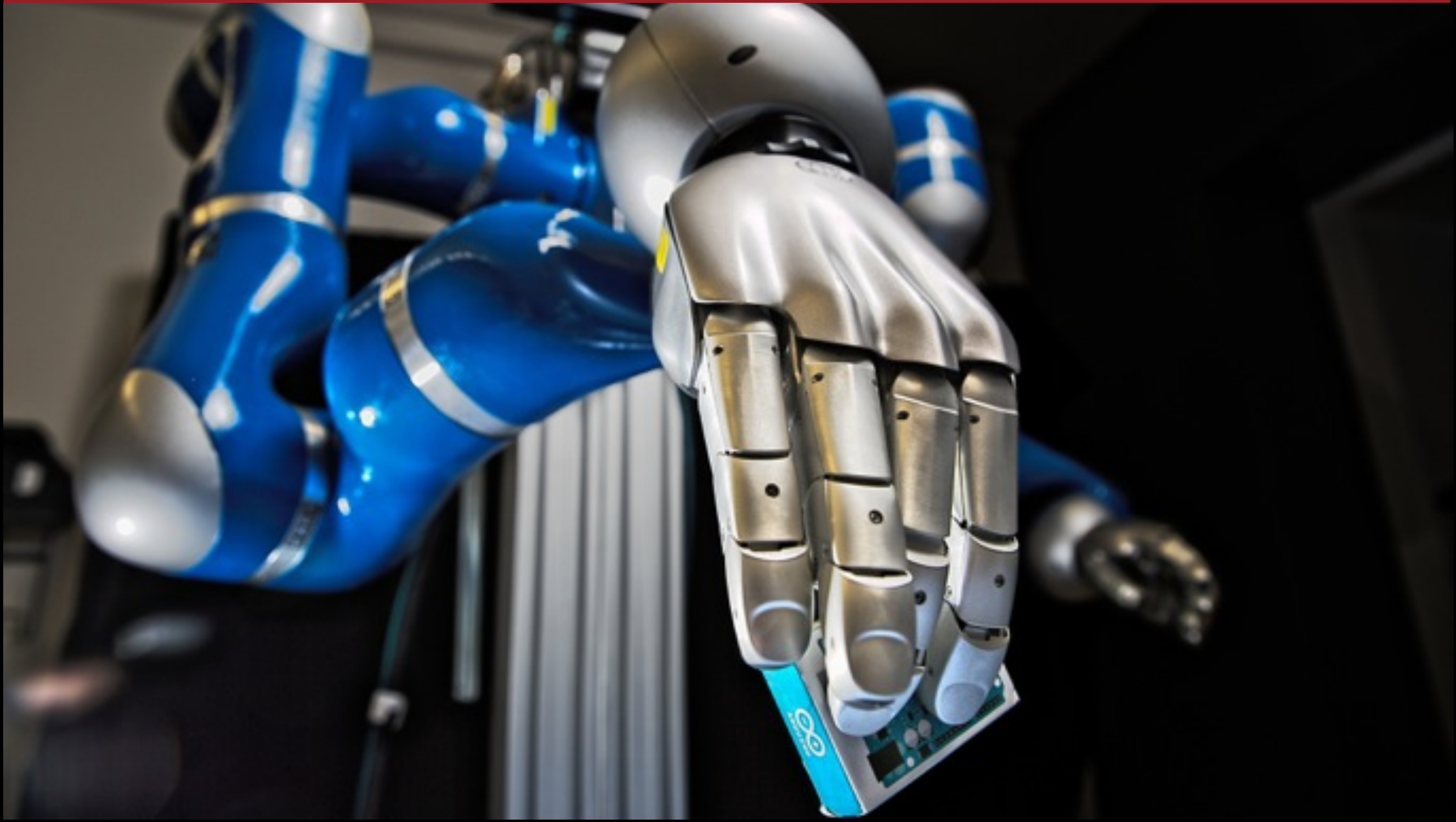
# Active Reward Learning

C. Daniel*, M. Viering*, J. Metz*, O. Kroemer*, J. Peters*†
* Technische Universität Darmstadt
† Max Planck Institut für Intelligente Systeme



