# Online adaptation
## *is a reality gap problem*

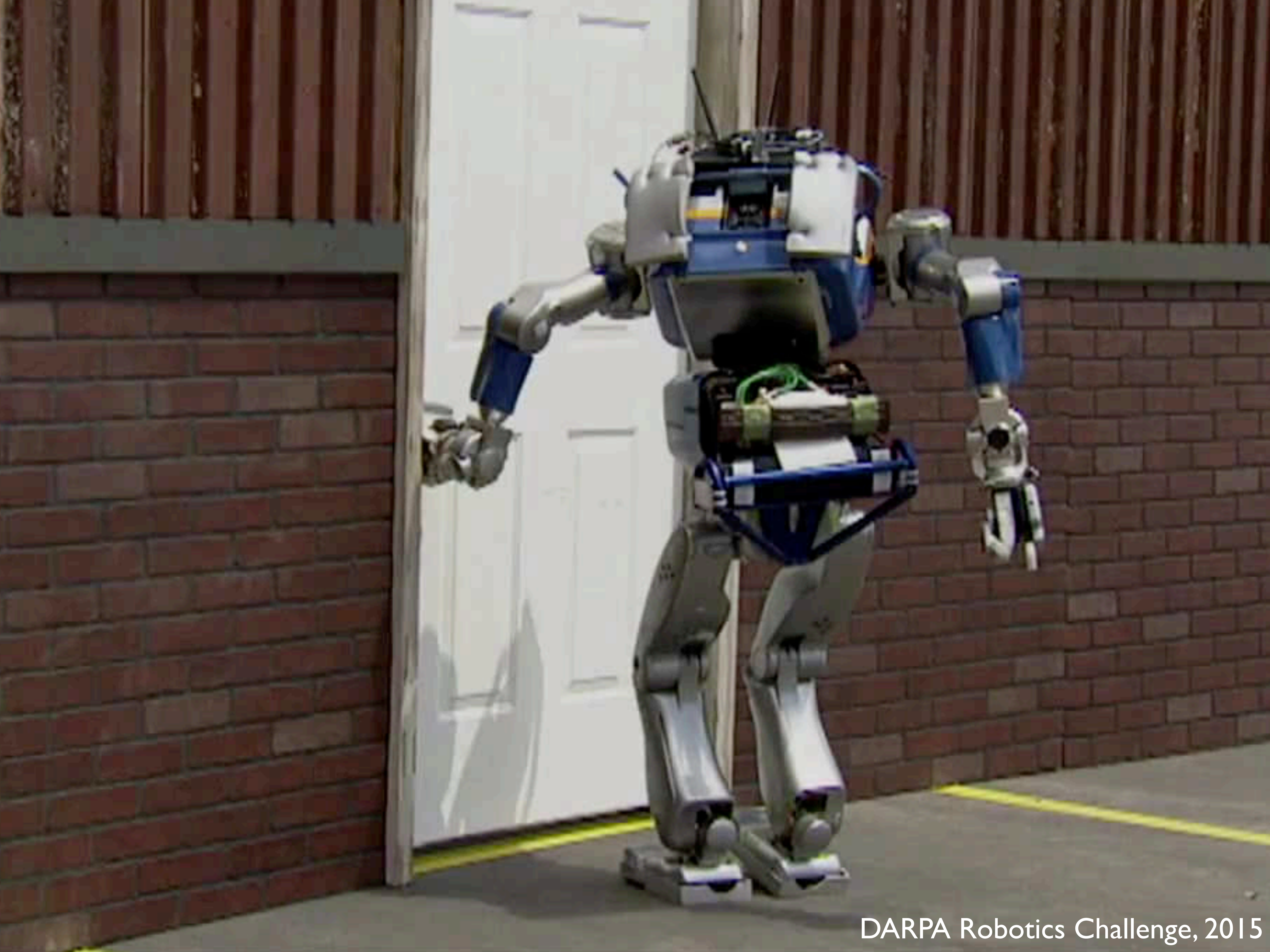
UNIVERSITÉ DE LORRAINE

cnrs

Inria informatics mathematics

erc
European Research Council
Established by the European Commission

**Jean-Baptiste Mouret (Inria)**



DARPA Robotics Challenge, 2015

**Online adaptation is a major "learning problem" of robotics**
**… more important that controller design / synthesis (?)**

The issue with current robots is not that they fail…
**… it is that they do not get back on their feet and try again**

> "Our Atlas Robot had mean time between failures of hours or, at most, days […]
> Behaviors that worked perfectly and robustly in the team's labs did not work or were erratic when tested on an "identical" setup at a DARPA test site.
> — **C. Atkeson *et al.* (2018).** What Happened at the DARPA Robotics Challenge Finals. In *The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue*. Springer."

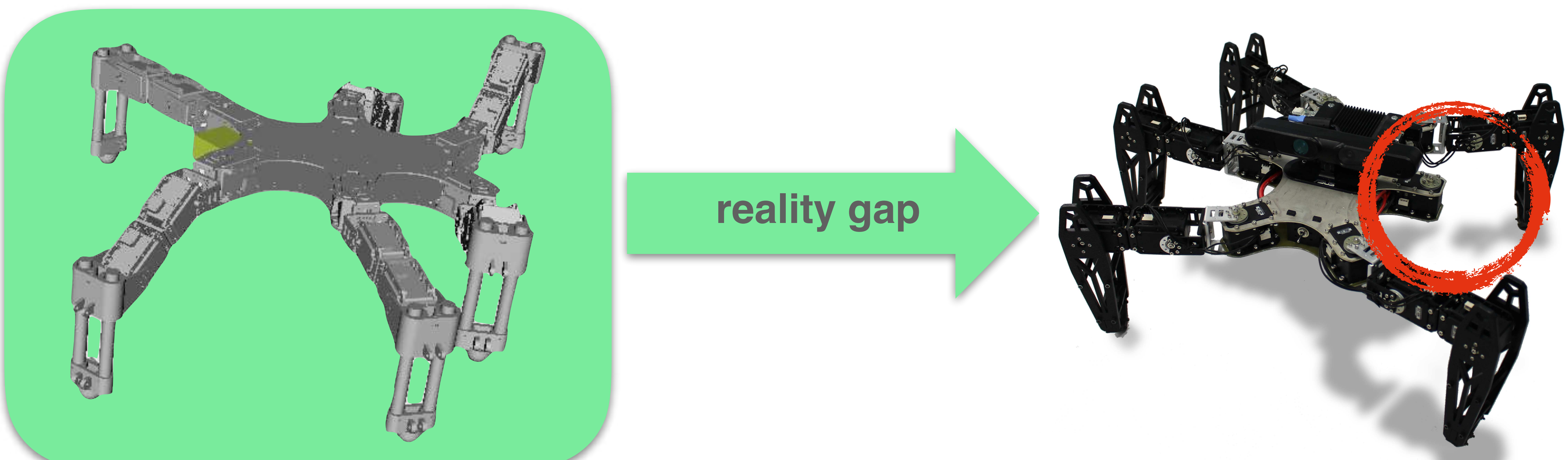
# Reality gap & online adaptation / damage recovery

- We <u>know</u> models of the intact robot in a nominal environment (simulator)
- We <u>do not know</u> the change / damage
- We need to adapt quickly to a change (*e.g.*, a damage)

… while minimising the number of episodes on the robot

➠ **If we can "solve" the reality gap problem, we might solve the adaptation problem**

➠ **we need to cross the reality gap, be data-efficient, and fast**




Simulator of the intact robot

Real (damaged) robot

## Crossing the reality gap

### Learn on the real robot (data-efficiently)

- Demonstrations (prior on policy parameters)
- Bayesian optimisation

- This is never the full environment
- Only for a few parameters

### Improve *robustness* of policies

- Domain randomization
- Meta-learning

Distribution of possible differences between simulation and reality?

### Improve simulators (*models*)

- Model identification
- Model-based policy search

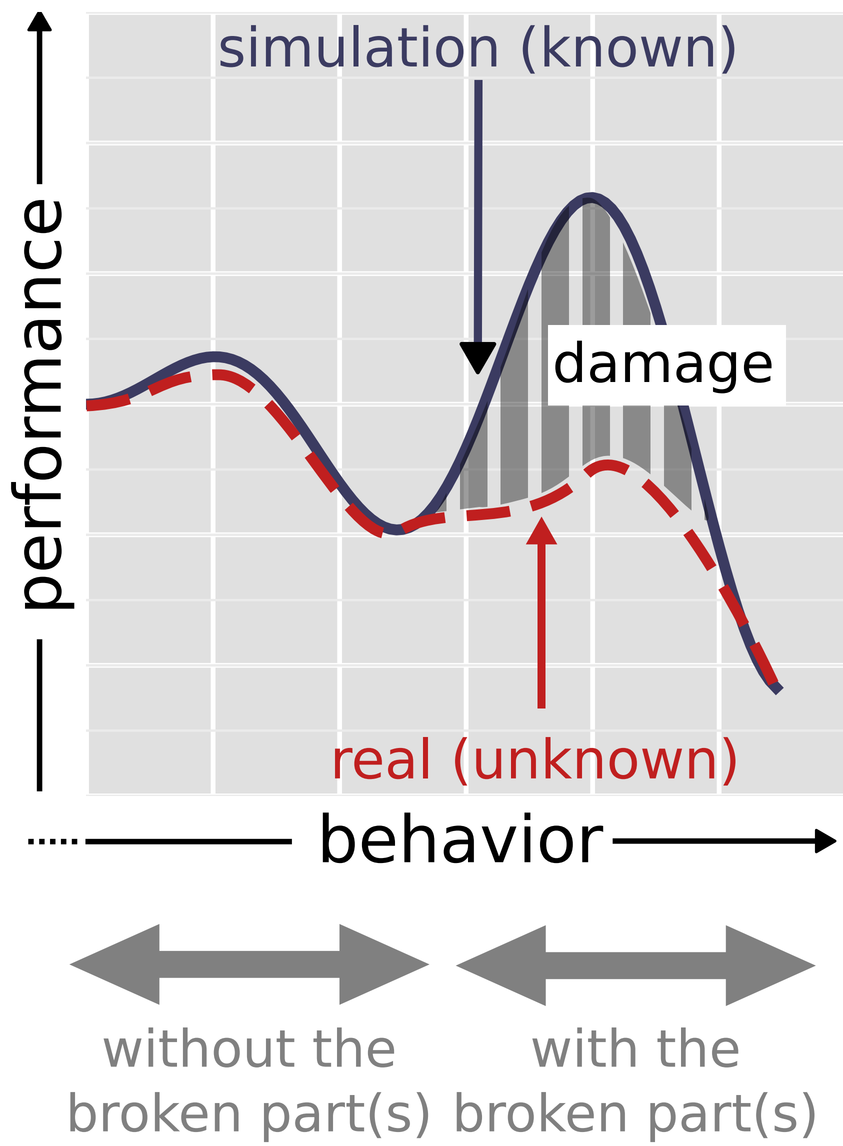
- When is it easier to learn a simulator than a policy?
- Need to wait between episodes

**Combinations are possible (and useful!)**

**Chatzilygeroudis, K., Vassiliades, V., Stulp, F., Calinon, S., & Mouret, J. B. (2018).** A survey on policy search algorithms for learning robot controllers in a handful of trials. *arXiv preprint arXiv:1807.02303*.

# Crossing the reality gap

| Learn on the real robot | Improve *robustness* of policies | Improve simulators (*models*) | **Learn the transferability function** |
|---|---|---|---|




**Transferability hypothesis:** *some* controllers will work similarly in simulation and in reality

**Transferability approach**

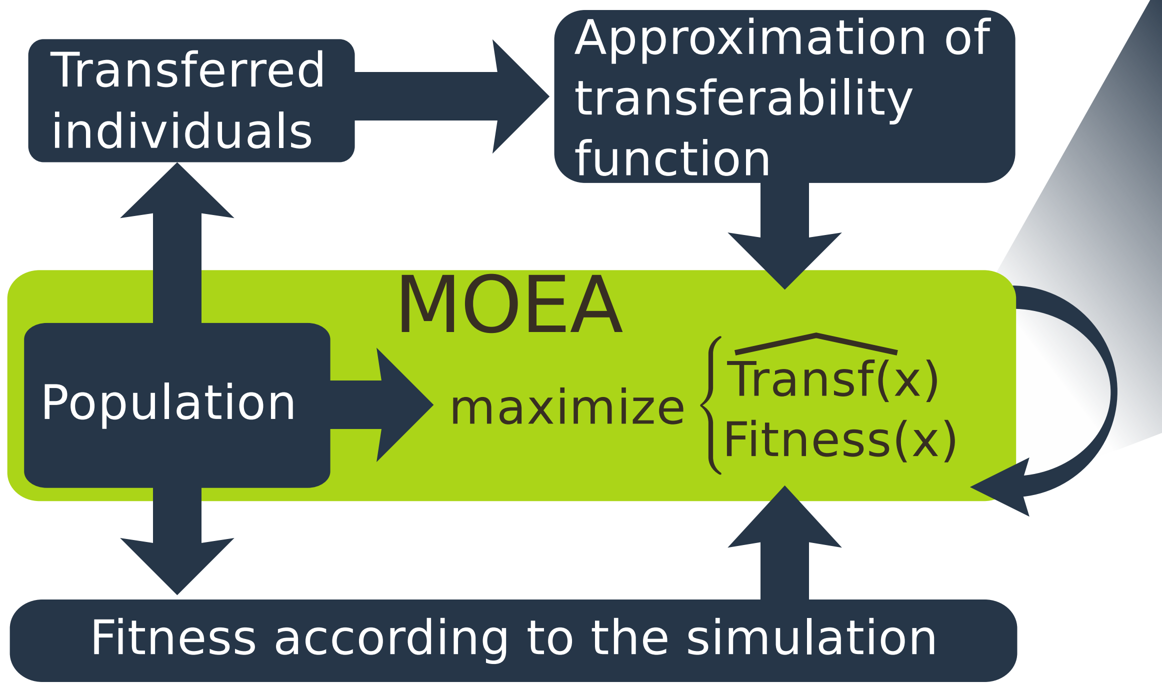
- learn a model that predicts the transferability score
  ~ learn the limits of the simulation
- search for a policy with a good reward and a good score

**Why?**

- transferability easier to learn that the dynamics
  ➠ know something is wrong vs knowing the correct answer
- learn a constraint (no crazy predictions)

**Koos, S., J.-B. Mouret, and S. Doncieux.** "The transferability approach: Crossing the reality gap in evolutionary robotics." IEEE Transactions on Evolutionary Computation 17.1 (2013): 122-145.

# The transferability approach




Maximize reward

## How can we adapt faster and with embedded hardware?

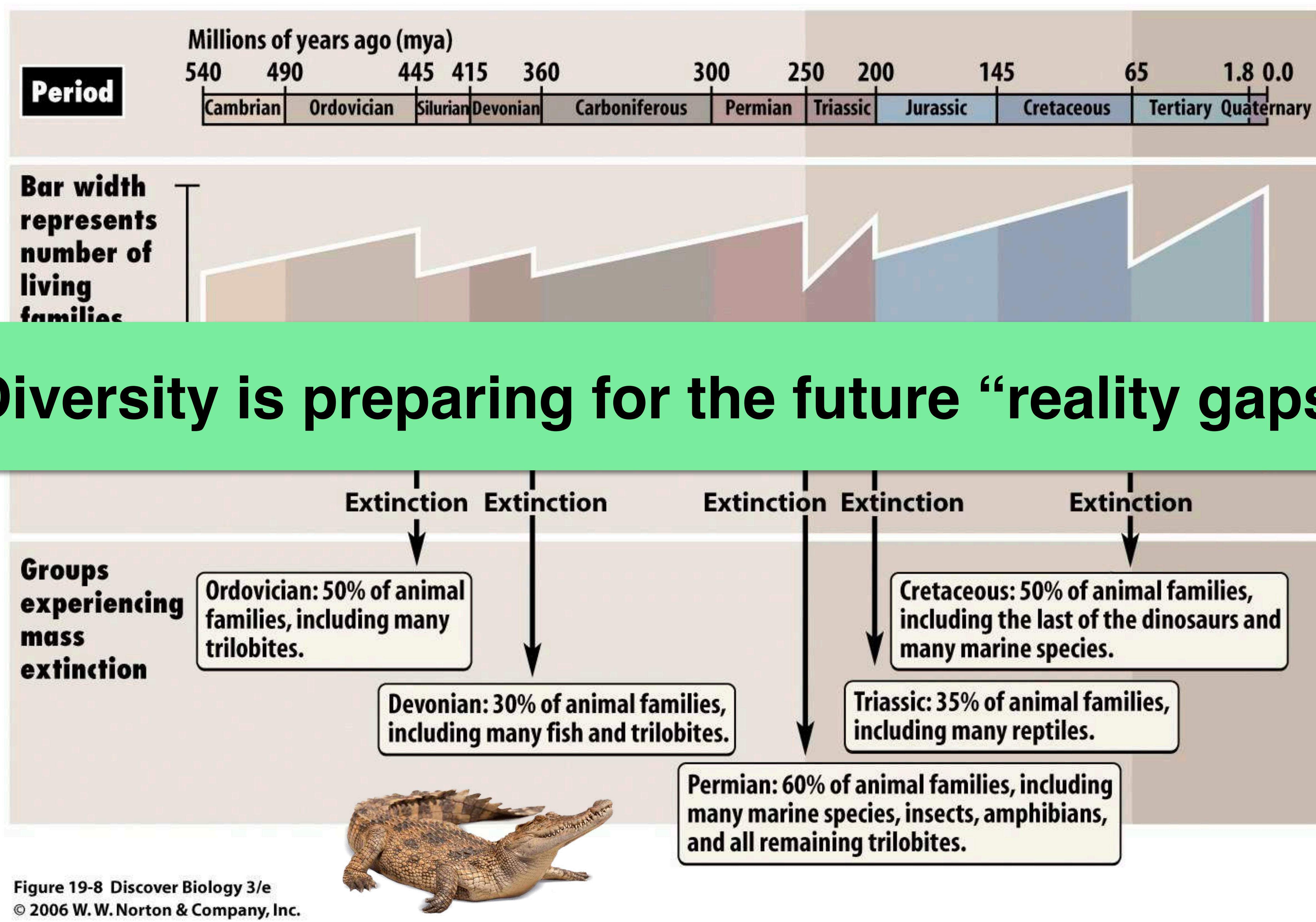

15 transfers (motion capture)

**… but a lot of computation between episodes (several minutes)**


**Koos, S., Cully, A., & Mouret, J. B. (2013).** Fast damage recovery in robotics with the T-Resilience algorithm. *The International Journal of Robotics Research*, *32*(14), 1700-1723.

