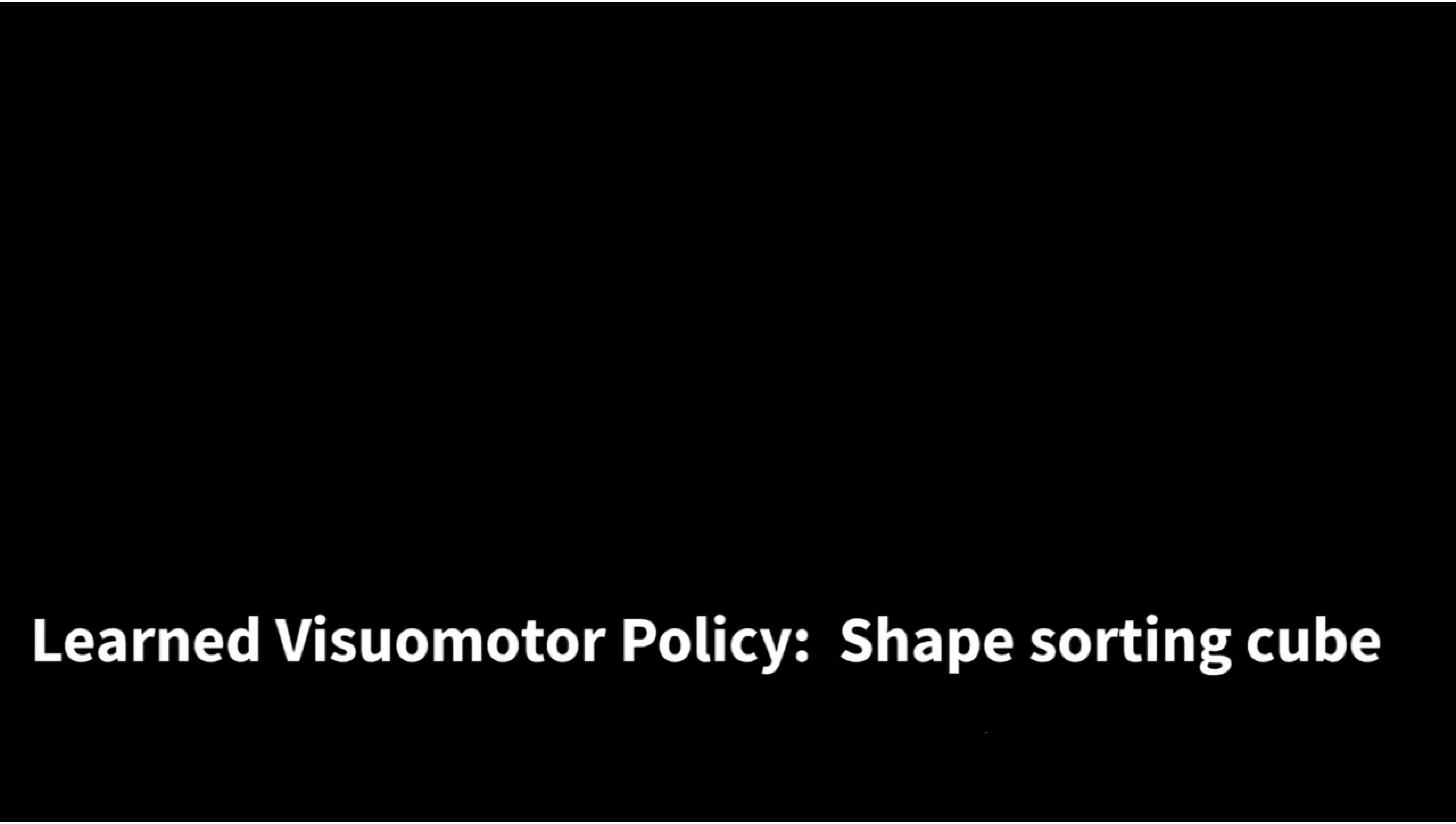
Guided Policy Search

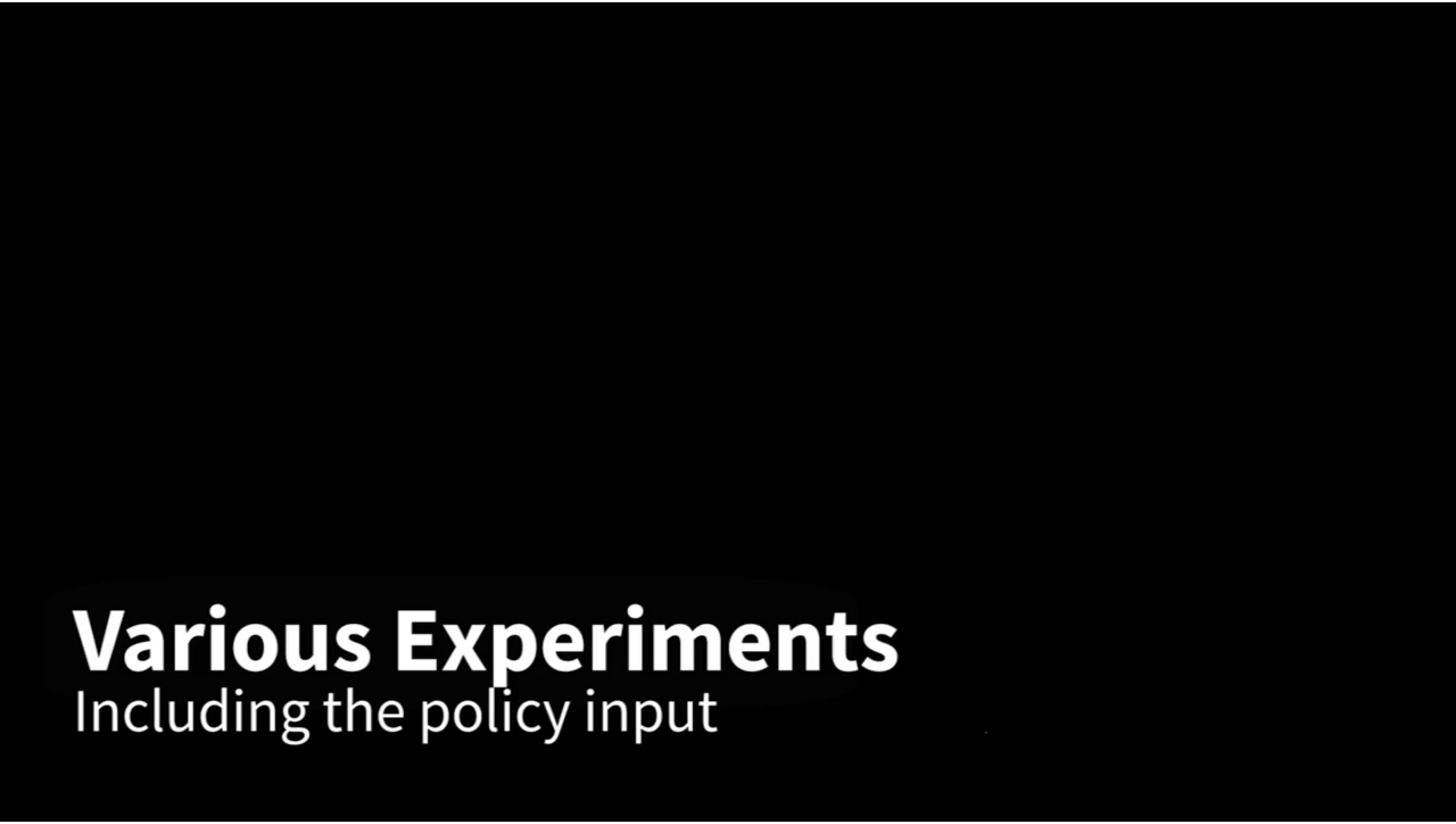
Sergey Levine

Learning on PR2

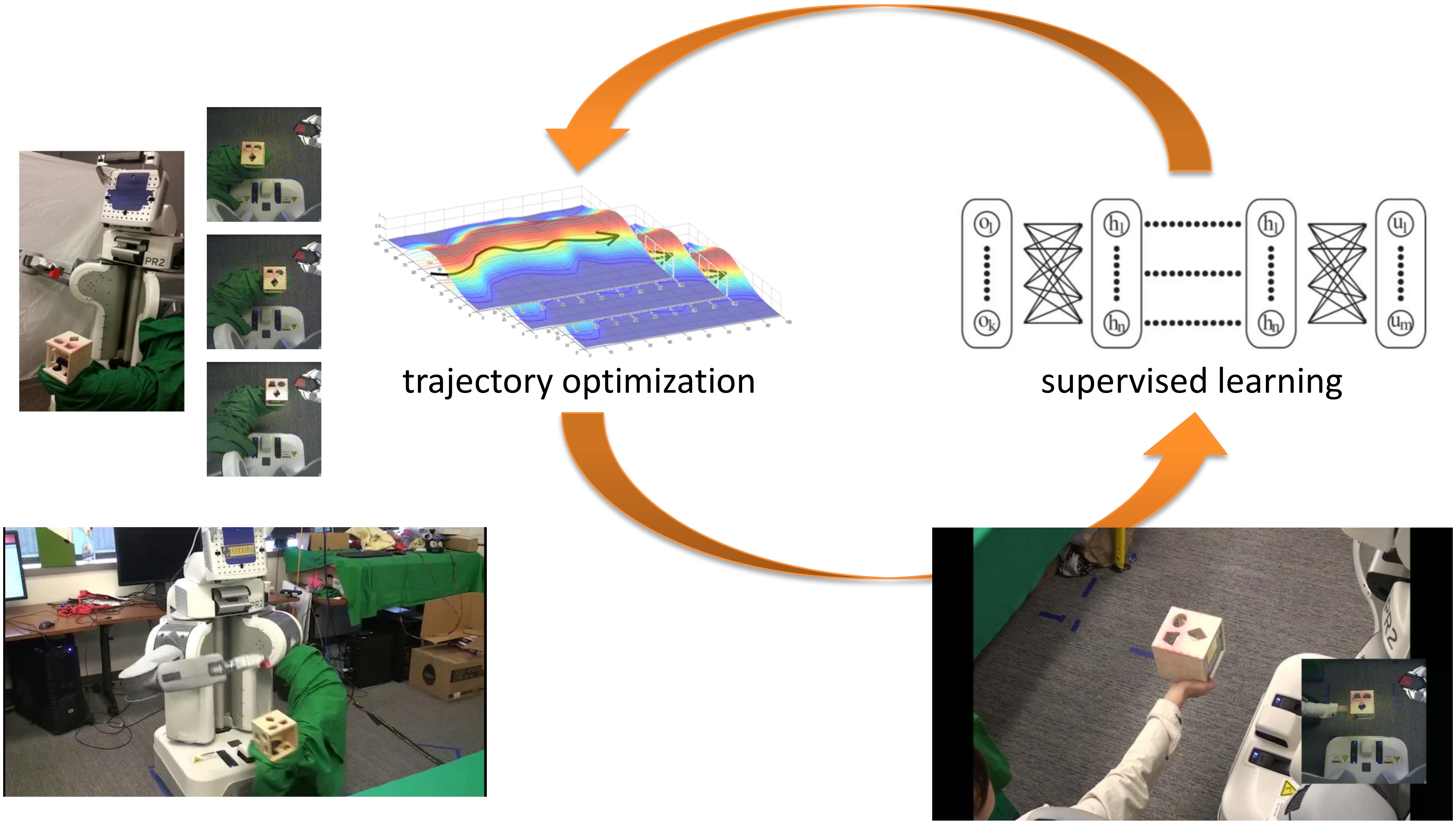
Shape sorting cube



Visuomotor Policies



Guided Policy Search