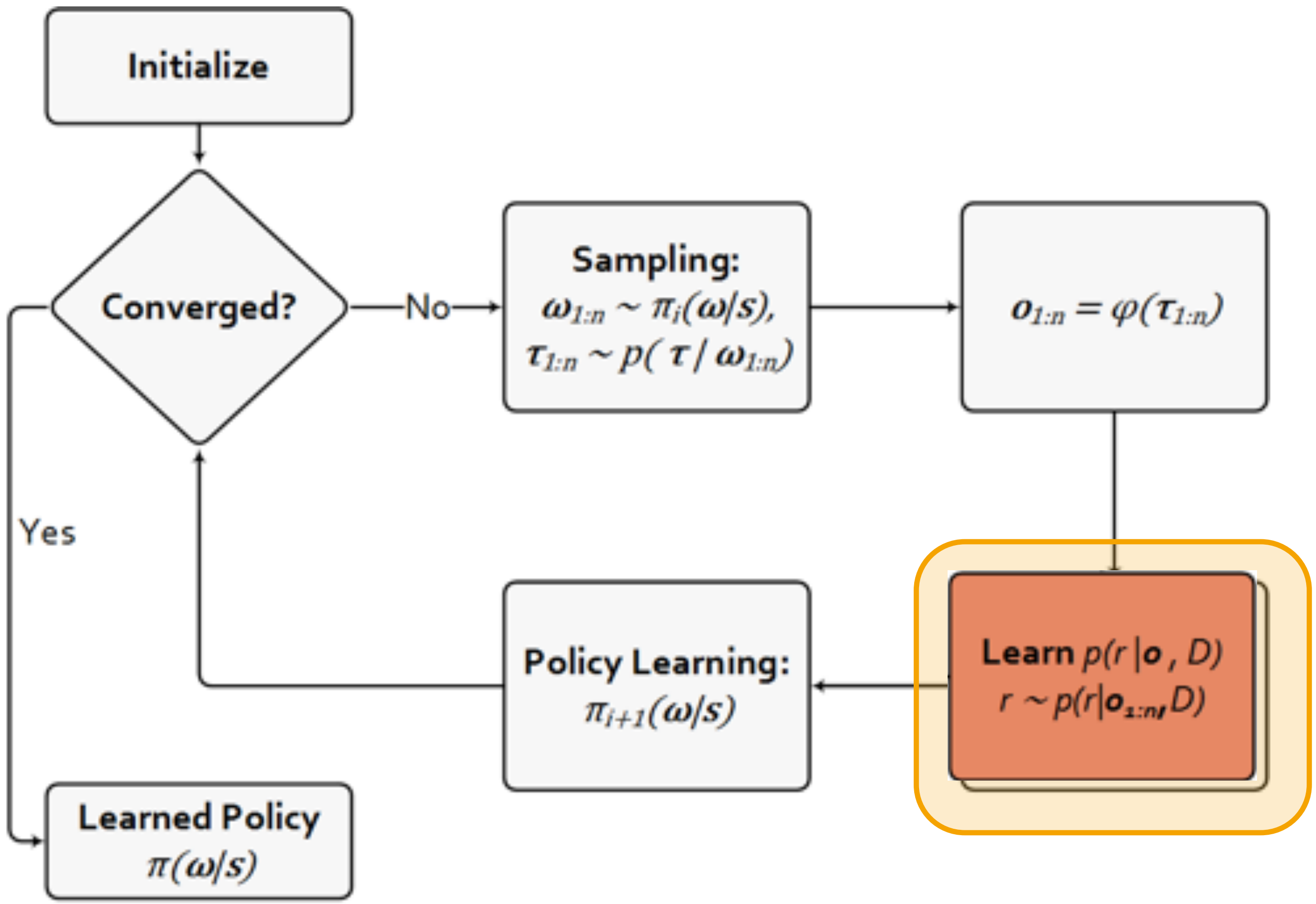
# Acquiring Reward Functions

- Reward functions are hard to design by hand.

➡Let the robot learn a reward model.

➡Human rates skill executions.