**Koos, S., Mouret, J.-B., & Doncieux, S. (2011).** The Transferability Approach : Crossing the Reality Gap in Evolutionary Robotics. *IEEE Transaction on Evolutionary Computation.*

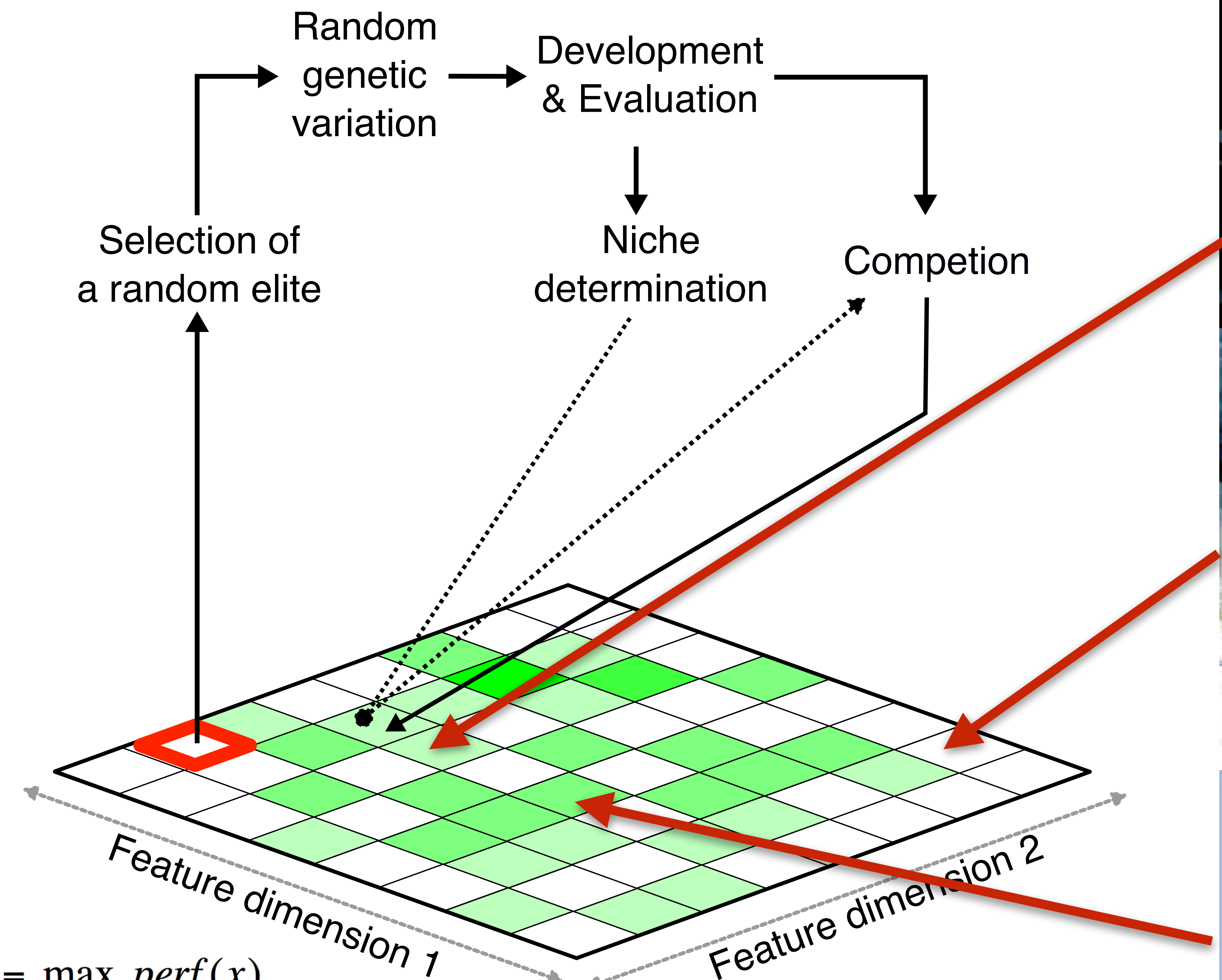
# Adaptation to sudden changes in nature




**Diversity is preparing for the future "reality gaps"**

# Generating species: The MAP-Elites algorithm

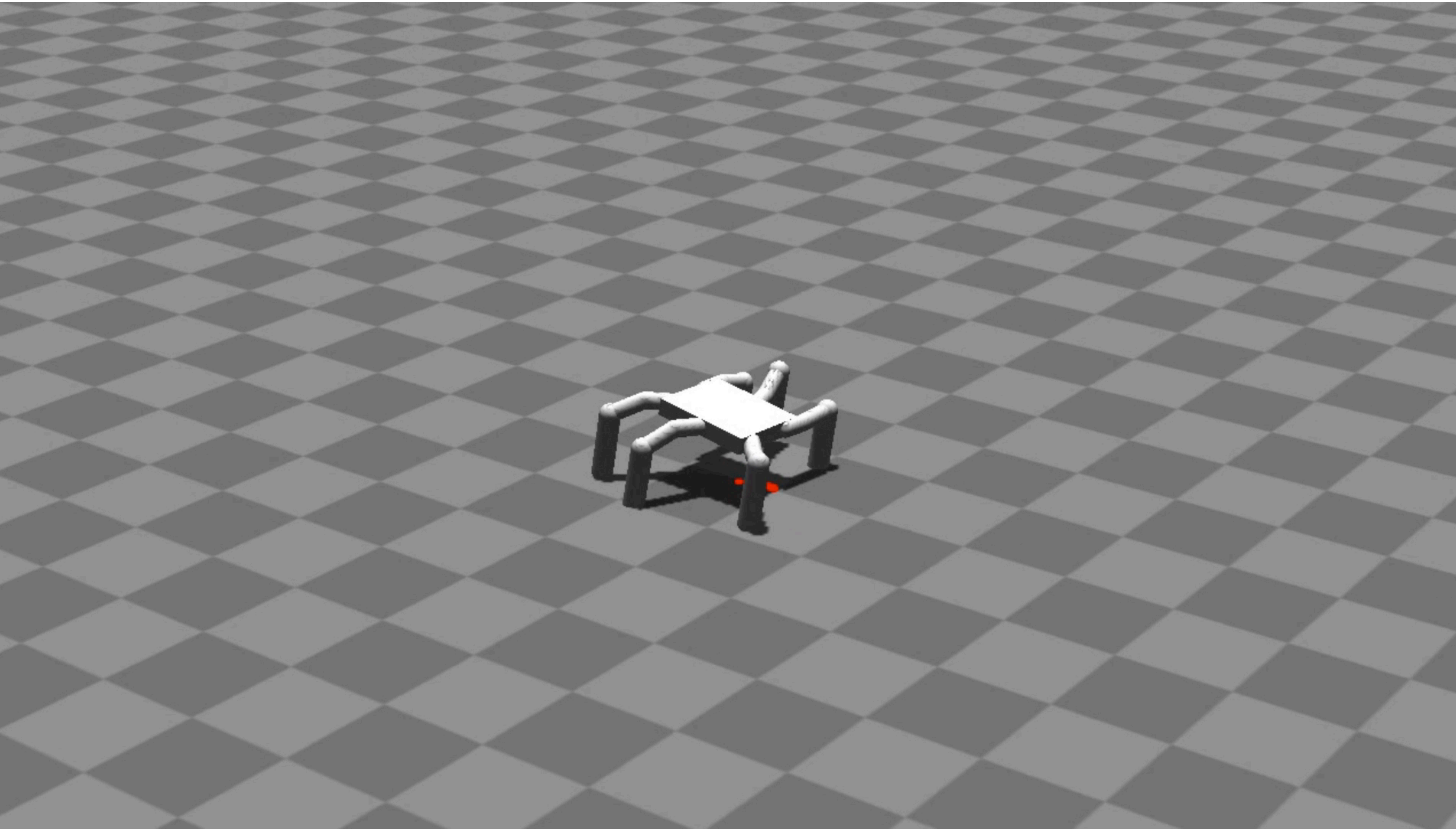
Multi-dimensional Archive of Phenotypic Elites: Quality Diversity / illumination algorithm



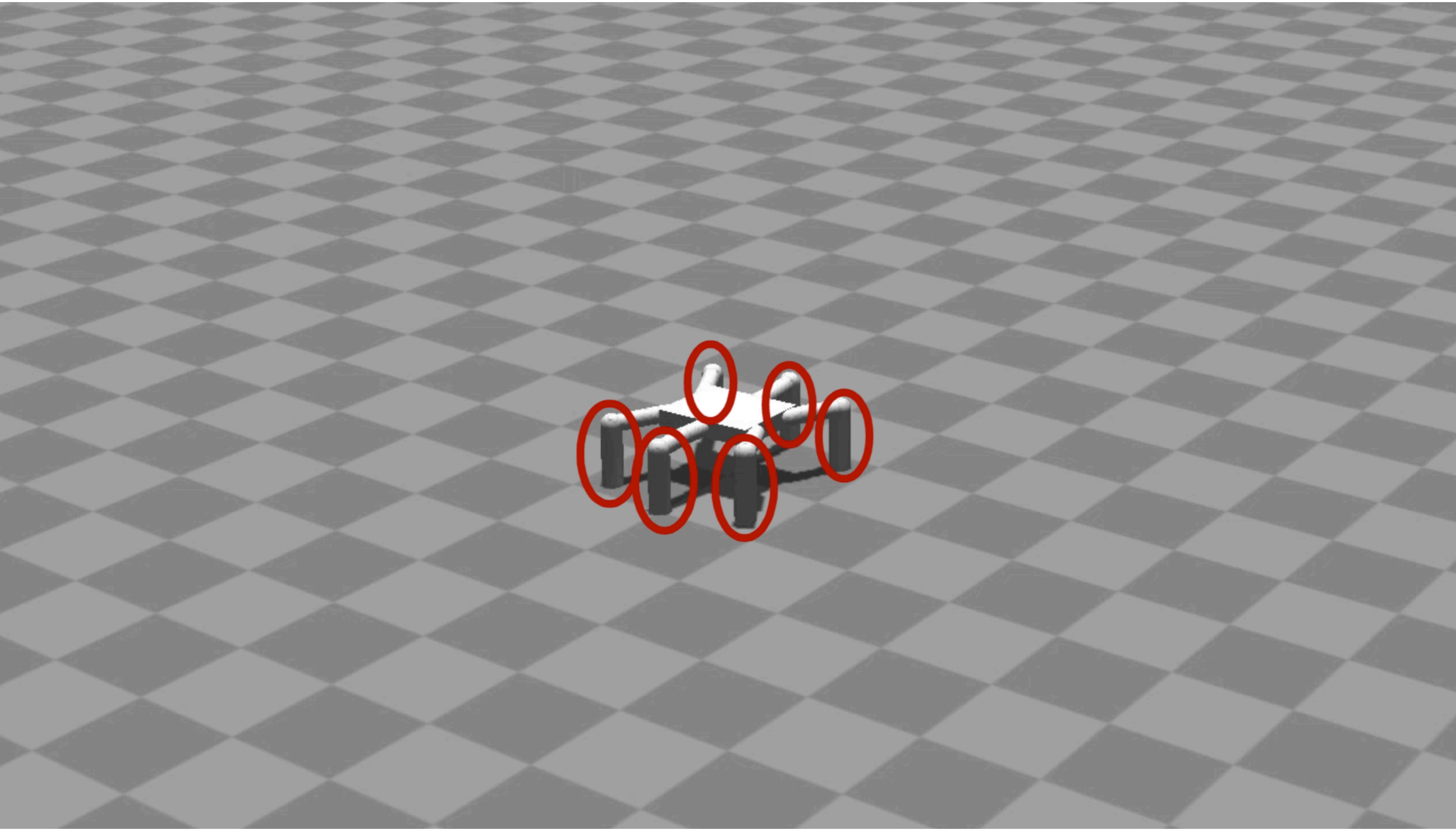

$$M_i = \max_{x \in Bi(S)} perf(x)$$

**Mouret, J.-B., and J. Clune. (2015)** "Illuminating search spaces by mapping elites." arXiv preprint arXiv:1504.04909

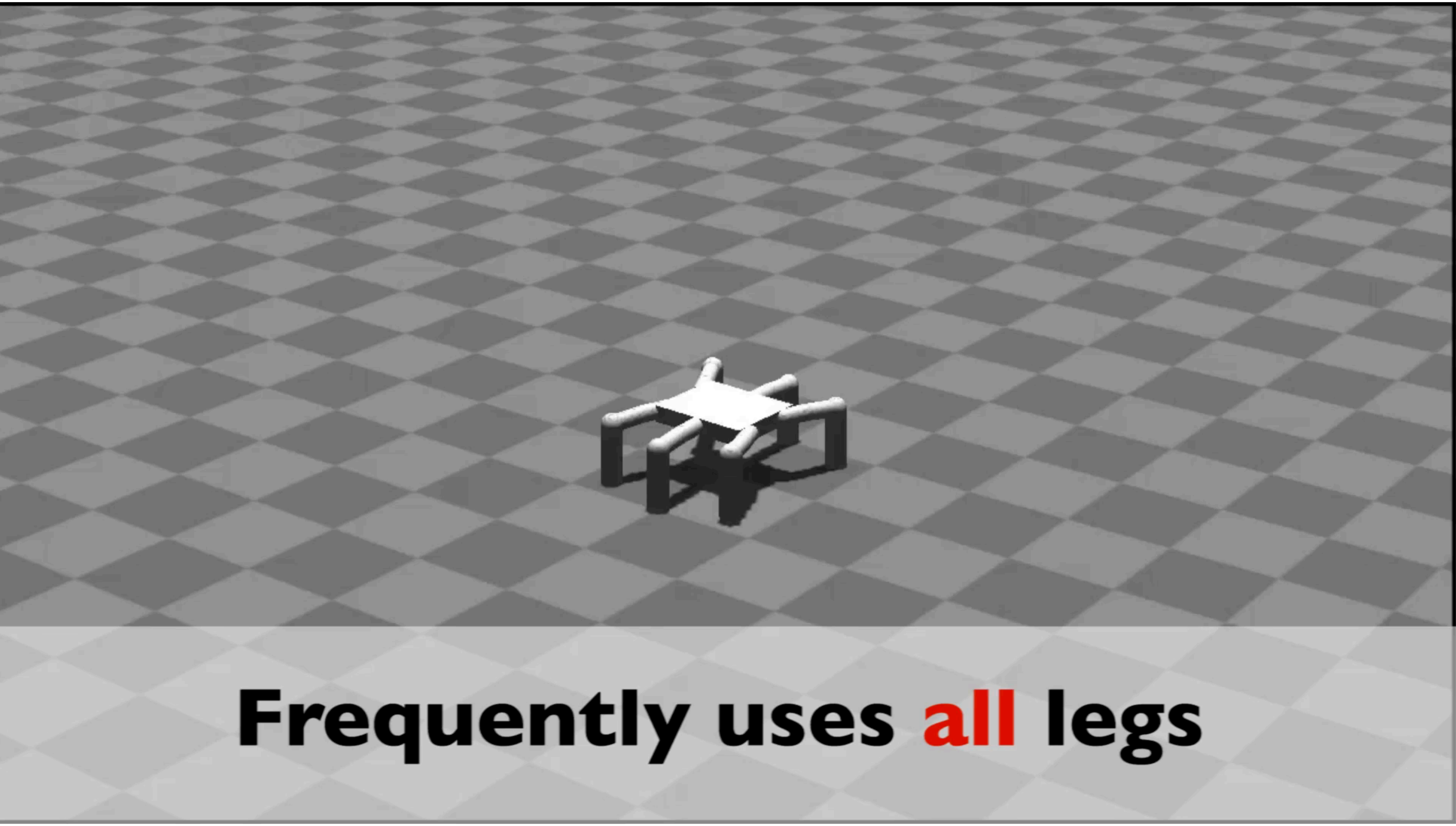
# **MAP-Elites:** 6-legged locomotion



# **MAP-Elites:** 6-legged locomotion



# **MAP-Elites:** 6-legged locomotion



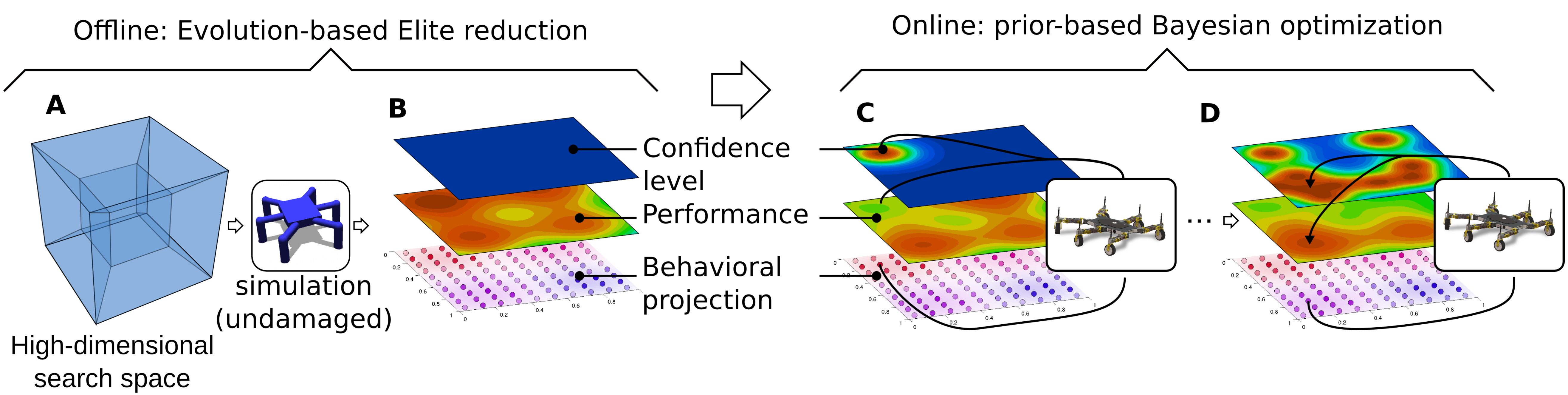
# Intelligent Trial & Error: Map-Elites + BO

The **MAP-Elites algorithm** generates the search space (prior)

- in simulation, with an intact robot
- many evaluations [simulation]
- "take the needles out of the haystack"
- provide an expected performance for the "needles"

**Prior-based Bayesian optimization** does the online learning

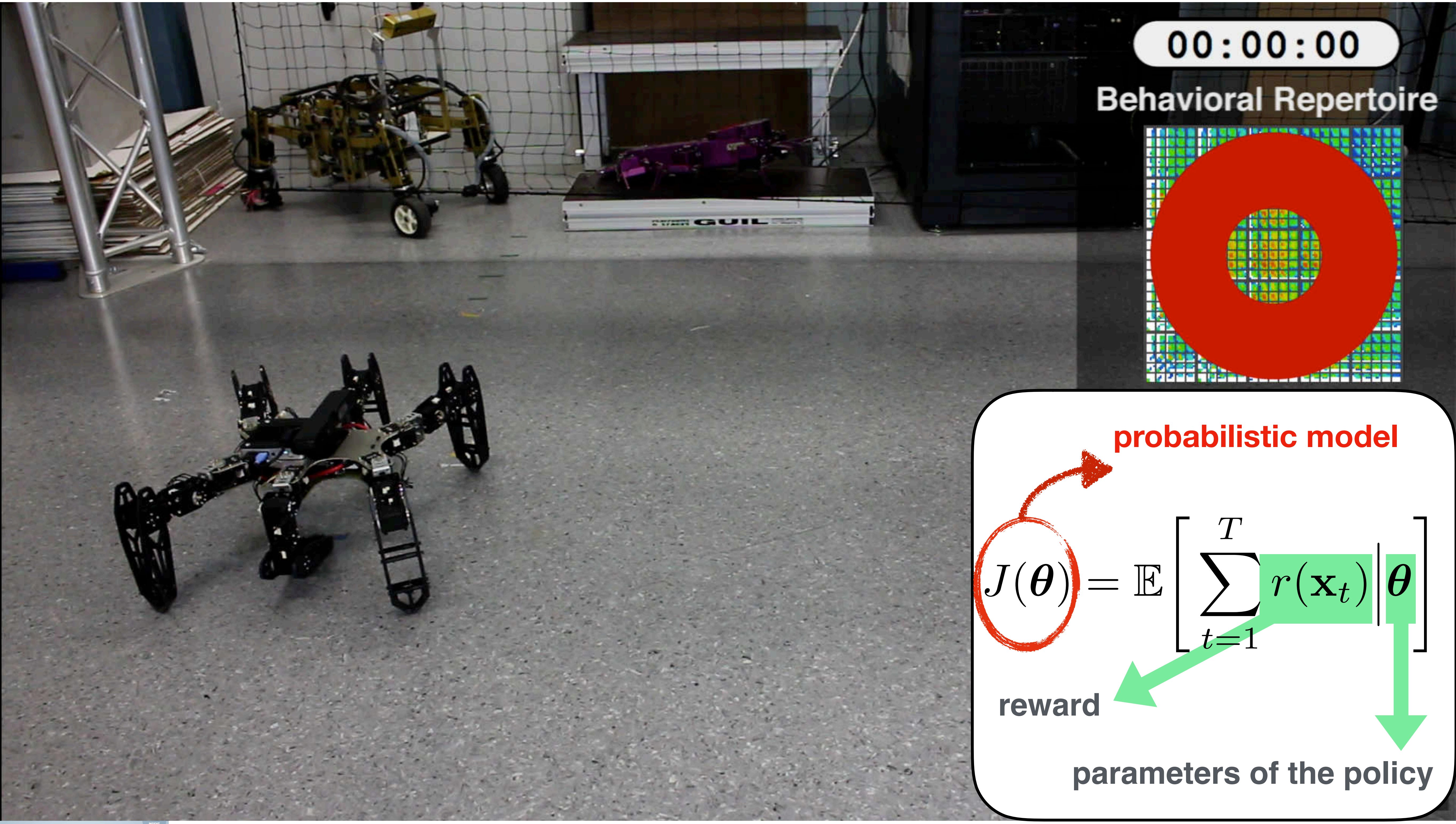
- search only among good solutions ("needles")
- trial-and-error
- few evaluations [real robot]