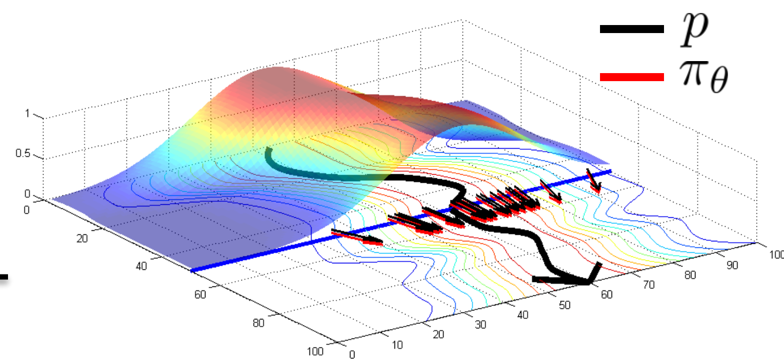


expectation under current policy

$$\min_{\theta} \overbrace{E_{\pi_\theta}}[c(\tau)]$$

$$\min_{\theta, p(\tau)} \quad E_p[c(\tau)]$$

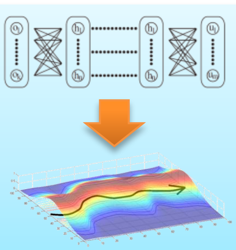
trajectory distribution(s)