# Setup




$s$ Context

$\boldsymbol{\omega}$ Parameters

$\pi$ Policy

$\boldsymbol{\tau}$ Trajectory

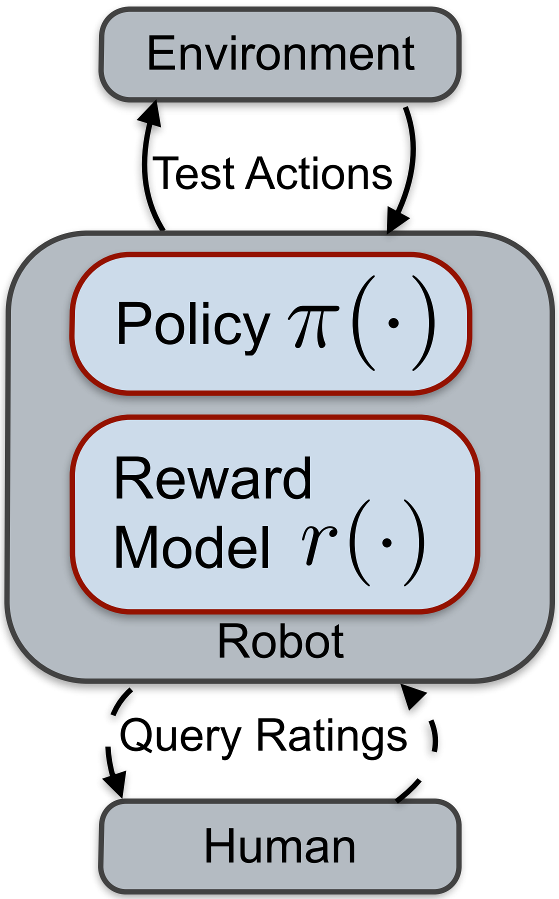
$\varphi$ feature

$o$ outcome

$r$ reward

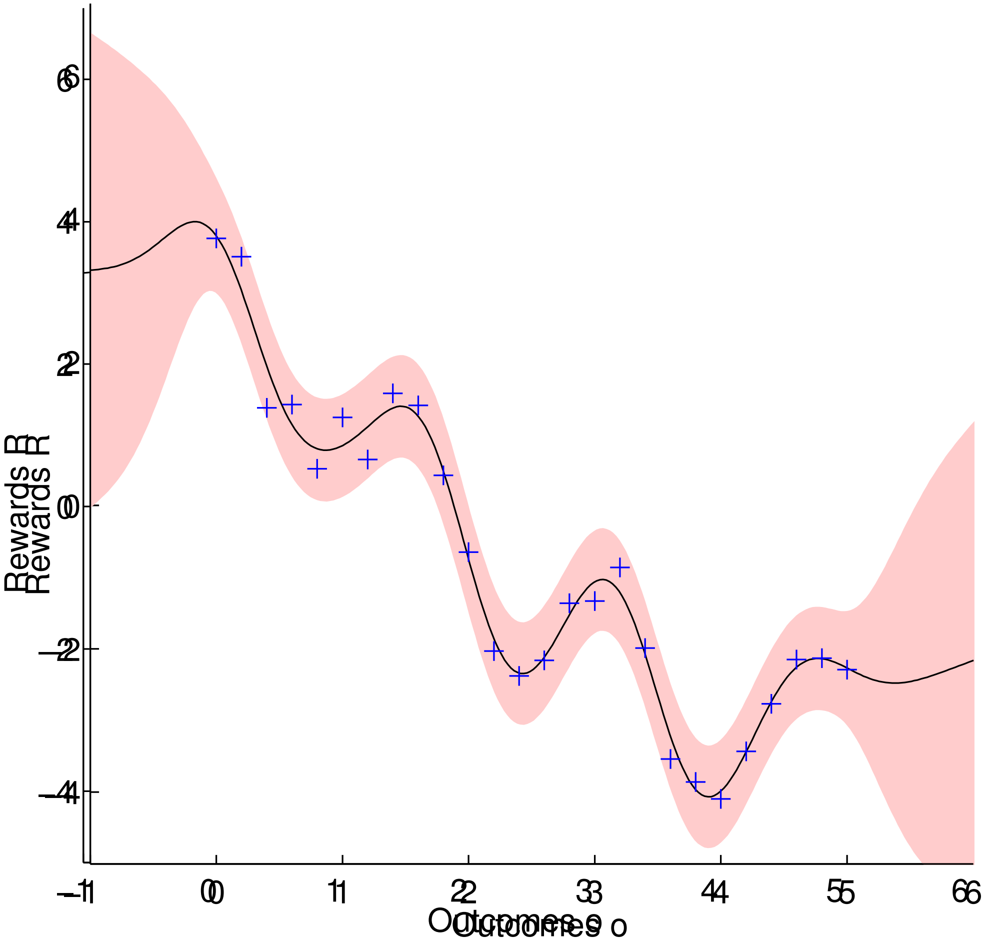
$$\max: \ \mathbb{E}_{\boldsymbol{s},\boldsymbol{\omega},\boldsymbol{\tau}}[f(\boldsymbol{o})] = \int\int\int f(\boldsymbol{o} = \phi(\boldsymbol{\tau}))p(\boldsymbol{\tau}|\boldsymbol{s},\boldsymbol{\omega})\pi(\boldsymbol{\omega}|\boldsymbol{s})\mu^{\pi}(\boldsymbol{s})d\tau ds d\omega$$

# Jointly Learn Reward Model and Policy





# Bayesian Regression Model




We want to model the reward given an outcome.

$$R(\boldsymbol{o}) = f(\boldsymbol{o})$$

Additionally, we want to model the noise of the human experts.

$$R(\boldsymbol{o}) = f(\boldsymbol{o}) + \eta, \;\; \eta \sim \mathcal{N}(0, \beta)$$

➡Use Gaussian Processes (GPs)

$$R(\boldsymbol{o}) \sim \mathcal{GP}\left(m(\boldsymbol{o}), k(\boldsymbol{o}, \boldsymbol{o}')\right)$$

# Bayesian Regression Model

Probabilistic model with explicit representation of the noise:

Gaussian Process $\mathcal{GP}\left(m(\boldsymbol{o}), k(\boldsymbol{o}, \boldsymbol{o}')\right)$

Kernel function $$k(\boldsymbol{o}, \boldsymbol{o}') = \boldsymbol{\theta}_0^2 \exp\left(-\frac{||\boldsymbol{o} - \boldsymbol{o}'||^2}{2\boldsymbol{\theta}_1^2}\right)$$

Kernel matrix $$\boldsymbol{K} = \begin{bmatrix} k(\boldsymbol{o}_1, \boldsymbol{o}_1) & \dots & k(\boldsymbol{o}_1, \boldsymbol{o}_n) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{o}_n, \boldsymbol{o}_1) & \dots & k(\boldsymbol{o}_n, \boldsymbol{o}_n) \end{bmatrix} + \beta \boldsymbol{I}$$

Reward prediction $$p(R^+ | \boldsymbol{o}, \mathcal{D}) \sim \mathcal{N}\left(\mu(\boldsymbol{o}^+), \sigma^2(\boldsymbol{o}^+)\right)$$

Mean and variance $$\mu(\boldsymbol{o}^+) = \boldsymbol{k}^T \boldsymbol{K}^{-1} \boldsymbol{R}_{1:n},$$ $$\sigma^2(\boldsymbol{o}^+) = k(\boldsymbol{o}^+, \boldsymbol{o}^+) - \boldsymbol{k}^T \boldsymbol{K}^{-1} \boldsymbol{k},$$

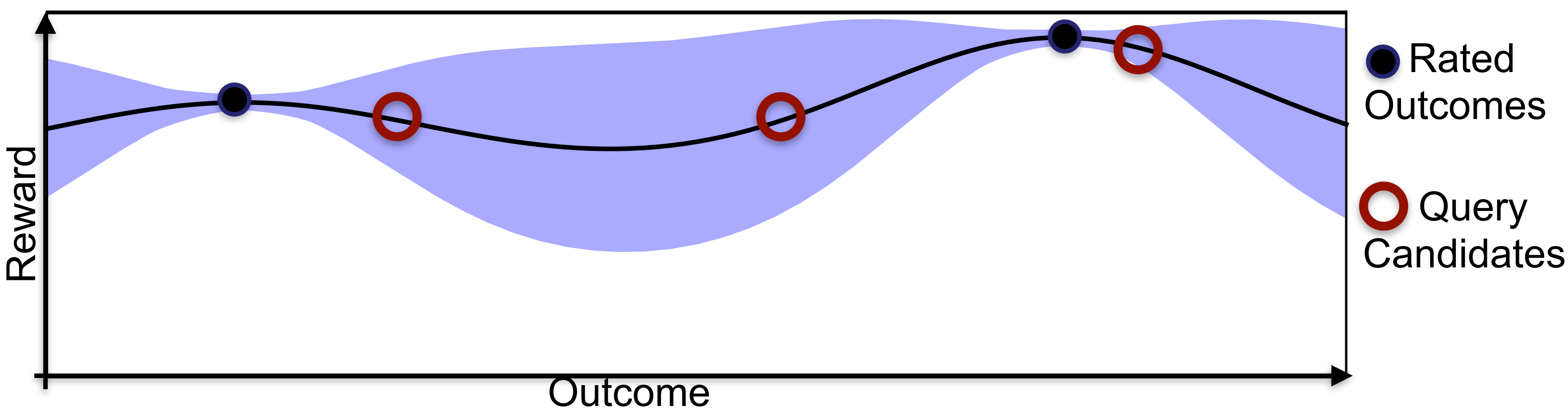
Hyperparameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \beta\}$ Human noise estimated through hyper parameter optimization