Diversity

Adaptation

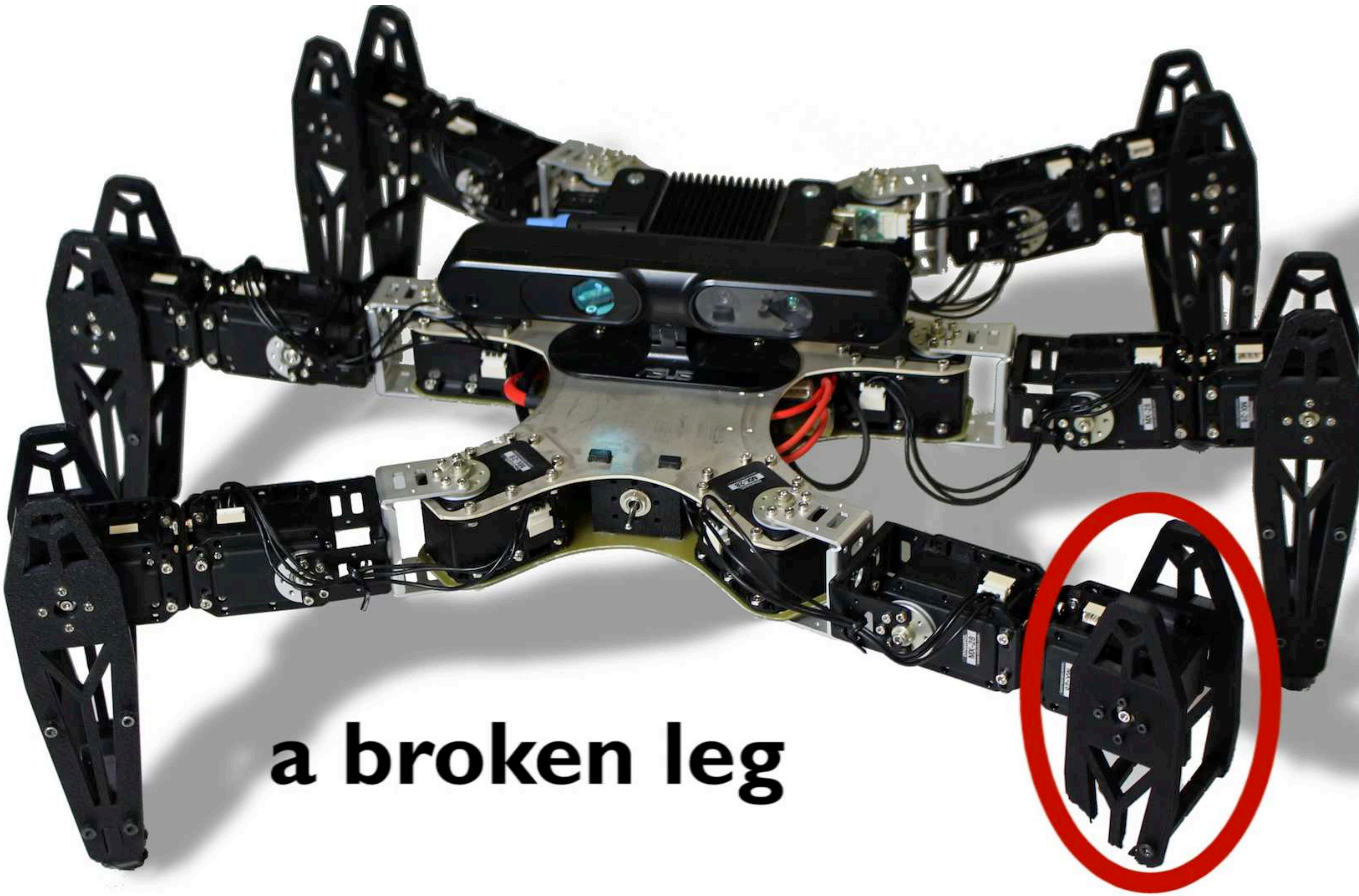
# Adaptation: Bayesian optimization

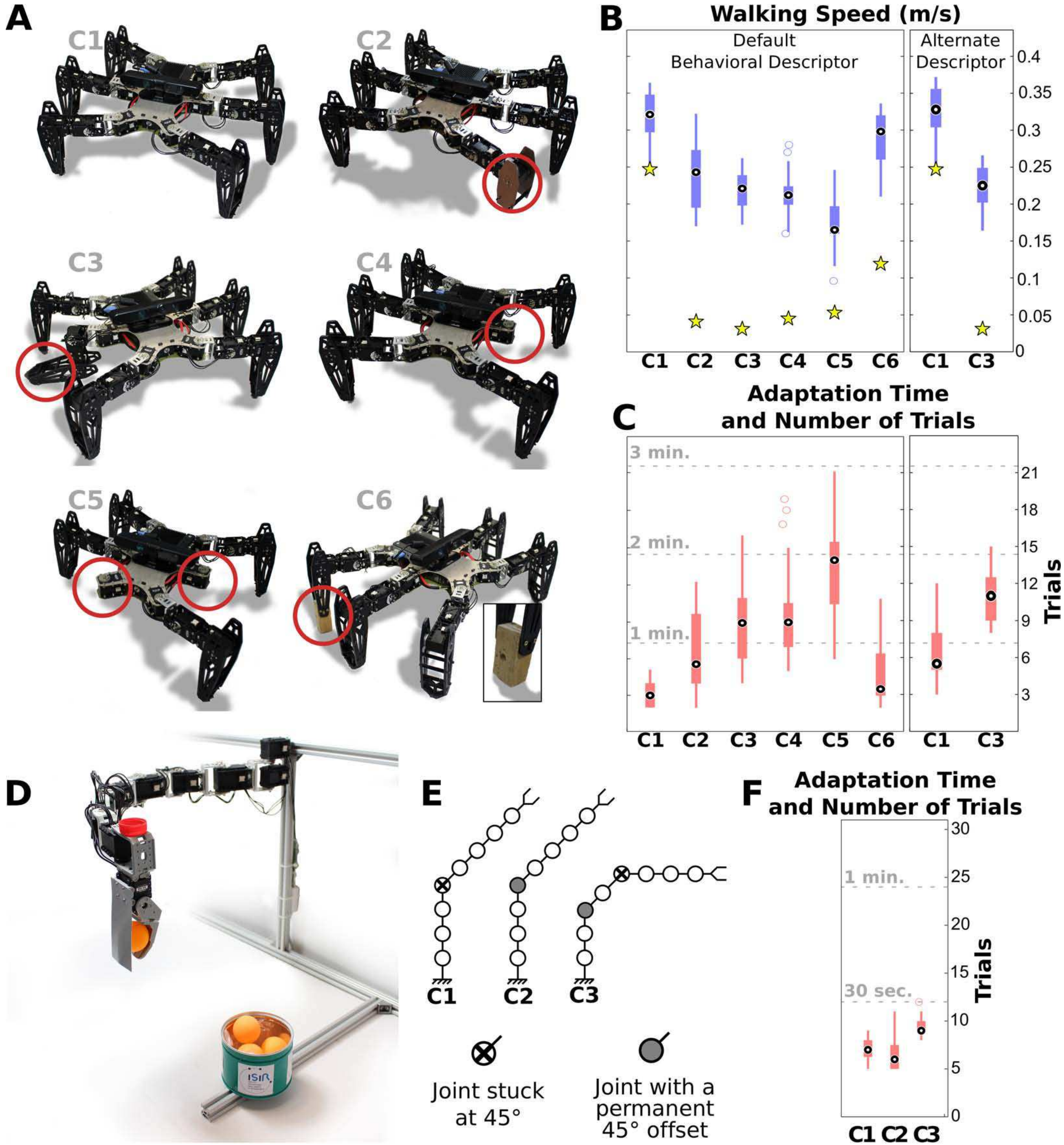


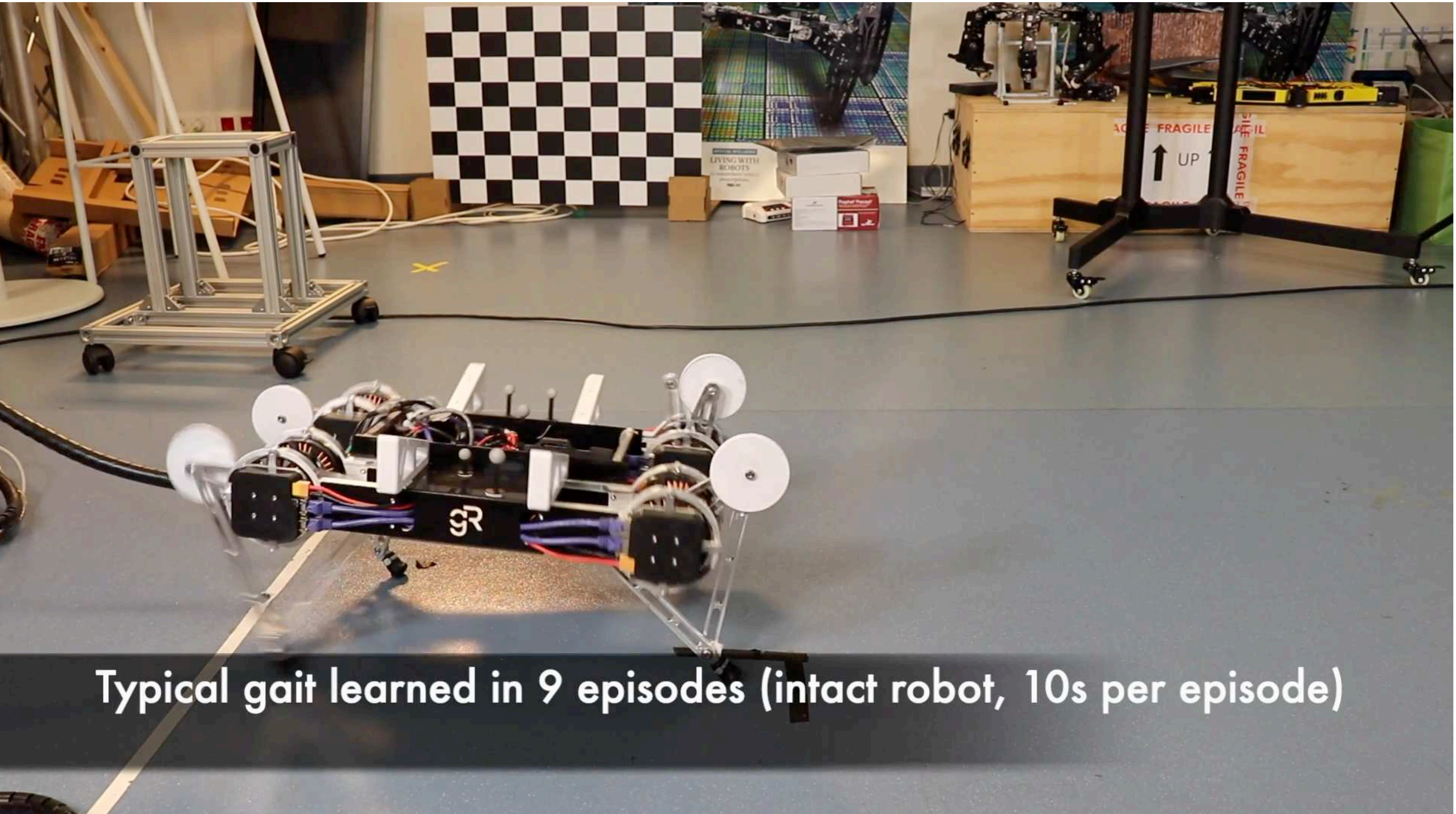


- policy : periodical signals (36 parameters)
- <u>No information about the damage</u>

**Cully, A. and Clune, J. and Tarapore, D. and Mouret, J.-B. (2015).** *Robots that can adapt like animals*. Nature. Vol 521 Pages 503-507.


a broken leg


A
C1
C2
C3
C4
C5
C6
B
Walking Speed (m/s)
Default
Behavioral Descriptor
Alternate
Descriptor
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
C1 C2 C3 C4 C5 C6 C1 C3
C
Adaptation Time
and Number of Trials
3 min.
2 min.
1 min.
21
18
15
12
9
6
3
Trials
C1 C2 C3 C4 C5 C6 C1 C3
D
E
C1
C2
C3
Joint stuck
at 45°
Joint with a
permanent
45° offset
F
Adaptation Time
and Number of Trials
30
25
20
15
10
5
0
1 min.
30 sec.
Trials
C1 C2 C3





- Policy : periodical signals (24 parameters)
- <u>No information about the damage</u>

**Dalin, P. Desreumaux, J.-B. Mouret. (2019)** Learning and adapting quadruped gaits with the “Intelligent Trial & Error” algorithm. ICRA Workshop on “Learning Legged Locomotion”.

# Recent extension: Multiple priors

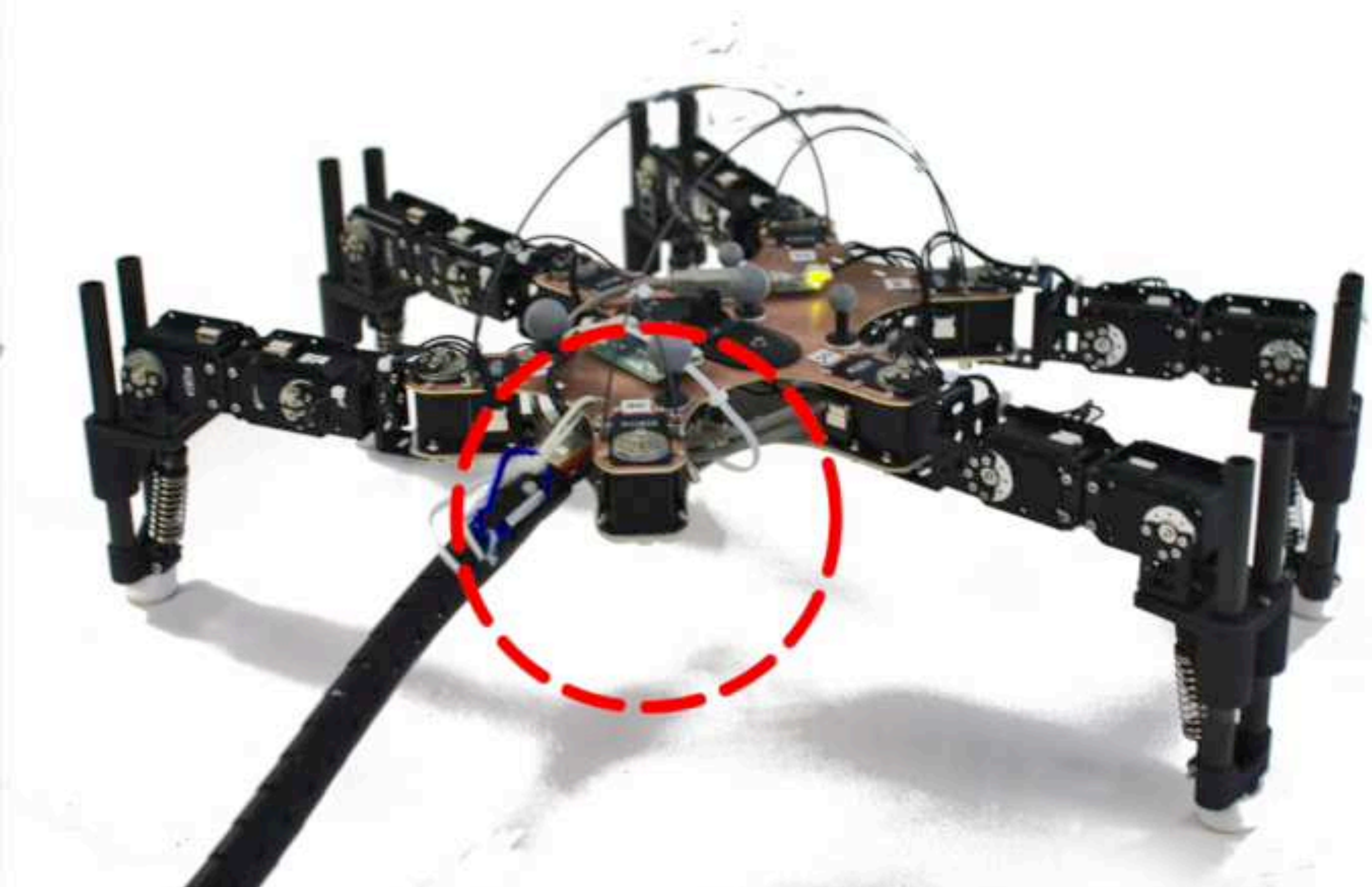
## Experiment 1
damage recovery

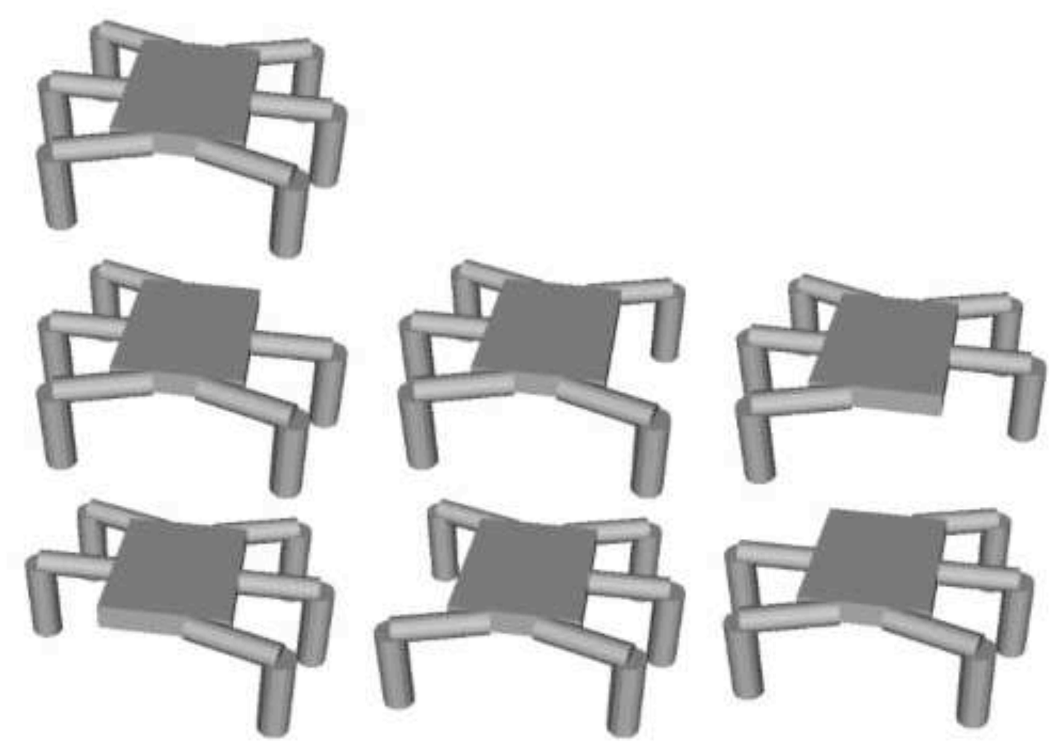
Unknown damage condition

105 priors (15 for each condition)

Reward: walking distance

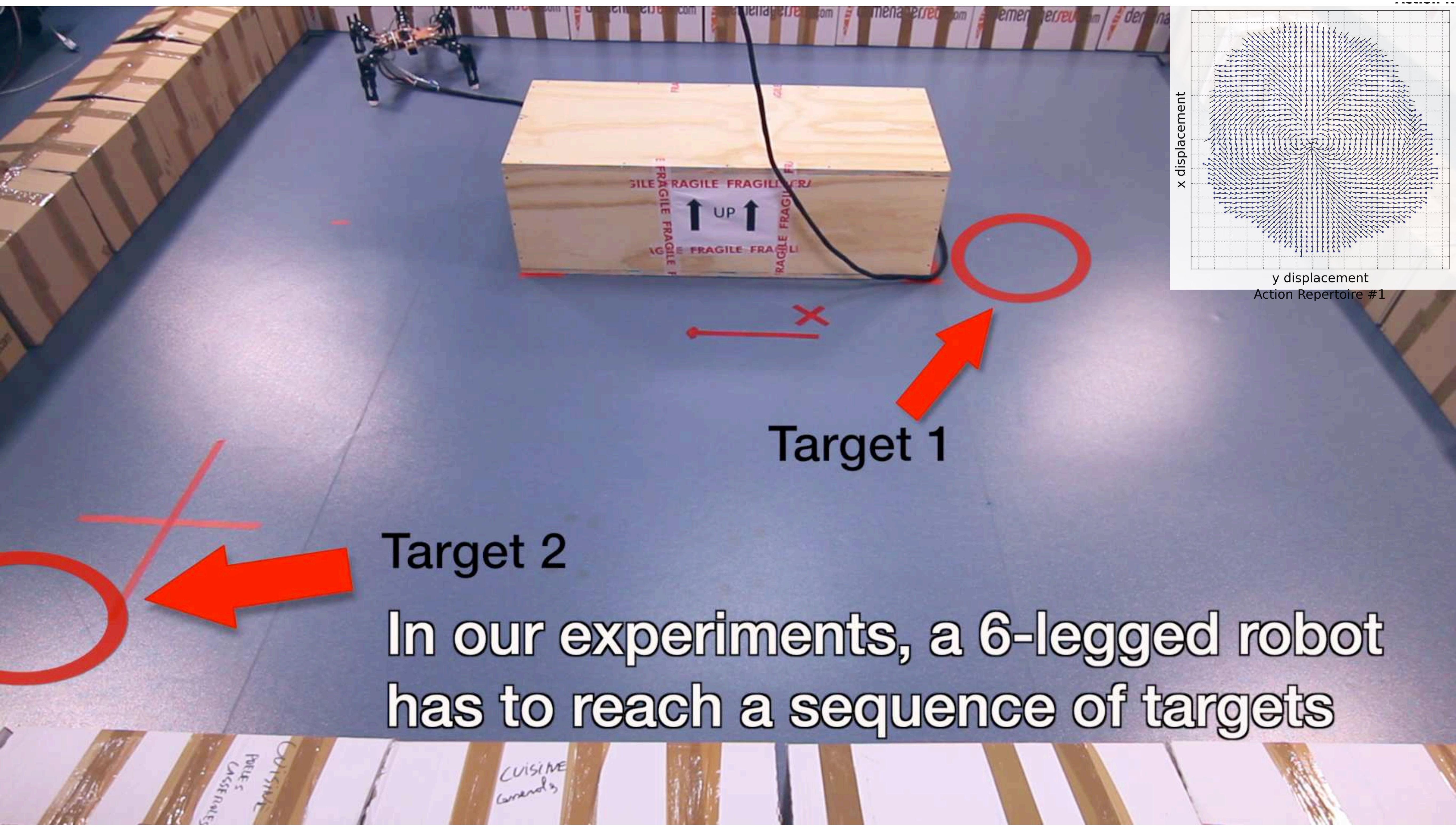
BO with MLEI acquisition function





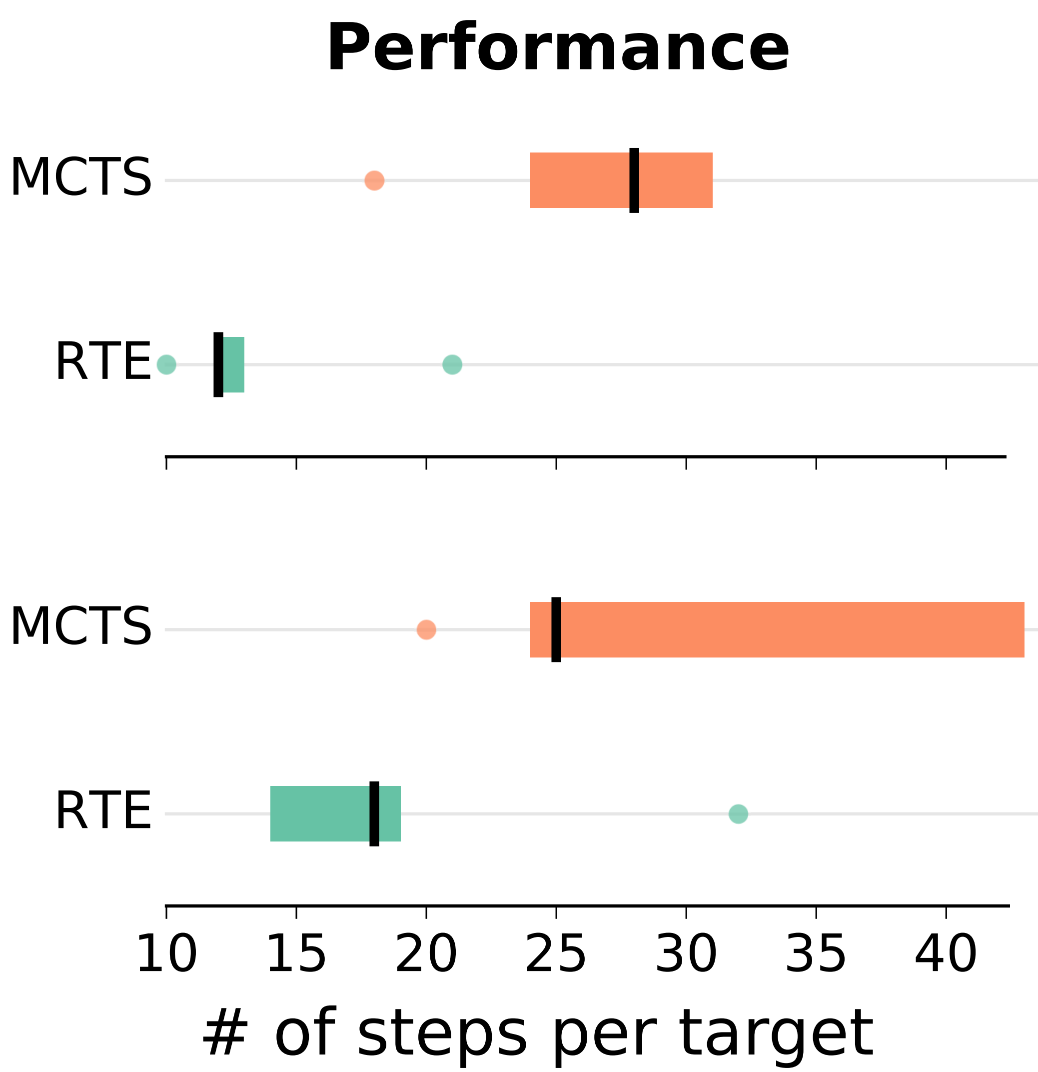
**Pautrat, R., Chatzilygeroudis, K., & Mouret, J.-B. (2018).** Bayesian Optimization with Automatic Prior Selection for Data-Efficient Direct Policy Search. Proc. of IEEE ICRA.

