Aalto University
School of Engineering

DEEP REINFORCEMENT LEARNING in ROBOTIC APPLICATIONS

Aalto University

December 2019

Ali Ghadirzadeh
algh@kth.se

Reinforcement Learning

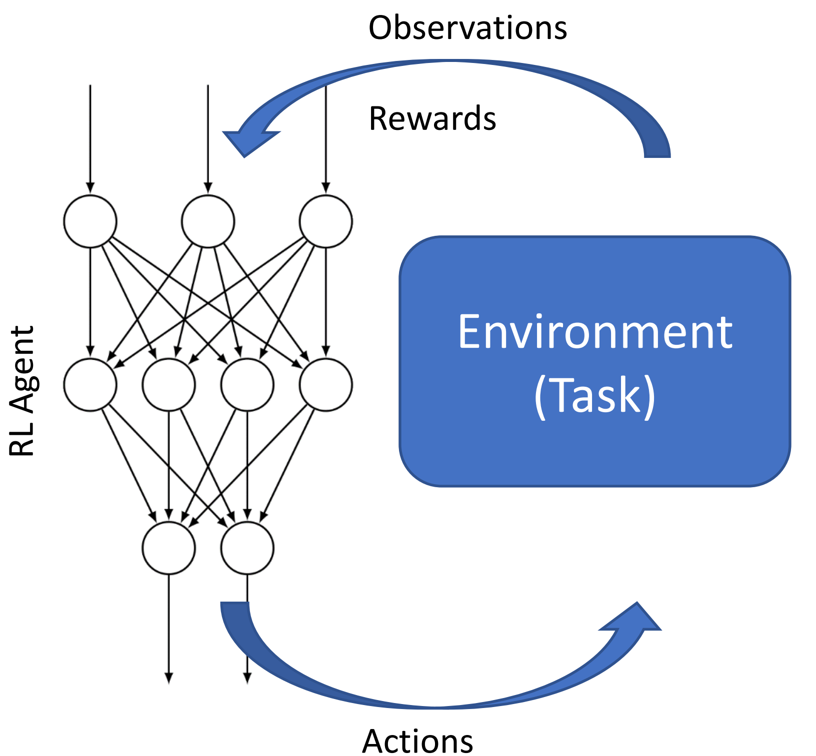
- Perform exploratory actions a_t
- Observe
  - the state s_t
  - the reward r_t
- Compare accumulated rewrad with your expectation at state s_t

$$R_t - V\ (s_t)$$

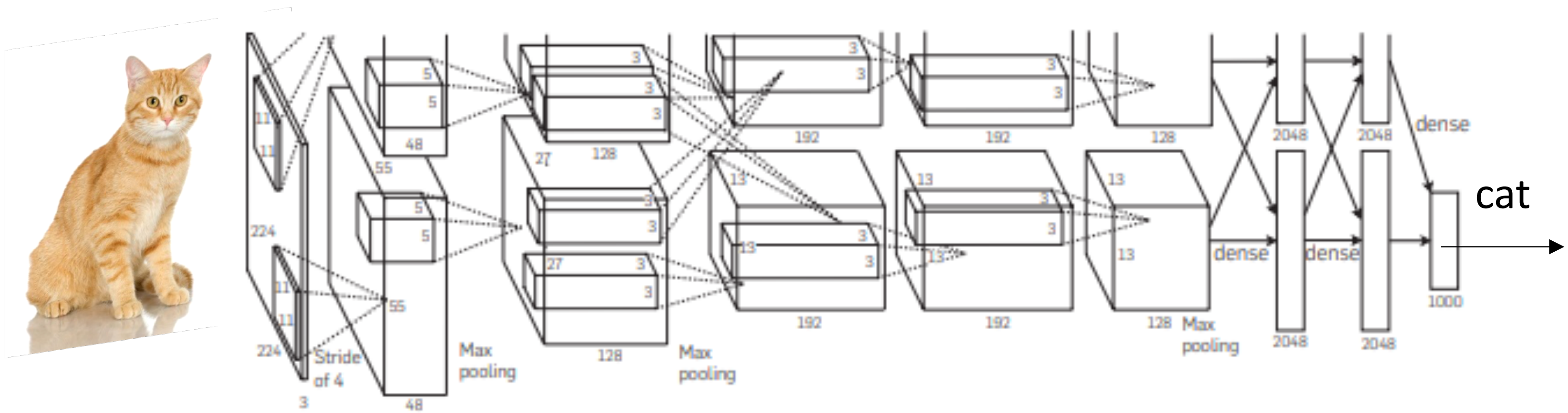
Accumulated reward

Your expectation

- Better than expected? Reinforce the action



Deep Learning



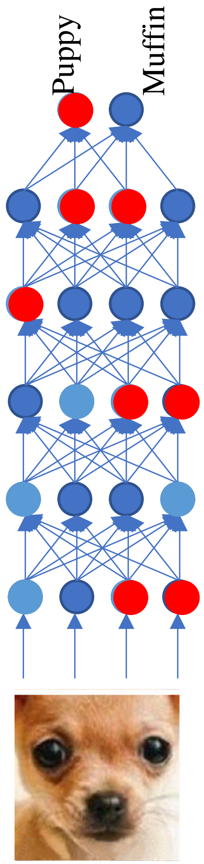
- High capacity models
- Highly diverse datasets
- Train end-to-end
- Powerful gradient-based opt.
- Powerful computations

Deep Learning

Prevent the Answer

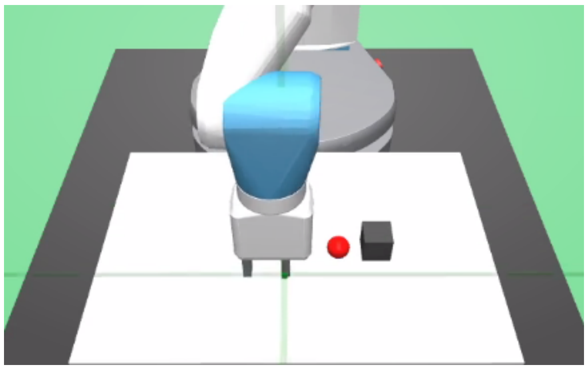
Reinforce the Answer

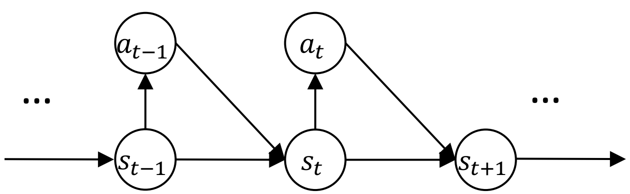
Puppies or Muffins?



Reinforcement Learning

Move the box to the target



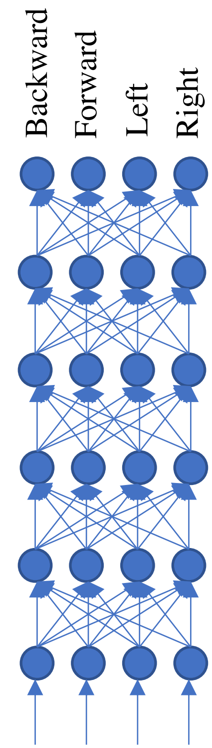


$$\tau = \{s_0, a_0, r_0, \ldots, s_{T-1}, a_{T-1}, r_{T-1}\}$$

Trajectory

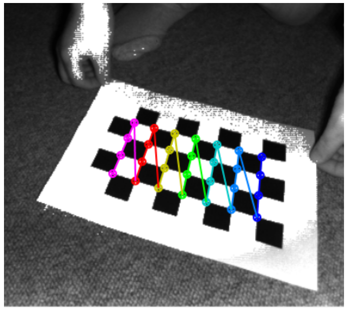
$$\pi_\theta(a_t|s_t)$$

Action-selection policy

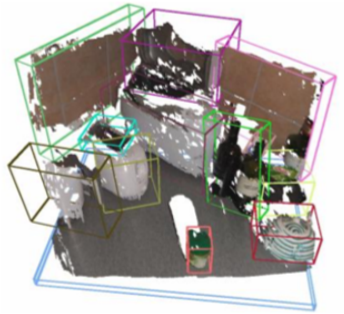


s_t

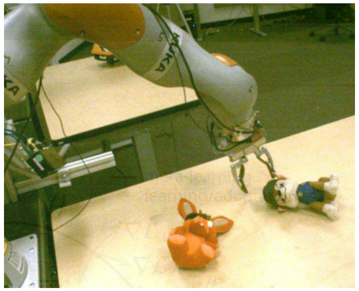
Reinforcement Learning - Robotics



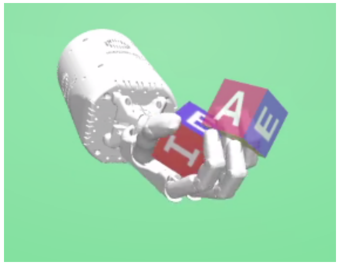
Calibration



Visual perception



Robustness



Dexterous manipulations

Experts

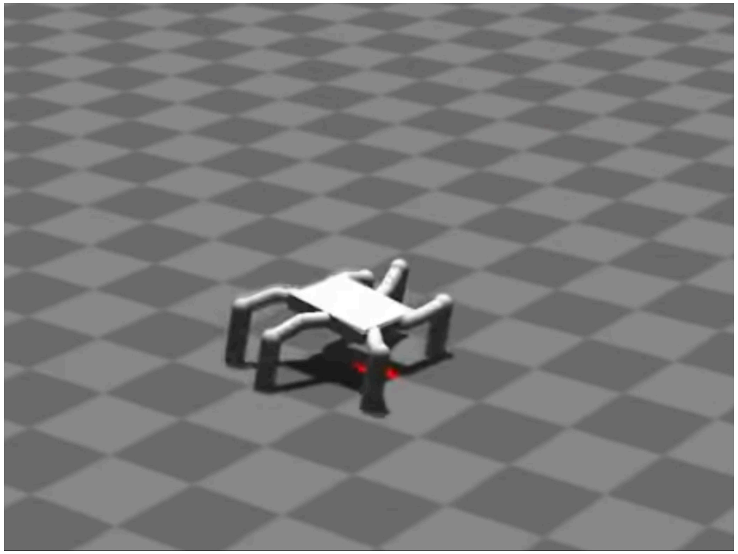
Complex dynamics to model

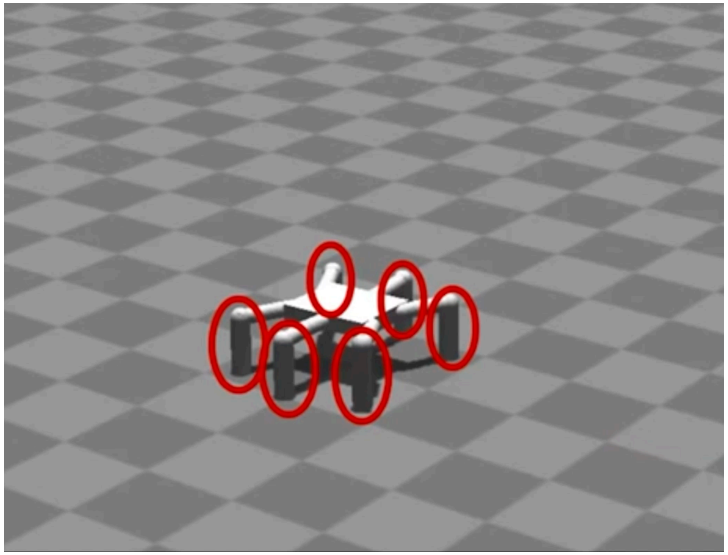
Reinforcement Learning

- Continuous score
- Reset after each trial
- Sufficient training data

Deep Q Learning AI playing Space Invaders
https://youtu.be/Qvco7ufsX_0

Reinforcement Learning

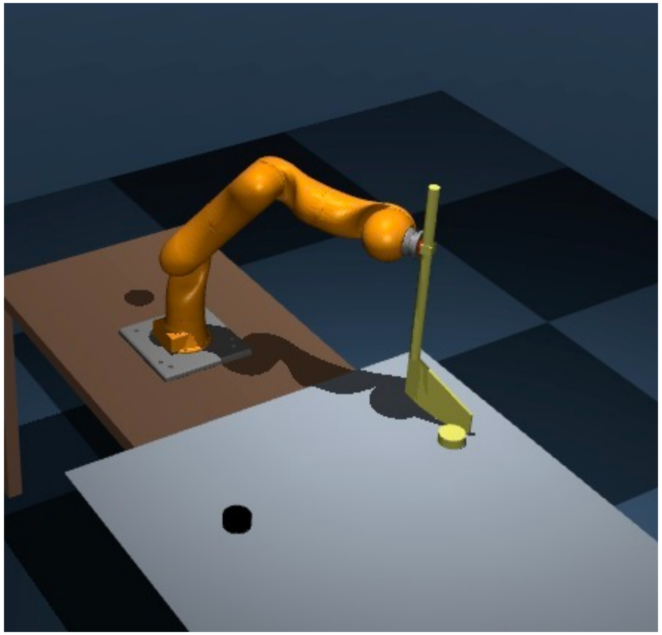




- Move as fast as possible
- Minimize foot contacts with ground

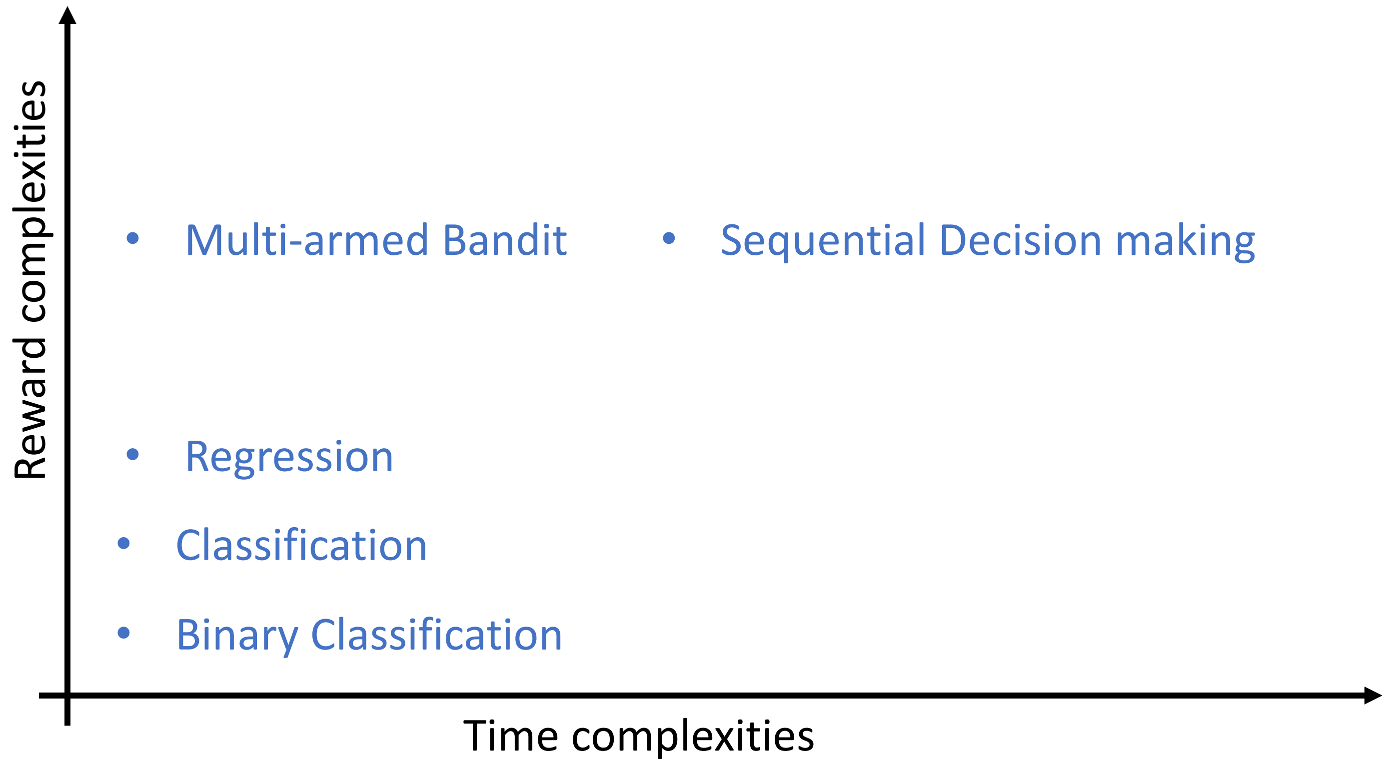
Reward Shaping

- Puck final position
- Puck moves
- Blade tip to puck distance
- Collision with the table and self
- Energy consumption
- Hitting as fast as possible
- …