# Building the Model

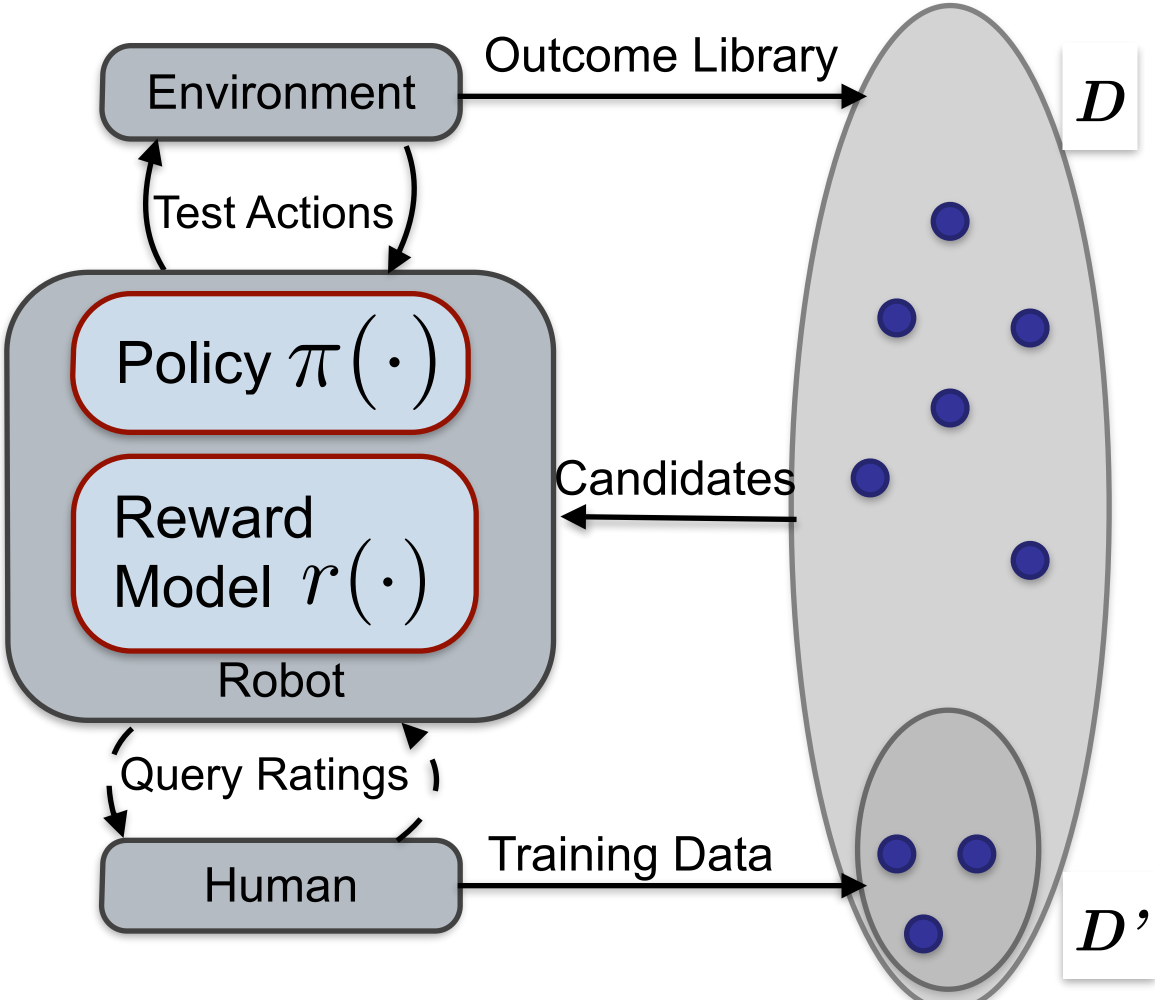
- Optimize using acquisition functions $\boldsymbol{o}^+ = \mathrm{argmax}(u(\boldsymbol{o} \in \boldsymbol{D}))$.

- No mapping $p(\boldsymbol{\omega}|\boldsymbol{s}, \boldsymbol{o})$.

➡Select sample from library of observed outcomes $D$.



# Minimize Human Interaction




➡ Select candidate

$$o^+ = \operatorname{argmax}(u(o \in D))$$

➡ Only sample until encounter known sample $o^+ \notin D'$

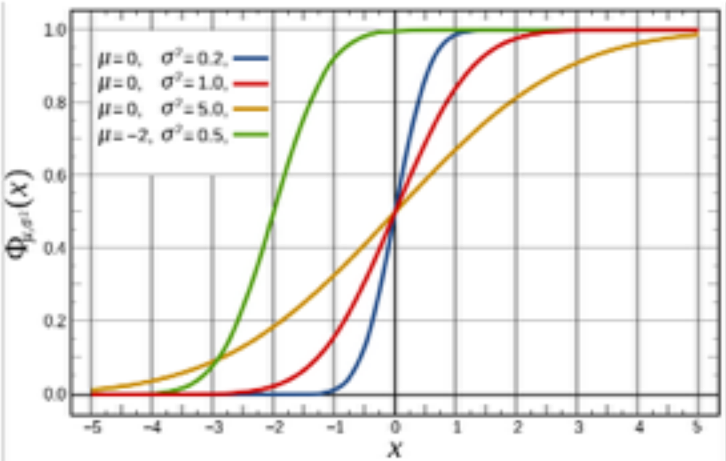
➡ Improvement threshold

$$\sigma^2(o^+) \,/\, \beta > \lambda$$

# Acquisition Functions

TECHNISCHE UNIVERSITÄT DARMSTADT

$$Z = \frac{\mu(\boldsymbol{o}) - f(\boldsymbol{o}^{*}) - \xi}{\sigma(\boldsymbol{o})}$$