# Planning (MCTS) + repertoire learning (priors)



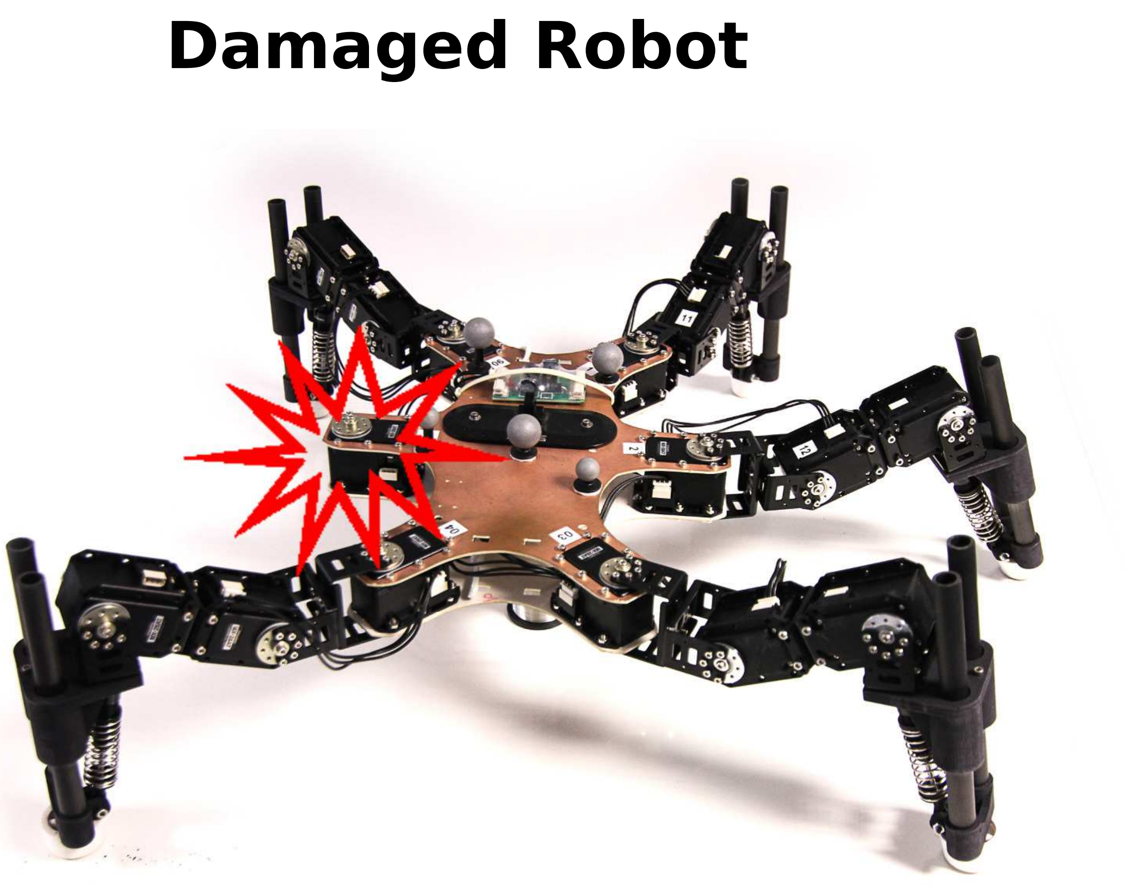

**K. Chatzilygeroudis, V. Vassiliades, and J.-B. Mouret (2018).** *Reset-free Trial-and-Error Learning for Data-Efficient Robot Damage Recovery.* Robotics and Autonomous Systems.

# Real robot / 5 replicates




Environment #1

Environment #2




1 step = 3 seconds

# Bridging the gap with model-based Policy Search

- Transferability function = learning the limits of an existing simulator
- … not far from a <u>probabilistic</u> model + prior

Gaussian processes

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \boxed{M(\mathbf{x}_t, \mathbf{u}_t, \boxed{\phi_M})} + \boxed{f(\mathbf{x}_t, \mathbf{u}_t, \boxed{\phi_K})} + \mathbf{w}$$

Simulator / model    parameters

Learning = maximize the <u>likelihood</u> of M+f

➟ **effects that can be captured** by the simulator will be included by tuning the simulator (model identification)

➟ **effects that cannot be captured** by changing the parameters are modelled by the Gaussian processes

**Chatzilygeroudis K, Rama R, Kaushik R, Goepp D, Vassiliades V, Mouret JB.** (2017) Black-Box Data-efficient Policy Search for Robotics. Proc. of IEEE IROS.

# The Black-DROPS algorithm

Model-based policy search

## Black-box Data-Efficient Robot Policy Search

**1.** Perform a few random trials

➠ new data

**2.** Learn a probabilistic model of the robot with Gaussian processes:

simulator ← → Gaussian processes

$$\mathbf{x}_{t+1} = \mathbf{x}_t + M(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\phi}_M) + f(\mathbf{x}_t, \mathbf{u}_t, \boldsymbol{\phi}_K) + \mathbf{w}$$

**3.** Optimize with CMA-ES a policy that maximizes the long-term reward according to the model

➠ one function evaluation = one rollout (propagate by sampling)

➠ treat each rollout as a measurement of a noisy function

**4.** Evaluate the policy on the robot

➠ new data

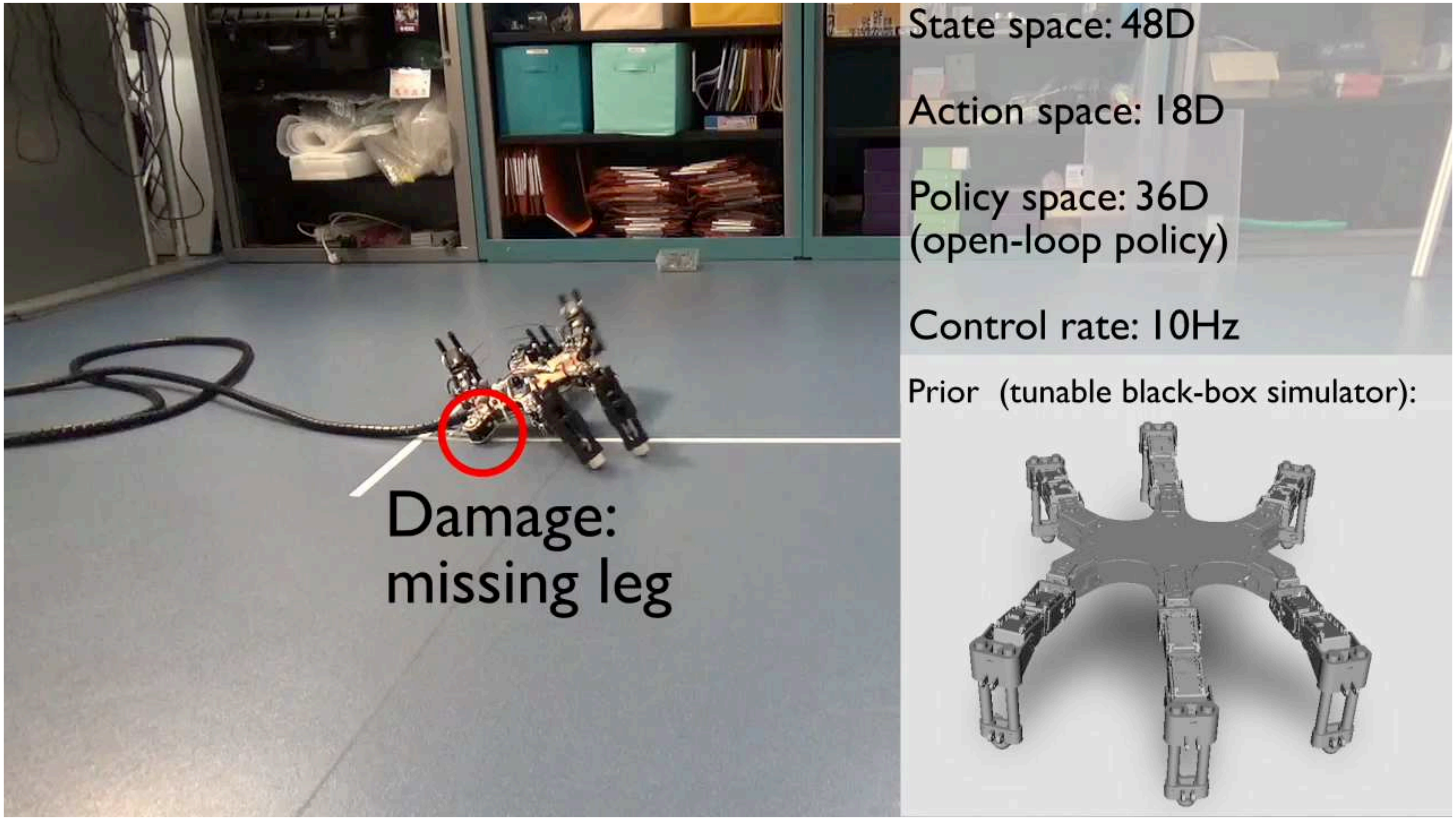
Most expensive step:

- benefit a lot from parallelization
- handle uncertainty as noise ➠ domain randomization

**Chatzilygeroudis K, Rama R, Kaushik R, Goepp D, Vassiliades V, Mouret JB.** (2017) Black-Box Data-efficient Policy Search for Robotics. Proc. of IEEE IROS.

# Black-DROPS + priors + identification

Model-based policy search




$$\mathbf{x}_{t+1} = \mathbf{x}_t + M(\mathbf{x}_t, \mathbf{u}_t, \phi_M) + f(\mathbf{x}_t, \mathbf{u}_t, \phi_K) + \mathbf{w}$$

**probabilistic model**

next state

Simulator / model

Other dynamics

**policy**

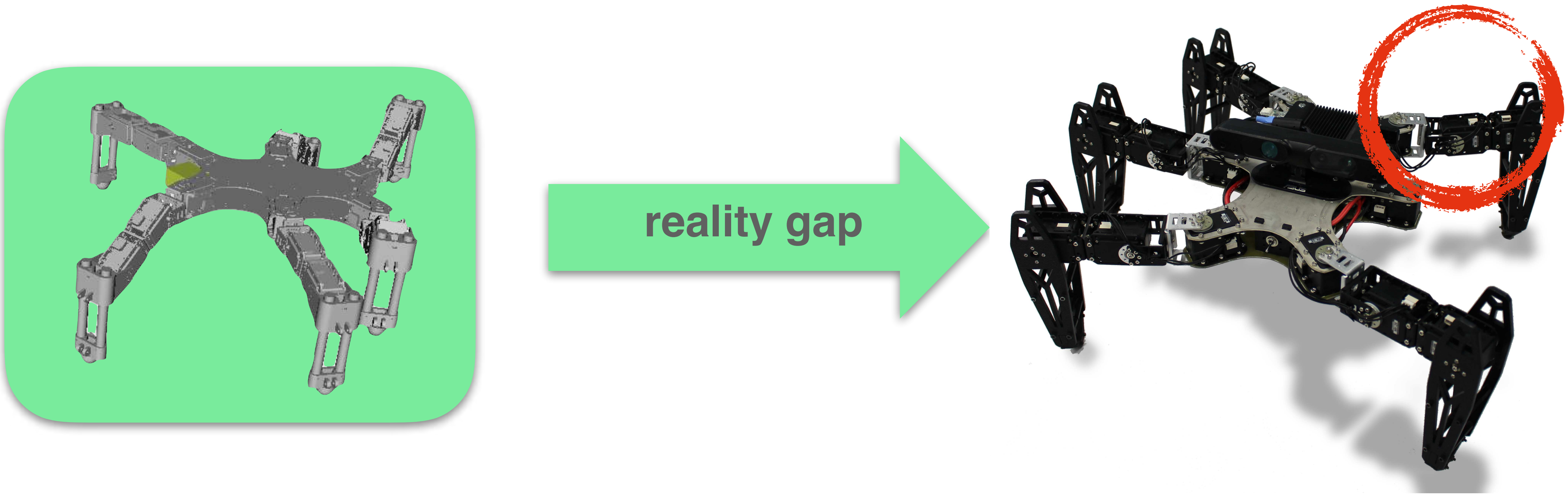
**Chatzilygeroudis K, Mouret JB. (2018)** Using Parameterized Black-Box Priors to Scale Up Model-Based Policy Search for Robotics. Proc. of ICRA.

# Conclusions

- **Adapting to damage is a reality gap problem**

… and adaptation is critical for robotics

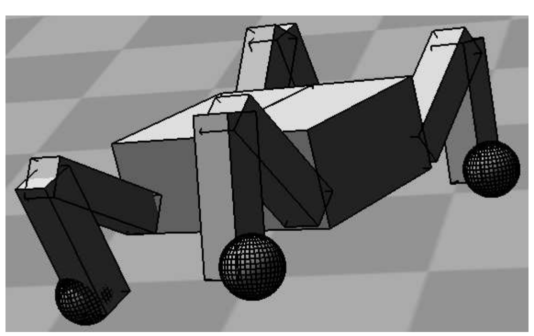
… we know how to design controllers, but not how to adapt

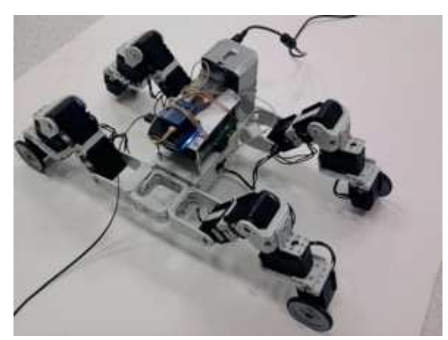


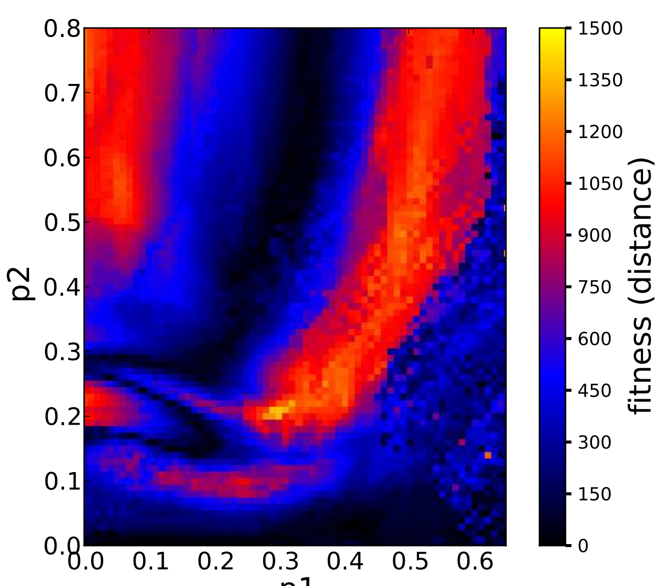

- <u>Diversity + Adaptation (IT&E)</u>: fast algorithm on the robot, but limited by the simulator
- <u>Model-based Policy Search with prior and uncertainty</u> (Black-DROPS): slow algorithm but could learn "anything"

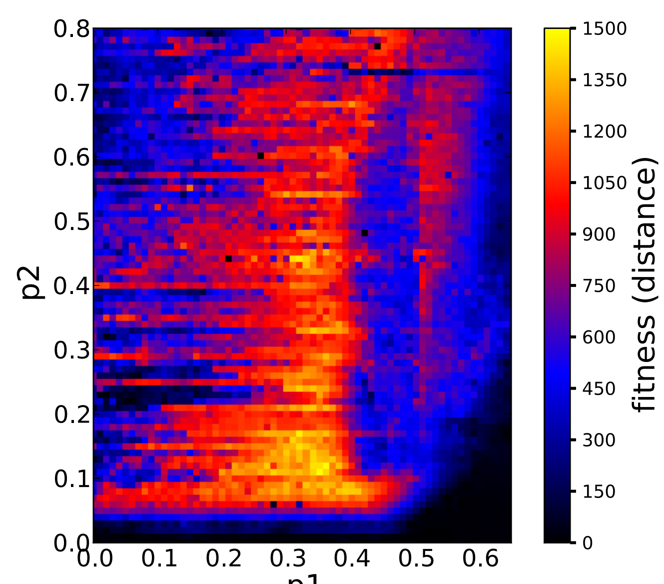
# Conclusions (2)

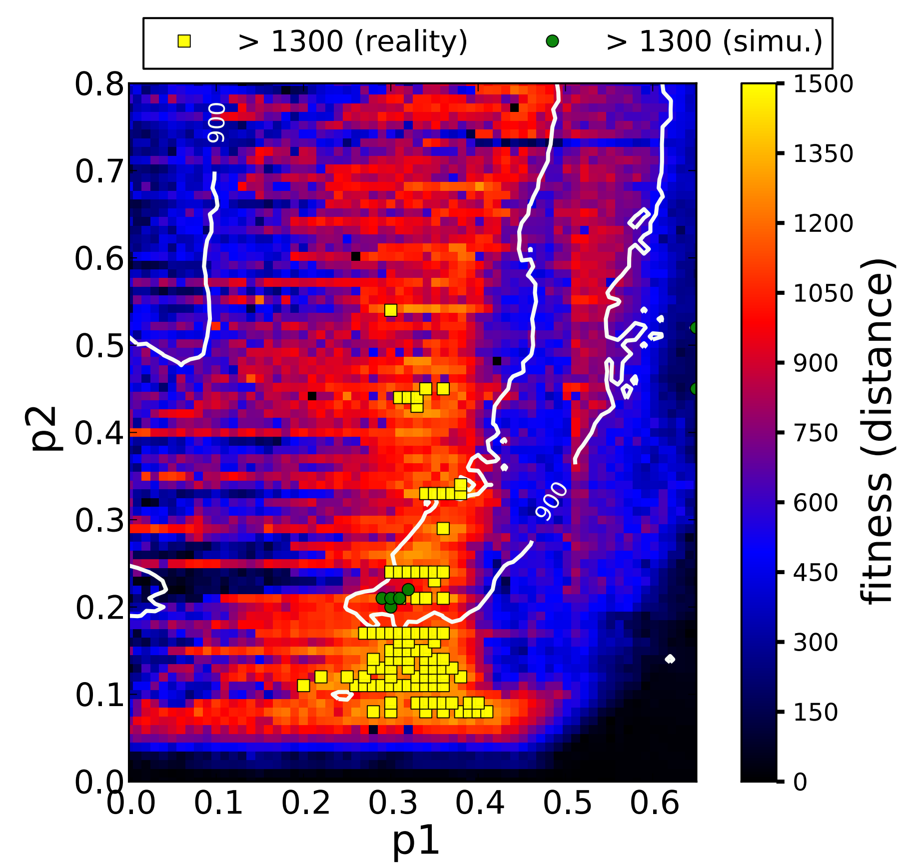
- **Simulators are often right (for rigid bodies)!**
- **Simulators are good *priors***
- **Every simulator (model) prediction should come with a measure of transferability or uncertainty**
  - this can be learned from data
  - **crowd-source a model for bullet/dart/ode?**
- We should map the reality gap








**Mouret, J. B., Koos, S., & Doncieux, S. (2013).** Crossing the reality gap: a short introduction to the transferability approach. arXiv preprint arXiv: 1307.1870.