$$w_{\ell_2} d_t^2 + w_{\log} \log(d_t^2 + \alpha)$$

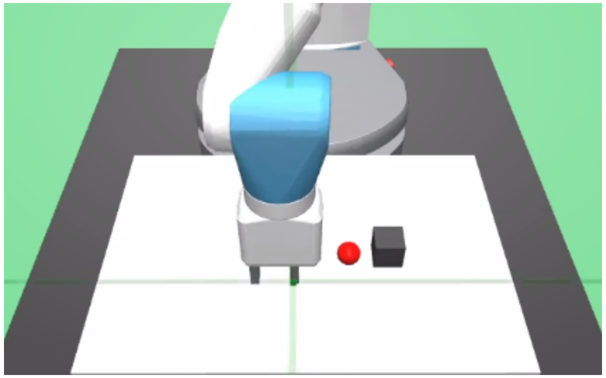
Machine Learning - Complexities

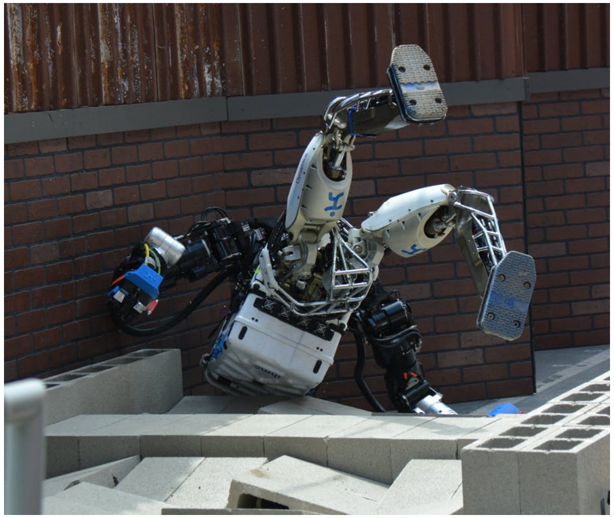


Reinforcement Learning - Challenges

- Sample efficiency
- Generalization
- Reward sparsity
- Credit assignment problem
- Safe exploration

Move the box to the target





DARPA robotic challenge

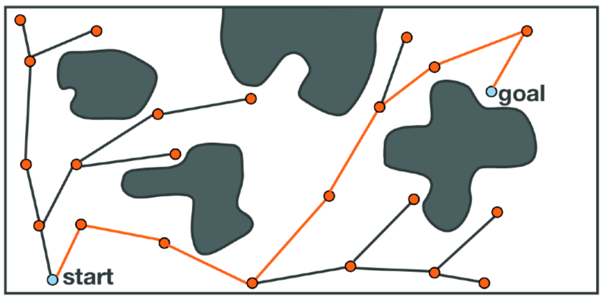
Learning Action-Selection Policies in Robotics

Today's Lecture

- Behavior Cloning
  - Feedforward Policy Training using VAE
  - Guided Policy Search
- Meta-Learning
  - Model-based RL
  - Sim-to-real transfer learning
  - Multi-objective RL
- Perception Training

Behavior Cloning

Motion Planning



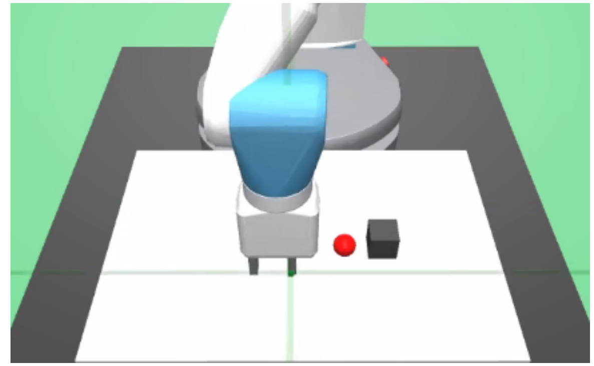
Open motion planning library

Optimal Control

$$s_{t+1} = As_t + Ba_t$$

$$J = (s_T - s^*)'Q(s_T - s^*) + \sum_t (s_t - s^*)'Q(s_t - s^*) + a_t' R a_t$$

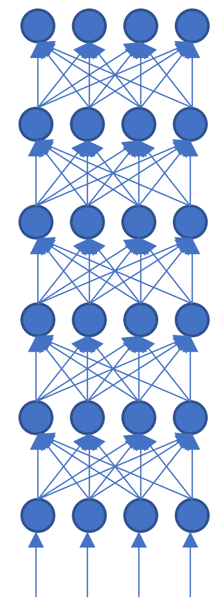
Linear Quadratic Regulator



Training Dataset

Supervised Learning

a_t

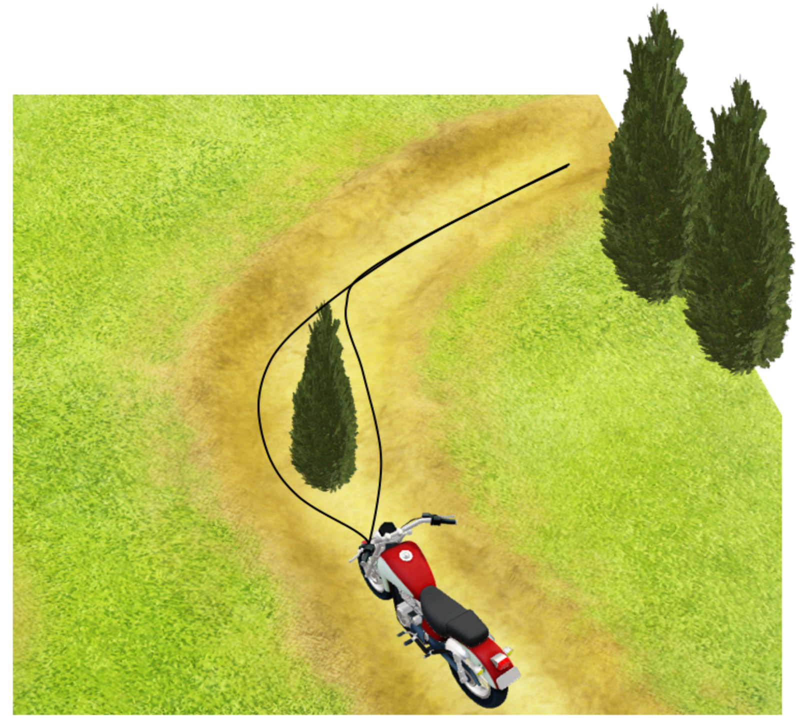


$\pi_\theta(a_t | s_t)$

s_t

Behavior Cloning - Challenges

Non-stationarity of data distribution



Inconsistency of data

Behavior Cloning – Today's Lecture

- Variational Methods for Feedforward Policy Training
- Guided Policy Search

Behavior Cloning – Variational Autoencoders

- Teleoperation
- Kinesthetic teaching
- Generic Motion Planners
- Optimal Control
- Blind controllers (trajectory shaping)

Training Dataset

$$\tau = \{u_0, u_1, \dots, u_{T-1}, r_{T-1}\}$$

trajectory of actions

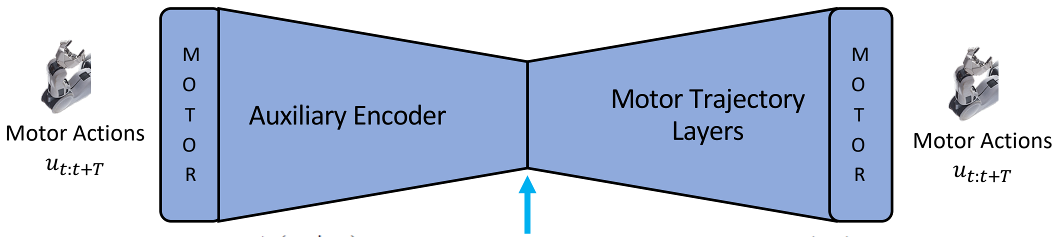
Latent Variable Generative Model

Variational Autoencoder

$$p(\tau \mid \alpha)$$

Deep predictive policy training using reinforcement learning
Ghadirzadeh, et al., IROS17.

Behavior Cloning – Feedforward Policy Training



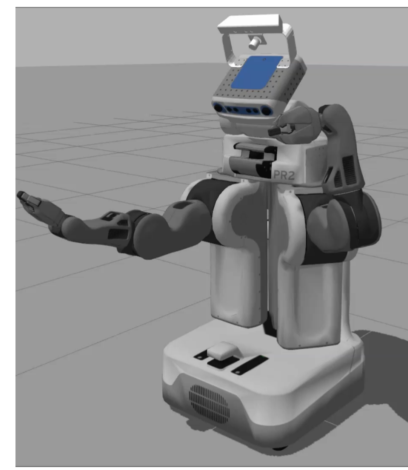
$\psi(\alpha|\tau)$ Action-manifold α $\tau \leftarrow g(\alpha)$

$$\mathcal{L}_{ae} = \sum_{i=1}^{N_\tau} |\tau_i - g(\alpha_i)|$$

Variational Auto-encoder

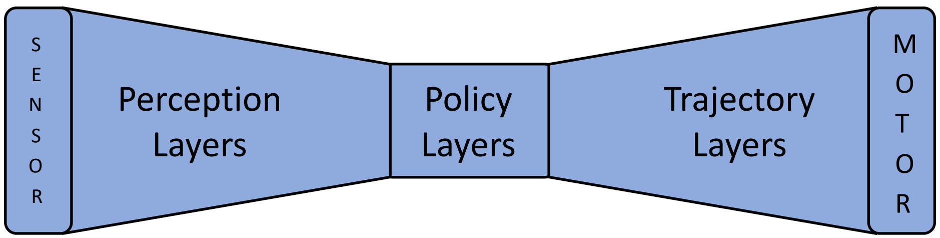
$$\mathcal{L}_{vae} = \sum_{i=1}^{N_\tau} |\tau_i - g(\alpha_i)| + D_{KL}(\,\psi(\alpha_i|\tau_i)\,||\,\mathcal{N}(0,I)\,)$$

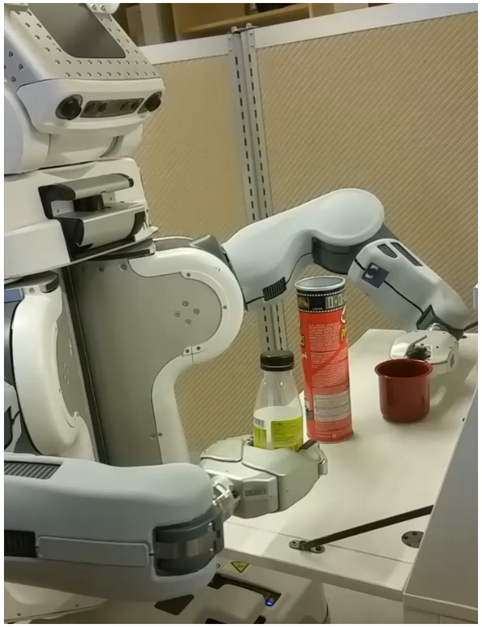
- Sampling efficiently
- Continuous mapping



A blind controller in simulation

Behavior Cloning – Feedforward Policy Training



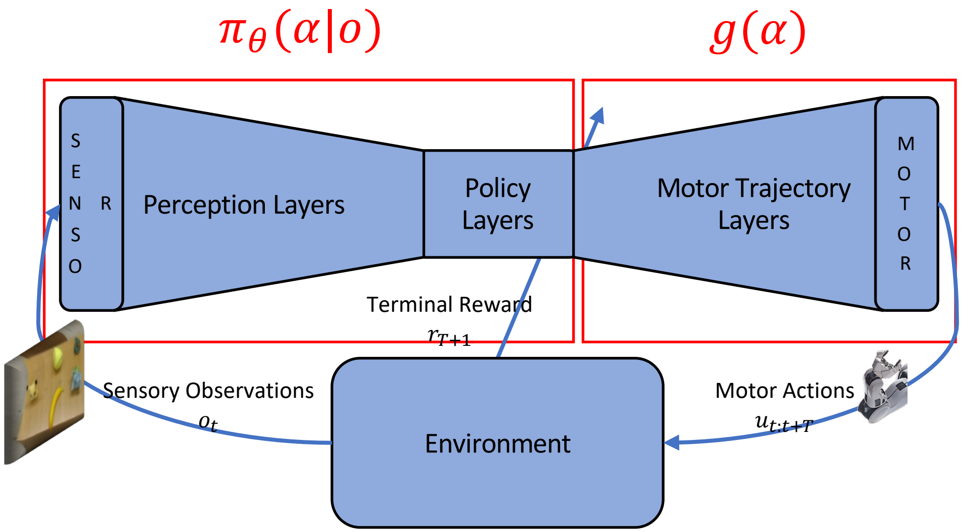


$$\log p(r|o) = \log \int p(r|o,\tau)\pi_\theta(\tau|o)d\tau$$

$$\tau = \{o, u_0, u_1, \dots, u_{T-1}, r_{T-1}\}$$

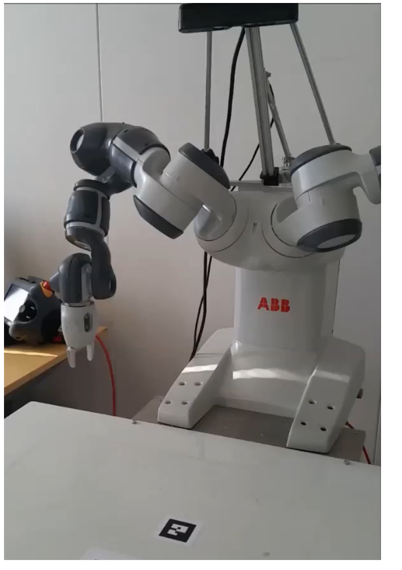
Feedforward trajectory

Behavior Cloning – Feedforward Policy Training



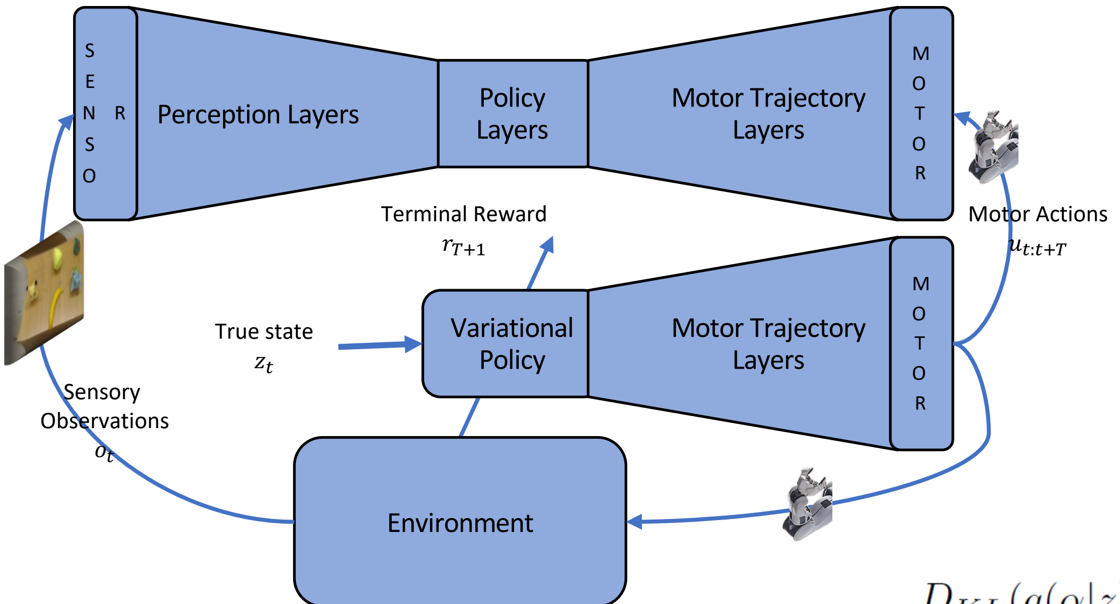
$$\log p(r|o) = \log \int p(r|o,\tau)\pi_\theta(\tau|o)d\tau$$

$$\log p(r|o) = \log \int p(\, r \,|\, o,\, g(\alpha)\,)\pi_\theta(\alpha|o)d\alpha$$