Probability of Improvement (PI):

$$PI(\boldsymbol{o}) = \Phi(Z)$$

$\Phi(\cdot)$ Cumulative Distribution Function (CDF)

Expected Improvement (EI):

$$\mathbb{E}(I, \boldsymbol{o}) = \int_{I=0}^{I=\infty} I \frac{1}{2\sqrt{2\pi}\sigma(\boldsymbol{o})} \exp\left(\frac{(-\mu(\boldsymbol{o}) - f(\boldsymbol{o}^{*}) - I)^2}{2\sigma^2(\boldsymbol{o})}\right) \mathrm{d}I$$

$$EI(\boldsymbol{o}) = \sigma(\boldsymbol{o}) \left[Z\Phi(Z) + \phi(Z)\right]$$

$\phi(\cdot)$ Probability Density Function (PDF)

Upper Confidence Bound (UCB):

$$UCB(\boldsymbol{o}) = \mu(\boldsymbol{o}) + \kappa\sigma(\boldsymbol{o})$$

Hedge

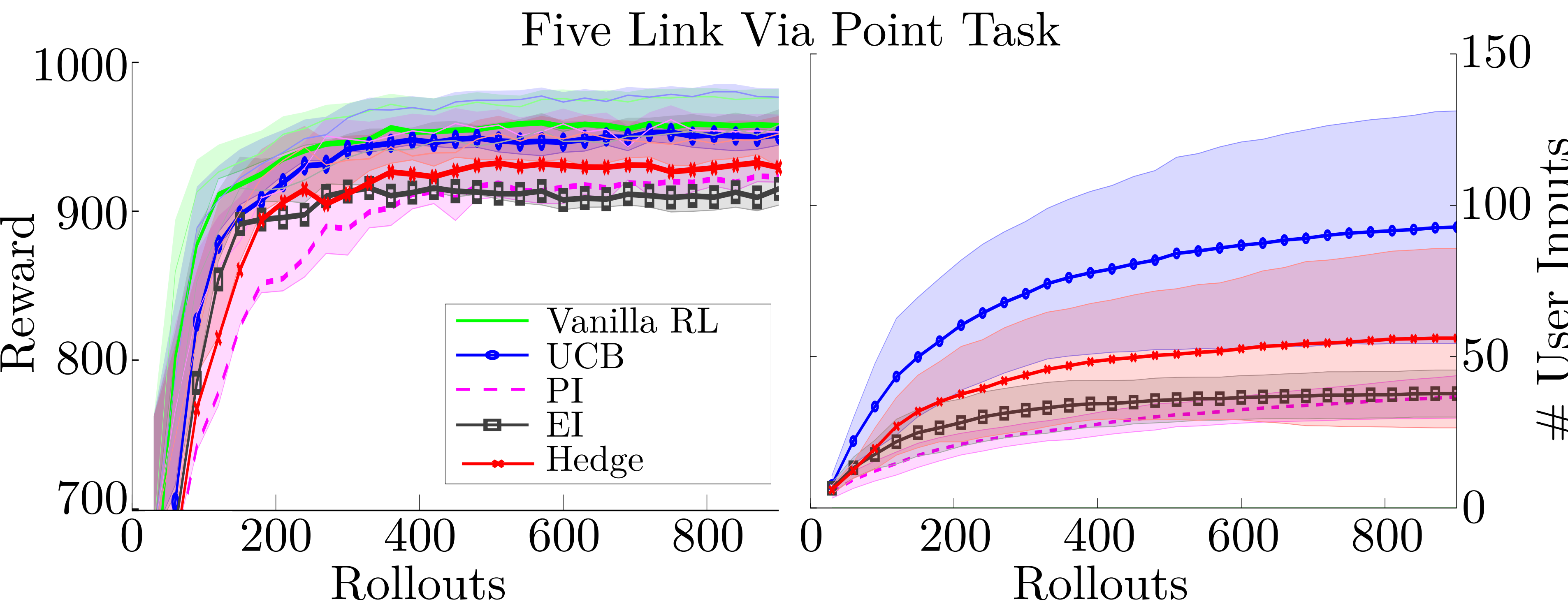


IAS

# Empirical Evaluations

- Programmed noisy expert to help us evaluate
- Evaluations of different Acquisition Functions
- Evaluations of Noisy Expert
- Evaluations of sample efficiency methods
- Real Robot Evaluations
- Evaluations of reward function transfer