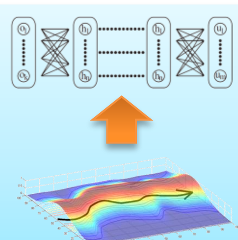
$$s.t. \quad \pi_\theta(\mathbf{u}_t | \mathbf{o}(\mathbf{x}_t)) = p(\mathbf{u}_t | \mathbf{x}_t) \quad \forall t, \mathbf{x}_t, \mathbf{u}_t$$

Lagrange multiplier

$$\mathcal{L}(\theta, p, \lambda) = E_p[c(\tau)] + \sum_{t=1}^{T} \overbrace{\lambda_t} D_t(\pi_\theta, p)$$



optimize $\mathcal{L}(\theta, p, \lambda)$ w.r.t. $p(\tau)$



optimize $\mathcal{L}(\theta, p, \lambda)$ w.r.t. θ

update λ with subgradient descent:

$$\lambda_t \leftarrow \lambda_t + \eta D_t(\pi_\theta, p)$$

Supervised Learning Objective

$$\theta = \arg\min_{\theta} \sum_{t=1}^{T} E_{p(\mathbf{x_t})}[\rho_t D_{KL}(\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)\|p(\mathbf{u}_t|\mathbf{x}_t)) + \lambda_t^T E_{\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)}[\mathbf{u}_t]]$$

$$\pi_\theta(\mathbf{u}_t|\mathbf{x}_t) = \mathcal{N}(\mu_\pi(\mathbf{x}_t), \Sigma_\pi(\mathbf{x}_t))$$

$$p(\mathbf{u}_t|\mathbf{x}_t) = \mathcal{N}(\mu_p(\mathbf{x}_t), \Sigma_p(\mathbf{x}_t))$$

generate samples from $p(\mathbf{x}_t)$ by executing $p(\mathbf{u}_t|\mathbf{x}_t)$

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} \rho_t D_{KL}(\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)\|p(\mathbf{u}_t|\mathbf{x}_t)) + \lambda_t^T \mu_\pi(\mathbf{x}_t)$$

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} \frac{\rho_t}{2}\left[(\mu_\pi(\mathbf{x}_t) - \mu_p(\mathbf{x}_t))^T \Sigma_p(\mathbf{x}_t)^{-1}(\mu_\pi(\mathbf{x}_t) - \mu_p(\mathbf{x_t})) + \mathrm{tr}\left(\Sigma_\pi(\mathbf{x}_t)\Sigma_p(\mathbf{x}_t)^{-1}\right) + \log\frac{|\Sigma_p(\mathbf{x}_t)|}{|\Sigma_\pi(\mathbf{x}_t)|}\right] + \lambda_t^T \mu_\pi(\mathbf{x}_t)$$

$$\mu_\pi : \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} \frac{\rho_t}{2} \|\mu_\pi(\mathbf{x}_t) - \mu^\star(\mathbf{x}_t)\|_{\Sigma_p(\mathbf{x}_t)^{-1}} \qquad \mu^\star(\mathbf{x_t}) = \mu_p(\mathbf{x}_t) - \Sigma_p(\mathbf{x}_t)\lambda_t$$

Trajectory Optimization (without GPS)

Goal: optimize Gaussian trajectory distribution $p(\tau)$ w.r.t. $E_p[c(\tau)]$

Must optimize time-varying linear-Gaussian controller $p(\mathbf{u}_t|\mathbf{x}_t)$

Controller has form $p(\mathbf{u}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{K}_t\mathbf{x}_t, \Sigma_t)$

Use LQR to get $\mathbf{K}_t$, but what is Σ_t?

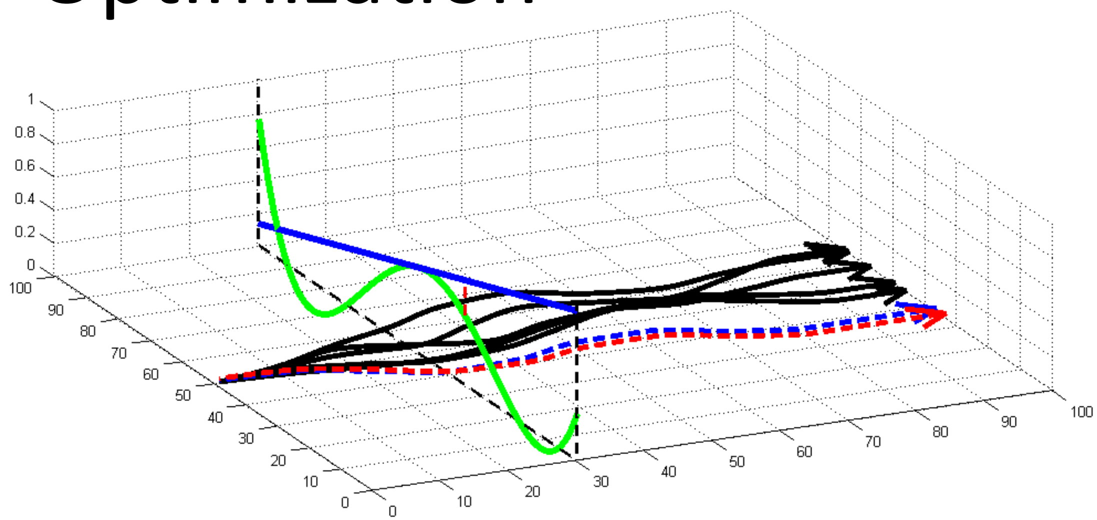
If $\mathbf{x}_t$ is Markovian, $\Sigma_t = 0$ always (but this is boring...)