- Efficient sampling due to low-dimensionality of α
- Highly possible reward outcome
- Safe exploration
- No temporal credit assignment issue

Behavior Cloning – Feedforward Policy Training



E-step

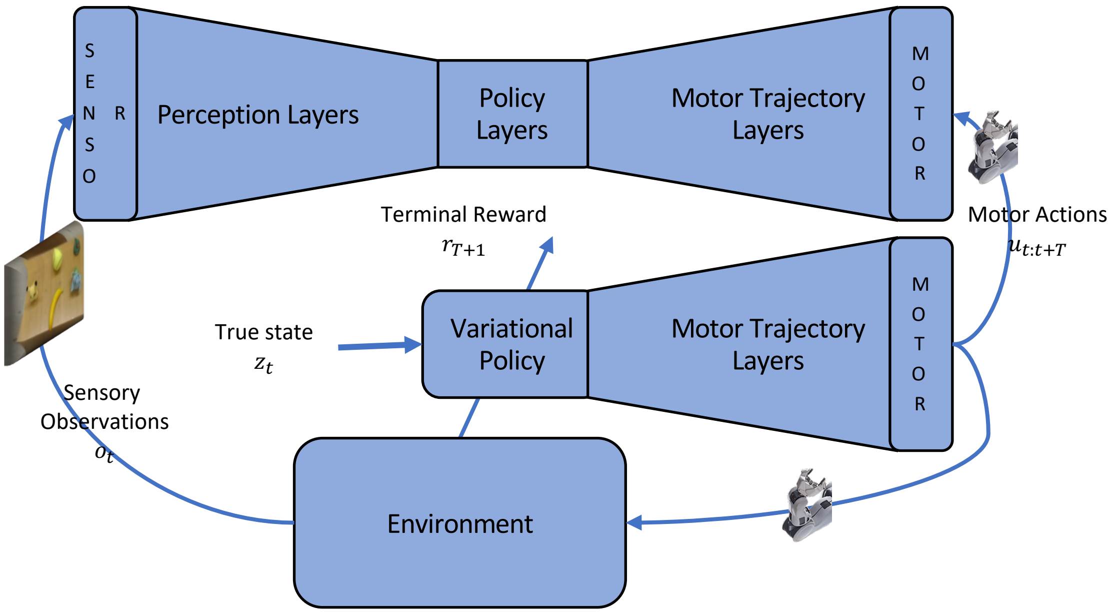
- Optimizes variational policy

$$q = \underset{q'}{\arg\min} \int q'(\alpha|z) \log \frac{q'(\alpha|z)}{p(\alpha|r,o,\theta)} d\alpha$$

$$= \underset{q'}{\arg\min} \{ \overbrace{\boxed{D_{KL}(q'(\alpha|z) \,||\, \pi_\theta(\alpha|o))}}^{\text{Trust region}} \underbrace{\boxed{- \mathbb{E}_{q(\alpha|z)}[\log p(r|\alpha, o)] \}}}_{\text{Cost averse}}$$

$$\log p(r|o,\theta) = \overbrace{\boxed{\int q(\alpha|z) \log \frac{p(r,\alpha|o,\theta)}{q(\alpha|z)} d\alpha \; (1)}}^{\text{Lower bound}} + \overbrace{\cancel{\int q(\alpha|z) \log \frac{q(\alpha|z)}{p(\alpha|r,o,\theta)} d\alpha.}}^{D_{KL}(q(\alpha|z) \,||\, p(\alpha|r,o,\theta))} \; (2)$$

Behavior Cloning – Feedforward Policy Training



M-step

- Optimizes deep policy

$$\theta = \underset{\theta'}{\operatorname{argmax}} \int q(\alpha|z) \log \frac{p(r,\alpha|o,\theta')}{q(\alpha|z)} d\alpha$$

$$= \boxed{\underset{\theta'}{\operatorname{argmin}}\, D_{KL}(q(\alpha|z) \,||\, \pi_{\theta'}(\alpha|o))}$$

Supervised learning

$$\log p(r|o,\theta) = \int q(\alpha|z) \log \frac{p(r,\alpha|o,\theta)}{q(\alpha|z)} d\alpha \; 1 + \int q(\alpha|z) \log \frac{q(\alpha|z)}{p(\alpha|r,o,\theta)} d\alpha. \; 2$$

Behavior Cloning – Feedforward Policy Training

Trust region term

1. Update q

$$q = \underset{q'}{\operatorname{argmin}} \{ D_{KL}(q'(\alpha|z) \,||\, \pi_\theta(\alpha|o)) - \mathbb{E}_{q(\alpha|z)}[\log p(r|\alpha, o)] \}$$

Cost averse term

Input remapping trick

2. Update π_θ

$$\theta = \underset{\theta'}{\operatorname{argmin}} \, D_{KL}(q(\alpha|z) \,||\, \pi_{\theta'}(\alpha|o))$$

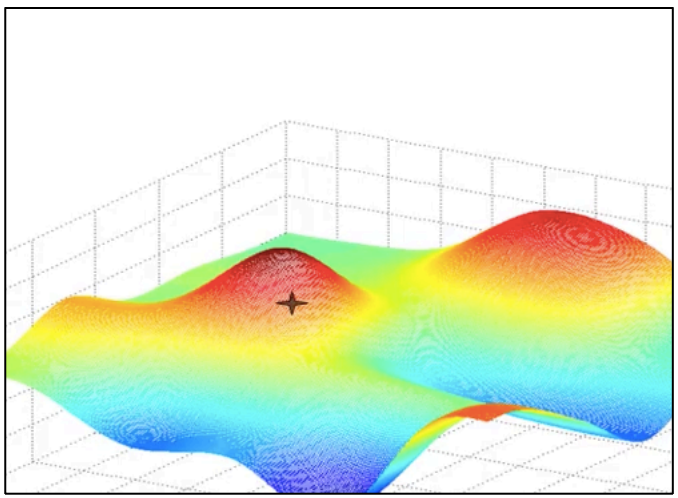
Behavior Cloning – Feedforward Policy Training

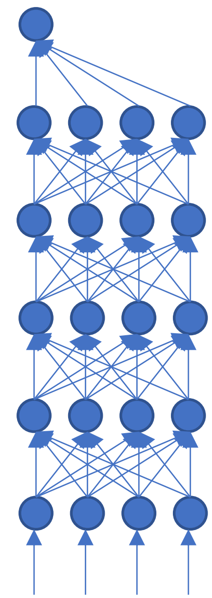
Update q such that

$\mathbb{E}_{q(\alpha|z)}[\log p(r|z,\alpha)]$

is maximized

Reward probability $p(r|z,\alpha)$

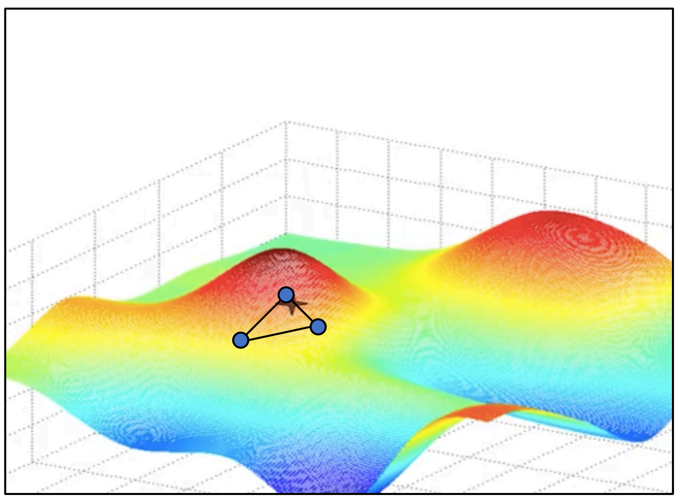


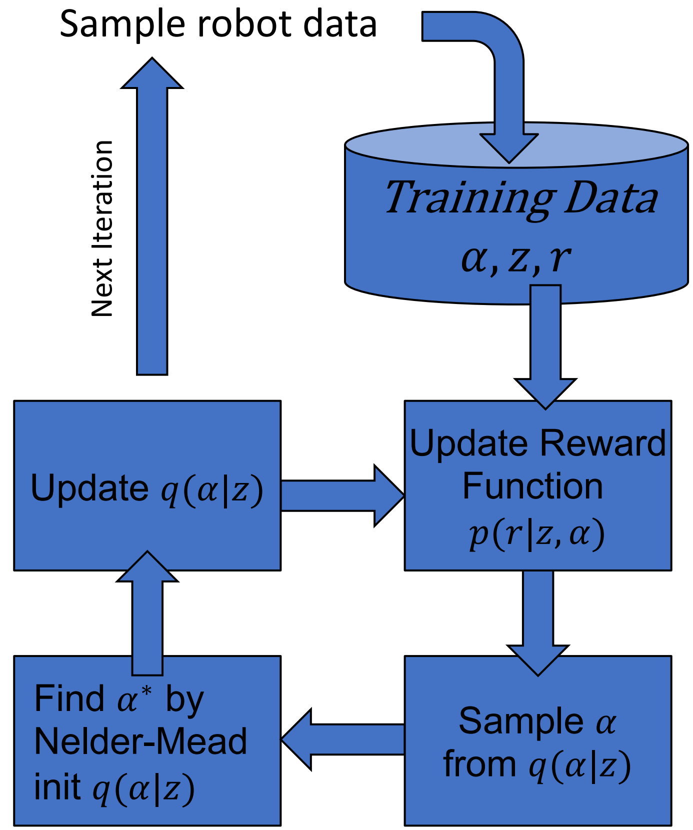


Behavior Cloning – Feedforward Policy Training

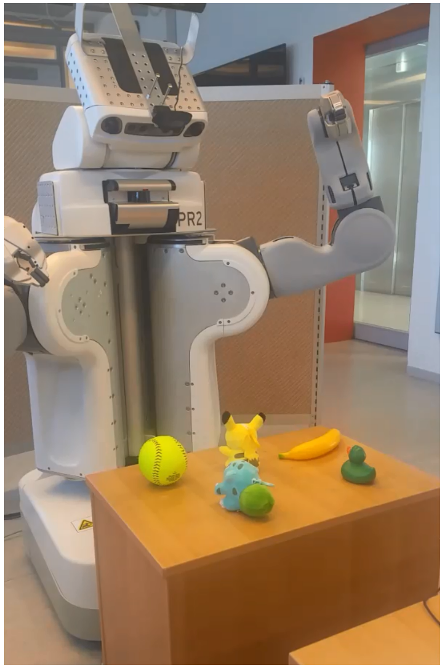
1. Get initial α by sampling $q(\alpha|z)$
2. Find $\alpha^* = argmax_\alpha \log p(r|\alpha, z)$
3. Update q to increase loglikelihood of $\{\alpha^*, z\}$

Reward probability $p(r|z, \alpha)$





Behavior Cloning – Feedforward Policy Training



Behavior Cloning

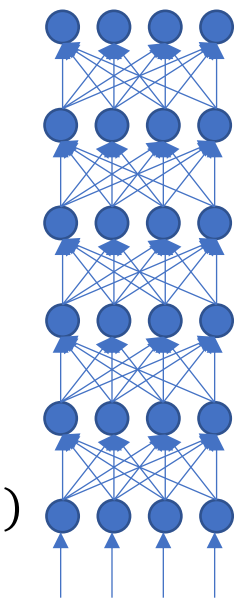
Optimal Control

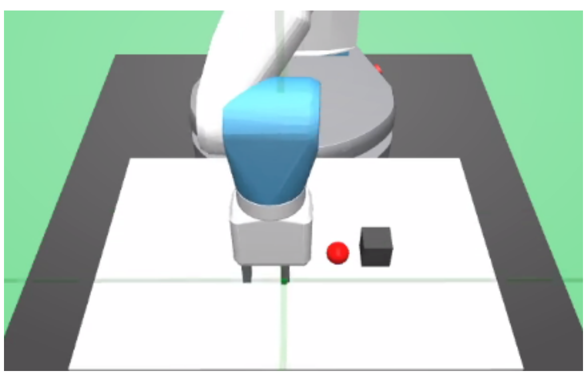
$$\mathrm{x_{t+1} = Ax_t + Bu_t}$$

$$J = (x_T - x^*)'Q(x_T - x^*) + \sum_t (s_t - s^*)'Q(s_t - s^*) + a_t' R a_t$$

Linear Quadratic Regulator

Training Dataset





Behavior Cloning – Guided Policy Search

$$\min_{\tau,\theta} J(\tau) \qquad s.t. \qquad u_t = \pi_\theta(x_t)$$

Finding the trajectory, $\boldsymbol{\tau = \{x_0, u_0, \ldots, x_{T-1}, u_{T-1}\}}$ and the policy $\boldsymbol{\pi_\theta}$ such that the **objective function** is minimized

Can be solved by **Dual Gradient Descent**

Dual Gradient Descent - Review

Goal $\min f(x) \quad s.t. \quad C(x) = 0$

- Construct the Lagrangian $\mathcal{L}(x, \lambda) = f(x) + \lambda C(x)$
- Construct the dual Lagrange function $g(\lambda) = \mathcal{L}(x^*, \lambda)$
- Repeat the followings:
  - Obtain $x^* \leftarrow \arg\min_x \mathcal{L}(x, \lambda)$
  - Compute $\frac{dg}{d\lambda} = \frac{d\mathcal{L}(x^*,\lambda)}{d\lambda}$
  - $\lambda \leftarrow \lambda + \alpha \frac{dg}{d\lambda}$

Behavior Cloning – Guided Policy Search

$$\min_{\tau,\theta} J(\tau) \qquad s.t. \qquad u_t = \pi_\theta(x_t) \quad \forall t$$

$$\min_{\tau,\theta} J(\tau) \qquad s.t. \qquad \sum_t u_t - \pi_\theta(x_t) = 0$$

$$\mathcal{L}(\tau,\theta,\lambda) = J(\tau) + \lambda(\sum_t u_t - \pi_\theta(x_t)) \qquad \tau_\lambda^*, \theta_\lambda^* = \arg\min_{\tau',\theta'} \mathcal{L}(\tau',\theta',\lambda)$$

$$g(\lambda) = \mathcal{L}(\tau_\lambda^*, \theta_\lambda^*, \lambda)$$

Lagangian

Lagrange Dual Function $\dfrac{dg(\lambda)}{d\lambda} = \dfrac{d\mathcal{L}(\tau_\lambda^*, \theta_\lambda^*, \lambda)}{d\lambda}$