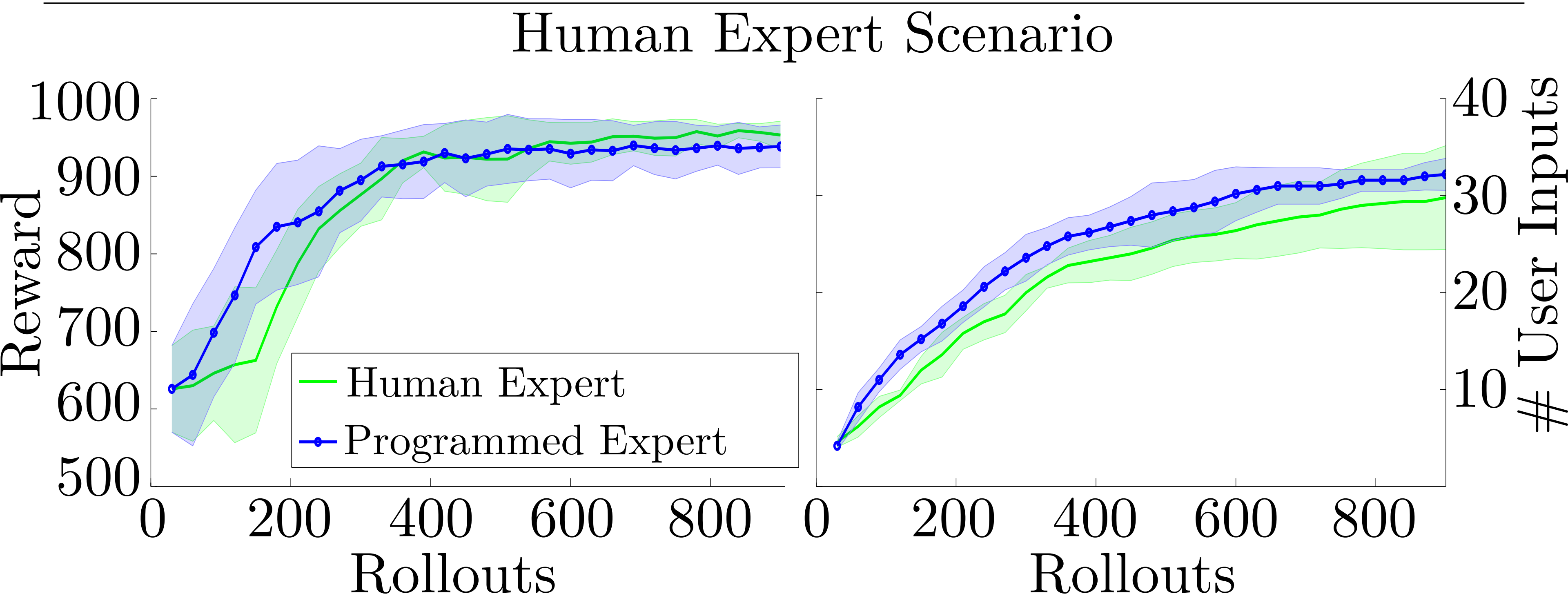
# Empirical Evaluations




- PI has worst performance but lowest # of user inputs.
- UCB has best performance but highest # of user inputs.

➡ We use PI for real robot experiments.