## Team

Vassilis Vassiliades
K. Chatzilygeroudis
Dorian Goepp
Adam Gaier
Brice Clément
Rituraj Kaushik
Jonathan Spitz
Eloïse Dalin
Pierre Desreumaux
Vladislav Tempez
Glenn Maguire
Remi Pautrat

## Collaborators

Serena Ivaldi

jean-baptiste.mouret@inria.fr

https://members.loria.fr/jbmouret

Inria

cnrs advancing the frontiers

UNIVERSITÉ DE LORRAINE

Loria Laboratoire lorrain de recherche en informatique et ses applications

erc European Research Council Established by the European Commission

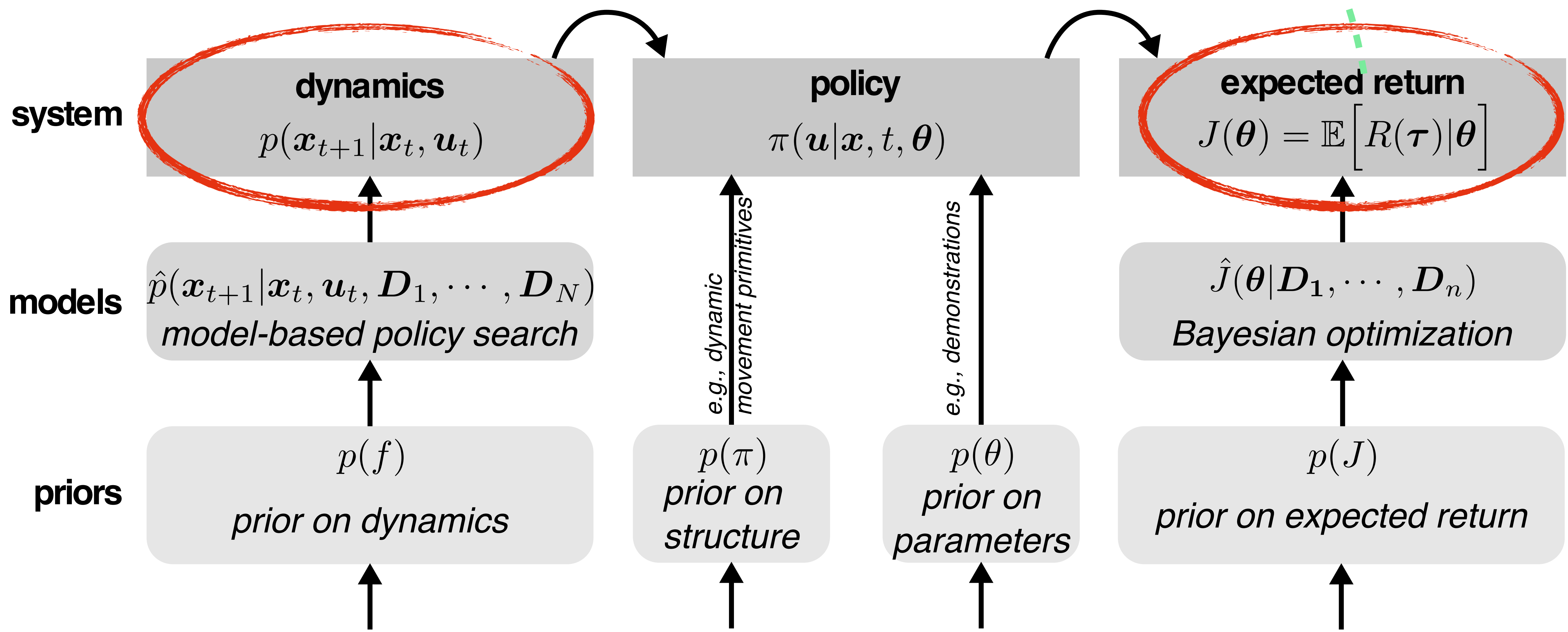
# Micro-data policy search

Parameters of the policy

**Optimize:** $$\boxed{J(\boldsymbol{\theta})} = \mathbb{E}\left[\sum_{t=1}^{T} \boxed{r(\mathbf{x}_t)} \,\middle|\, \boxed{\boldsymbol{\theta}}\right]$$

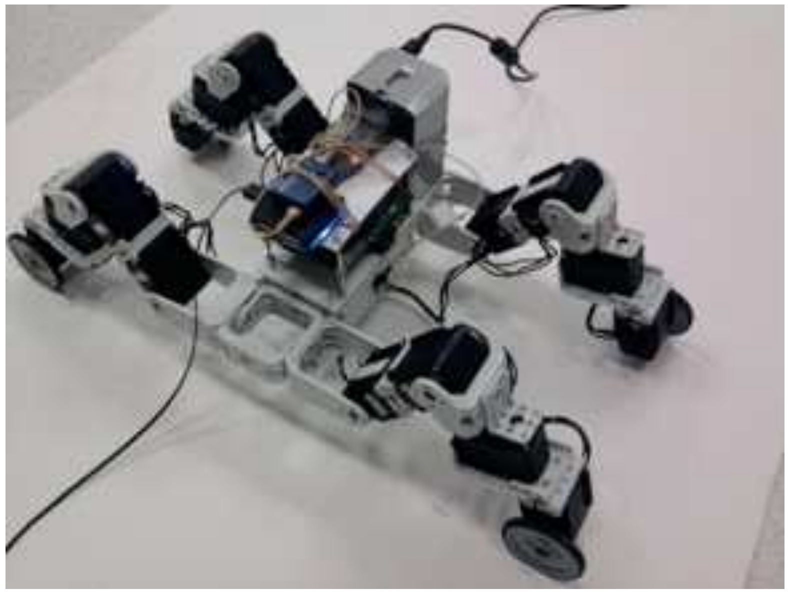
Objective

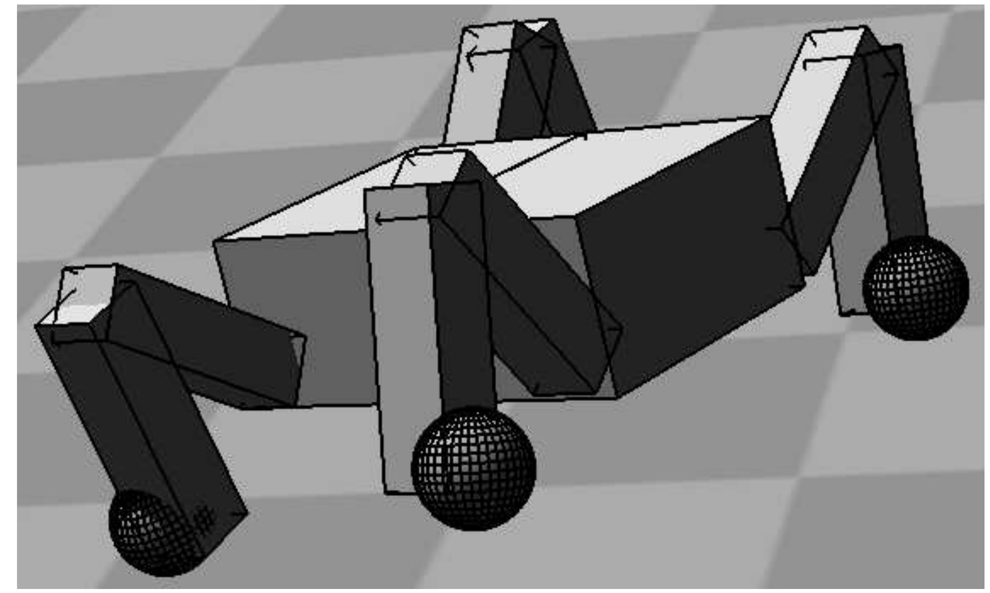
Reward for state $x_t$

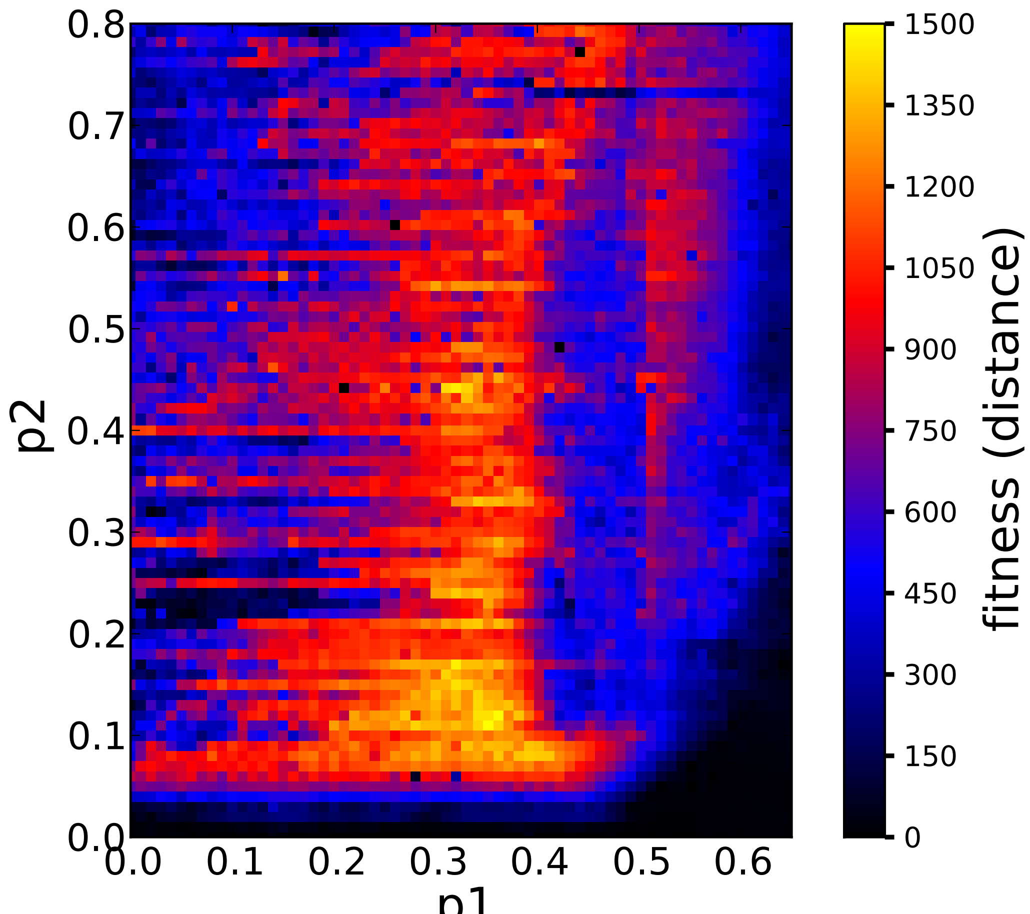


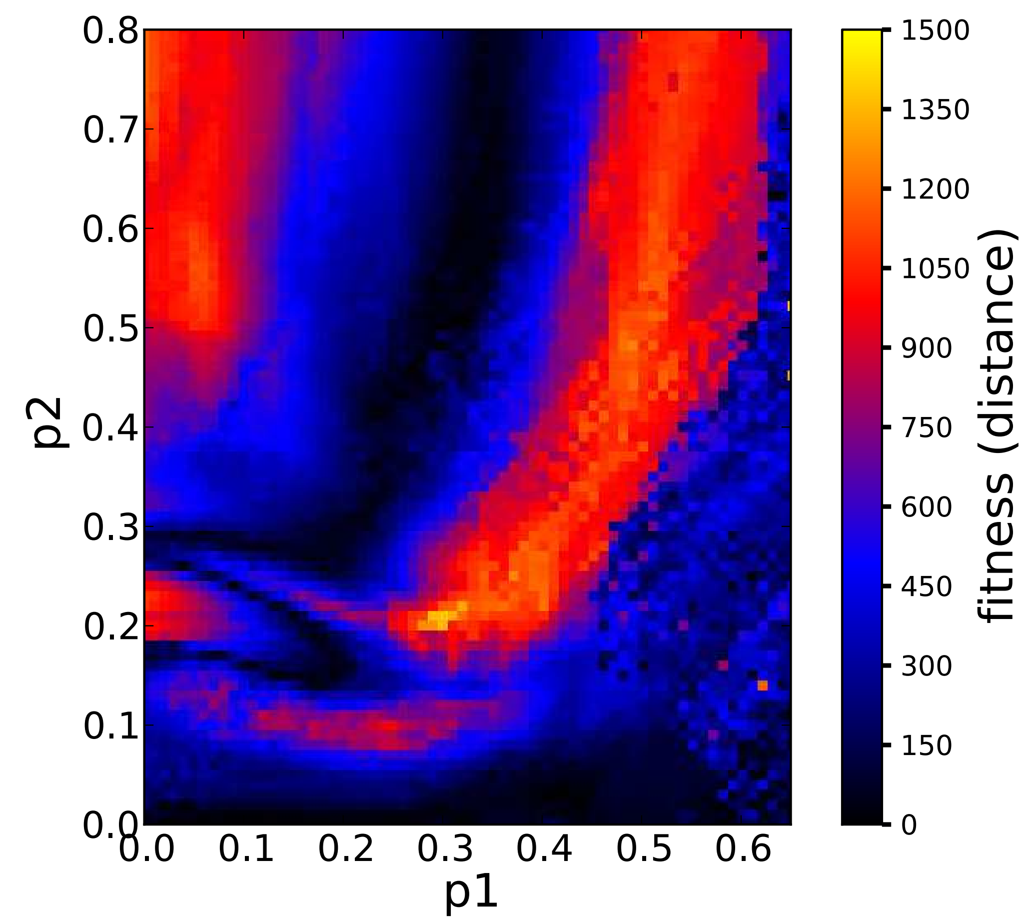

**In nature: evolution & experience**

**K. Chatzilygeroudis, V. Vassiliades, F. Stulp, S. Calinon, J.-B. Mouret. (2018).** A survey on policy search algorithms for learning robot controllers in a handful of trials. arXiv:1807.02303



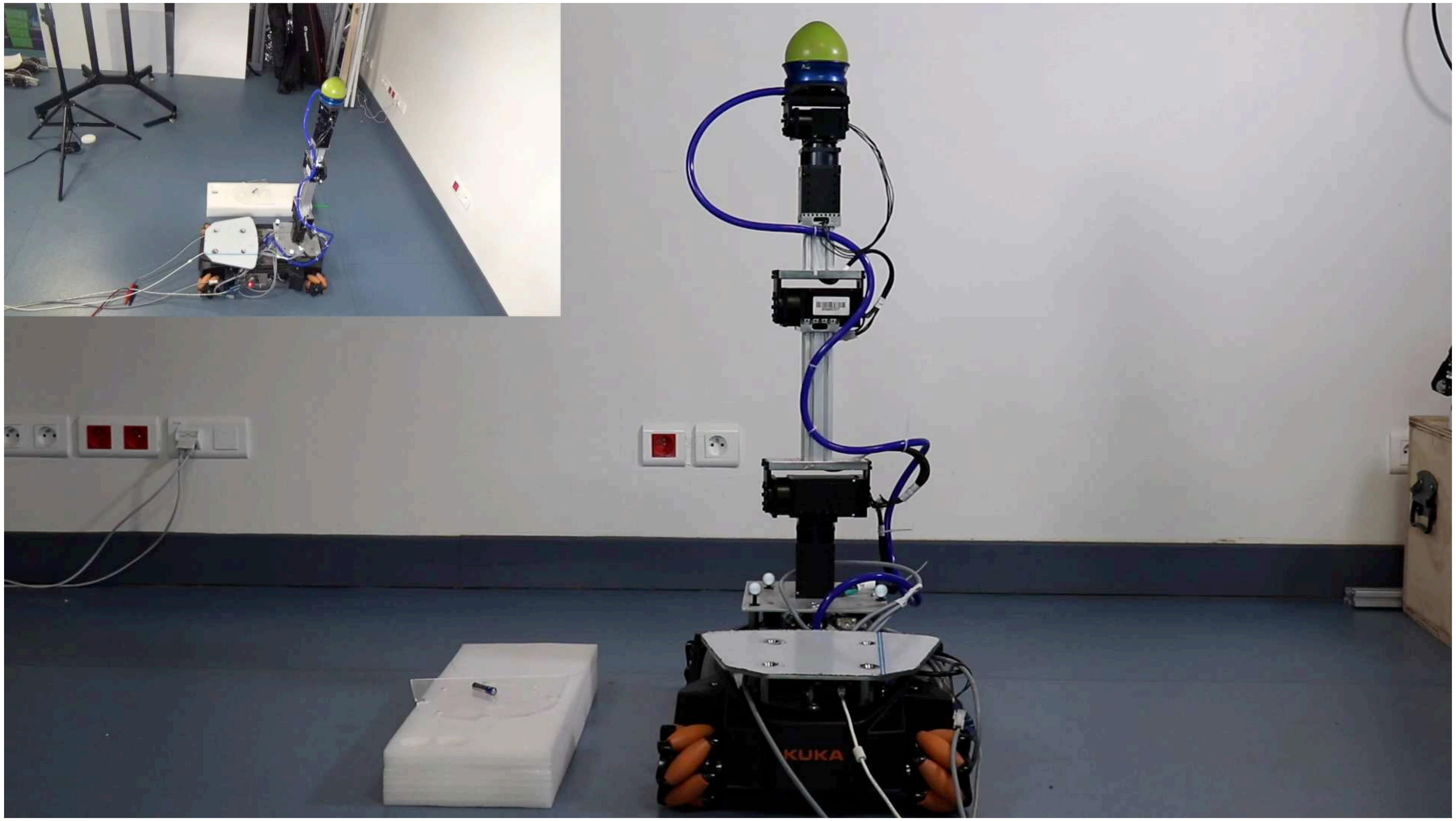





Mouret, J. B., Koos, S., & Doncieux, S. (2013). Crossing the reality gap: a short introduction to the transferability approach. arXiv preprint arXiv:1307.1870.

# Strategy 1: Learning the dynamical model




$$\mathbf{x}_{t+1} = \mathbf{x}_t + f(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{w}$$

next stat dynamic nois

Probabilistic model

Optimal policy (in the model)

Chatzilygeroudis K, Rama R, Kaushik R, Goepp D, Vassiliades V, Mouret JB. (2017) Black-Box Data-efficient Policy Search for Robotics. Proc. of IEEE IROS.

Deisenroth, M. P., Fox, D., & Rasmussen, C. E. (2015). Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2), 408-423.

# Scaling up model-based RL with priors

We cannot learn a model of a 6-legged robot

➟ the state space is too large

We can add the simulator as a prior

➟ we learn a <u>marginal model (residual)</u>

Gaussian processes

$$\mathbf{x}_{t+1} = \mathbf{x}_t + M(\mathbf{x}_t, \mathbf{u}_t, \phi_M) + f(\mathbf{x}_t, \mathbf{u}_t, \phi_K) + \mathbf{w}$$

Simulator / model

parameters

Learning = maximize the <u>likelihood</u> of M+f

We can <u>combine model learning and model identification</u>

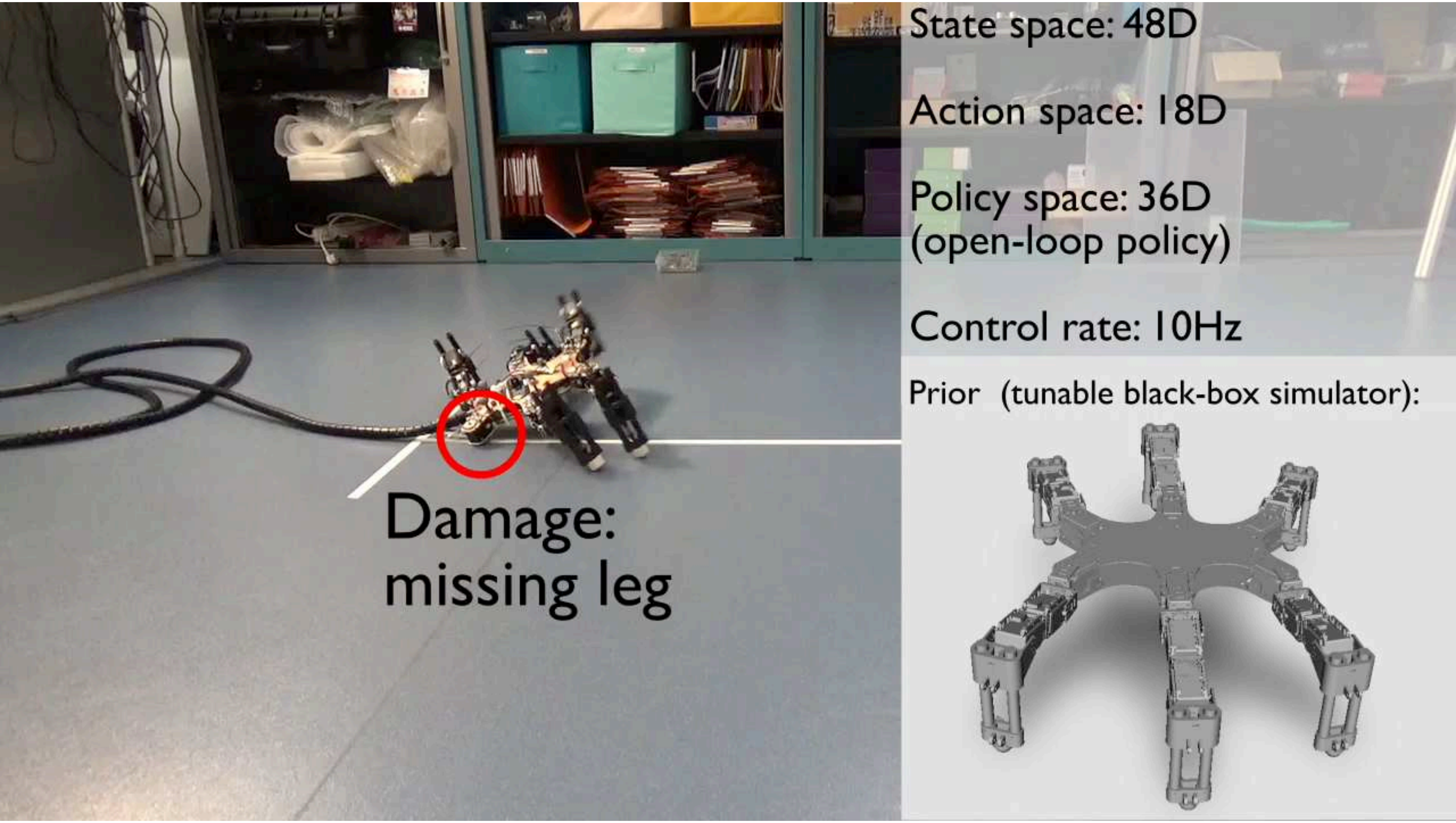
➟ **effects that can be captured** by the simulator will be included by tuning the simulator (model identification)

➟ **effects that cannot be captured** by changing the parameters are modelled by the Gaussian processes

**Chatzilygeroudis K, Mouret JB.** (2018) Using Parameterized Black-Box Priors to Scale Up Model-Based Policy Search for Robotics. Proc. of ICRA.