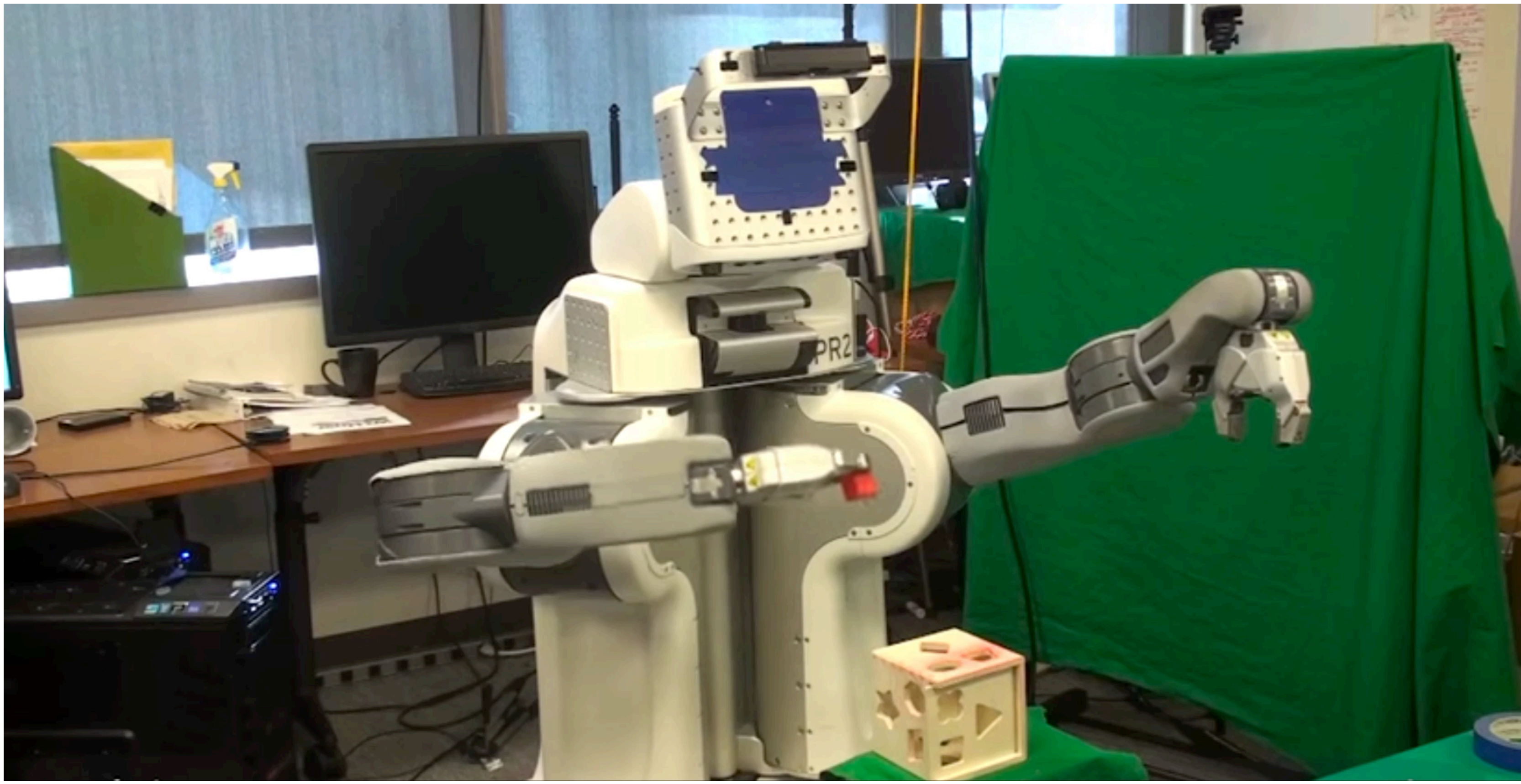
Behavior Cloning – Guided Policy Search

$$\min_{\tau,\theta} J(\tau) \qquad s.t. \qquad u_t = \pi_\theta(x_t) \quad \forall t$$

$$J(\tau) = (x_T - x^*)' Q (x_T - x^*) + \sum_t (x_t - x^*)' Q (x_t - x^*) + u_t' R u_t$$

- Construct the Lagrangian $\mathcal{L}(\tau, \theta, \lambda) = J(\tau) + \lambda(\sum_t u_t - \pi_\theta(x_t))$
- Construct the dual Lagrange function $g(\lambda) = \mathcal{L}(\tau_\lambda^*, \theta_\lambda^*, \lambda)$
- Repeat
  - $\tau \leftarrow \arg\min_{\tau'} \mathcal{L}(\tau', \theta, \lambda)$ Trajectory Optimization
  - $\theta \leftarrow \arg\min_{\theta'} \mathcal{L}(\tau, \theta', \lambda)$ Supervised Learning
  - Compute $\frac{dg}{d\lambda}$
  - $\lambda \leftarrow \lambda + \alpha \frac{dg}{d\lambda}$

Behavior Cloning – Guided Policy Search



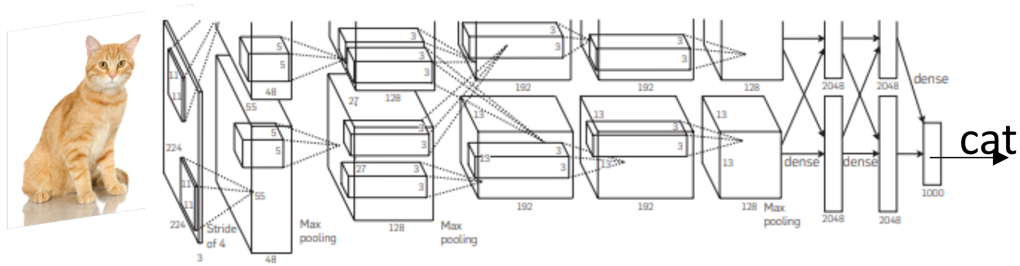
End-to-End Training of Deep Visuomotor Policies
Levine et al.

Learning Action-Selection Policies in Robotics

Today's Lecture

- Behavior Cloning
  - Feedforward Policy Training using VAE
  - Guided Policy Search
- Meta-Learning
  - Model-based RL
  - Sim-to-real transfer learning
  - Multi-objective RL
- Perception Training

Meta-Learning   Learn to Learn



ImageNet pretrained model → Add some more layers → Collect training data (new task) → Fine-tune/train end-to-end

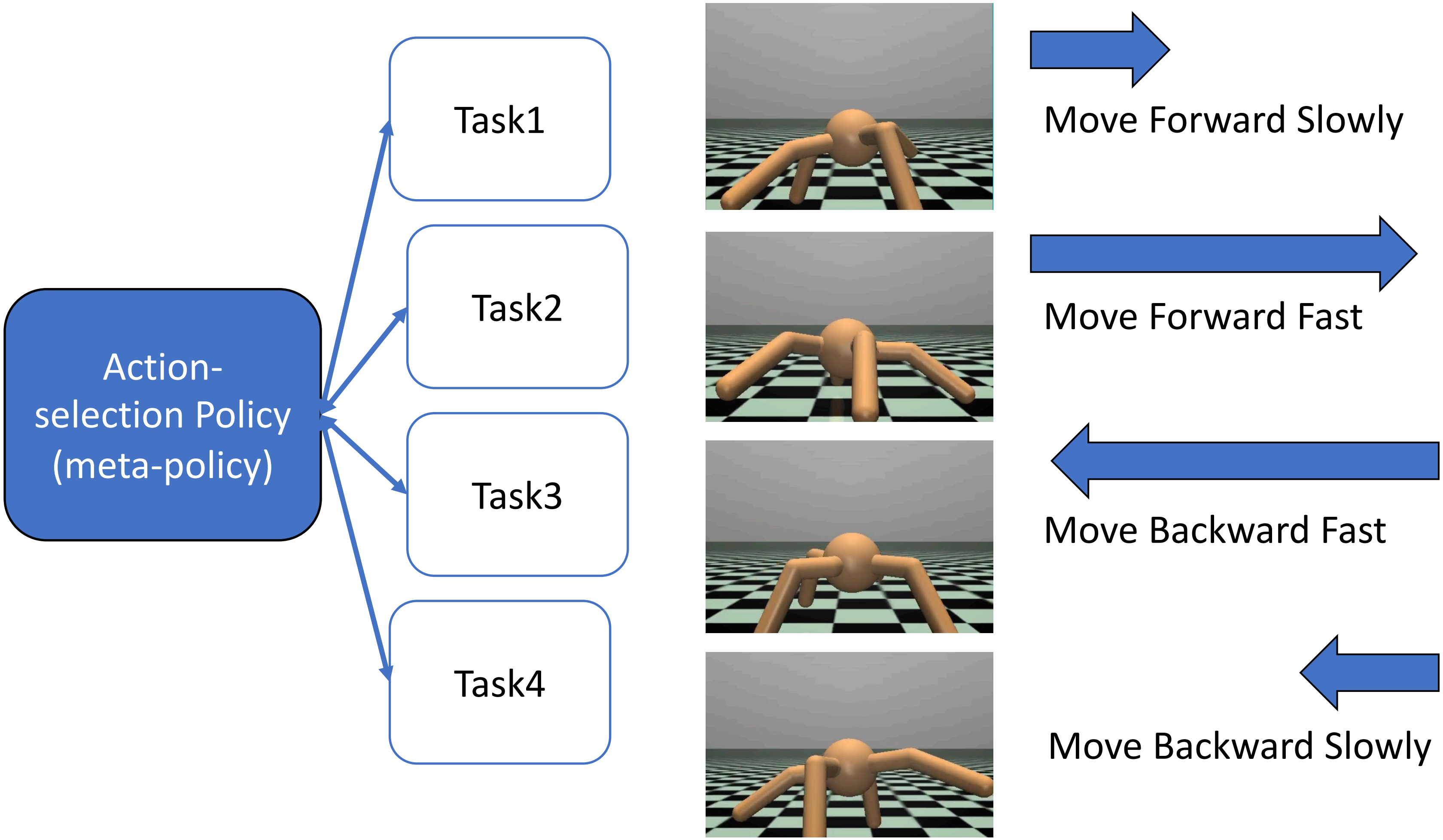
---

Meta-learner → Learner 1, Learner 2, Learner n

Learner 1	Task 1
Learner 2	Task 2
Learner n	Task n

Meta-Learning

Learn to Learn



Model-Agnostic Meta-Learning

Assuming K different tasks, the objective is:

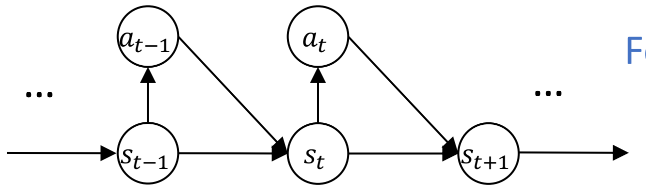
$$\max_{\boldsymbol{\theta}} \quad \frac{1}{K}\sum_{k=0}^{K} J_k(\boldsymbol{\theta}'_k) \qquad \text{s.t.:} \quad \boldsymbol{\theta}'_k = \boldsymbol{\theta} + \alpha\, \nabla_{\boldsymbol{\theta}} J_k(\boldsymbol{\theta})$$

where,

$$J_k(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{a}_t \sim \pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}\left[\sum_{t=0}^{H-1} r(\boldsymbol{s}_t, \boldsymbol{a}_t)\right]$$

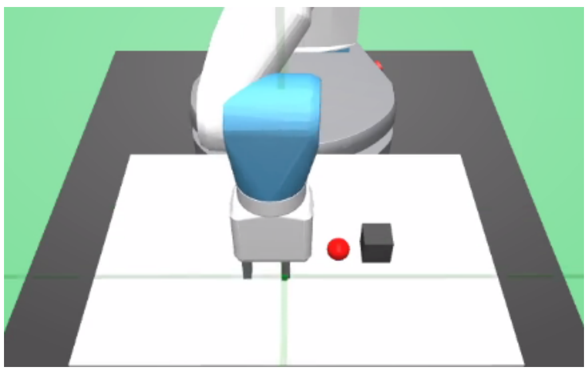
Meta-Learning

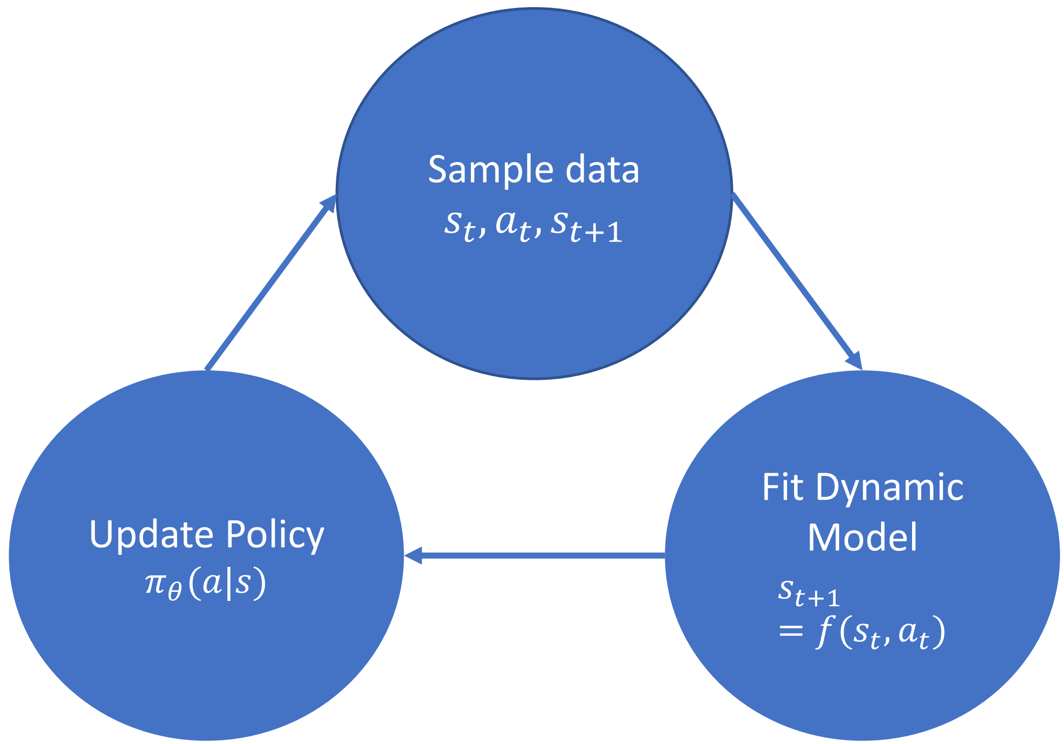
Robust Model-Based RL



Forward Dynamic Model

$$s_{t+1} = f(s_t, a_t)$$

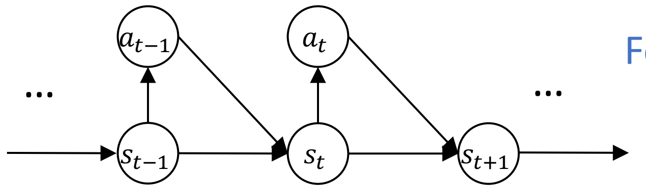




model bias problem

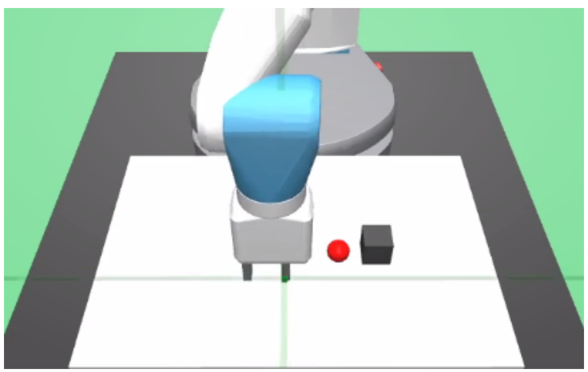
Meta-Learning

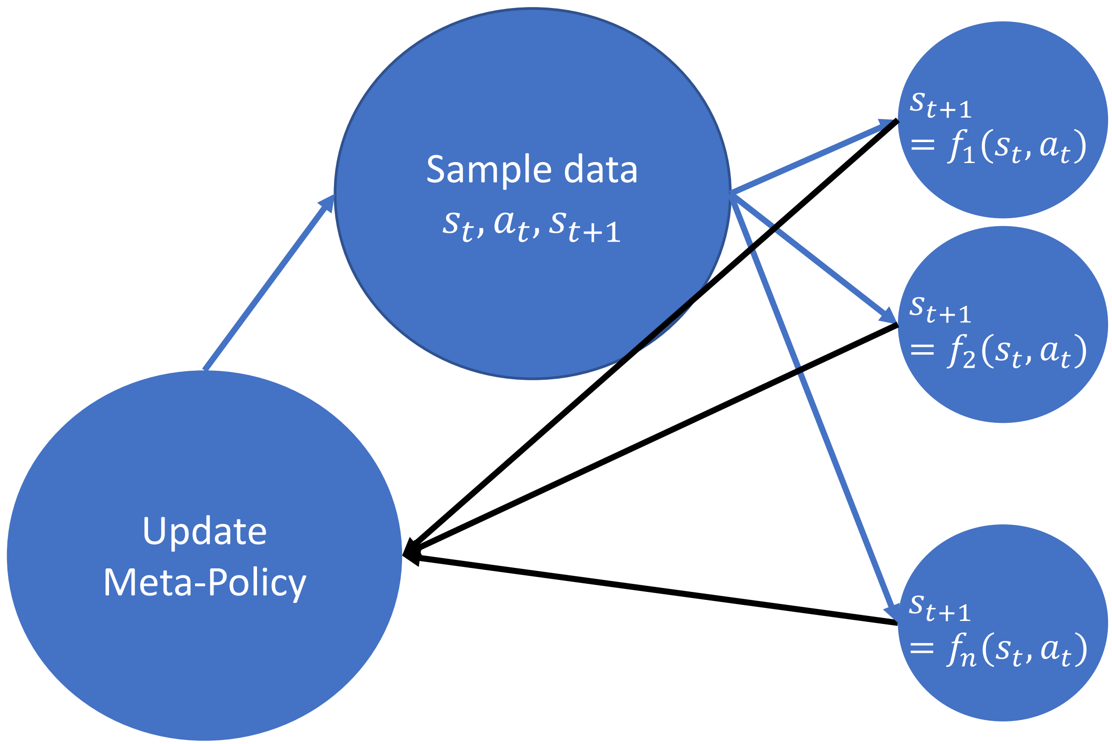
Robust Model-Based RL



Forward Dynamic Model

$$s_{t+1} = f(s_t, a_t)$$





$$J_k(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{a}_t \sim \pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}\left[\sum_{t=0}^{H-1} r(\boldsymbol{s}_t, \boldsymbol{a}_t)\middle|\boldsymbol{s}_{t+1} = \hat{f}_{\phi_k}(\boldsymbol{s}_t, \boldsymbol{a}_t)\right]$$