Let's instead optimize $E_p[c(\tau)] - \mathcal{H}(p) = \underbrace{\sum_{t=1}^T E_{p(\mathbf{x}_t,\mathbf{u}_t)}[c(\mathbf{x}_t,\mathbf{u}_t)] - \mathcal{H}(p(\mathbf{u}_t|\mathbf{x}_t))}_{E_p[c(\tau)] - \mathcal{H}(p)}$

(we'll see why soon...)

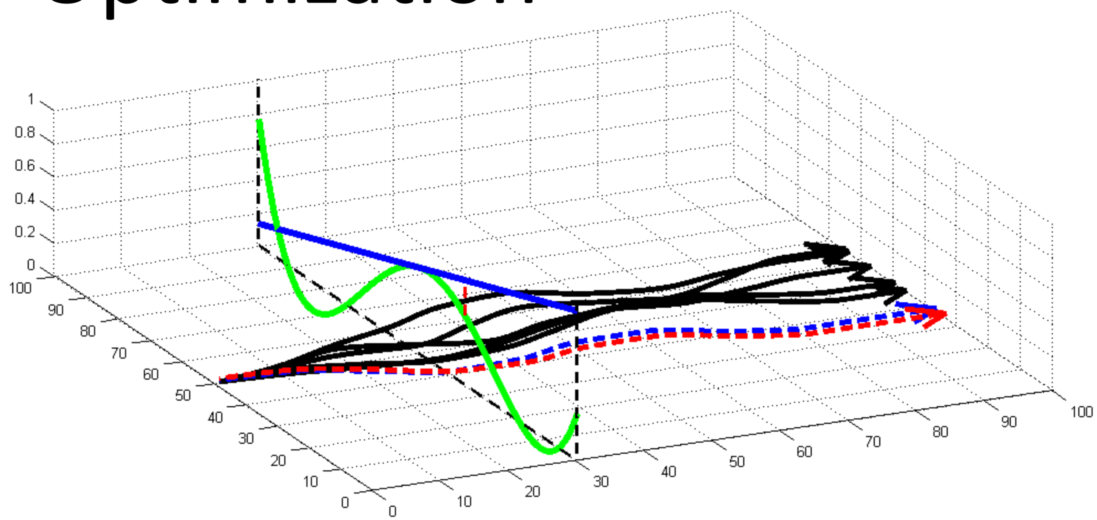
Maximum entropy solution is simply $p(\mathbf{u}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{K}_t\mathbf{x}_t, \underbrace{\mathbf{R}_t + \mathbf{B}_t^T\mathbf{P}_{t+1}\mathbf{B}}_{\text{LQR cost-to-go w.r.t. } \mathbf{u}_t\text{, sometimes written as } Q_{\mathbf{uu}t}})$

Trajectory Optimization



1. Run time-varying policy $p(\mathbf{u}_t|\mathbf{x}_t)$ on robot N times
2. Collect dataset $\mathcal{D} = \{\tau_i\}$ where $\tau_i = \{\mathbf{x}_{1i}, \mathbf{u}_{1i}, \ldots, \mathbf{x}_{Ti}, \mathbf{u}_{Ti}\}$
3. For each $t \in \{0, \ldots, T-1\}$, fit linear Gaussian $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$
4. Solve control problem to get new $p(\mathbf{u}_t|\mathbf{x}_t)$

Trajectory Optimization

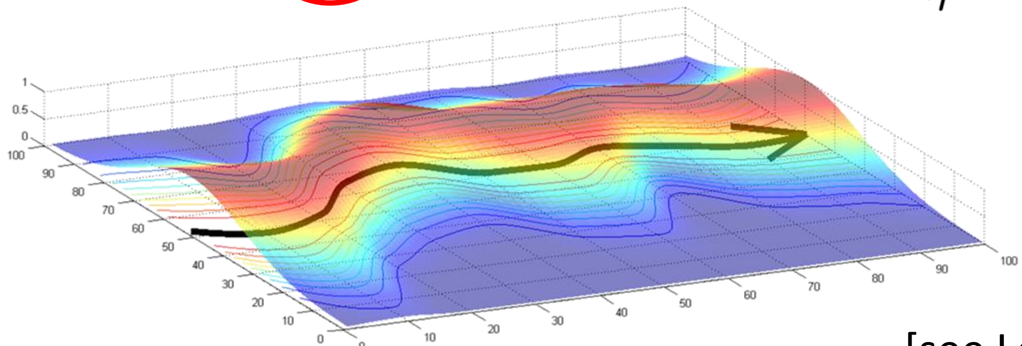


$$\min_{p(\tau)} E_p[c(\tau)] \text{ s.t. } D_{KL}(\underbrace{p(\tau)}_{\text{new}} \| \underbrace{\bar{p}(\tau)}_{\text{old}}) \leq \epsilon$$