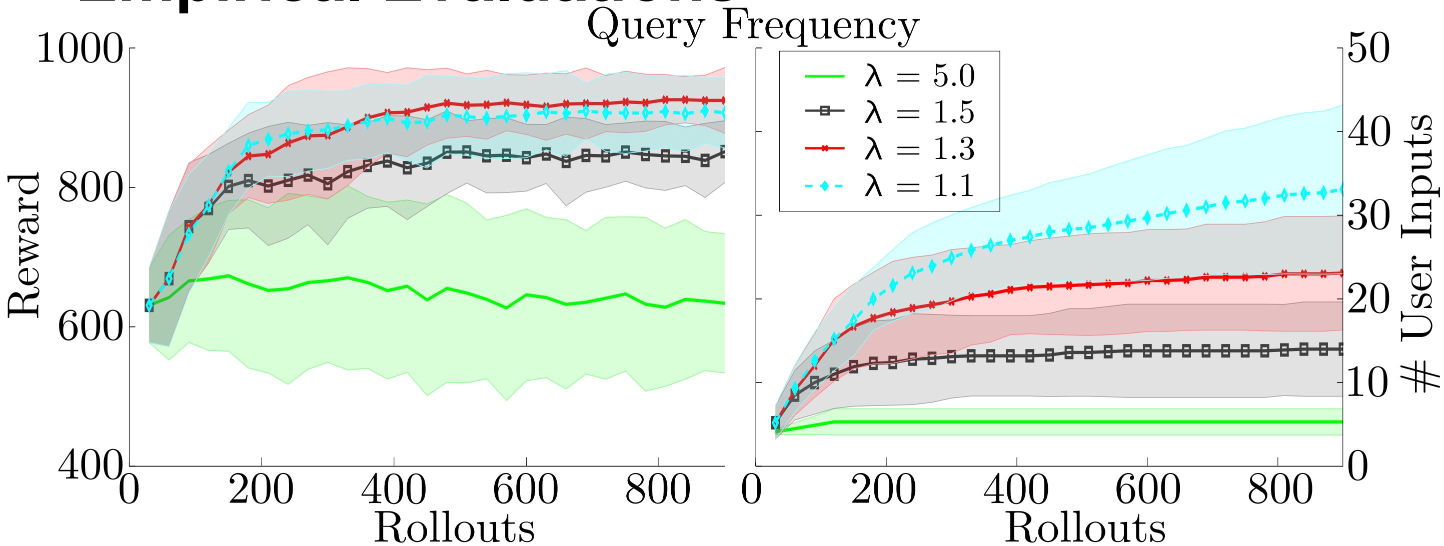
# Empirical Evaluations




- Similar performance
- Similar # user inputs

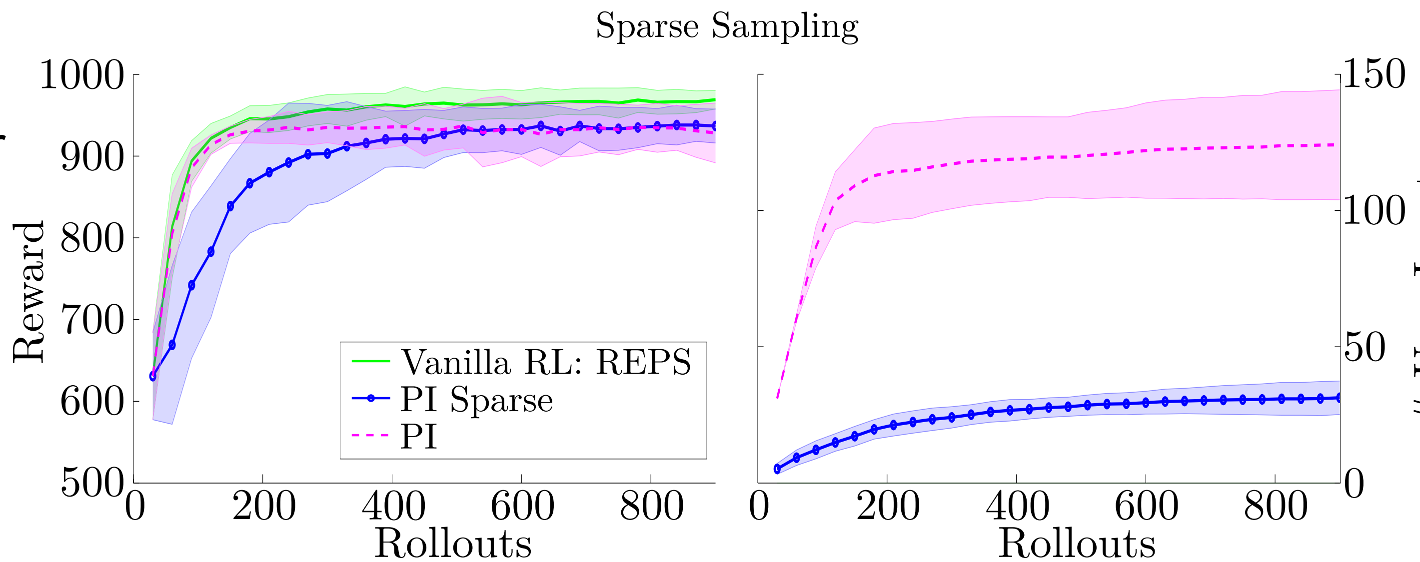
➡ Suitable model of human expert for our purposes

# Empirical Evaluations

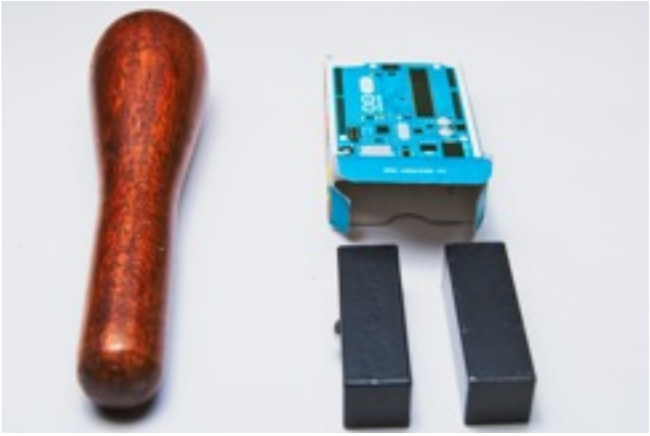



Reasonable trade-off at λ = 1.5

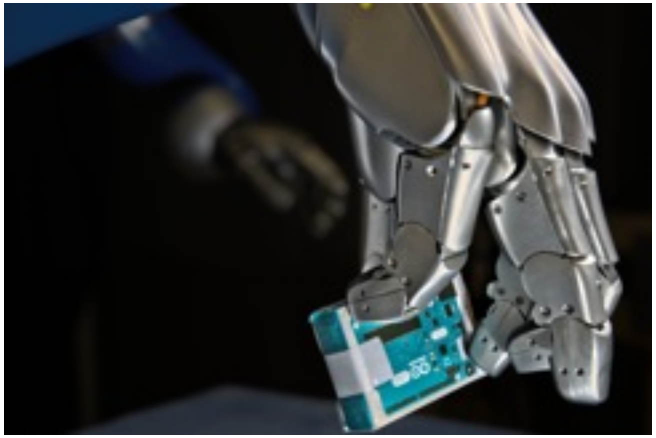
- Equivalent asymptotic behavior
- Approx. three times less user inputs