KTH VETENSKAP OCH KONST

Aalto University School of Engineering

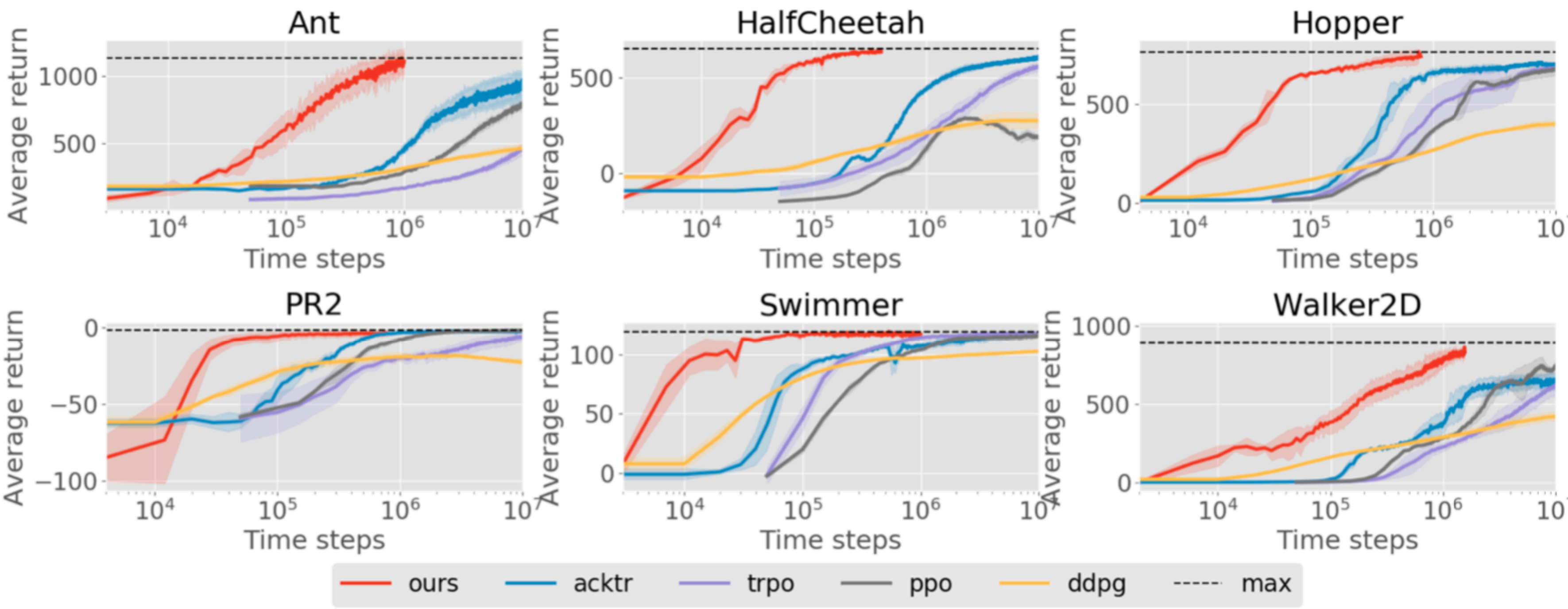
Meta-Learning Robust Model-Based RL

- Sample data from the real environment using adapted policies $\pi_{\theta_1}, \pi_{\theta_2}, \dots, \pi_{\theta_k}$
- Update $f_{\phi_1}, \dots, f_{\phi_k}$
- For every model f_{ϕ_i}
  - Sample imaginary data using meta-policy π_θ
  - Update π_{θ_i} using the data, $\theta_i' = \theta + \alpha \nabla_\theta J_i(\theta)$
  - Sample imaginary data from f_{ϕ_i} using $\pi_{\theta_i'}$
- Update meta-policy with the imaginary data

$$\boldsymbol{\theta} \rightarrow \boldsymbol{\theta} - \beta \, \frac{1}{K} \sum_k \nabla_{\boldsymbol{\theta}} J_k(\boldsymbol{\theta}_k')$$

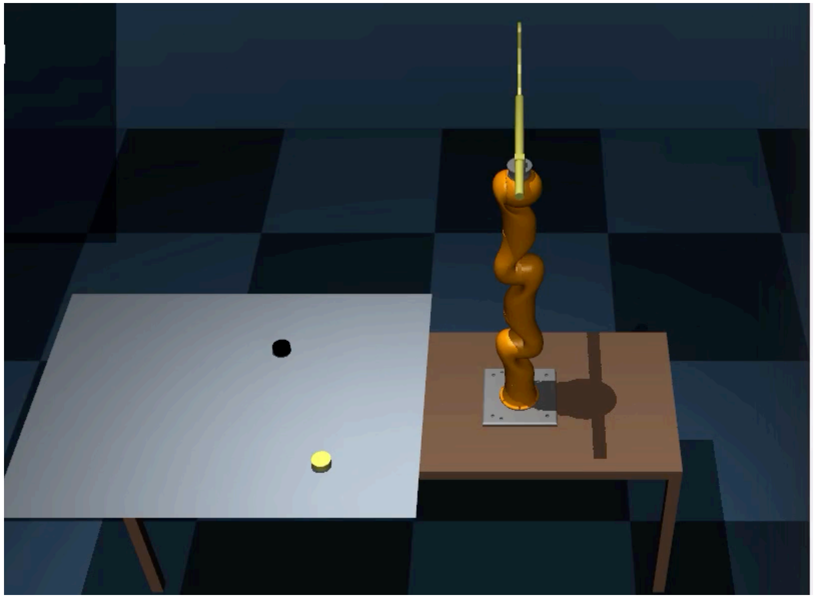
Model-Based Reinforcement Learning via Meta-Policy Optimization
Clavera et al, 2018.

Meta-Learning Robust Model-Based RL



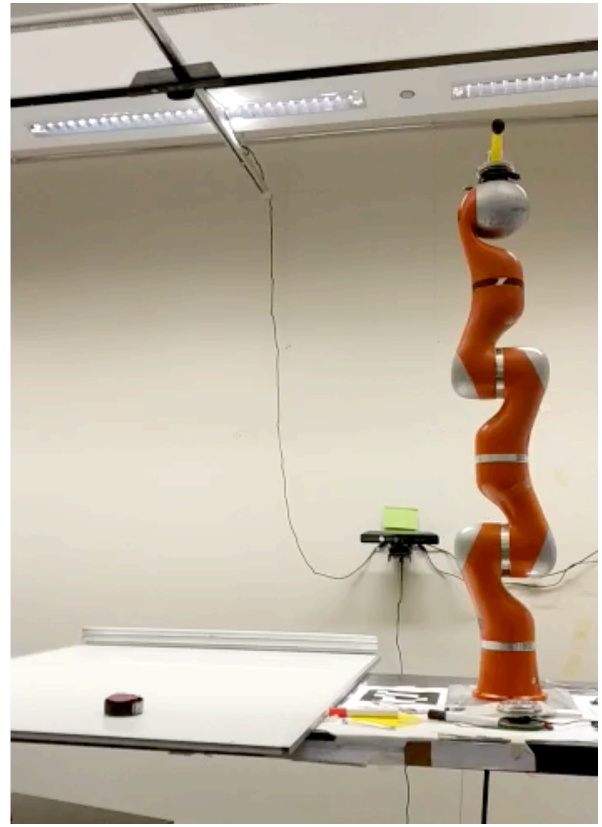
Model-Based Reinforcement Learning via Meta-Policy Optimization
Clavera et al, 2018.

Meta-Learning



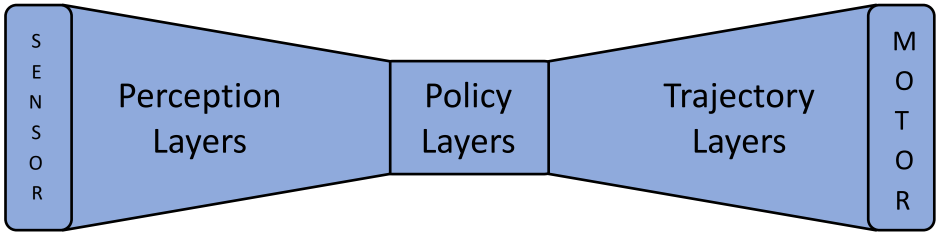
- Discrepancies in system dynamics
- Differences in the robot controllers
- Different sources of noise and uncertainty

