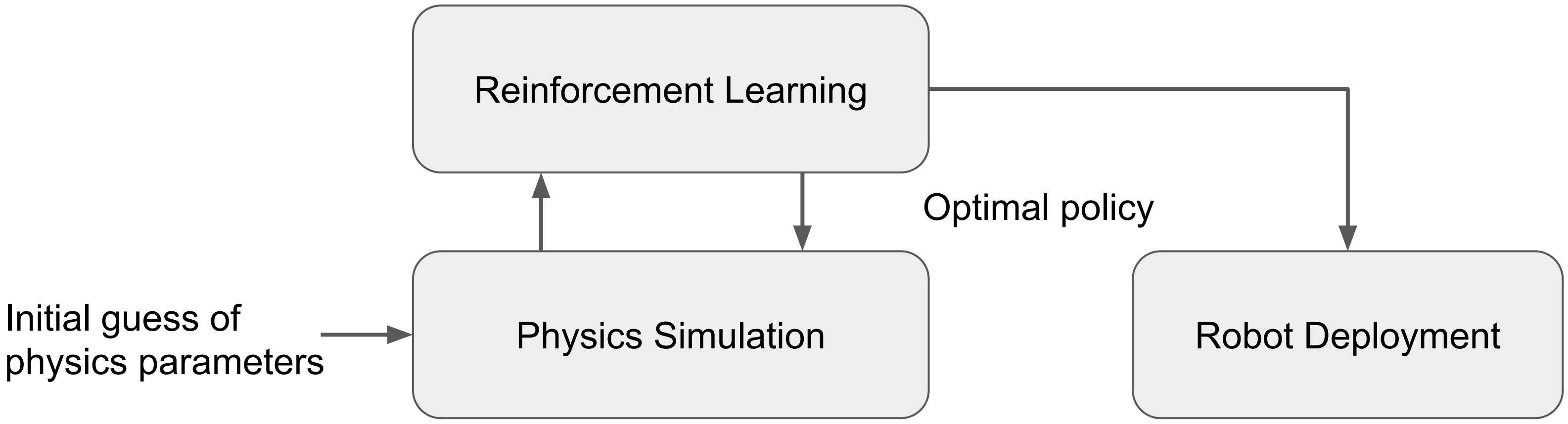
[Sim-to-Real: Learning Agile Locomotion For Quadruped Robots, RSS 2018]



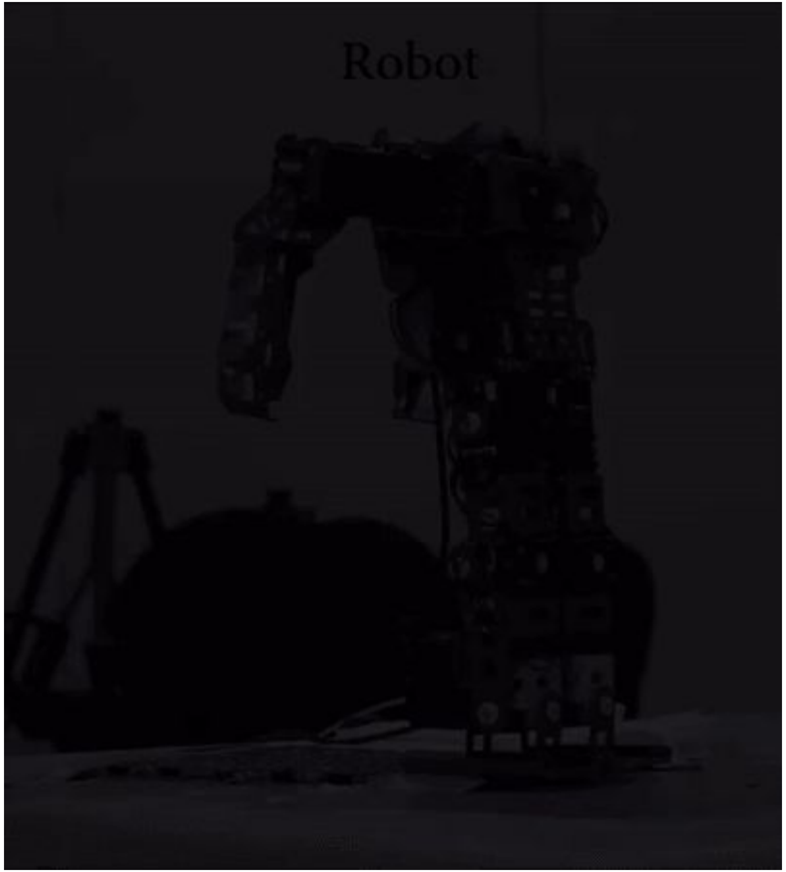
- Limitations
  - Disassemble the robot
  - Decide what parameters to identify
  - Design experiments for individual parameters
  - Lots of manual work



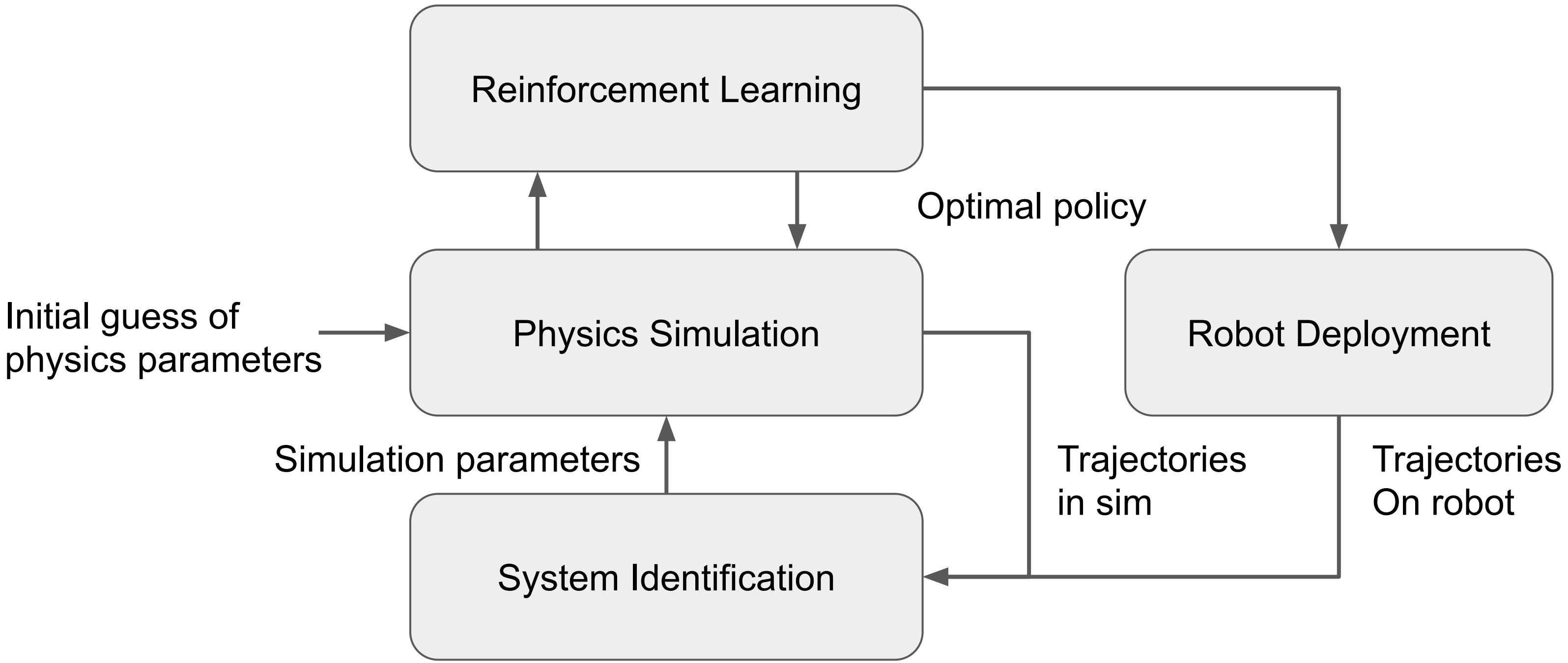
[Simulation-based design of dynamic controllers for humanoid balancing, IROS 2016]