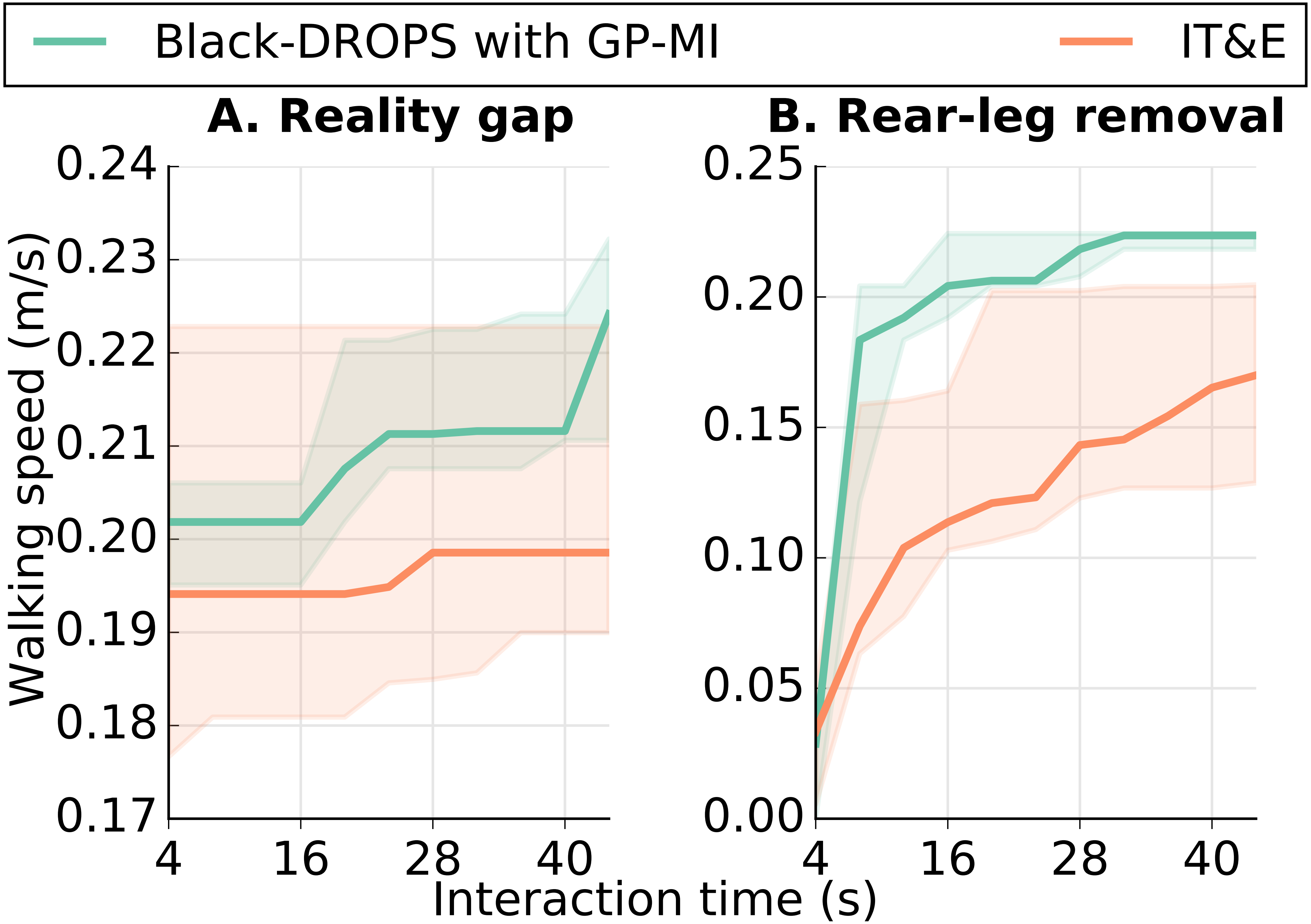
# Black-DROPS + priors + identification

Model-based policy search




**Chatzilygeroudis K, Mouret JB.** (2018) Using Parameterized Black-Box Priors to Scale Up Model-Based Policy Search for Robotics. Proc. of ICRA.

# Physical 6-legged robot




**Chatzilygeroudis K, Mouret JB.** (2018) Using Parameterized Black-Box Priors to Scale Up Model-Based Policy Search for Robotics. Proc. of ICRA.

# Conclusions — Micro-Data learning

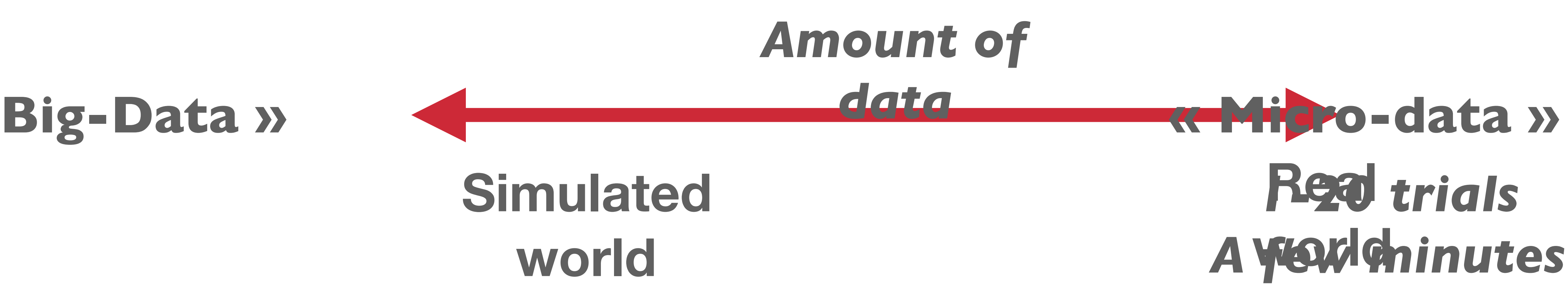
- We all want robots that can learn. But why? Is it *really* needed?

➟ Learning in robotics is most useful for online adaptation

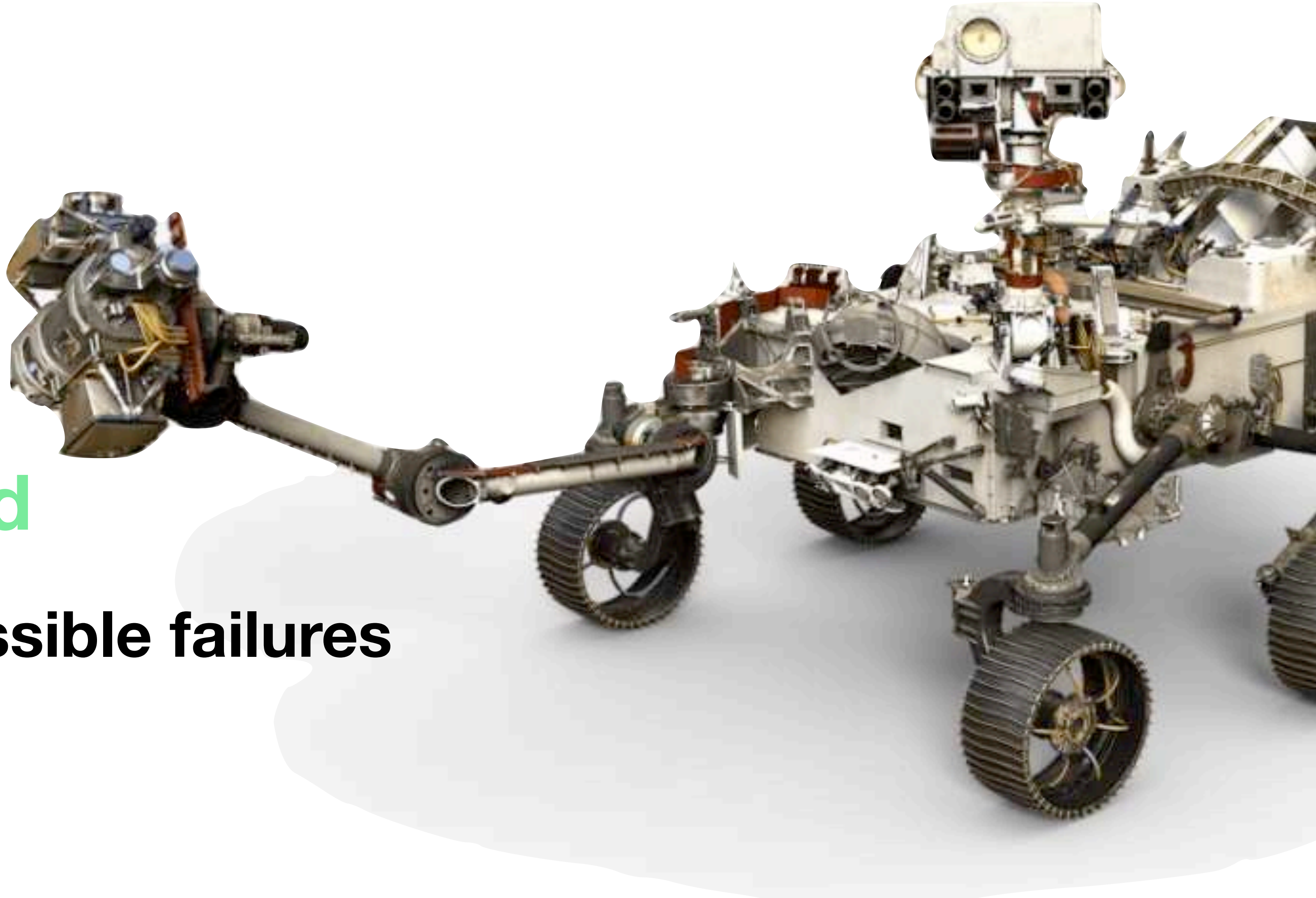
➟ Damage recovery: the "killer app" for robot learning?

**… but robots need to learn in a few minutes (micro-data)**

- ➟ priors: generic for robotics? automatic generation?
- ➟ *probabilistic* models: policies robust to inaccuracies



... diagnosis is hard
... robustness is hard



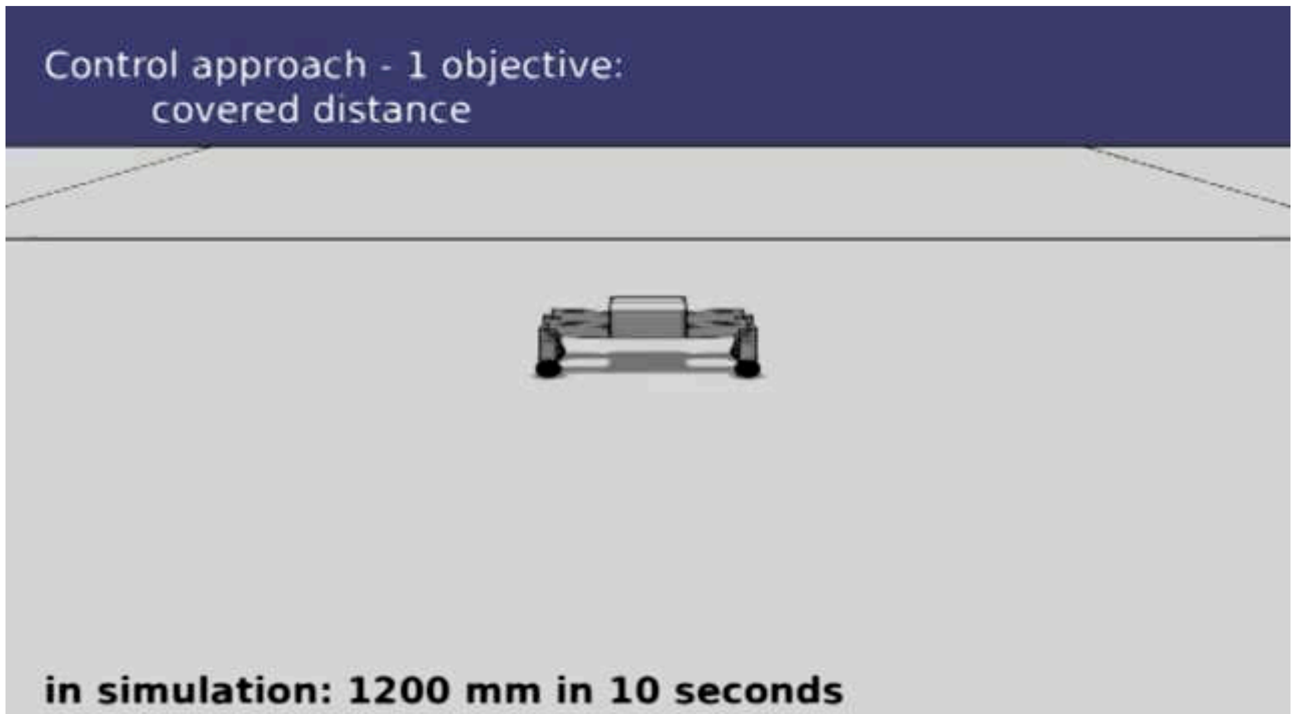
➡ **need to anticipate possible failures**

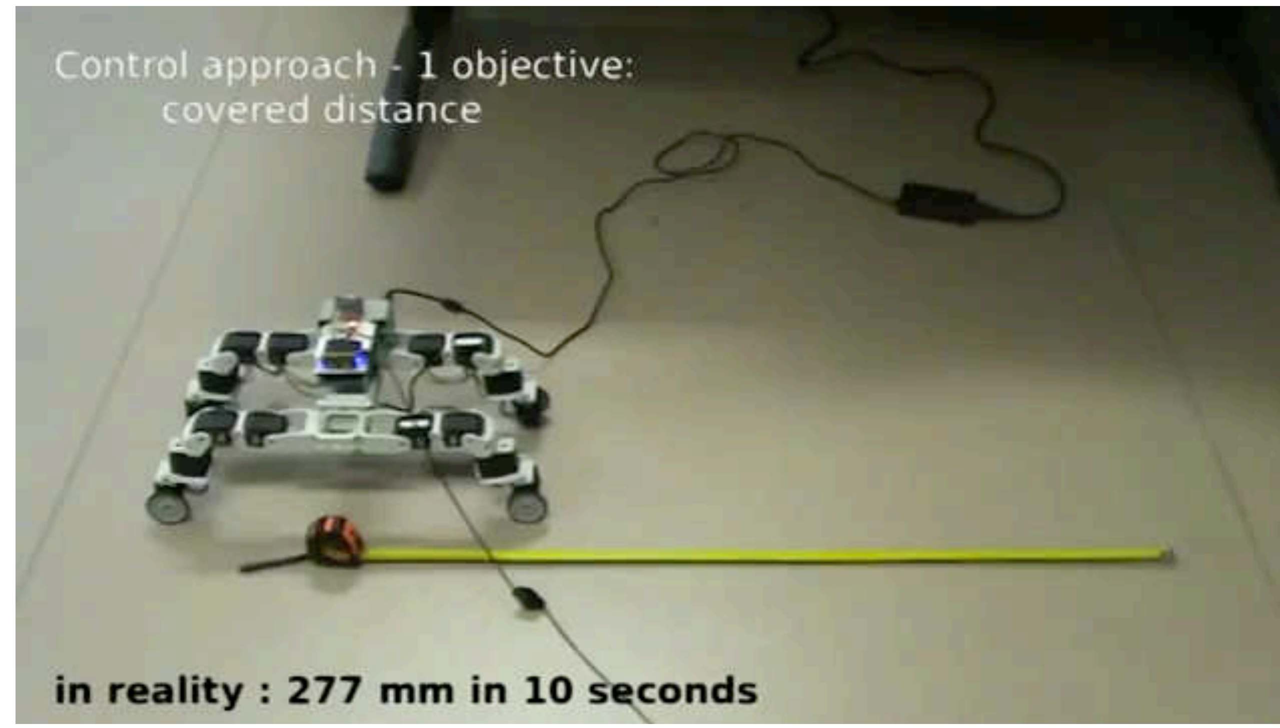
## Could trial-and-error learning offer an alternative?

➡ no need to understand the damage to find a compensatory behaviour
➡ “takes a shortcut”