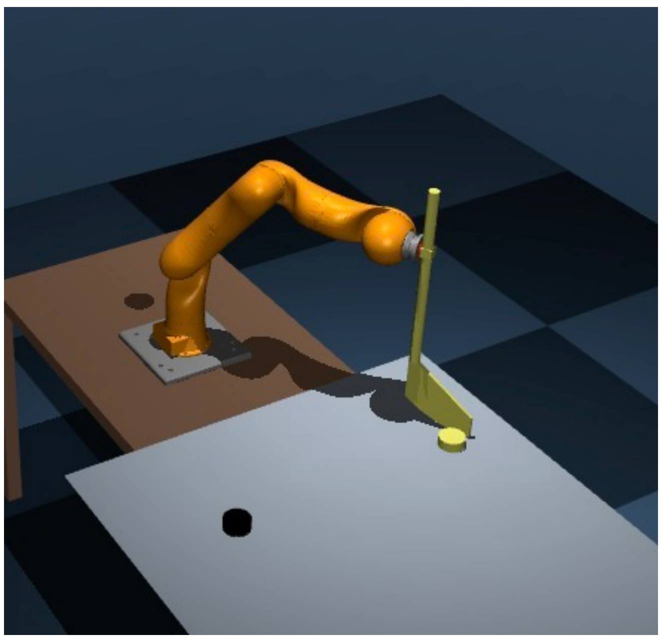
Sim-to-Real Transfer

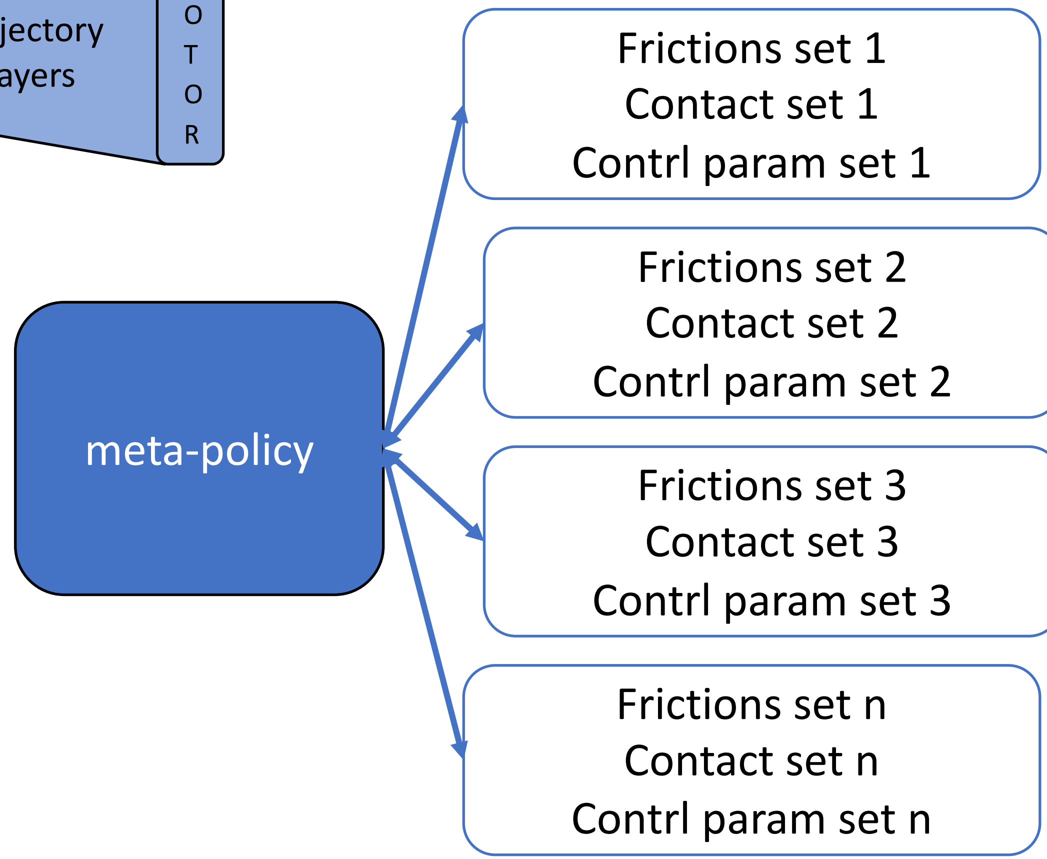


Meta-Learning

Sim-to-real Transfer



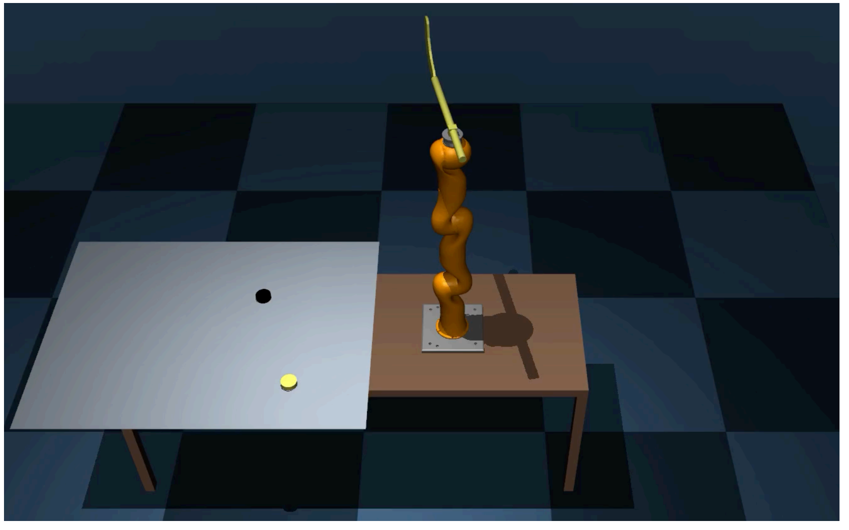




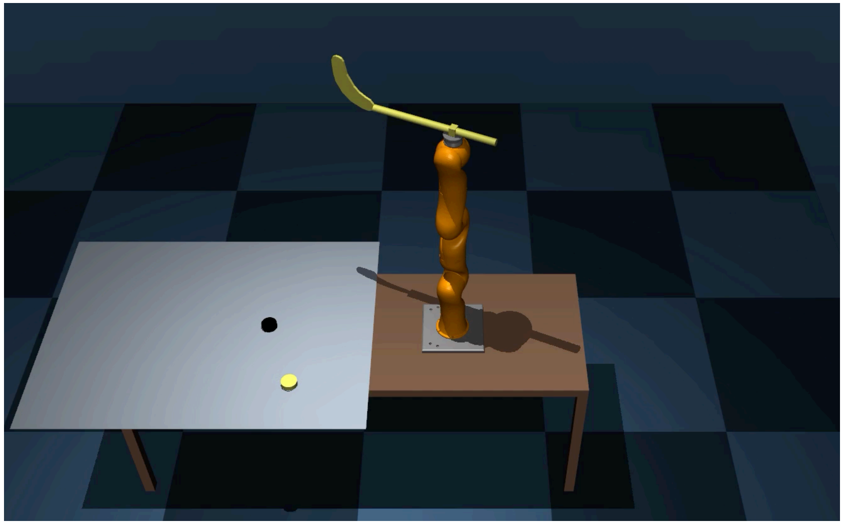
Domain Randomization

Arndt, Ghadirzadeh, Hazara, Kyrki
ICRA20

Meta-Learning Sim-to-Real Transfer



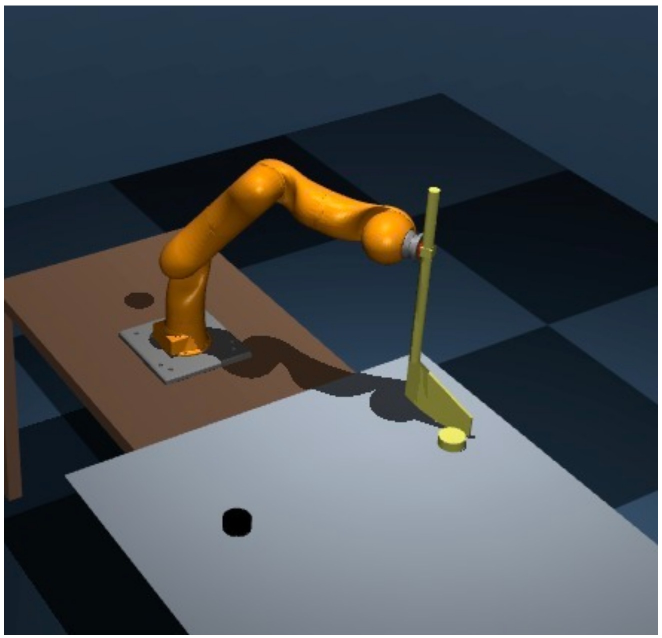
Before adaptation (meta-policy)

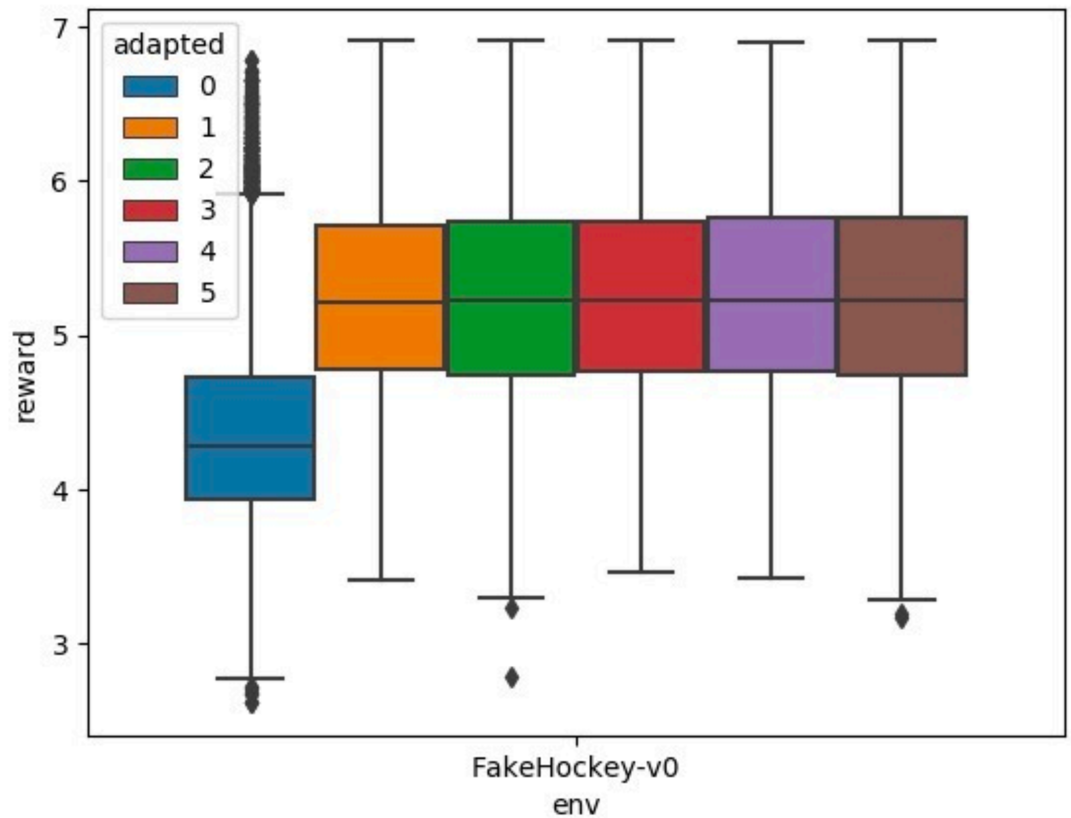


After single adaptation

Meta-Learning

Sim-to-Real Transfer

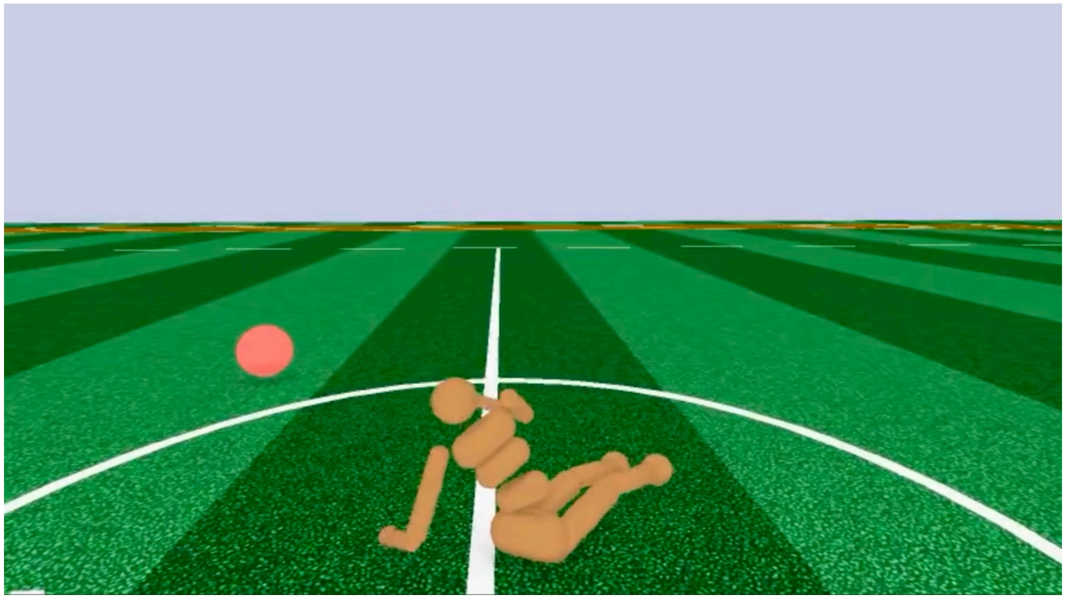




Arndt, Ghadirzadeh, Hazara, Kyrki
ICRA20

Meta-Learning

Multi-Objective RL

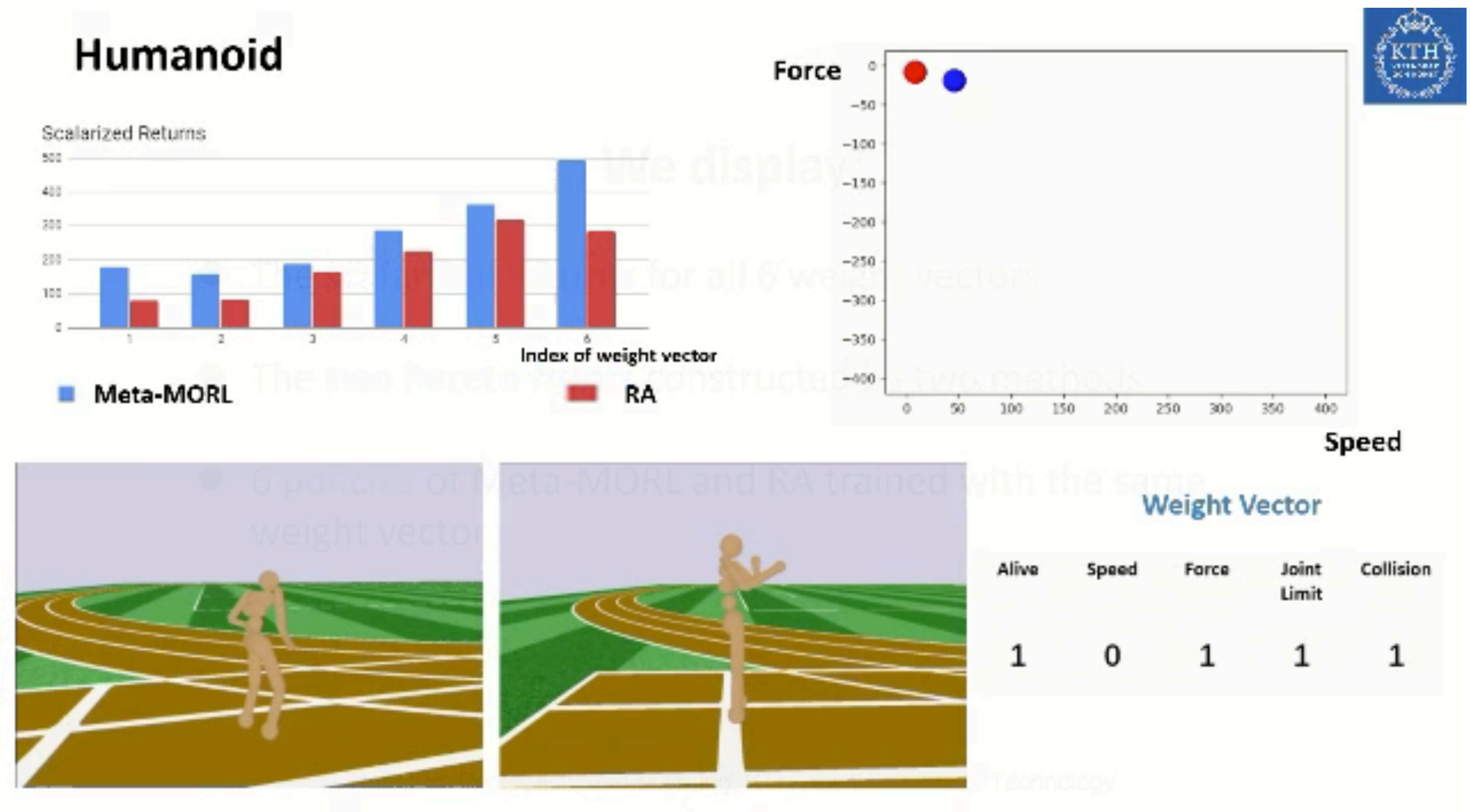


+ Stay upright
+ Forward speed
- Energy consumption
- Joint limit violation
- Collision

$$r = f(\sum \omega_i \, r_i)$$

Meta-Learning

Multi-Objective RL

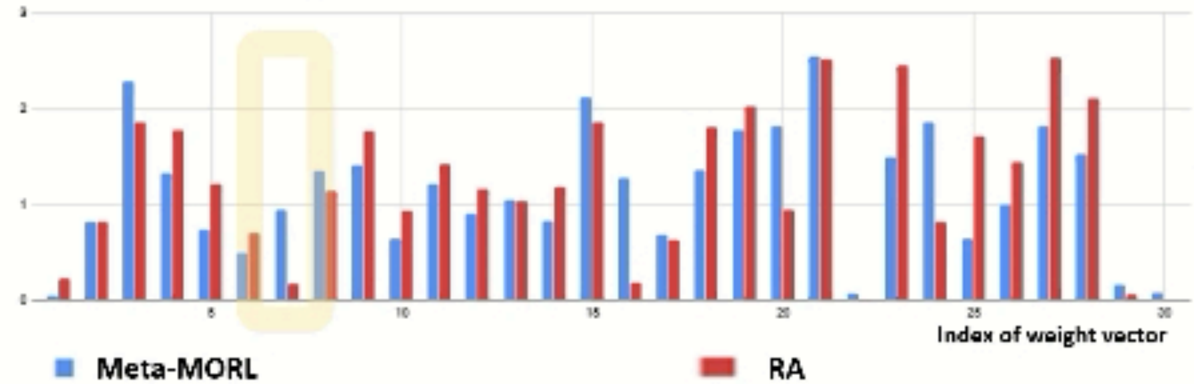


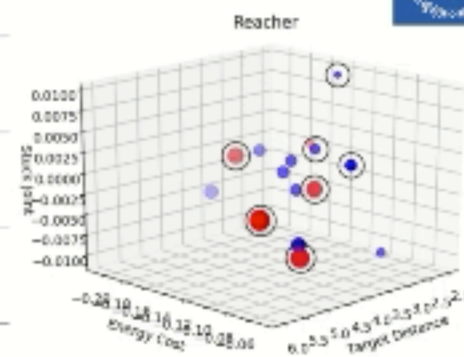
Meta-Learning for Multi-objective Reinforcement Learning
Chen, Ghadirzadeh, Bjorkman and Jensfelt, IROS19

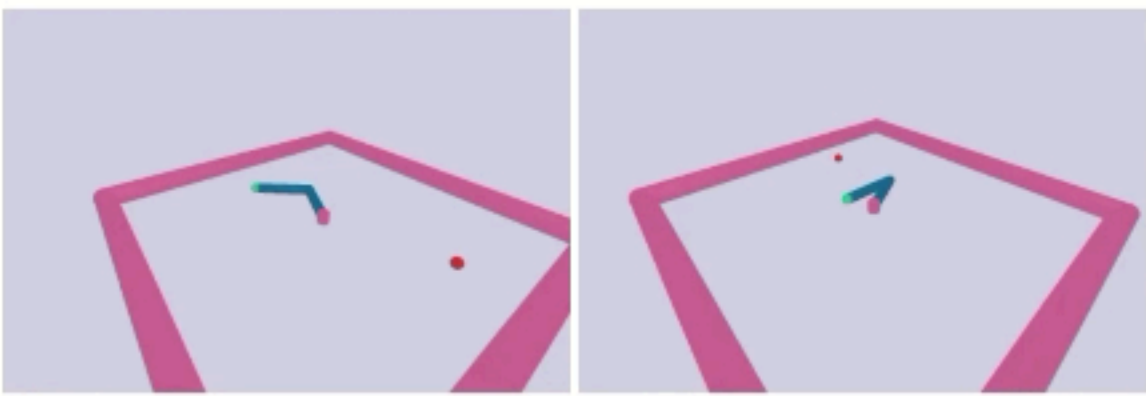
Meta-Learning Multi-Objective RL

Reacher

Scalarized returns of Reacher using 30 random weight vectors







Weight Vector

Target Distance	Energy Cost	Stuck Joint
0.61	0.94	0.22

Meta-Learning for Multi-objective Reinforcement Learning
Chen, Ghadirzadeh, Bjorkman and Jensfelt, IROS19