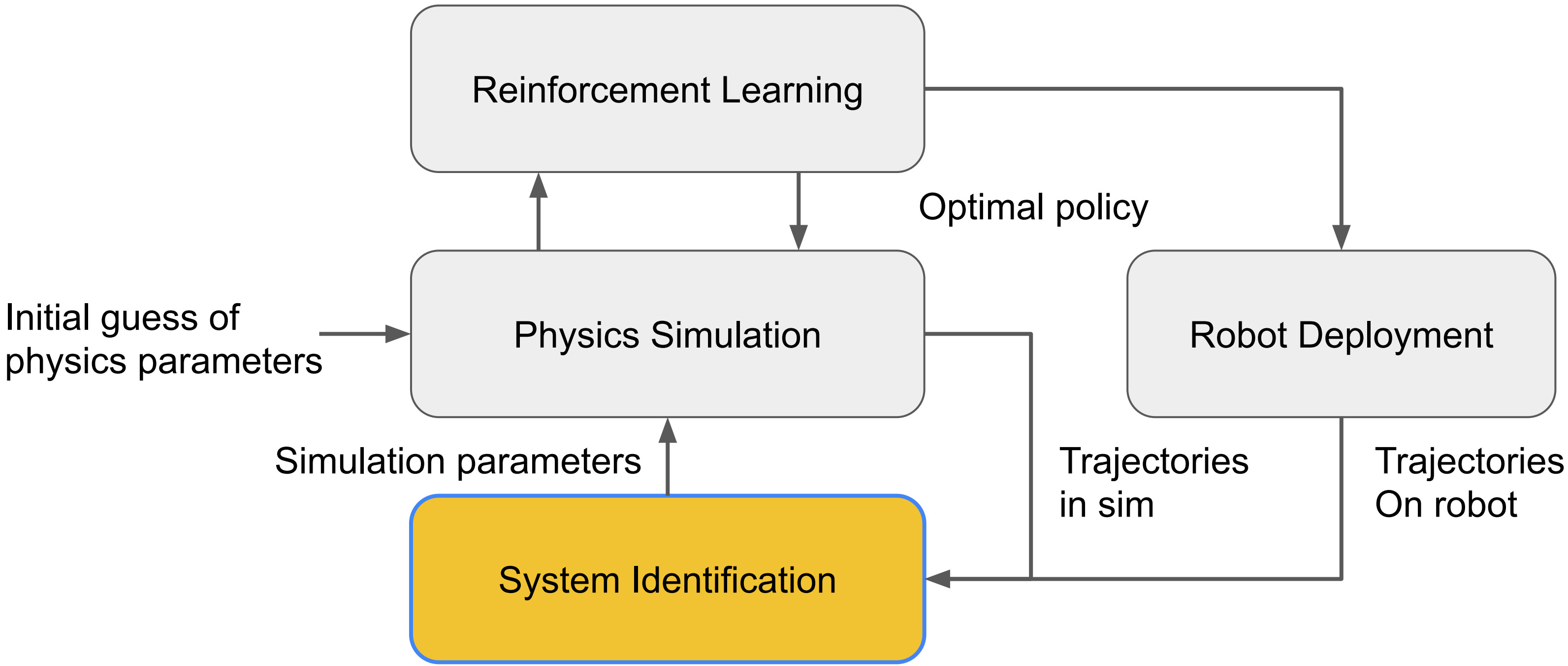
[Simulation-based design of dynamic controllers for humanoid balancing, IROS 2016]



[Simulation-based design of dynamic controllers for humanoid balancing, IROS 2016]



[Simulation-based design of dynamic controllers for humanoid balancing, IROS 2016]



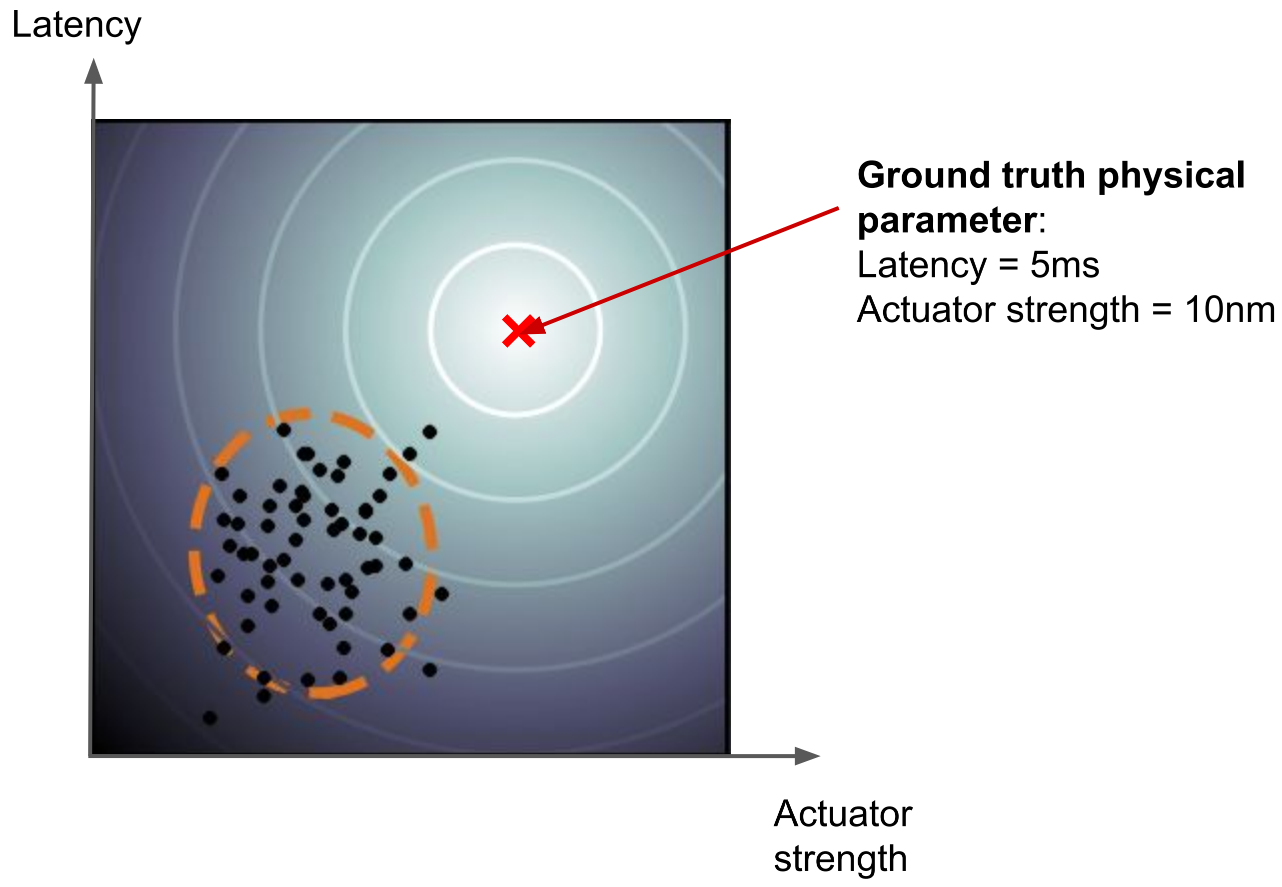
[Simulation-based design of dynamic controllers for humanoid balancing, IROS 2016]

Automatic System Identification

- Measure sim-to-real discrepancy

$$\boxed{\boldsymbol{\theta}} = \arg\min \boxed{\frac{1}{n}\sum_{i=1}^{n}} \int_0^{T+1} \|\boxed{\tilde{\mathbf{q}}_i(t)} - \boxed{\mathbf{q}_i(t;\boldsymbol{\theta})}\|_{\mathbf{W}}^2 \mathrm{d}t$$

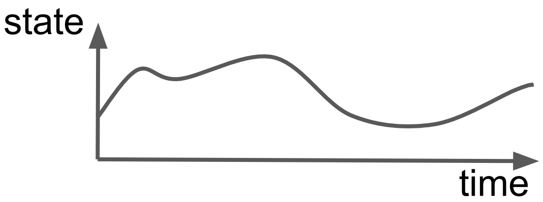
- Optimize the physics parameters
  - [Covariance Matrix Adaptation-Evolution Strategy]()



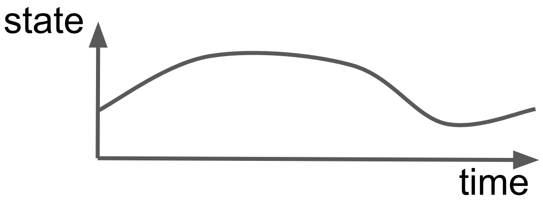
Latency
Actuator strength
Ground truth physical parameter:
Latency = 5ms
Actuator strength = 10nm

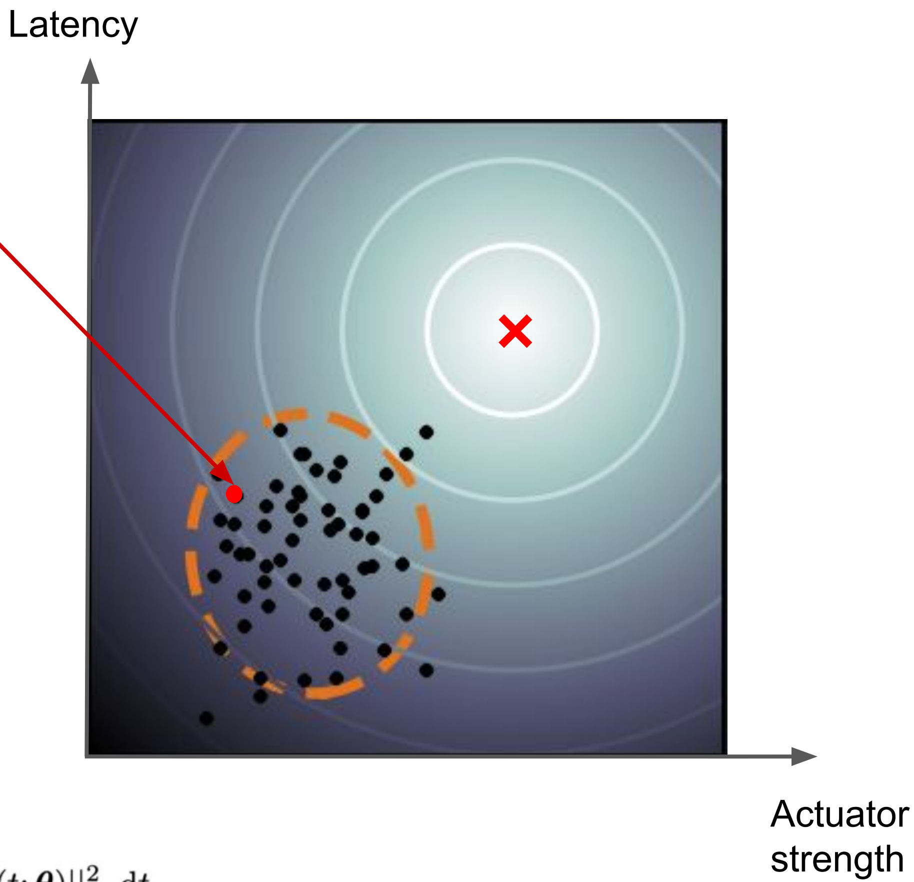
Randomly sampled physical parameter:
Latency = 1ms
Actuator strength = 2Nm