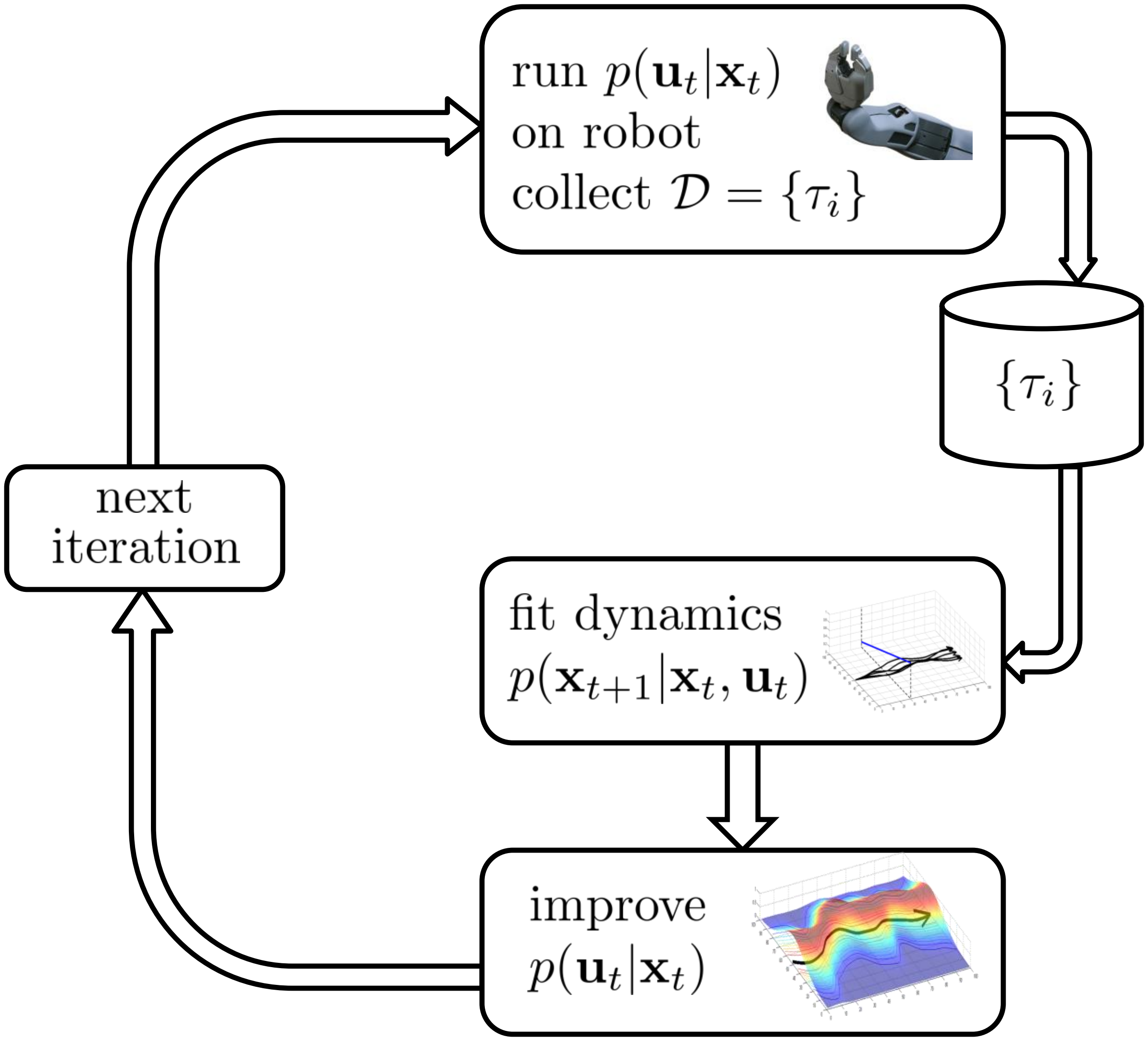
$$\frac{1}{\eta}\mathcal{L}(p,\eta) = E_p\left[\frac{1}{\eta}c(\tau) - \log \bar{p}(\tau)\right] - \mathcal{H}(p) - \epsilon$$

$$\min_{p(\tau)} E_p[\tilde{c}(\tau)] - \mathcal{H}(p)$$

$$\tilde{c}(\tau) = \frac{1}{\eta}c(\tau) - \log \bar{p}(\tau)$$



[see Levine & Abbeel '14 for details]



[see L. et al. NIPS '14 for details]

Trajectory Optimization (with GPS)

$$\min_{p(\tau)} \sum_{t=1}^{T} E_{p(\mathbf{x}_t,\mathbf{u}_t)}[c(\mathbf{x}_t,\mathbf{u}_t) + \mathbf{u}_t^T\lambda_t + \rho_t D_{KL}(p(\mathbf{u}_t|\mathbf{x}_t)\|\pi_\theta(\mathbf{u}_t|\mathbf{x}_t))]$$

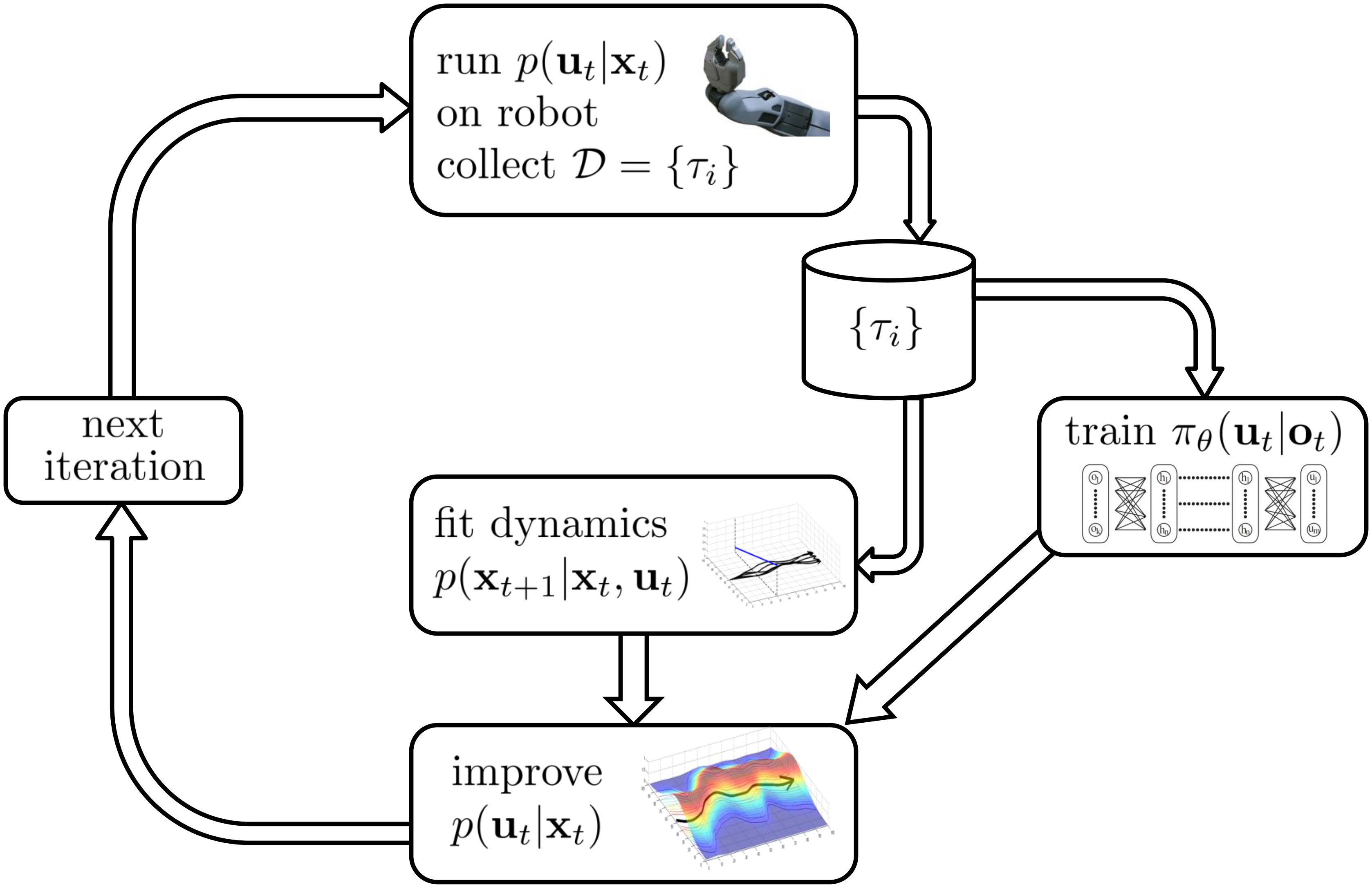
$$\text{s.t. } D_{KL}(p(\tau)\|\bar{p}(\tau)) \leq \epsilon$$