Meta-Learning Multi-Objective RL

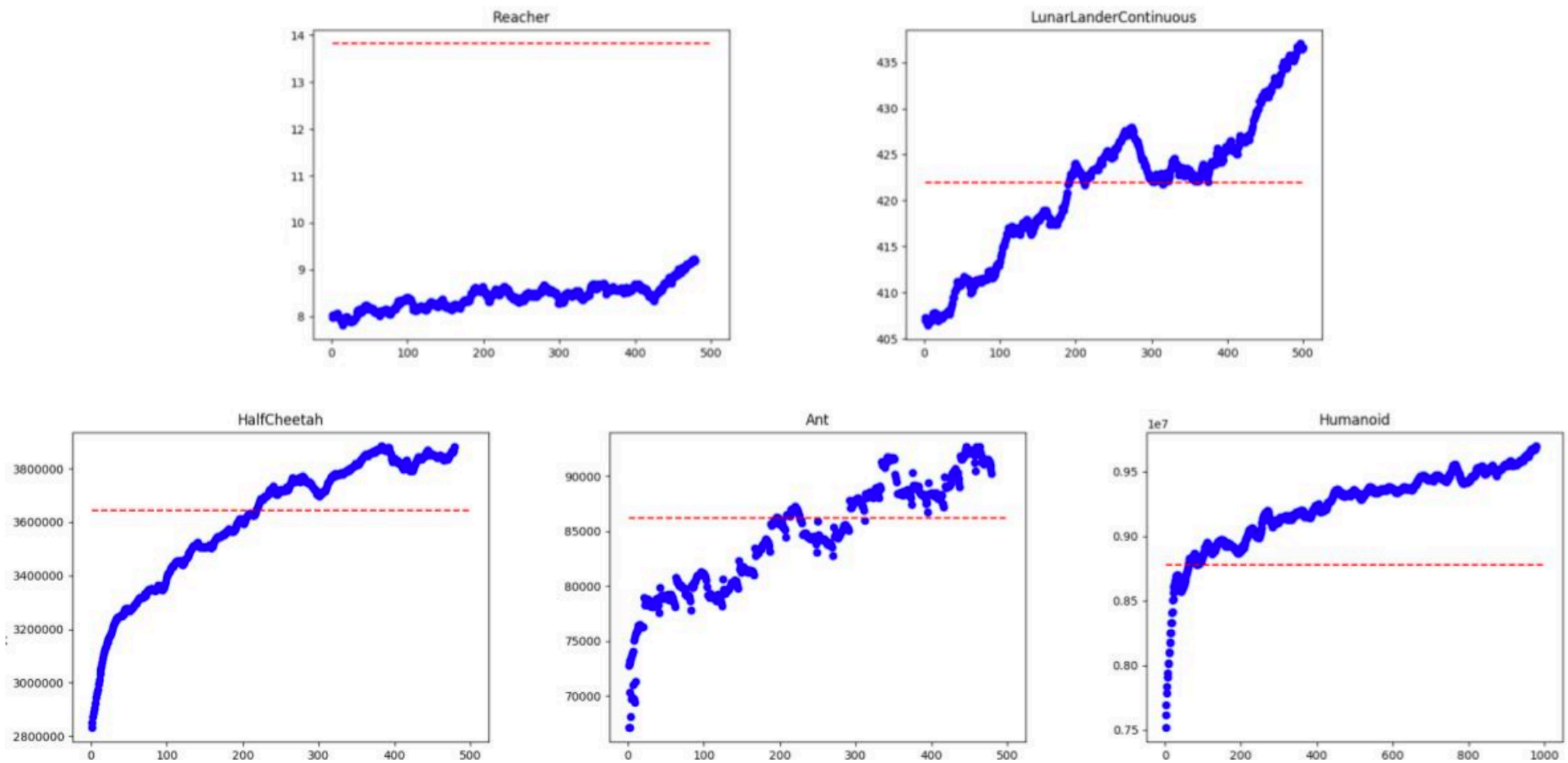


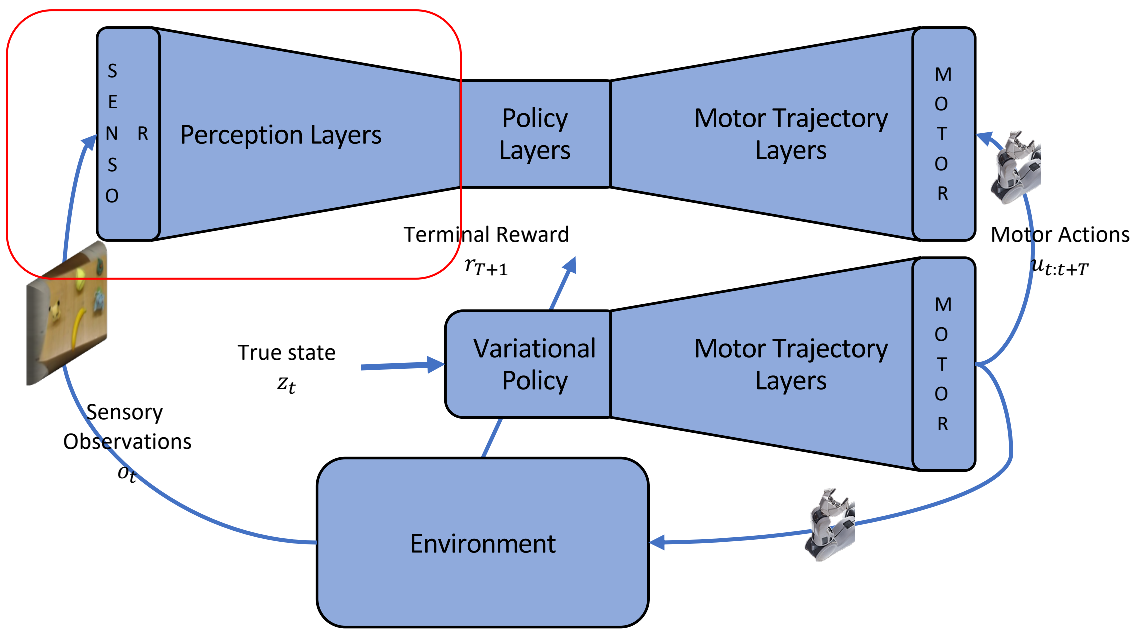
Fig. 4: The improvements of the hypervolume indicator (vertical axis) with respect to the iteration of fine-tuning the meta policy (horizontal axis). The blue curve denotes the hypervolume and the red line denotes the final hypervolume of the Pareto front estimated by RA.

Learning Action-Selection Policies in Robotics

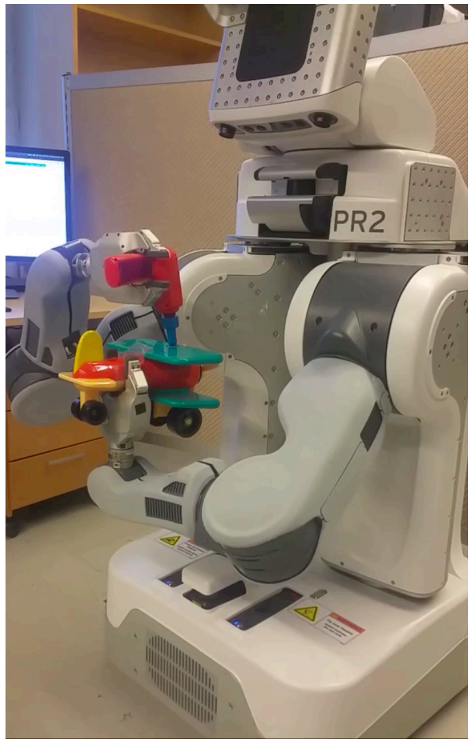
Today's Lecture

- Behavior Cloning
  - Feedforward Policy Training using VAE
  - Guided Policy Search
- Meta-Learning
  - Model-based RL
  - Sim-to-real transfer learning
  - Multi-objective RL
- Perception Training

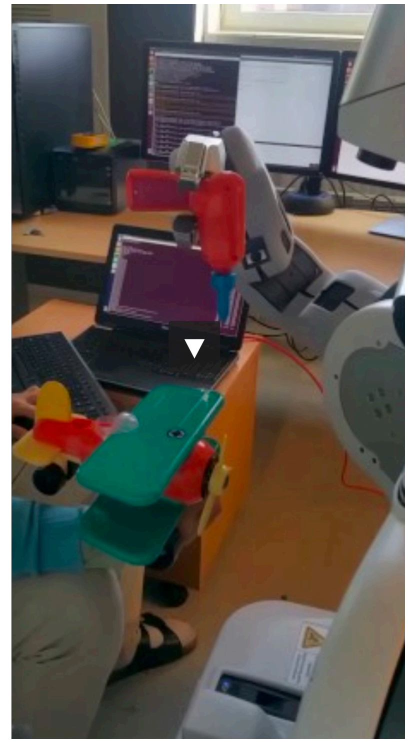
Perception Training



Input remapping trick



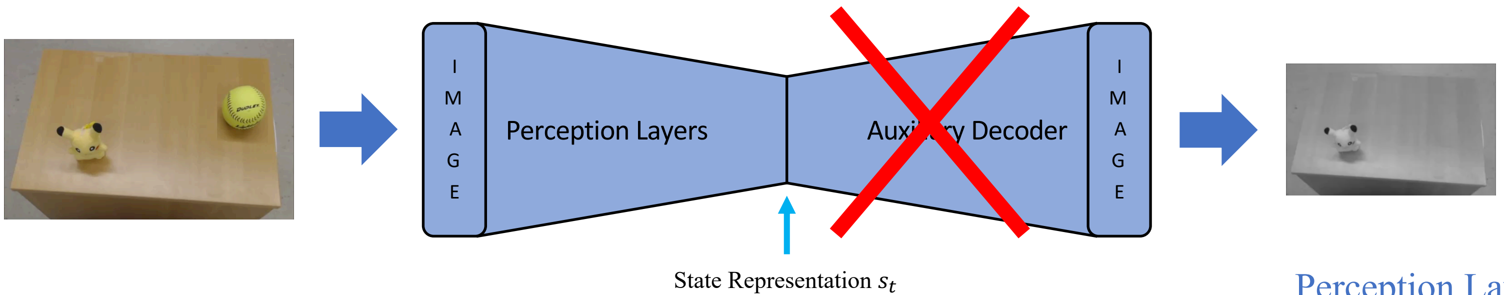
$q(\alpha|z)$

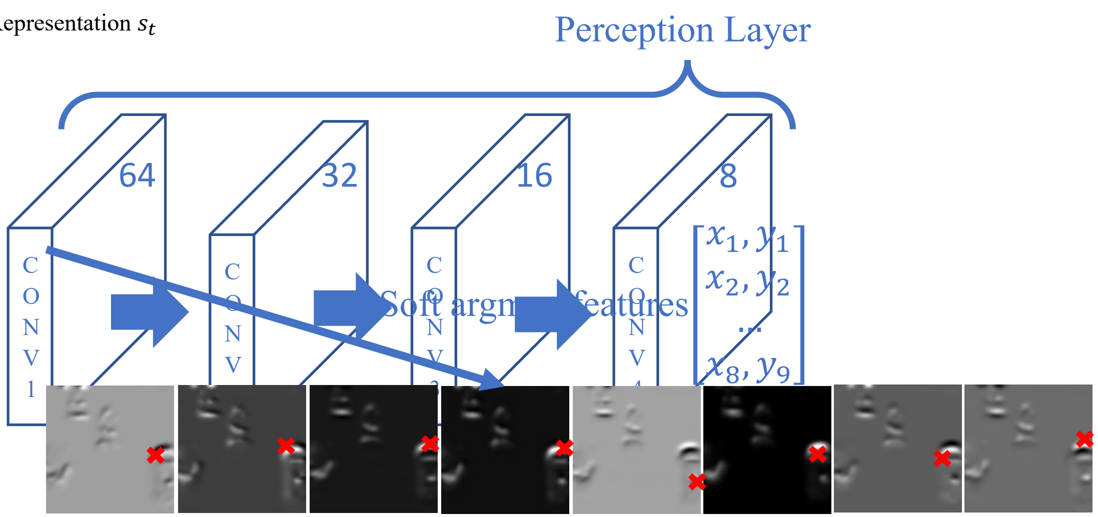


$\pi_\theta(\alpha|o)$

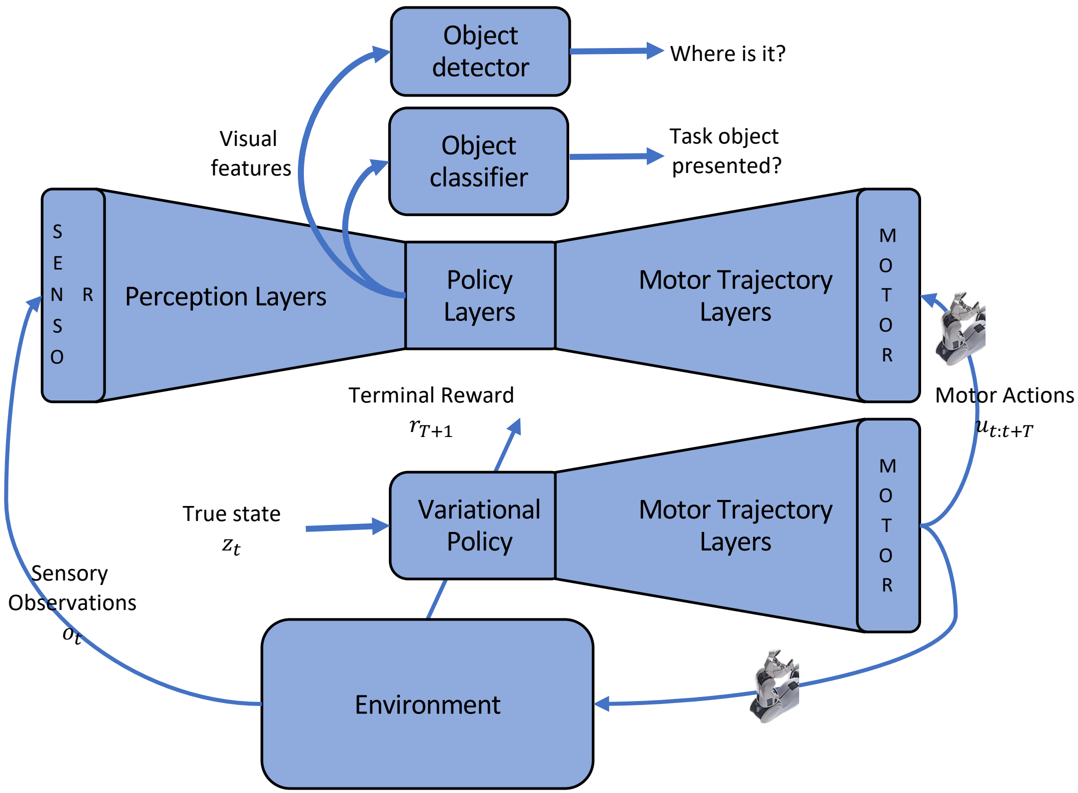
$$\theta = \underset{\theta'}{\operatorname{argmin}}\, D_{KL}(q(\alpha|z)\,||\,\pi_{\theta'}(\alpha|o))$$