## Where could “AI” could make a difference in robotics?

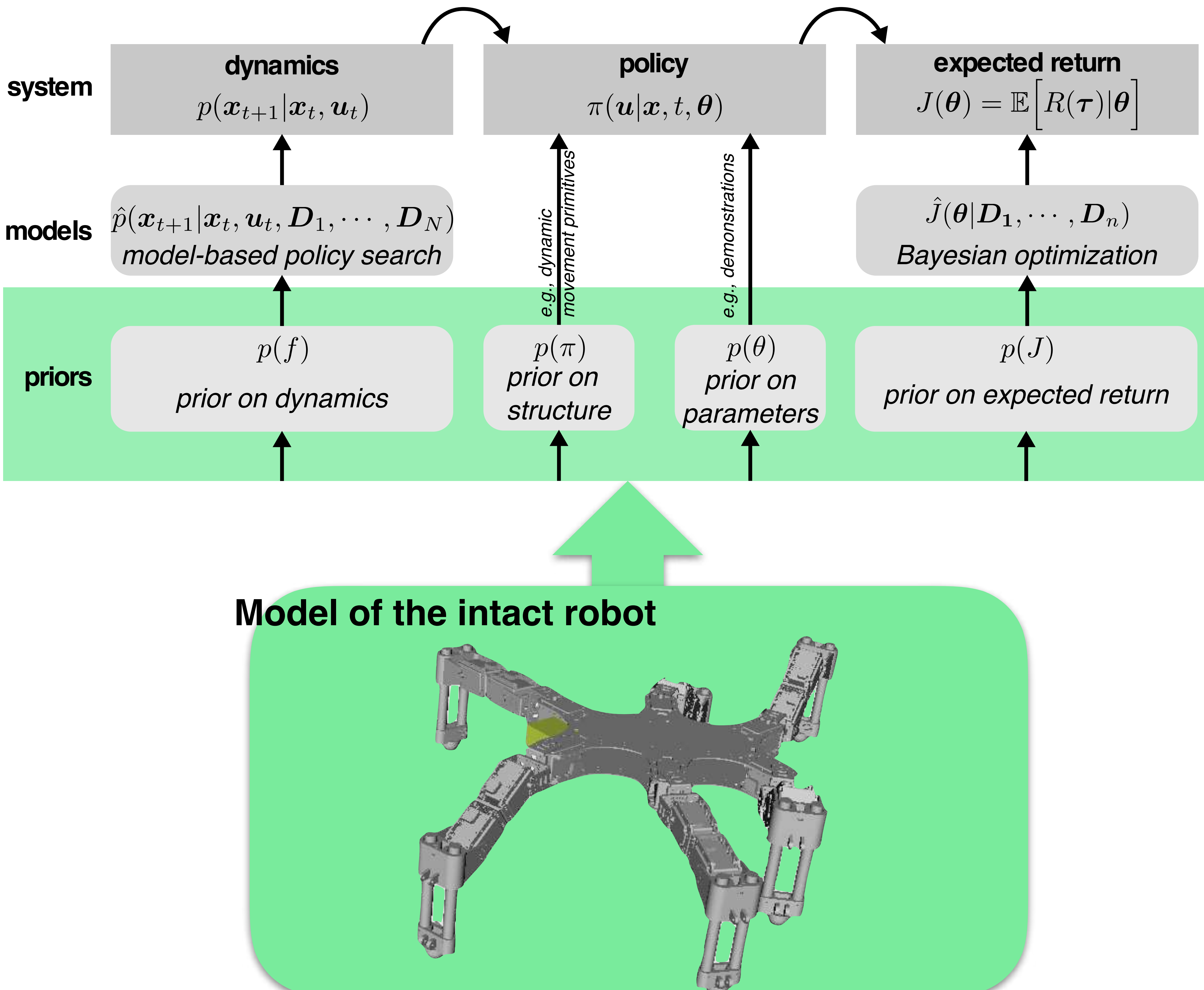
## subtitle



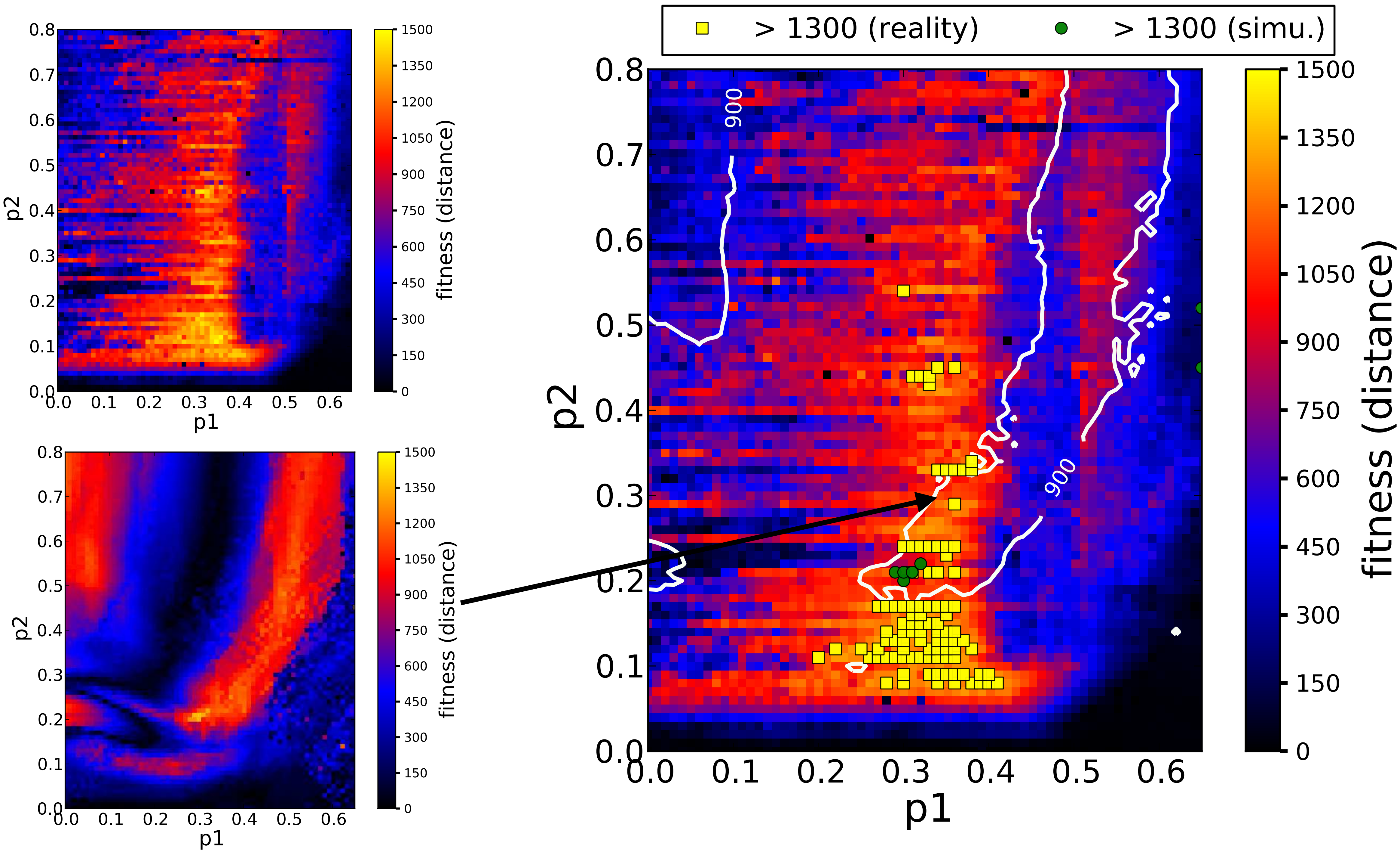




biblio

# Policy Search for damage recovery & adaptation

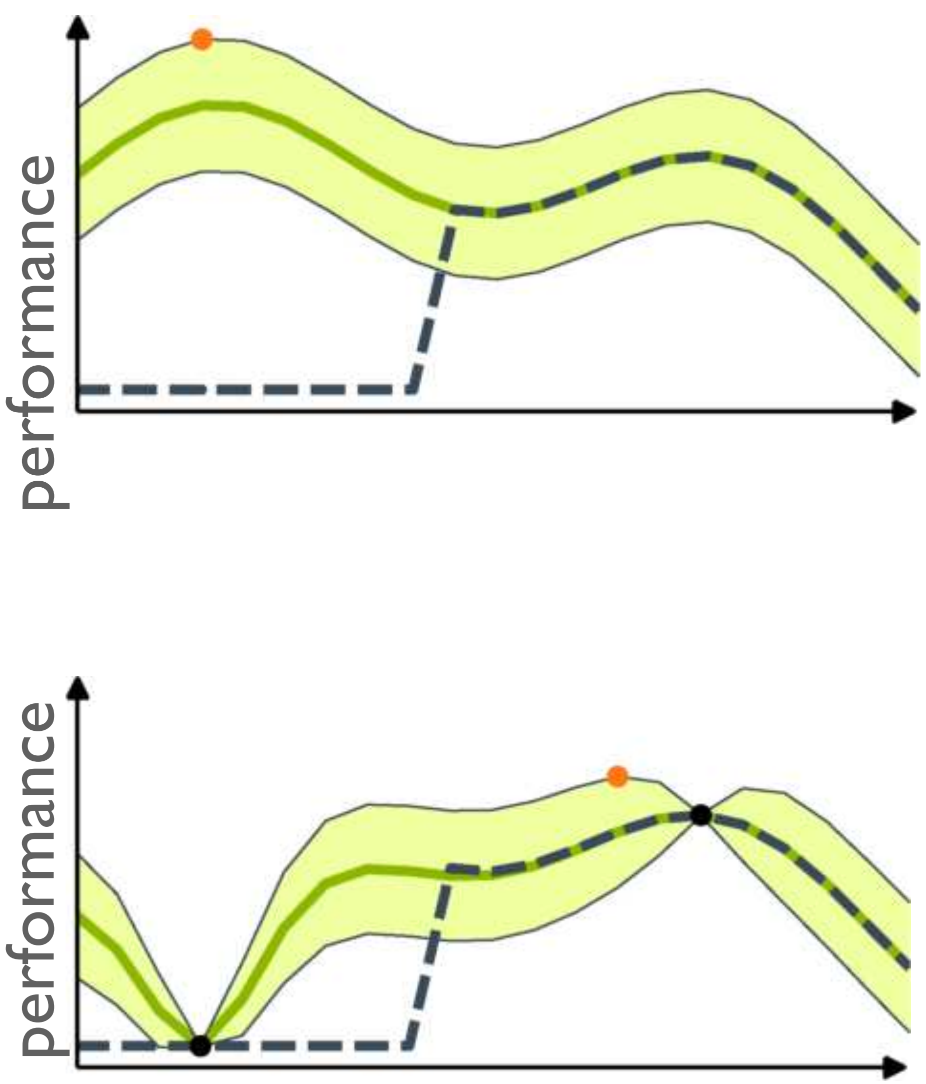


# But they can agree (sometimes)!



# BO + MAP-Elites

## "Intelligent Trial and Error"




$$P(f(\mathbf{x})|\mathbf{P}_{1:t+1}), \mathbf{x}) = \mathcal{N}(\mu_{t+1}(\mathbf{x}), \sigma^2_{t+1}(\mathbf{x}))$$

where

$$\mu_{t+1}(\mathbf{x}) = \mathcal{A}(\mathbf{x}) + \mathbf{k}^t\mathbf{K}^{-1}(\mathbf{P}_{1:t+1} - \mathcal{A}(\mathbf{y}_{1:t+1}))$$

$$\sigma^2_{t+1}(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}^t\mathbf{K}^{-1}\mathbf{k}$$

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{y}_1, \mathbf{y}_1) + \sigma^2_{noise} & \cdots & k(\mathbf{y}_1, \mathbf{y}_t) \\ \vdots & \ddots & \vdots \\ k(\mathbf{y}_t, \mathbf{y}_1) & \cdots & k(\mathbf{y}_t, \mathbf{y}_t) + \sigma^2_{noise} \end{bmatrix}$$

$$\mathbf{k} = \begin{bmatrix} k(\mathbf{x}, \mathbf{y}_1) & k(\mathbf{x}, \mathbf{y}_2) & \cdots & k(\mathbf{x}, \mathbf{y}_t) \end{bmatrix}$$

**Cully, A. and Clune, J. and Tarapore, D. and Mouret, J.-B.** (2015). *Robots that can adapt like animals.* **Nature**. Vol 521 Pages 503-507.

# Automatic selection of priors

- **Goal:** relax the need to have “the right prior”
- **Concept**:
  - Generate / create many priors (e.g. 100)
  - The likelihood of a prior can be computed (prior + obs.)
  - Choose after each episode
    - prior that is likely given the optimization
    - prior that can help the optimization

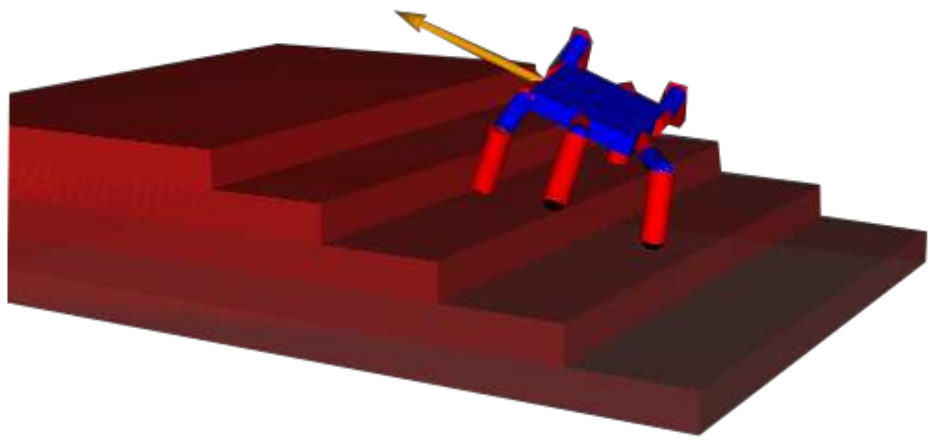
    ➠ trade-off
- **New acquisition function:** Most likely Expected Improvement

$$\mathrm{EIP}(\boldsymbol{x}, \mathcal{P}) = \mathrm{EI}(\boldsymbol{x}) \times P(\boldsymbol{f}(\boldsymbol{x}_{1..t}) \mid \boldsymbol{x}_{1..t}, \boldsymbol{\mathcal{P}}(\boldsymbol{x}_{1..t}))$$

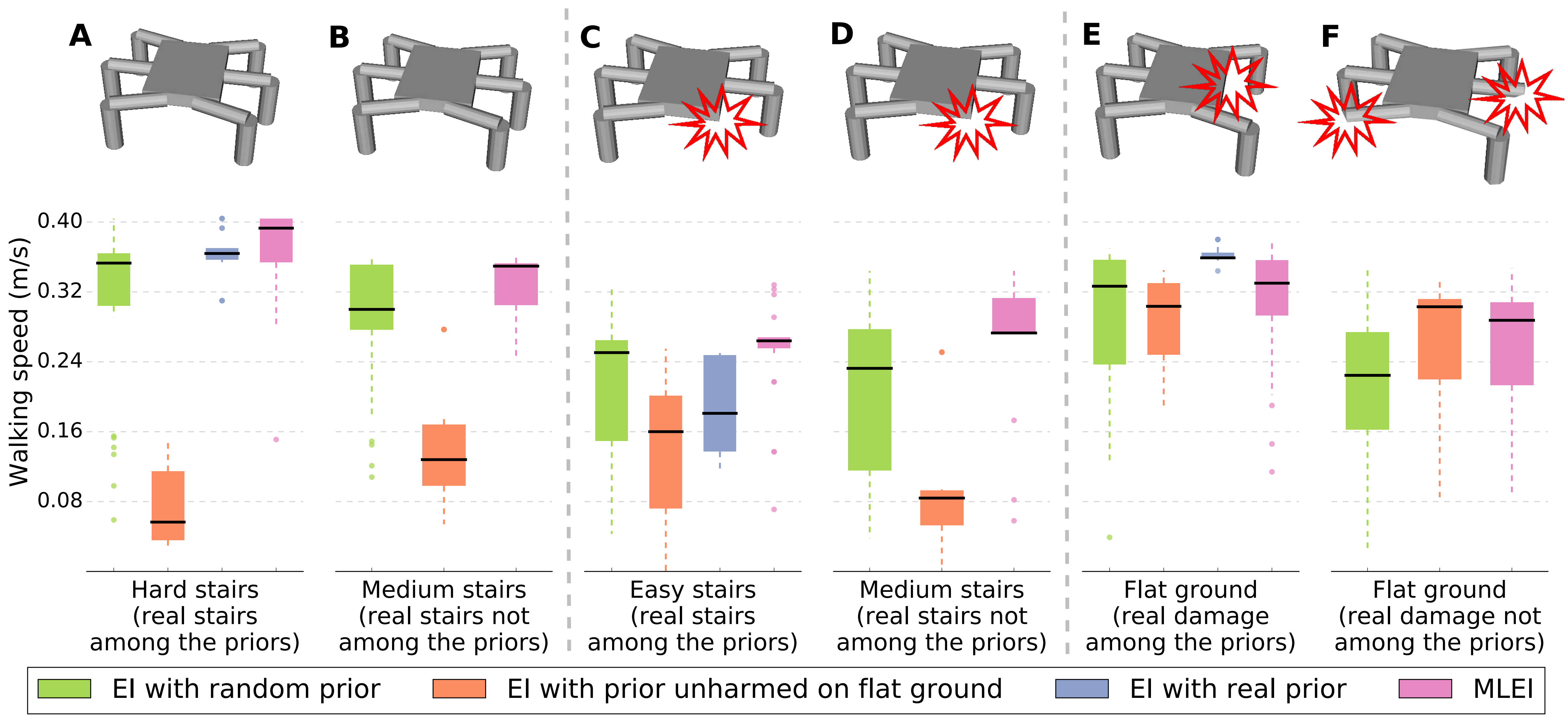
$$\mathrm{MLEI}(\boldsymbol{x}, \mathcal{P}_1, \cdots, \mathcal{P}_m) = \max_{p \in \mathcal{P}_1, \cdots, \mathcal{P}_m} \mathrm{EIP}(\boldsymbol{x}, p)$$

**Pautrat, R., Chatzilygeroudis, K., & Mouret, J. B. (2017).** Bayesian Optimization with Automatic Prior Selection for Data-Efficient Direct Policy Search. arXiv preprint arXiv:1709.06919.

# Damage recovery on stairs



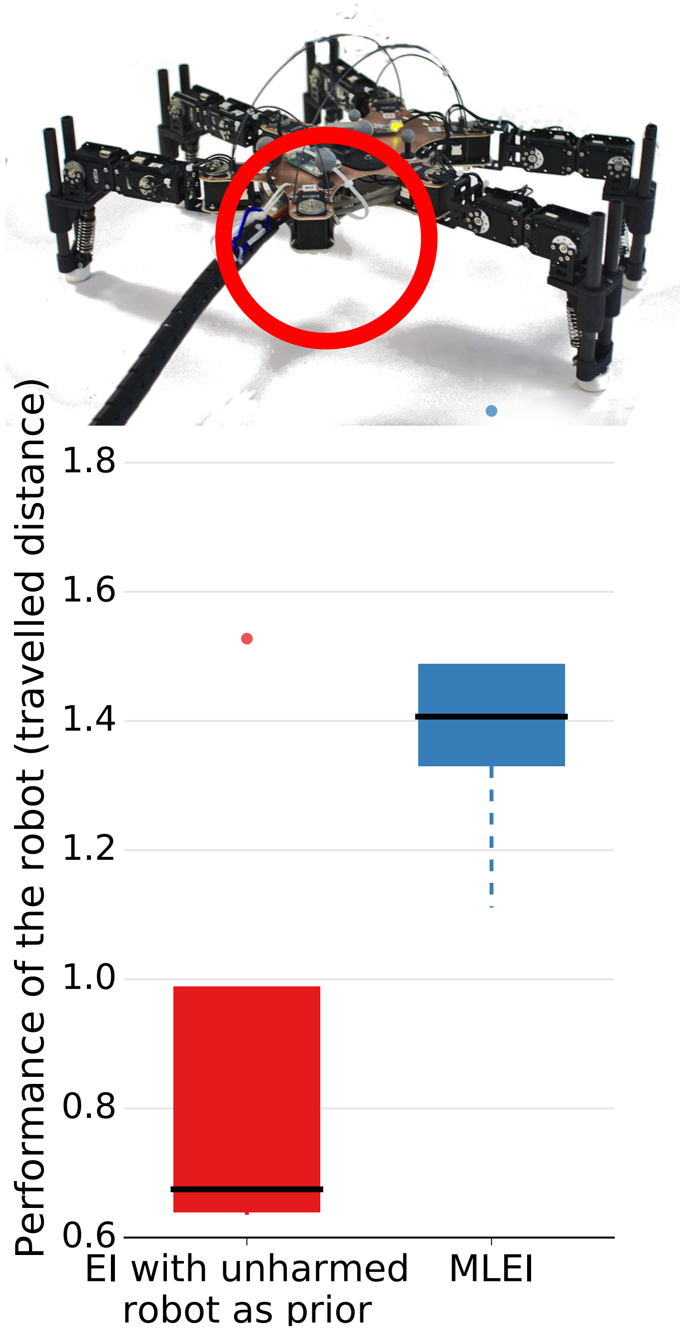
15 priors for 4 environments (stairs) = 60
15 priors for 6 damage conditions + intact = 105
5 iterations
30 replicates



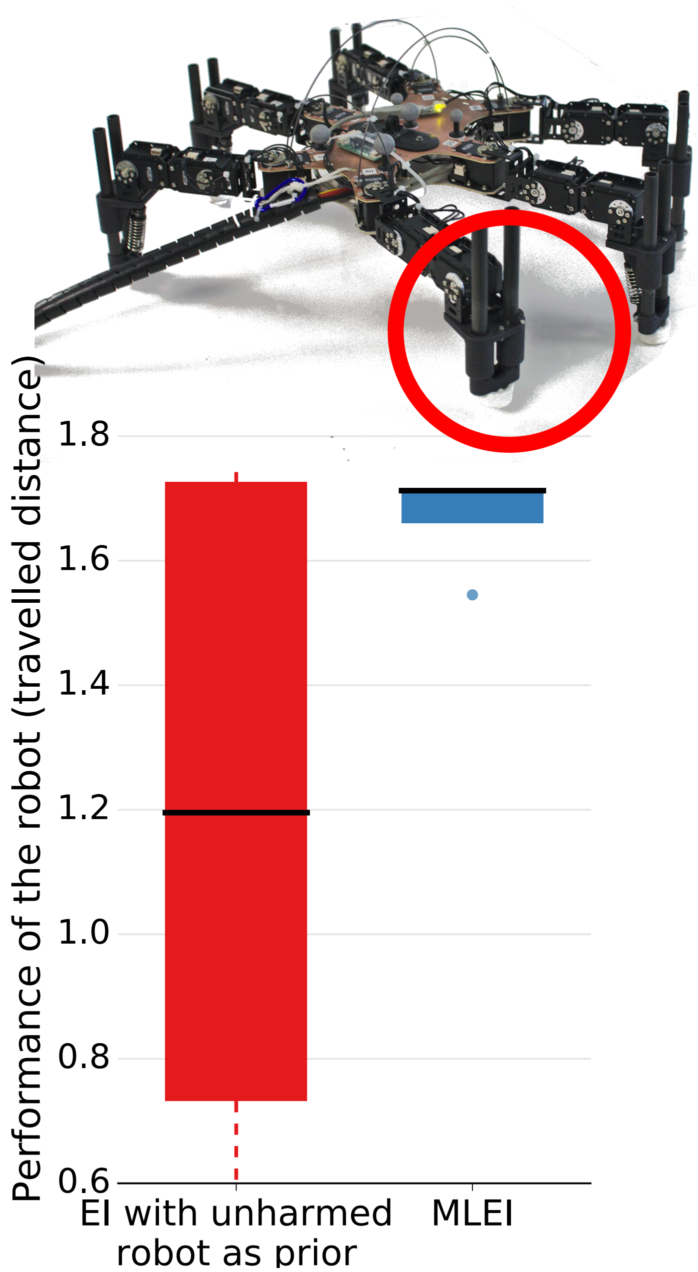

**Pautrat, R., Chatzilygeroudis, K., & Mouret, J. B. (2017).** Bayesian Optimization with Automatic Prior Selection for Data-Efficient Direct Policy Search. arXiv preprint arXiv:1709.06919.