Sim trajectory:

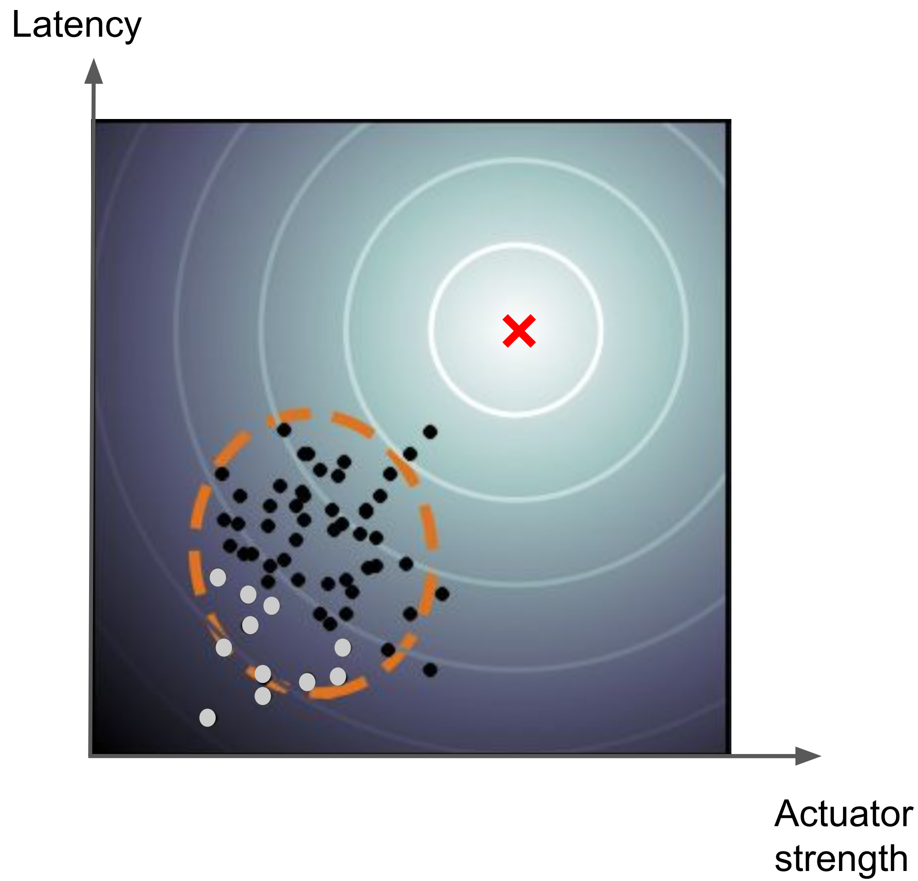


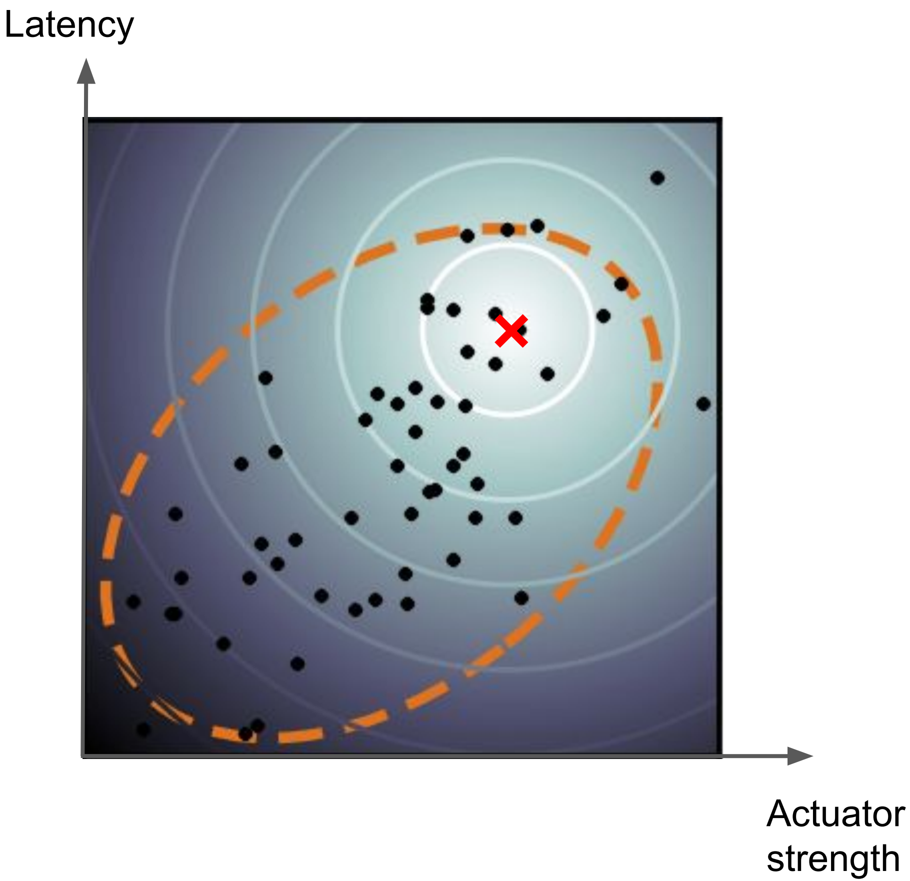
Real trajectory:

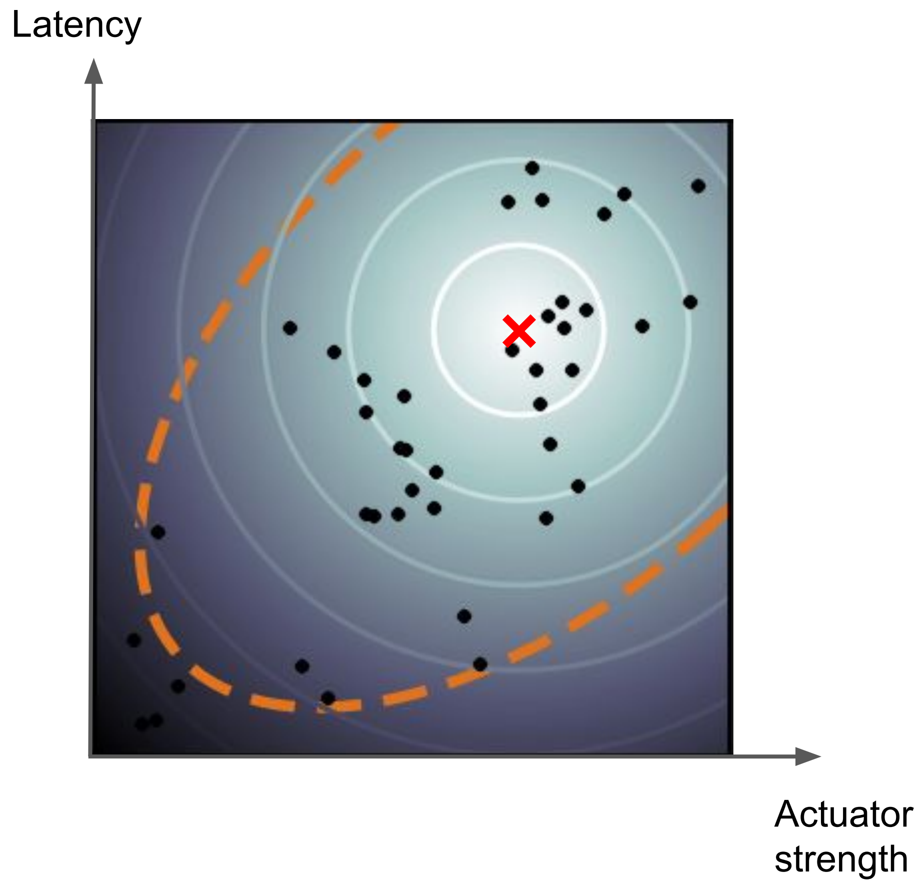


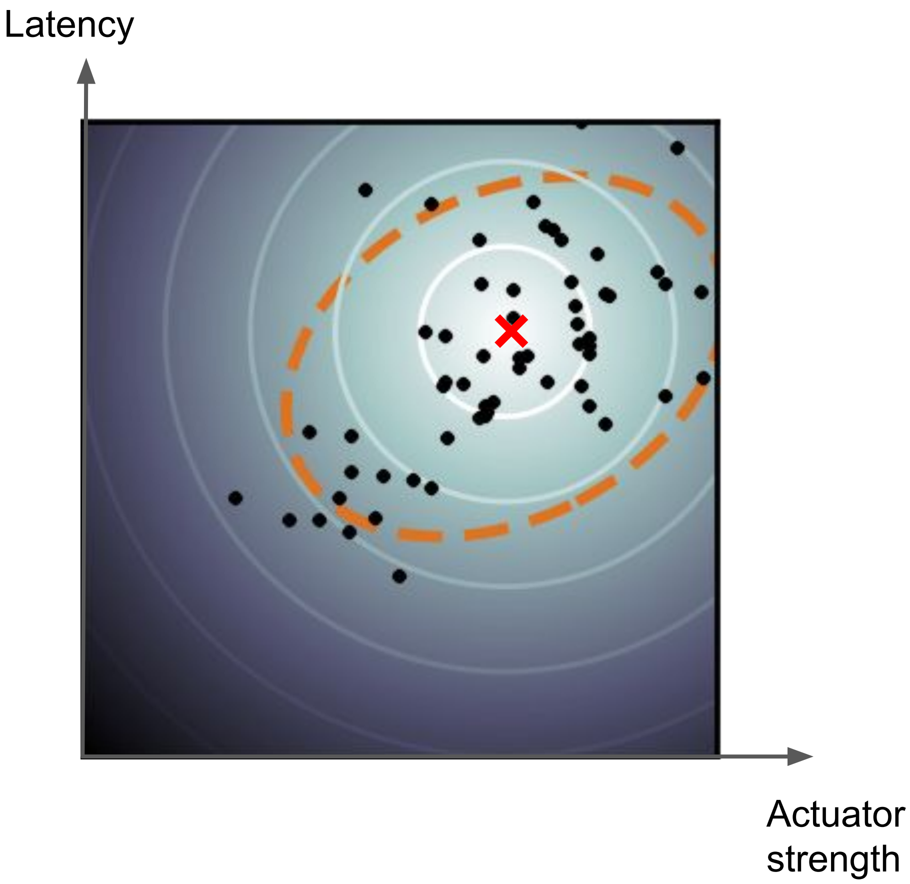


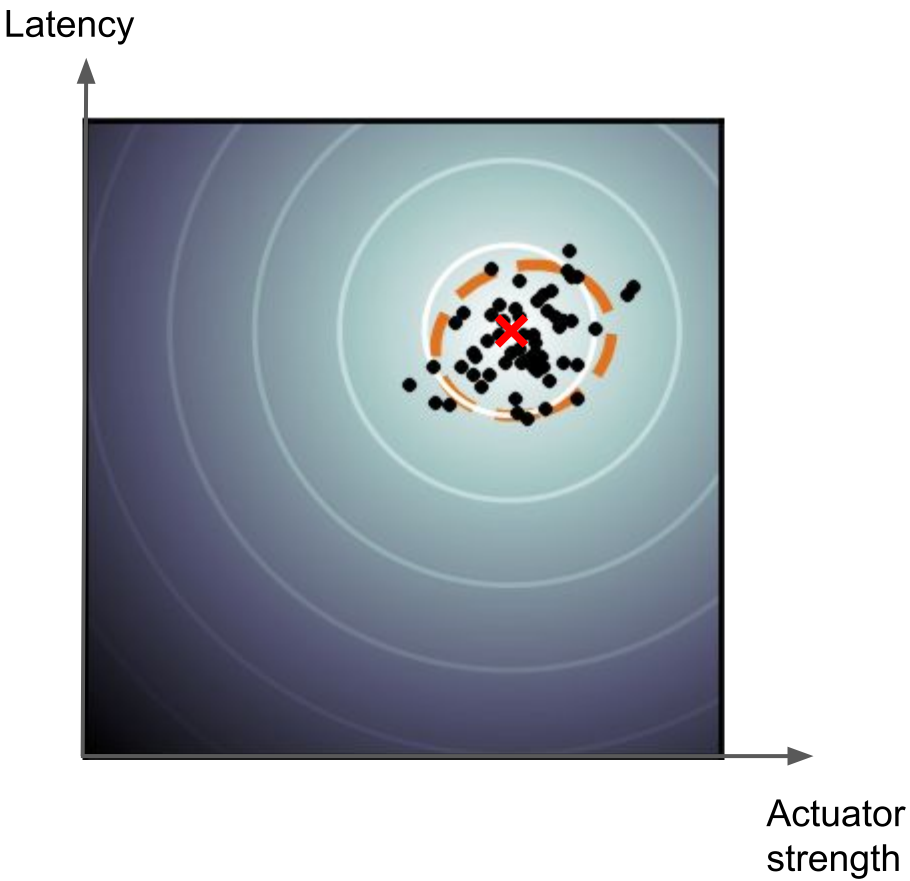
Loss: $\frac{1}{n}\sum_{i=1}^{n}\int_{0}^{T+1} ||\tilde{\mathbf{q}}_i(t) - \mathbf{q}_i(t;\boldsymbol{\theta})||^2_{\mathbf{W}}\,\mathrm{d}t$