$$\mathcal{L}(p,\eta) = \sum_{t=1}^{T} E_{p(\mathbf{x}_t,\mathbf{u}_t)}[c(\mathbf{x}_t,\mathbf{u}_t)+\mathbf{u}_t^T\lambda_t+\rho_t D_{KL}(p(\mathbf{u}_t|\mathbf{x}_t)\|\pi_\theta(\mathbf{u}_t|\mathbf{x}_t))+\eta D_{KL}(p(\mathbf{u}_t|\mathbf{x}_t\|\bar{p}(\mathbf{u}_t,\mathbf{x}_t))]$$

$$\mathcal{L}(p,\eta) = \sum_{t=1}^{T} E_{p(\mathbf{x}_t,\mathbf{u}_t)}[\underbrace{c(\mathbf{x}_t,\mathbf{u}_t)+\mathbf{u}_t^T\lambda_t-\rho_t\log\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)-\eta\bar{p}(\mathbf{u}_t,\mathbf{x}_t)}_{\tilde{c}(\mathbf{x}_t,\mathbf{u}_t)}]-(\rho_t+\eta)\mathcal{H}(p(\mathbf{u}_t|\mathbf{x}_t))$$

$$\mathcal{L}(p,\eta) = \sum_{t=1}^{T} E_{p(\mathbf{x}_t,\mathbf{u}_t)}[\tilde{c}(\mathbf{x}_t,\mathbf{u}_t)] - \nu_t\mathcal{H}(p(\mathbf{u}_t|\mathbf{x}_t))$$

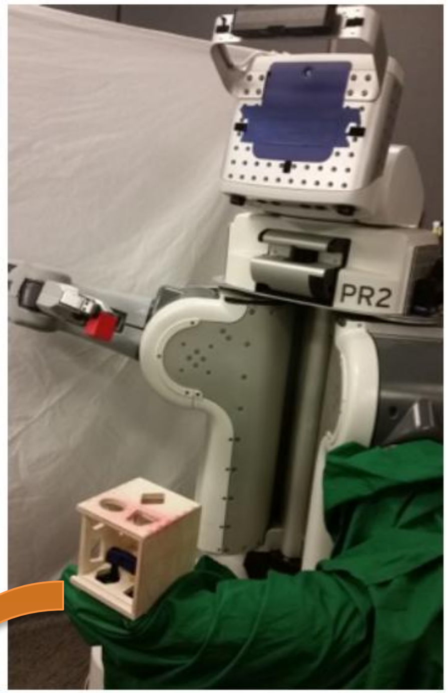
maximum entropy objective (like before)



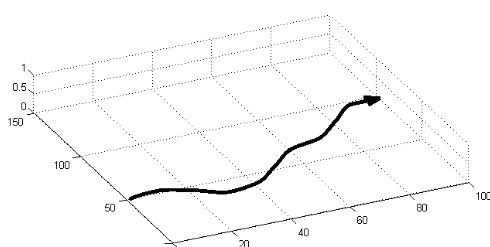
[see L. et al. NIPS '14 for details]

Instrumented Training

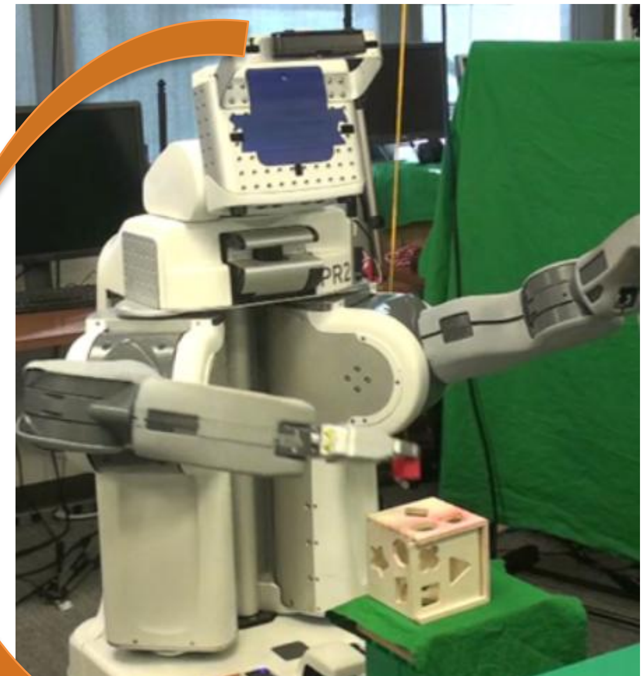
training time



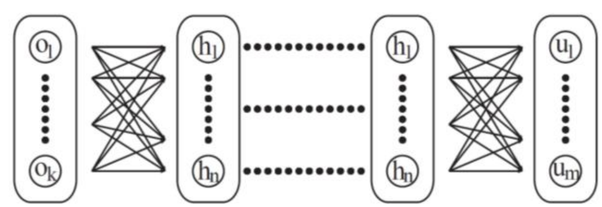
$$\mathbf{x}_t \rightarrow \mathbf{u}_t$$