# Empirical Evaluations



Pestle and Paper Box

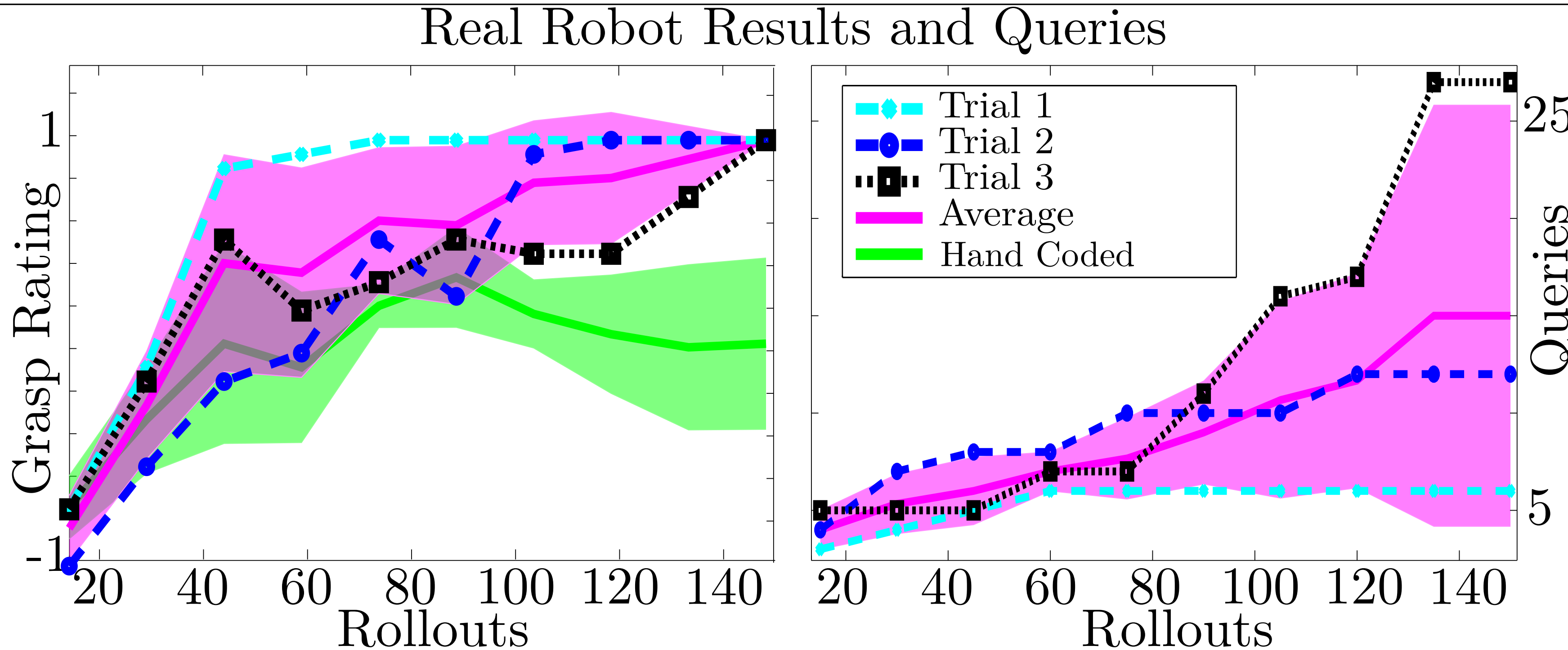


Unstable Grasp

Wrong Orientation

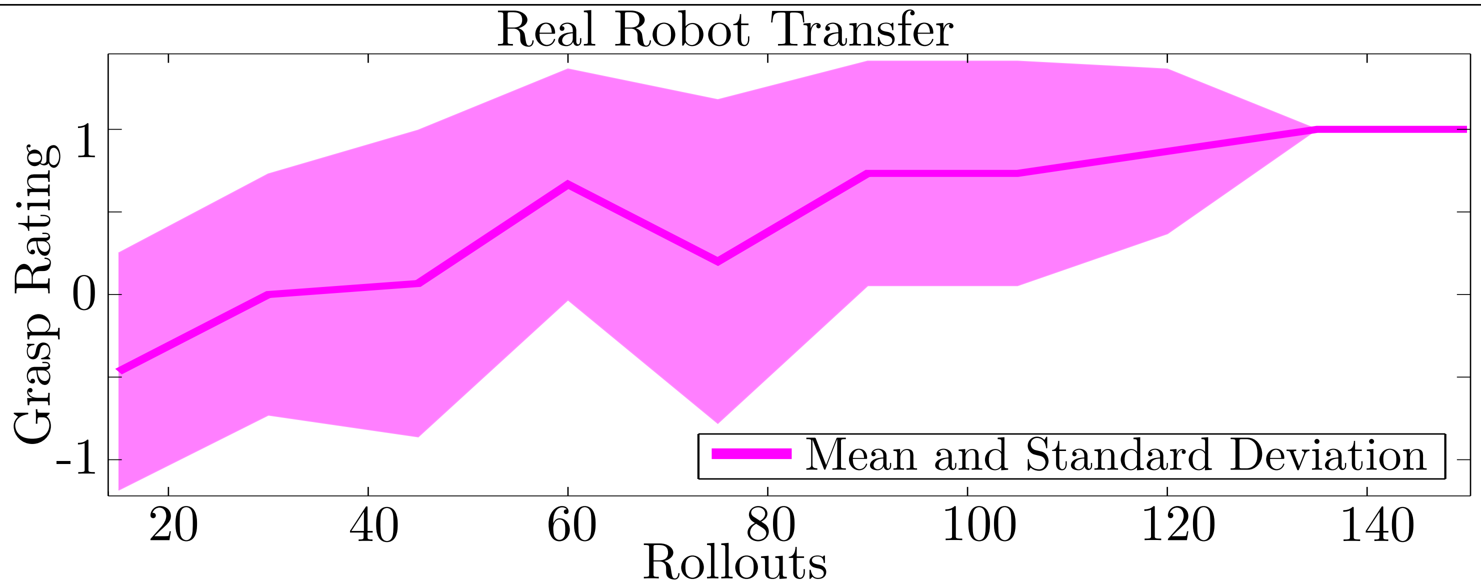
Stable Grasp

## Empirical Evaluations




Successful grasps (15/15) in all three trials after 150 rollouts.
Performance achieved with an average of 15 user inputs
Robot broke and recovered in trial three

# Empirical Evaluations




Robot learned to grasp new object with the same reward function.

# Related Work

Preference learning [Akrour 2011]

➡Only allows for binary ratings.

Inverse Reinforcement Learning [Ziebart 2008]

➡Requires access to reasonably good demonstrations.

Trajectory Preferences [Jain 2013]

➡Requires forward model of the system and environment.

# Conclusion & Future Work

**Conclusion:**

- Able to learn a reward model from a small set (~15) of human ratings.
- Learned reward models are sufficiently generalize to similar objects.

**Limitations:**

- Requires access to expressive features.

**Future Work:**

- Evaluate effect of different kernels (inspired by human expert data).
- Investigate specialized acquisition function.

# Thank you!