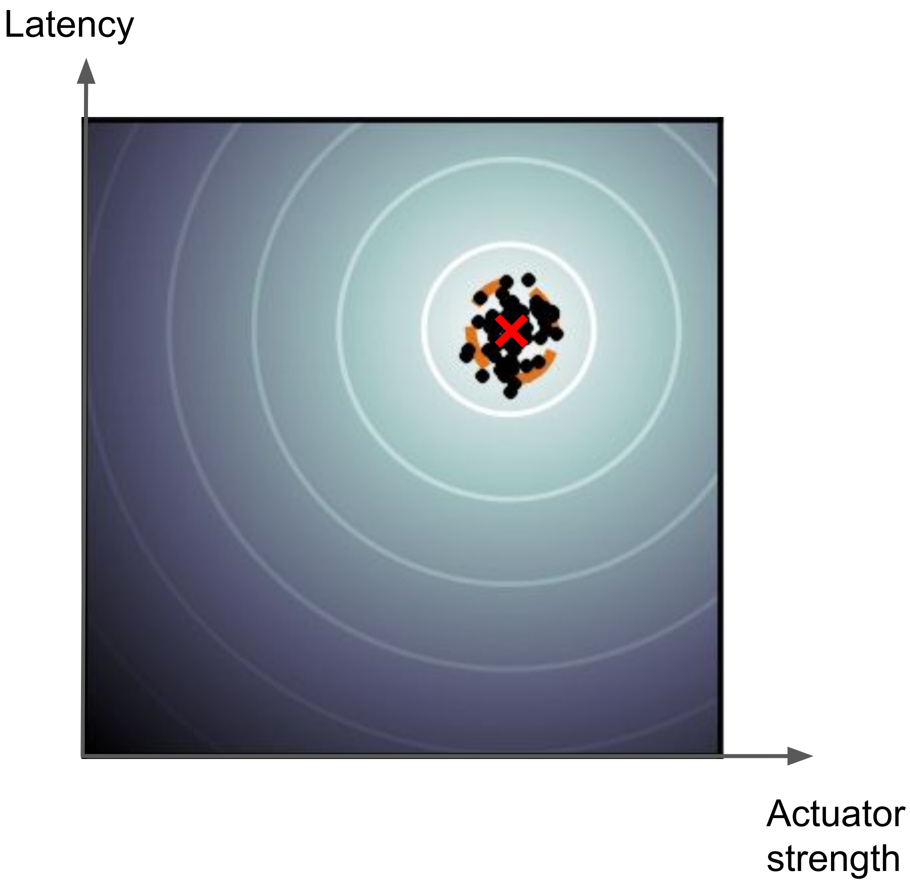

Latency
Actuator
strength

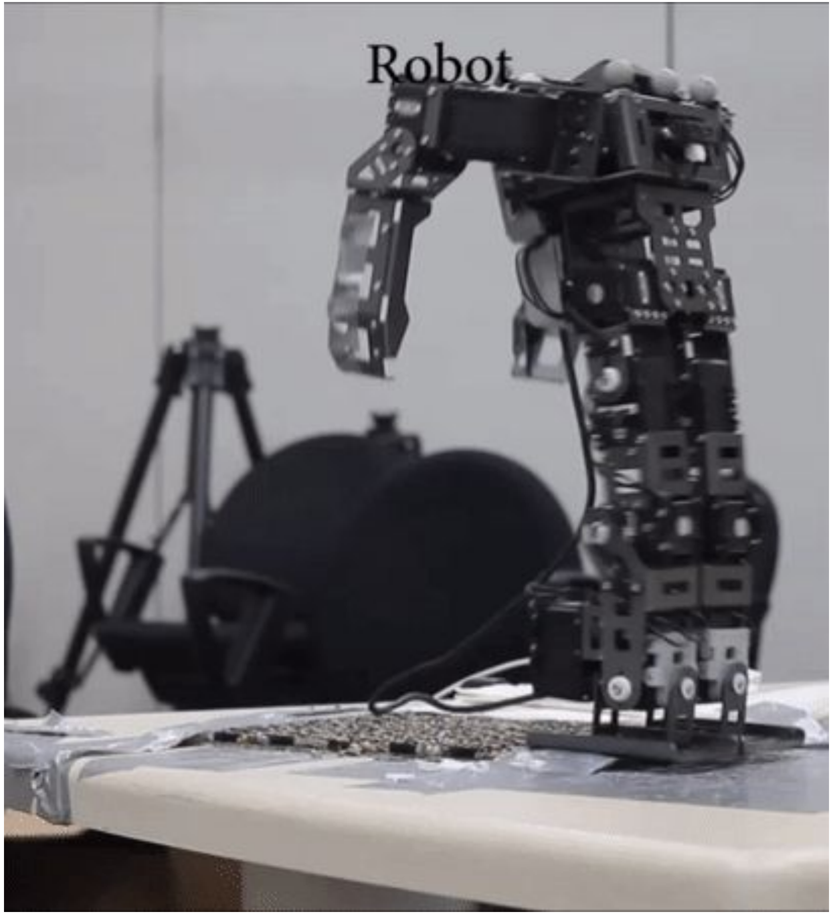

Latency
Actuator
strength


Latency
Actuator strength


Latency
Actuator strength

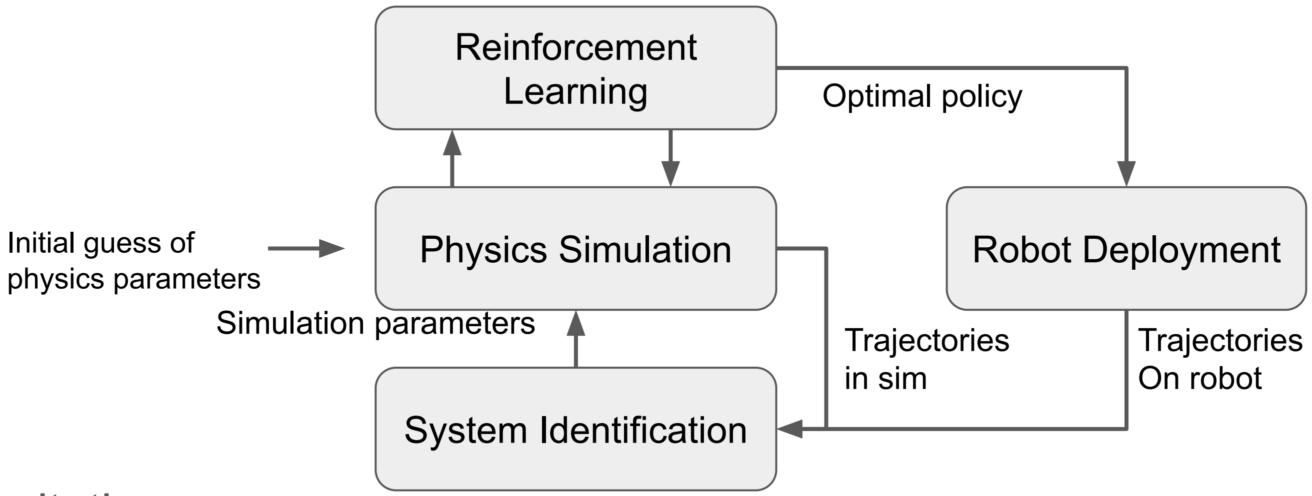

Latency
Actuator
strength


Latency
Actuator strength



[Simulation-based design of dynamic controllers for humanoid balancing, IROS 2016]

Automatic System Identification



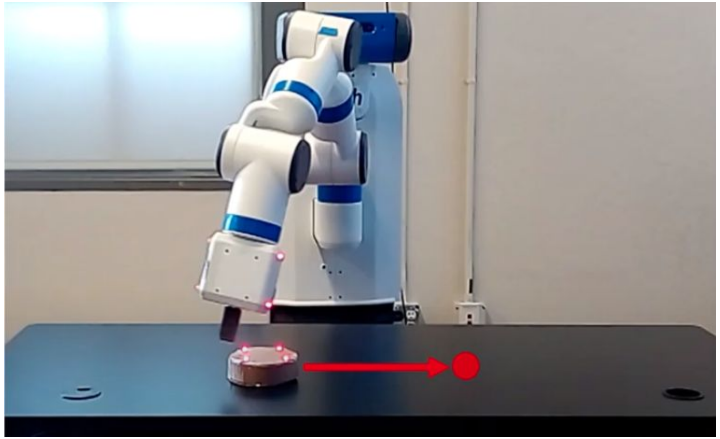
- Limitations
    - Manual selection of physical parameters needed
    - Do not work if sim and real trajectory diverge too quickly
    - Not account for unmodeled dynamics
    - Physical parameters overfit

Domain Randomization

Domain Randomization

- Original objective: reward maximization

$$\mathbb{E}_{\tau \sim p(\tau|\pi)} \left[\sum_{t=0}^{T-1} r(s_t, a_t) \right]$$



[Sim-to-Real Transfer of Robotic Control with Dynamics Randomization, ICRA 2018]

Domain Randomization

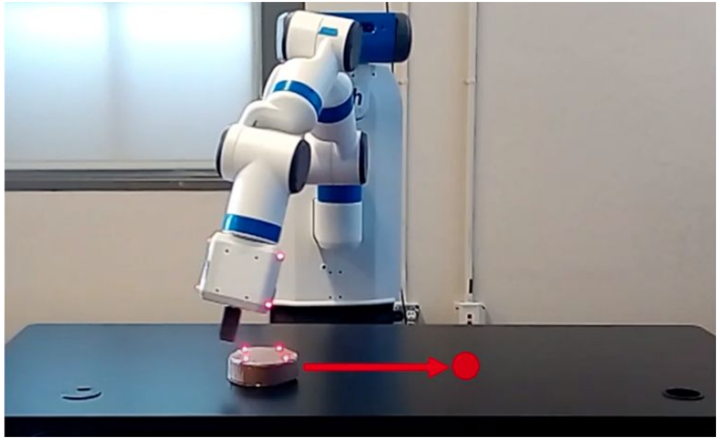
- Original objective: reward maximization

$$\mathbb{E}_{\tau \sim p(\tau|\pi)}\left[\sum_{t=0}^{T-1} r(s_t, a_t)\right]$$

- New objective with domain randomization

$$\boxed{\mathop{\mathbb{E}}_{\mu \sim \rho_\mu}}\left[\mathbb{E}_{\tau \sim p(\tau|\pi\boxed{,\mu})}\left[\sum_{t=0}^{T-1} r(s_t, a_t)\right]\right]$$

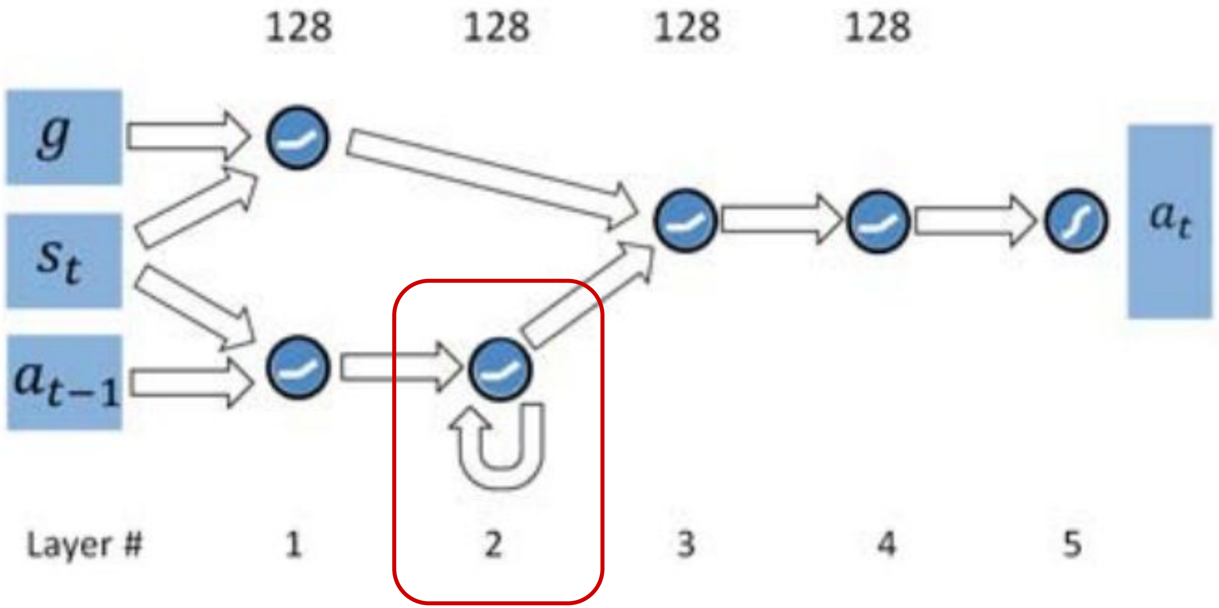
Physical parameters



Parameter	Range
Link Mass	$[0.25, 4] \times$ default mass of each link
Joint Damping	$[0.2, 20] \times$ default damping of each joint
Puck Mass	$[0.1, 0.4] kg$
Puck Friction	$[0.1, 5]$
Puck Damping	$[0.01, 0.2] Ns/m$
Table Height	$[0.73, 0.77] m$
Controller Gains	$[0.5, 2] \times$ default gains
Action Timestep λ	$[125, 1000] s^{-1}$

[Sim-to-Real Transfer of Robotic Control with Dynamics Randomization, ICRA 2018]