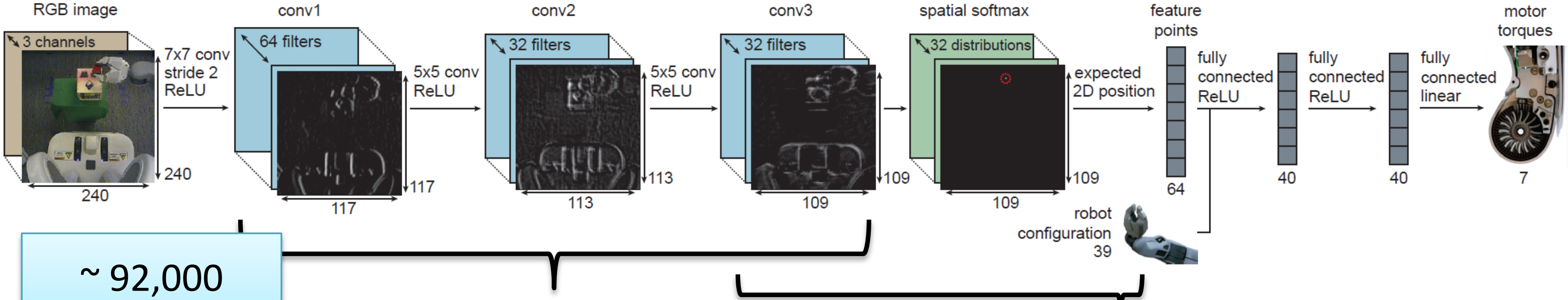


test time

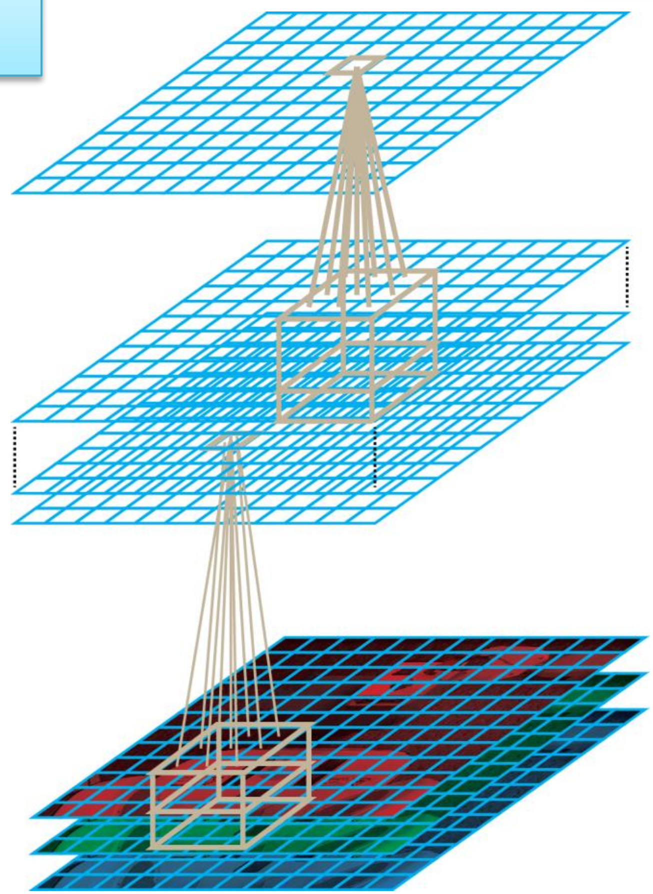


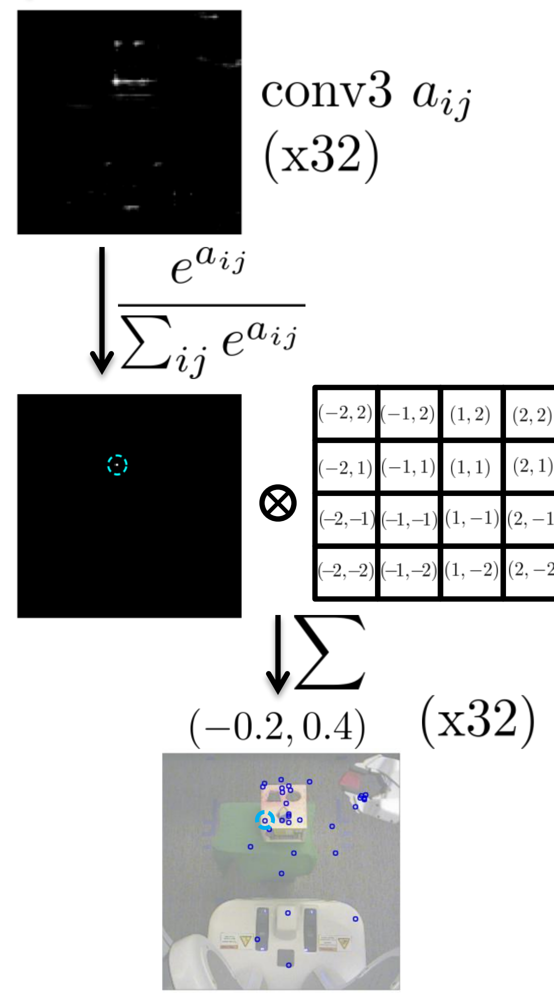
$$\mathbf{o}_t \rightarrow \mathbf{u}_t$$