Perception Training





Perception Training



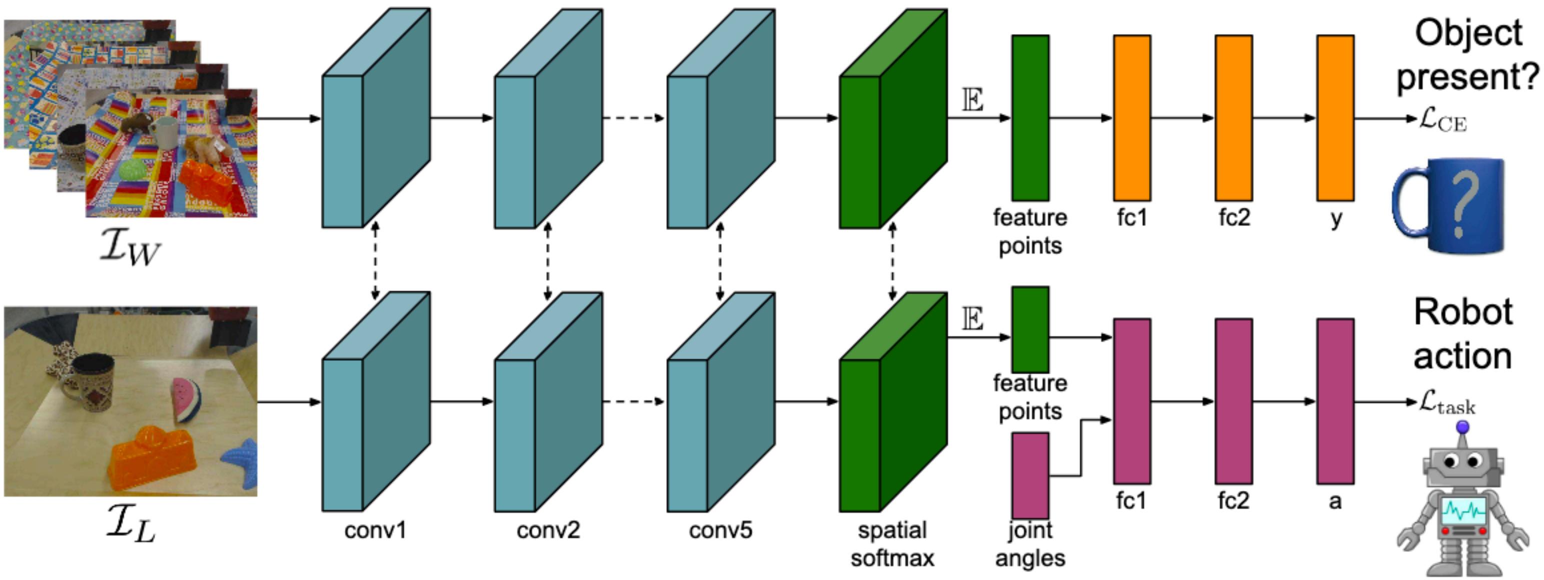
$$\mathcal{L}_M = D_{KL}(\, q(\alpha|z) \,||\, \pi_\theta(\alpha|o) \,)$$

$$\mathcal{L}_{det} = \sum_{i \in obj} \sum_{c \in cls} \sum_{p \in pos} -\mathbb{I}_{ip}^{c} \log f_{det}(o, c, p)$$

$$\mathcal{L}_{loc} = \sum_{i \in obj} \sum_{c \in cls} \sum_{p \in pos} \mathbb{I}_{ip}^{c} \,|\, f_{loc}(o, c, p) - \bar{p}_i \,|$$

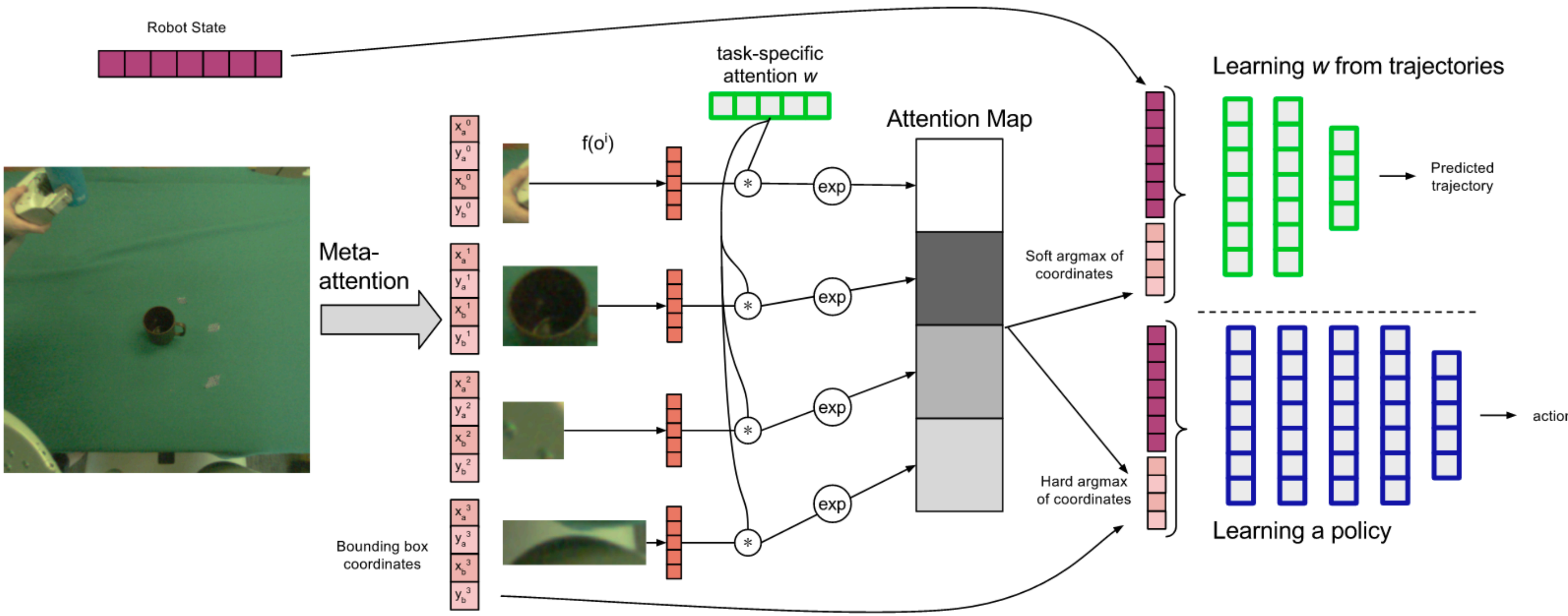
$$\theta = \underset{\theta'}{\operatorname{argmax}} \{ \mathcal{L}_M + \mathcal{L}_{loc} + \mathcal{L}_{det} \}$$

Perception Training



Singh et al., GPLAC
UC Berkeley

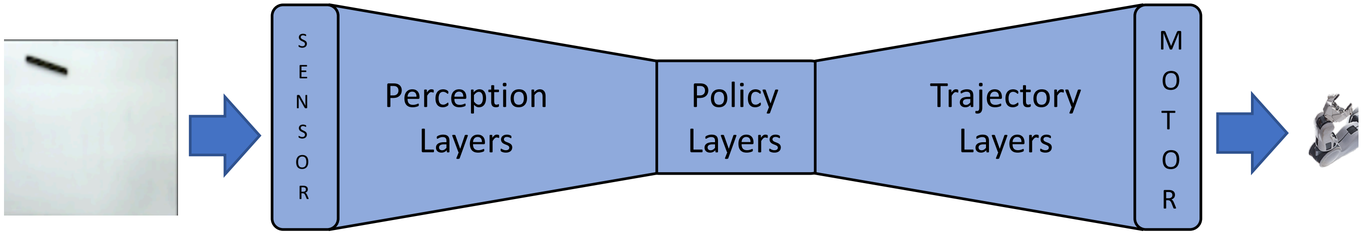
Perception Training

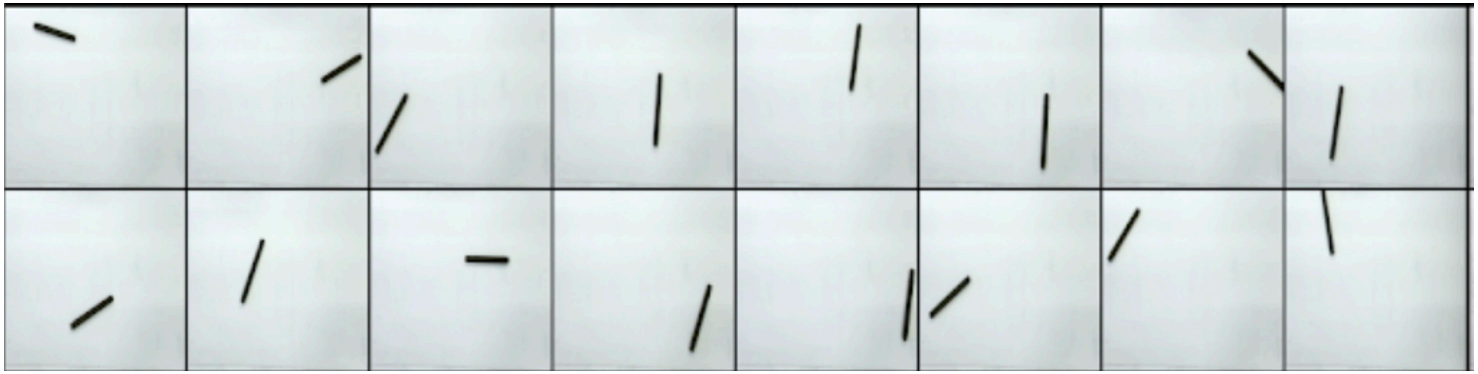


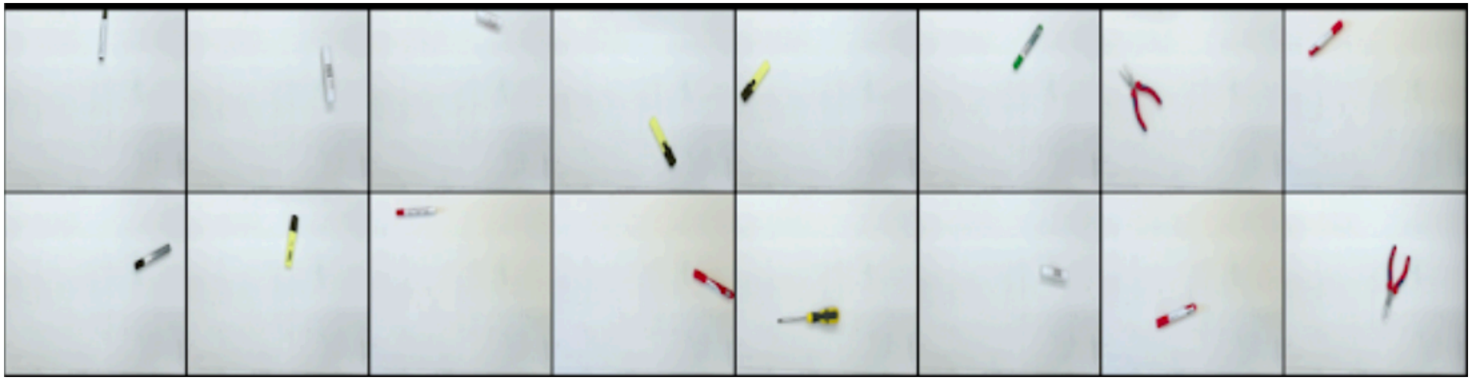
Devin et al., Deep object Centric Representations
UC Berkeley

Perception Training

Adversarial Training

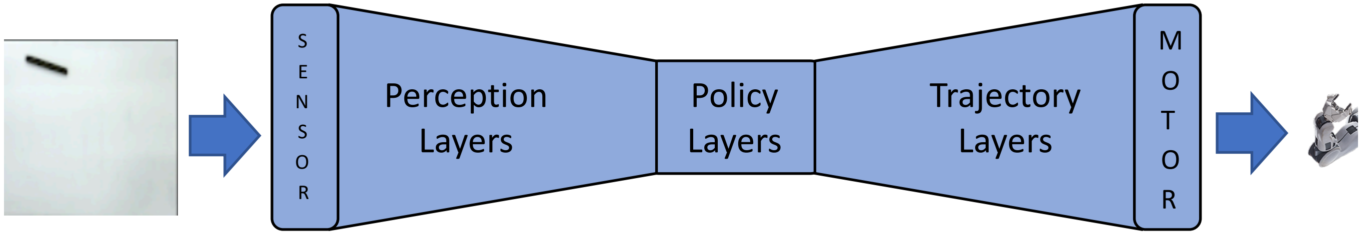






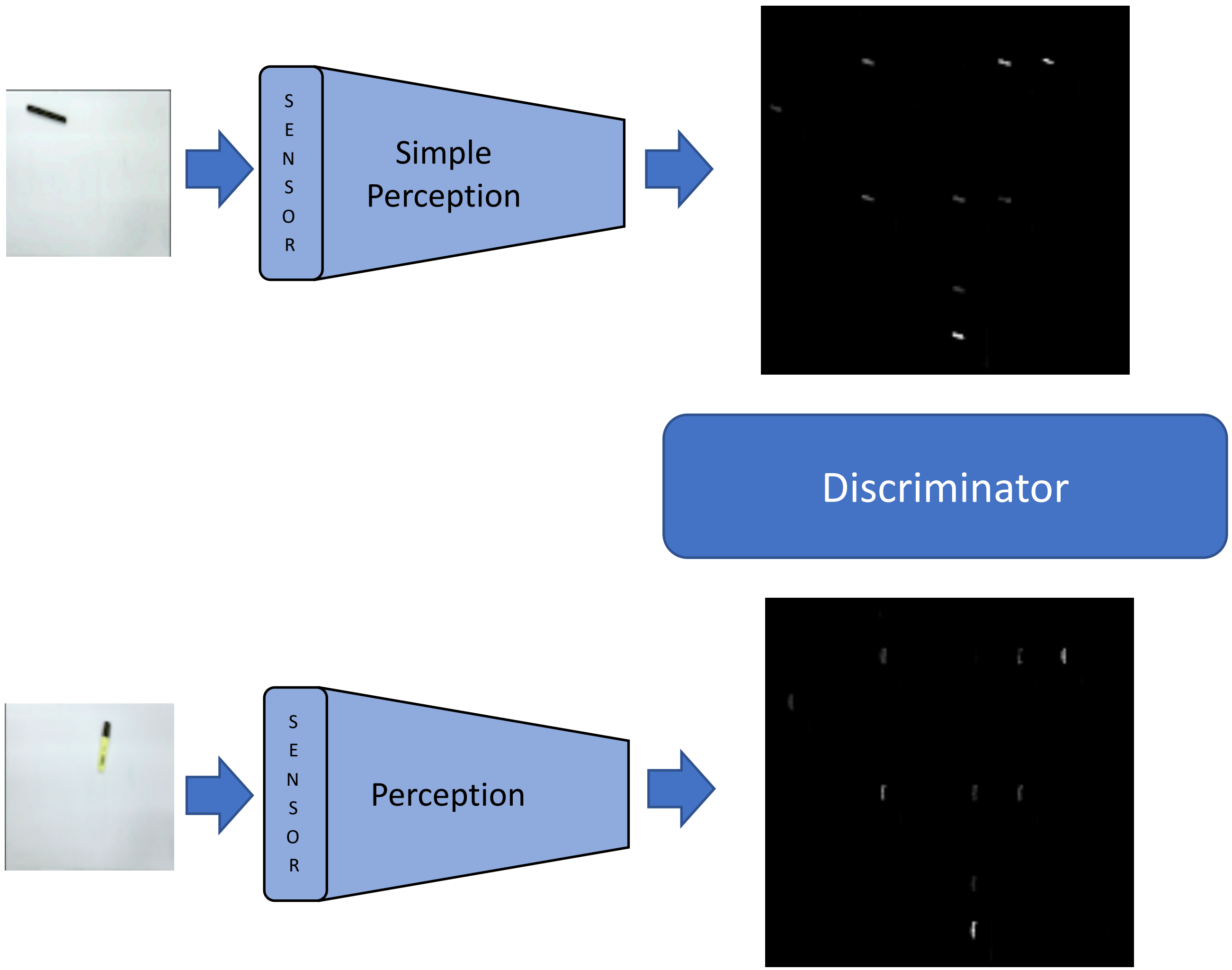
Perception Training

Adversarial Training



Perception Training

Adversarial Training



Chen, Ghadirzadeh, Bjorkman and Jensfelt