Memory (LSTM) in sim-to-real



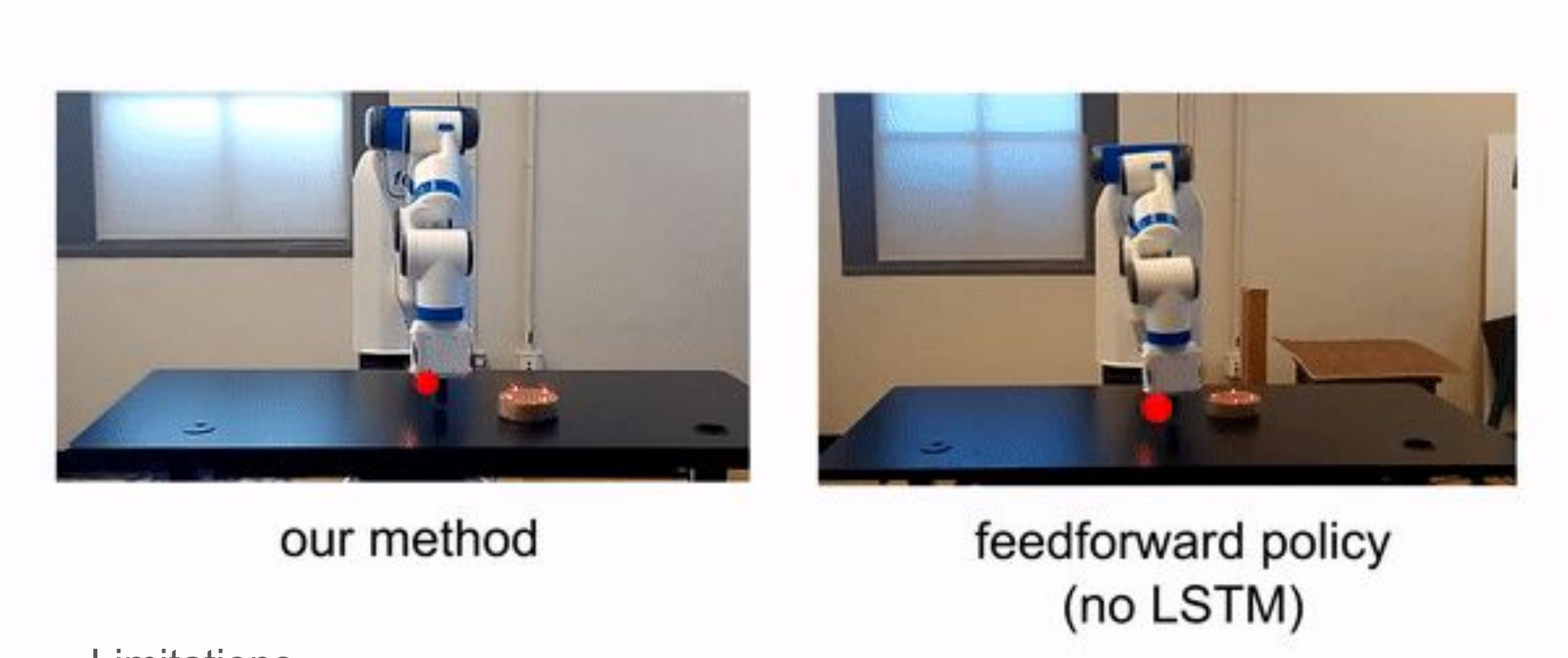
[Sim-to-Real Transfer of Robotic Control with Dynamics Randomization, ICRA 2018]

our method

no randomization during training

- Limitations
    - Trade optimality for robustness
    - Careful tuning needed for the range of randomization

[Sim-to-Real Transfer of Robotic Control with Dynamics Randomization, ICRA 2018]



- Limitations
  - Trade optimality for robustness
  - Careful tuning needed for the range of randomization

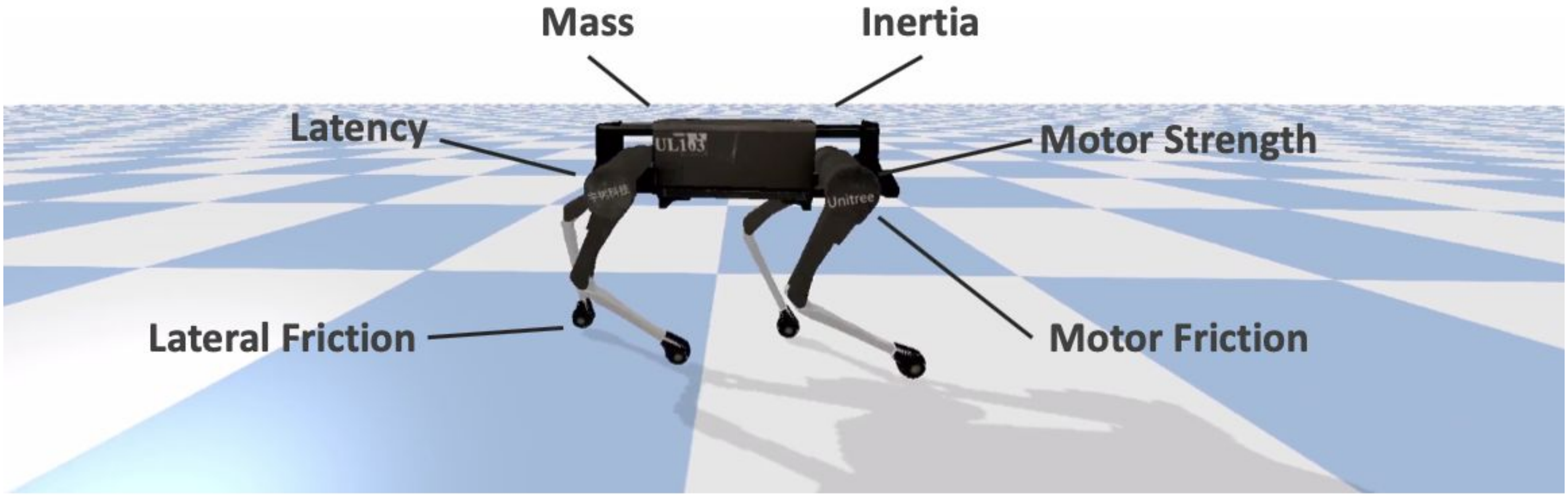
[Sim-to-Real Transfer of Robotic Control with Dynamics Randomization, ICRA 2018]

Can we use a small amount of real data to mitigate these limitations?

- Limitations
    - Trade optimality for robustness
    - Careful tuning needed for the range of randomization

Domain Adaptation

Domain Adaptation



[Learning Agile Robotic Locomotion Skills by Imitating Animals, RSS 2020]

Domain Adaptation

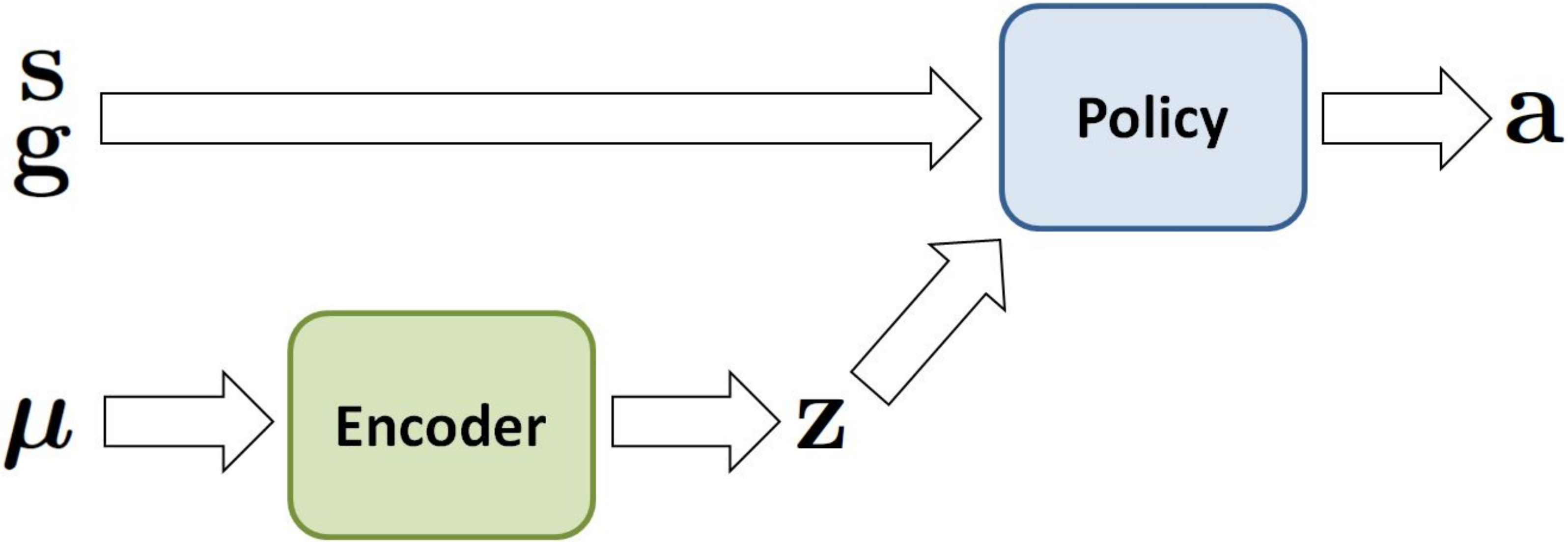
$$\mu = \begin{bmatrix} \text{Mass} \\ \text{Inertia} \\ \text{Motor Strength} \\ \text{Motor Friction} \\ \text{Latency} \\ \text{Lateral Friction} \\ \text{Etc...} \end{bmatrix}$$