# Damage recovery

10 episodes / 5 replicates / 15 priors for 6 damage conditions + intact = 105



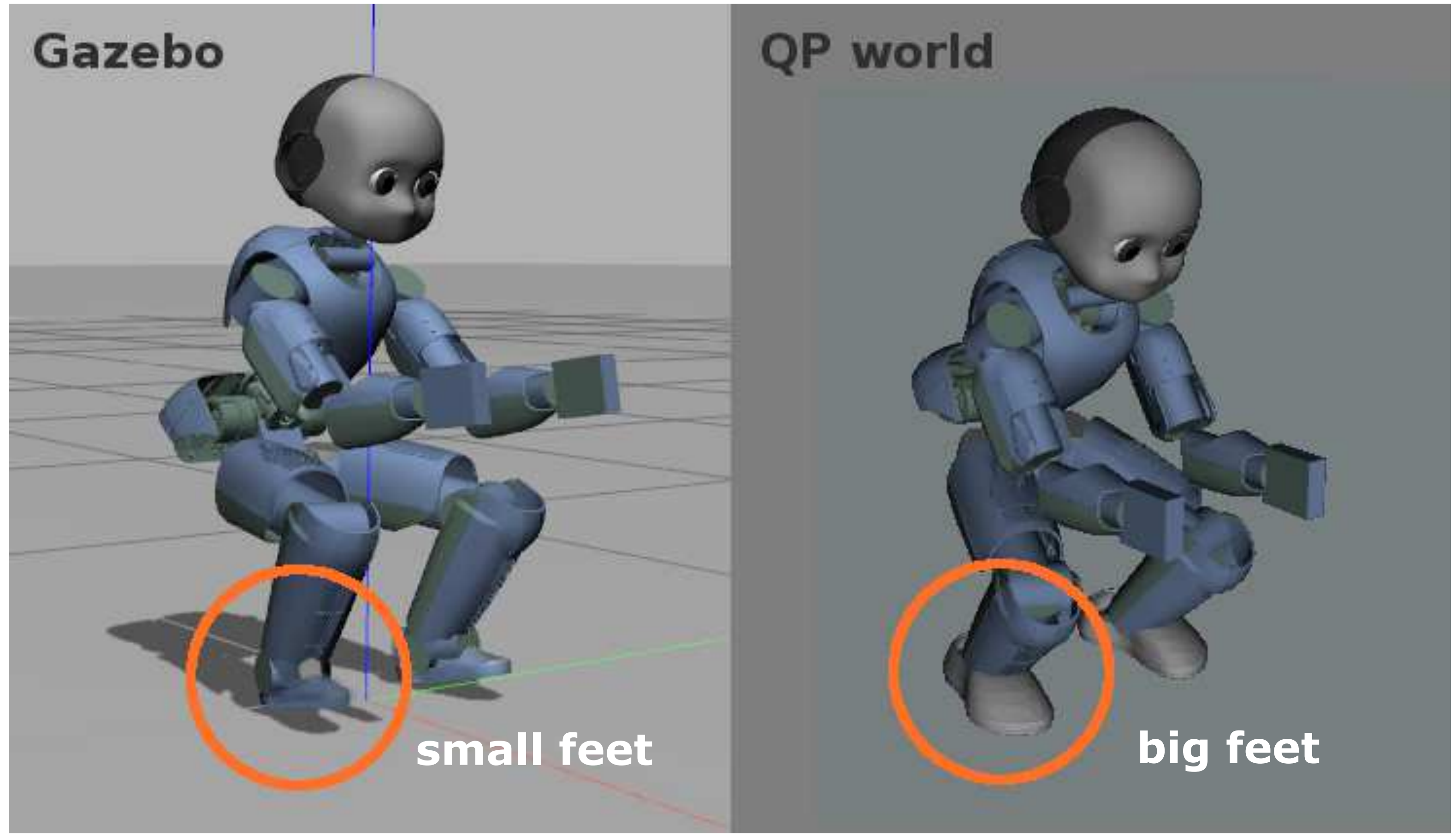

***damage among the priors***

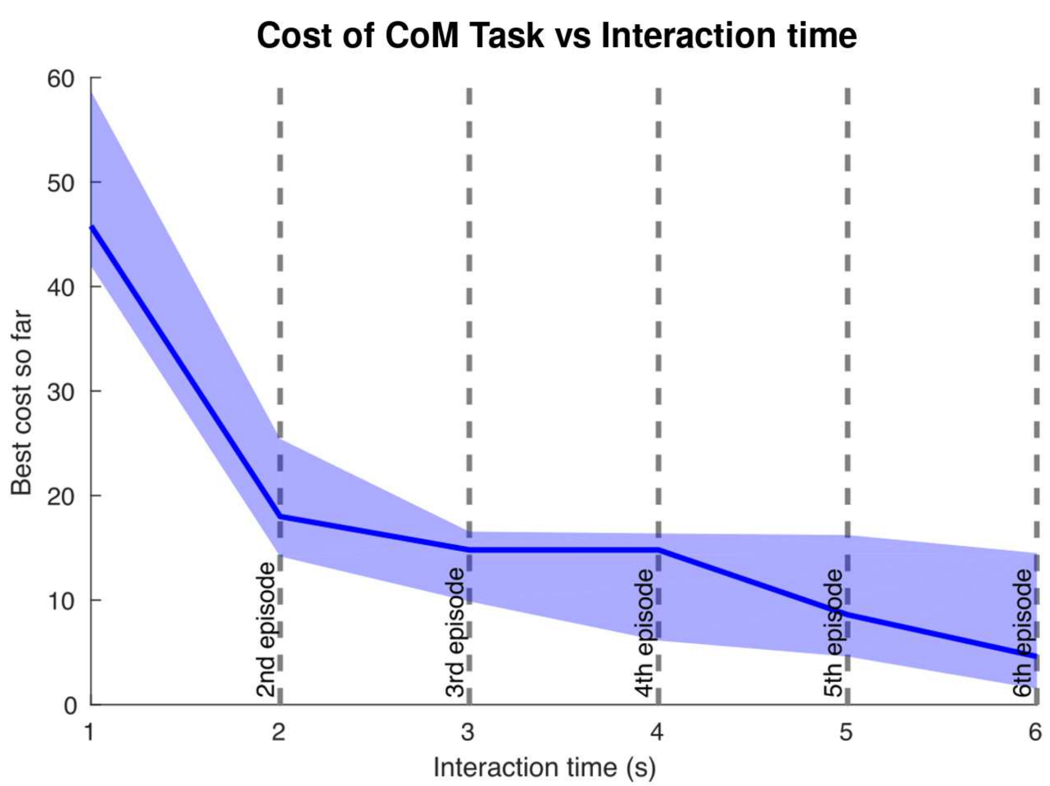


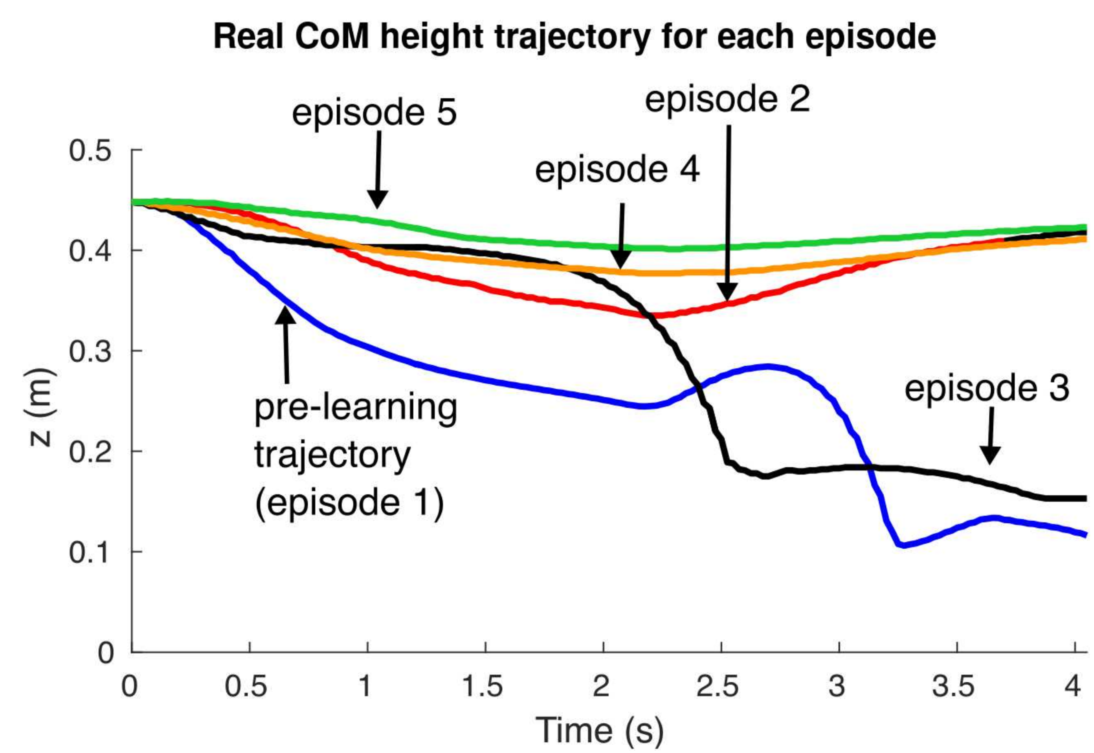

***damage NOT among the priors***

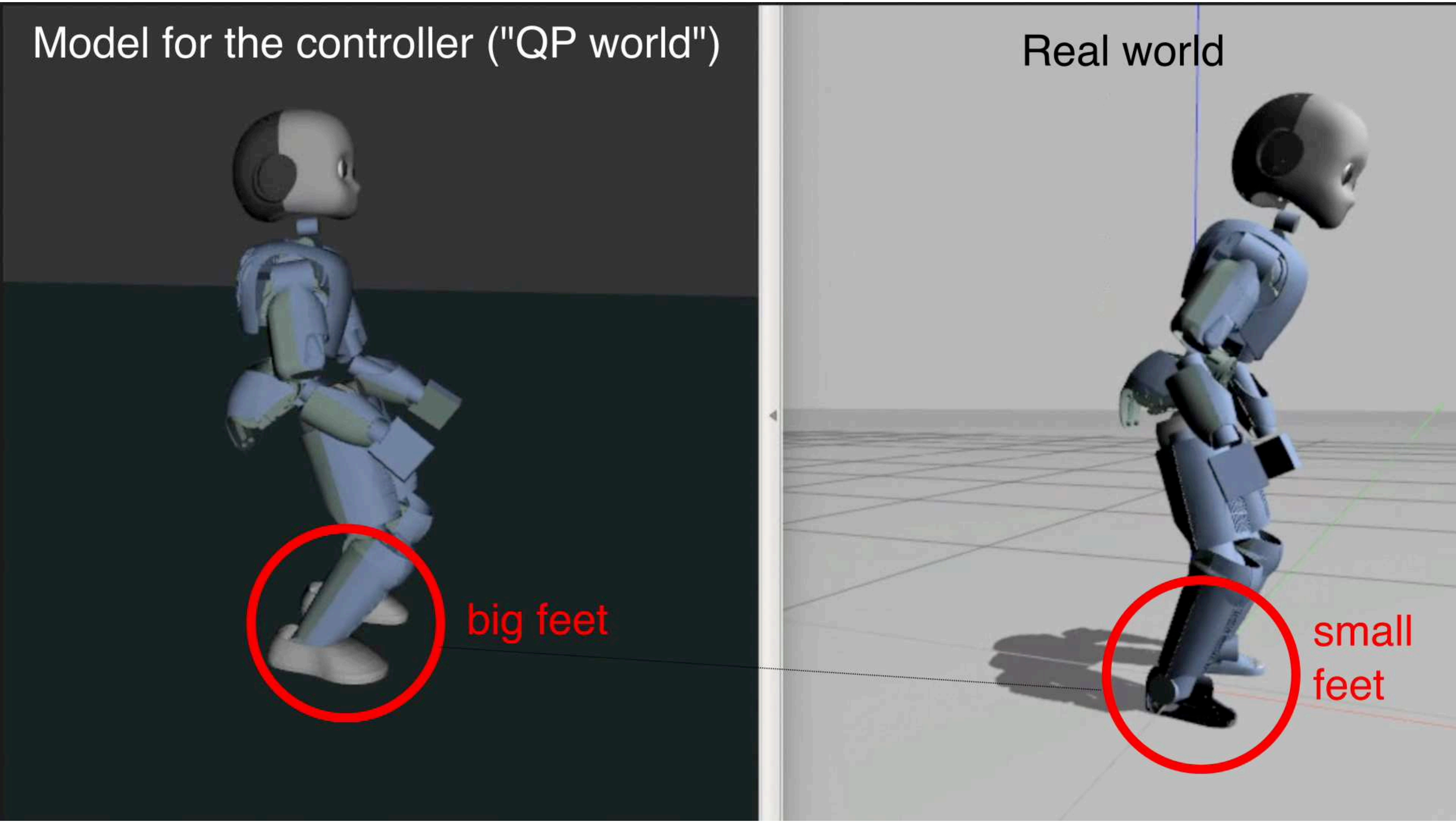

**Pautrat, R., Chatzilygeroudis, K., & Mouret, J. B. (2017).** Bayesian Optimization with Automatic Prior Selection for Data-Efficient Direct Policy Search. arXiv preprint arXiv:1709.06919.

"real" physical world model for the controller



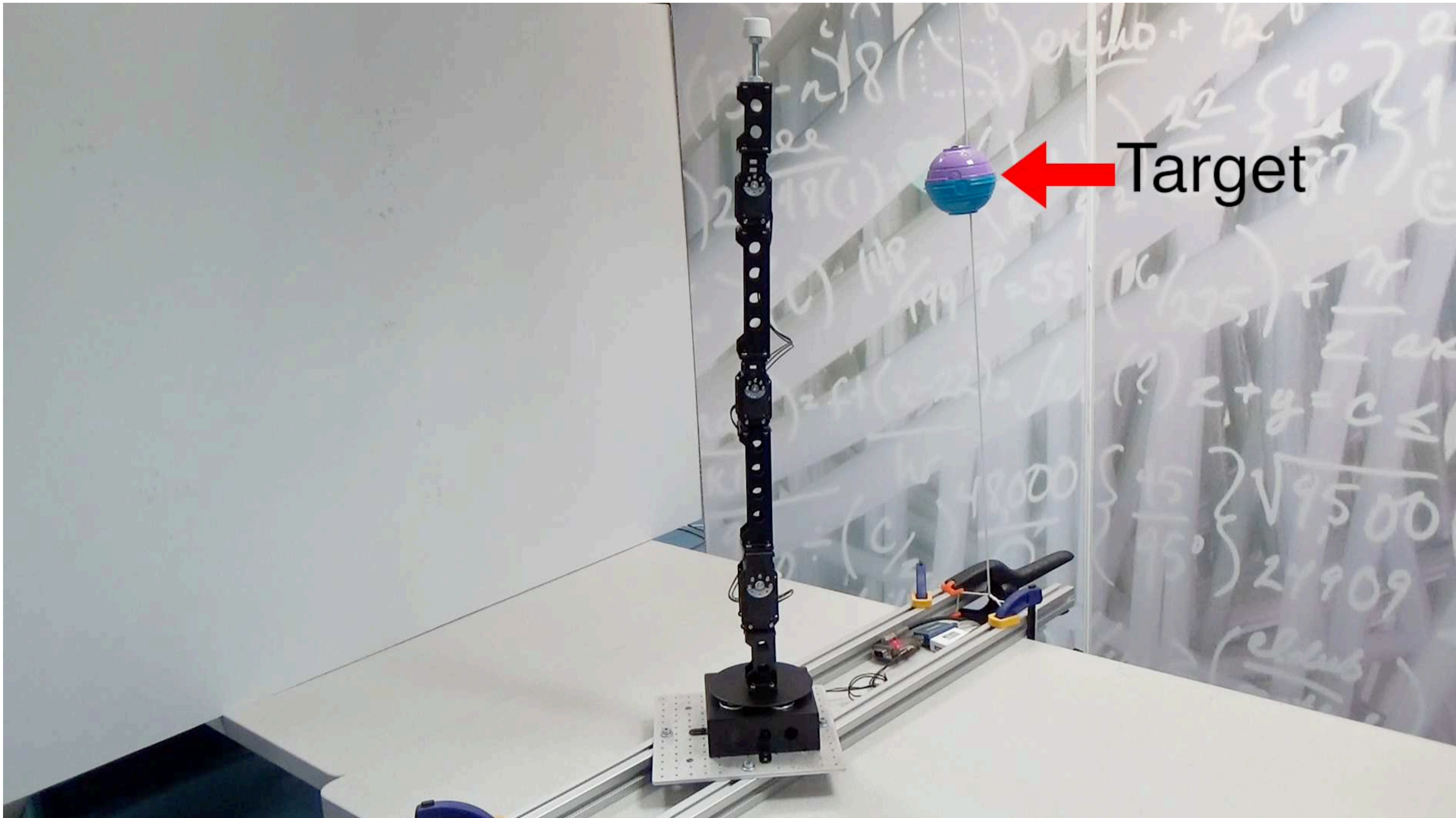



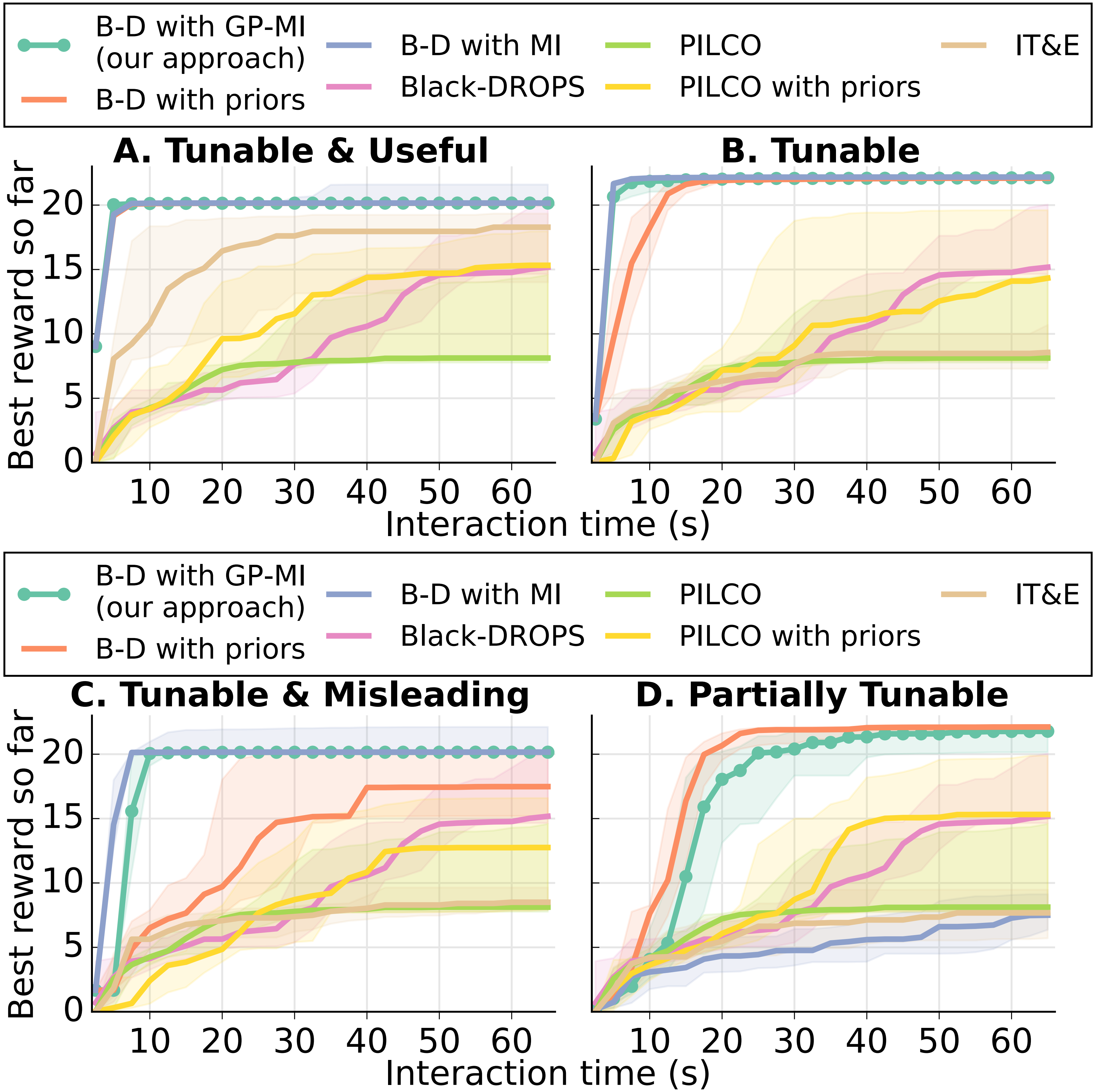

Model for the controller ("QP world")
Real world
big feet
small
feet

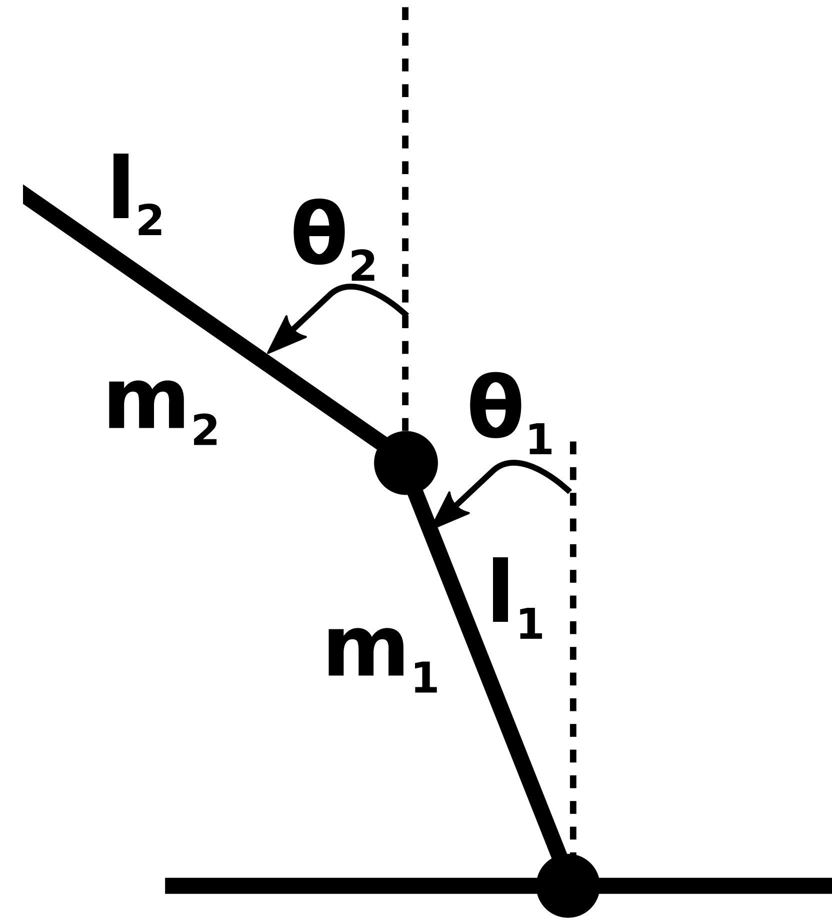
# Model-based policy search

## The Black-DROPS algorithm / 160 parameters (neural net.)




**Chatzilygeroudis K, Rama R, Kaushik R, Goepp D, Vassiliades V, Mouret JB.** (2017) Black-Box Data-efficient Policy Search for Robotics. Proc. of IEEE IROS.




Pendubot system