[Learning Agile Robotic Locomotion Skills by Imitating Animals, RSS 2020]

Domain Adaptation

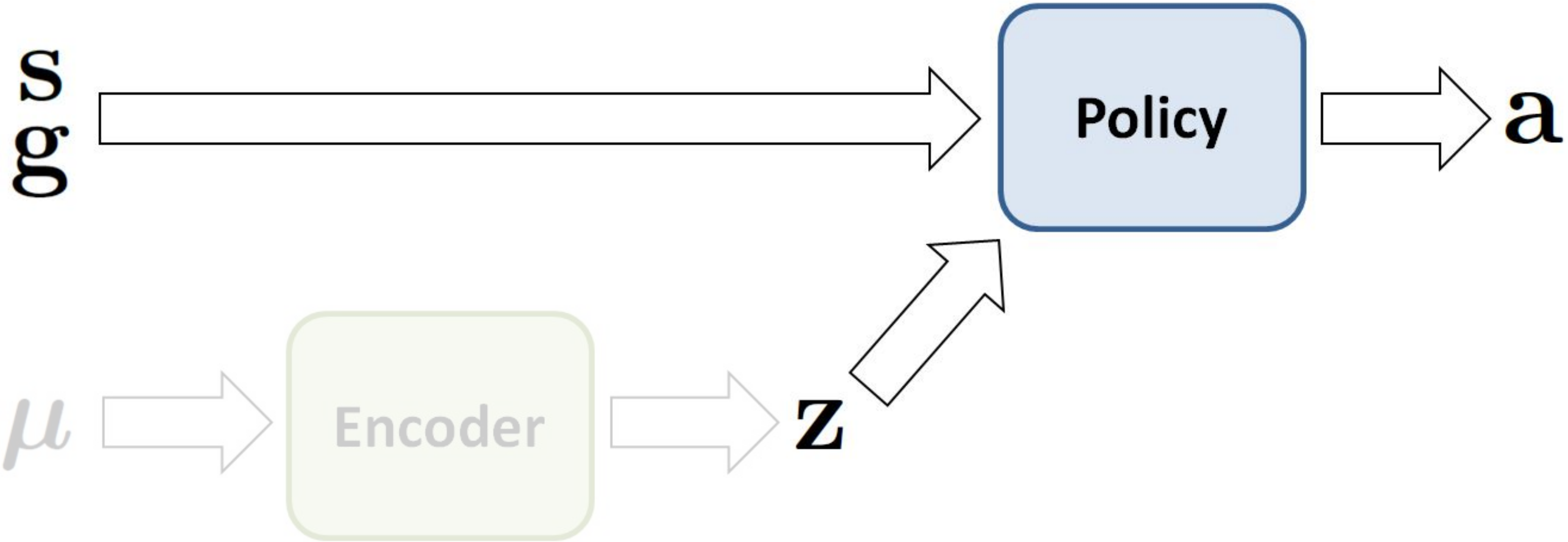
[Learning Agile Robotic Locomotion Skills by Imitating Animals, RSS 2020]

Domain Adaptation



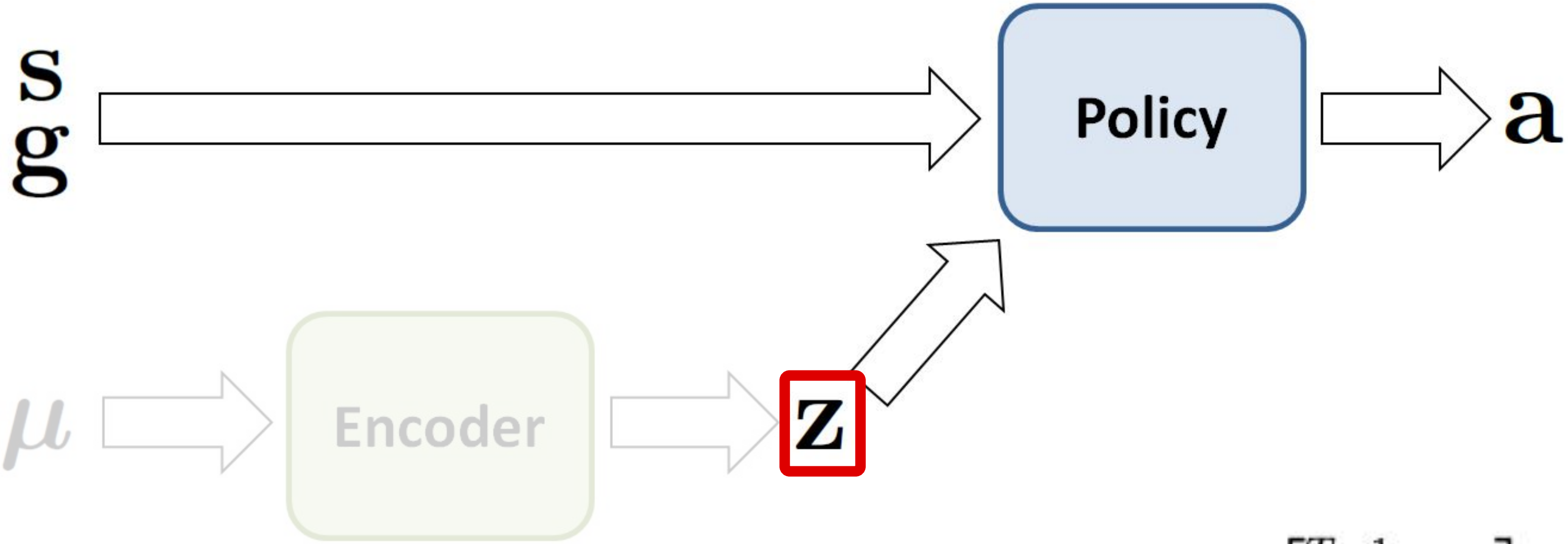
[Learning Agile Robotic Locomotion Skills by Imitating Animals, RSS 2020]

Domain Adaptation



[Learning Agile Robotic Locomotion Skills by Imitating Animals, RSS 2020]

Domain Adaptation



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

[Learning Agile Robotic Locomotion Skills by Imitating Animals, RSS 2020]

Domain Adaptation vs. Domain Randomization

Dog Pace

No Randomization

Randomization

Domain Adaptation (Ours)

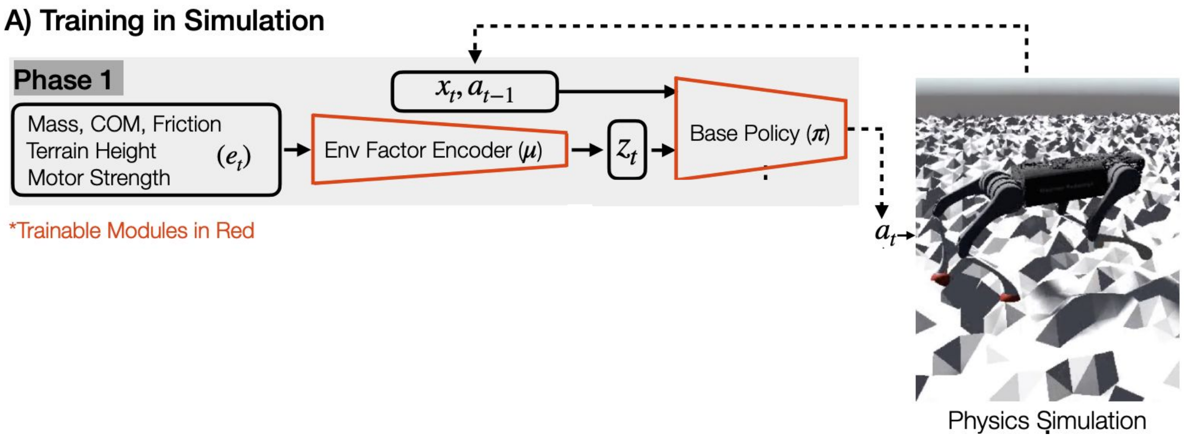
Domain Adaptation vs. Domain Randomization

Dog Spin

No Randomization | Randomization | Domain Adaptation (Ours)

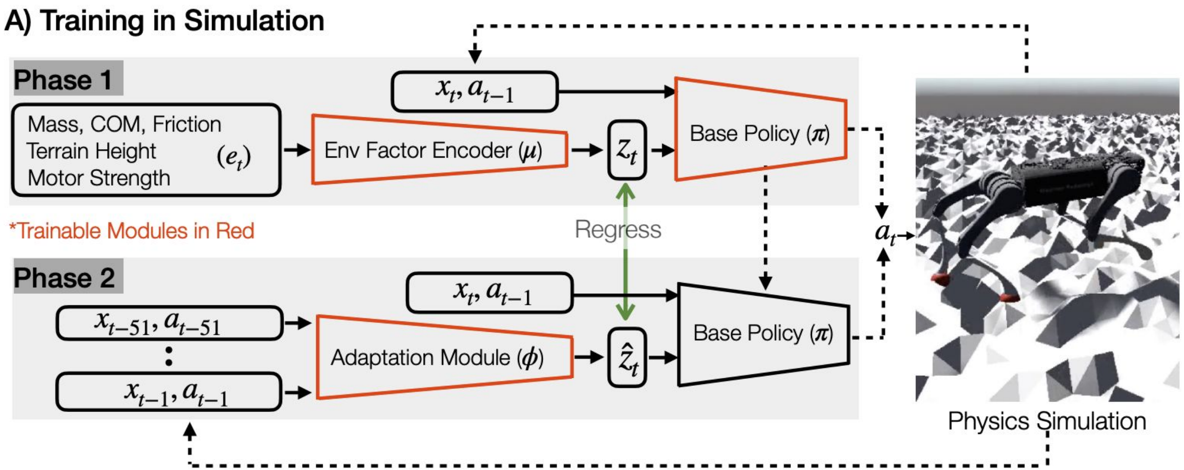
- Limitations
  - The latent space may not contain the optimal vector for the real world
  - Policy is not updated: Performance does not necessarily improve with more real data
  - Adaptation is slow (requires a few episodes)

Domain Adaptation: Rapid Motor Adaptation (RMA)



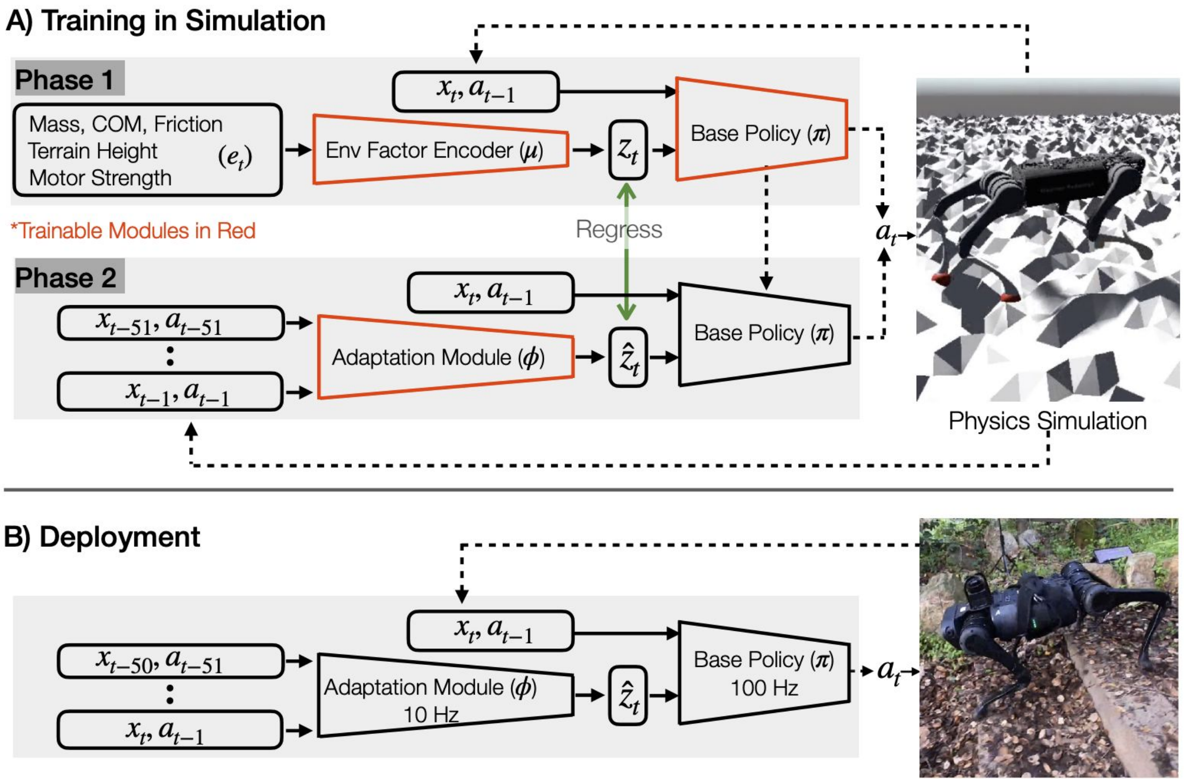
[RMA: Rapid Motor Adaptation for Legged Robots, RSS 2021]

Domain Adaptation: Rapid Motor Adaptation (RMA)



[RMA: Rapid Motor Adaptation for Legged Robots, RSS 2021]

Domain Adaptation: Rapid Motor Adaptation (RMA)

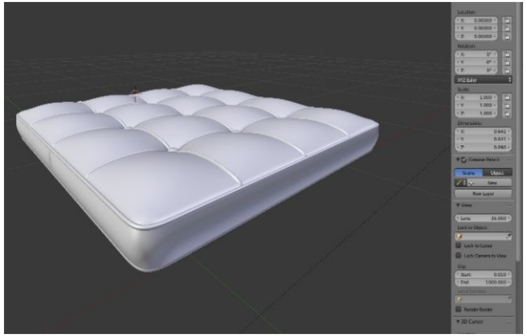


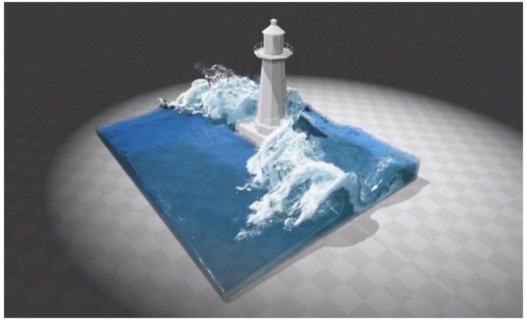
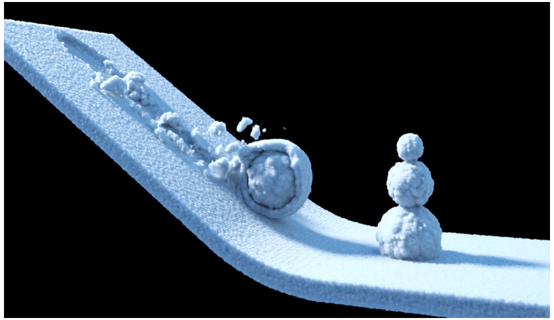
[RMA: Rapid Motor Adaptation for Legged Robots, RSS 2021]


Vegetation on uneven surface

Discuss: Does sim-to-real solve everything?

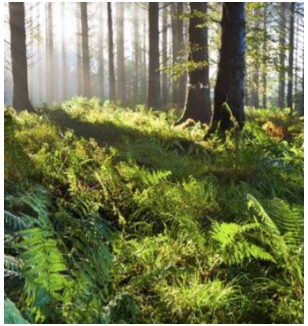
- Some physical phenomena are difficult to model



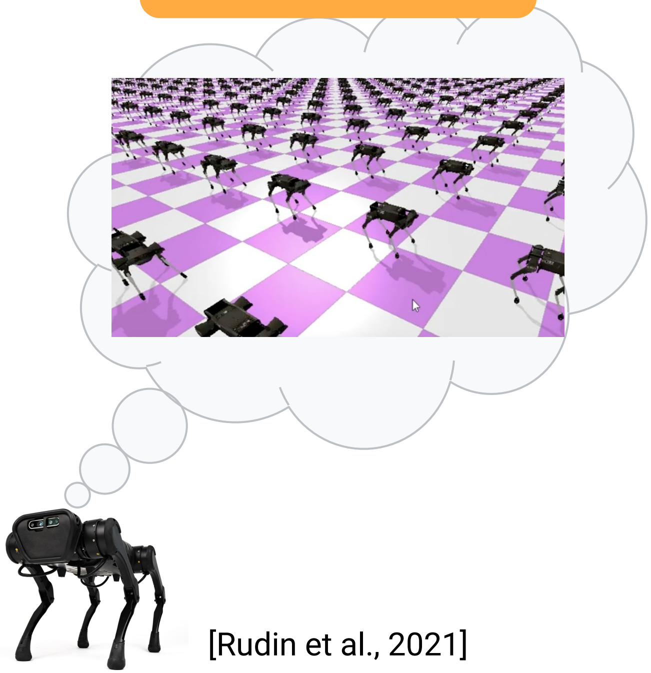
Discuss: Does sim-to-real solve everything?

- Some physical phenomena are difficult to model
- Impossible to capture the diversity of real-world scenarios



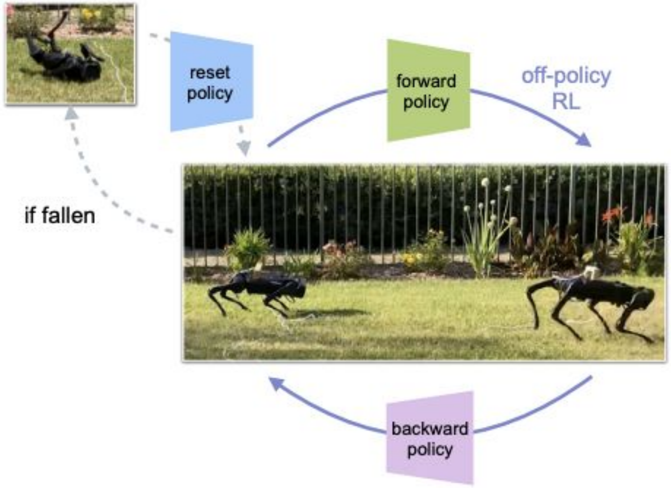

Sim-to-Real: A Complete Picture

Training in large-scale simulation → Rapid adaptation via online sysID → Safe and autonomous real-world fine-tuning



[Rudin et al., 2021]

[RMA, Kumar et al., 2021]



[Smith et al., 2022]

Questions?