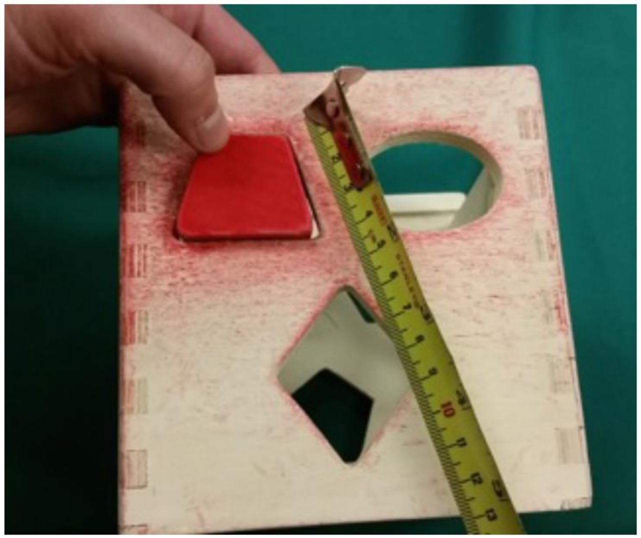
~ 92,000 parameters

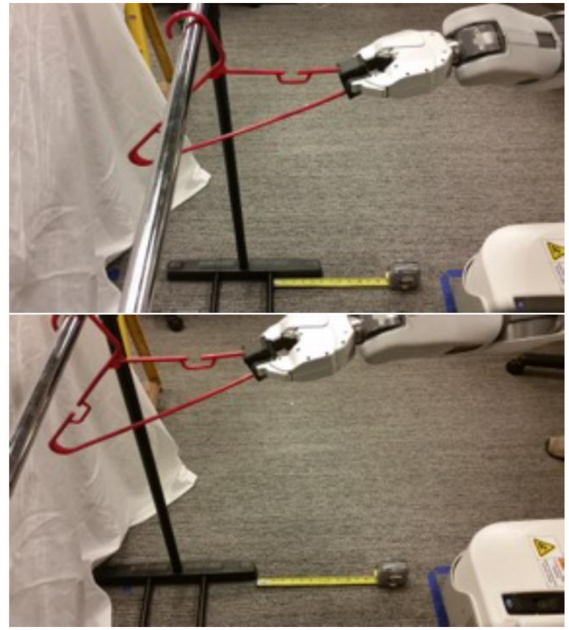




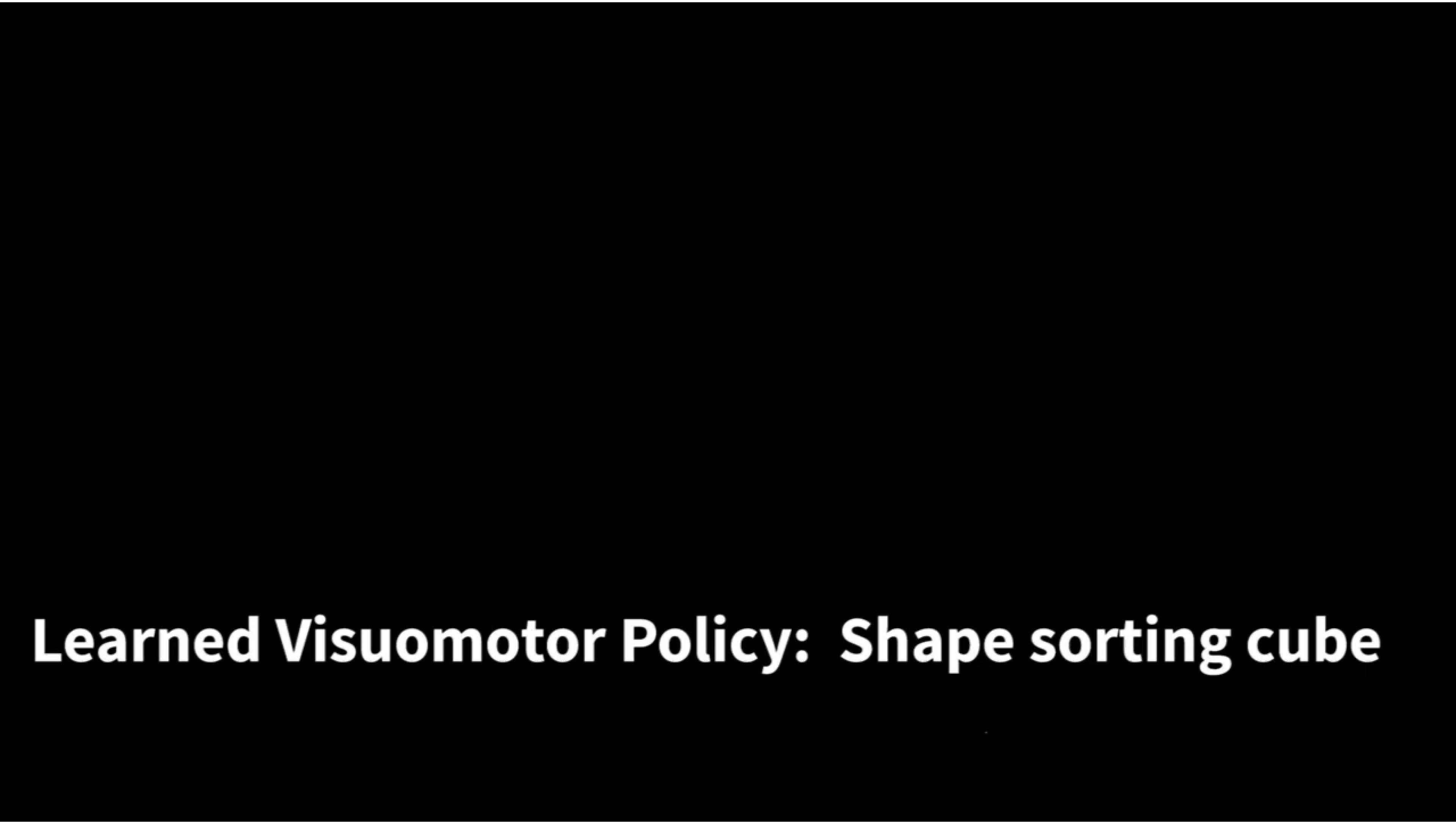
Chelsea Finn

Experimental Tasks





Shape sorting cube



Hanger

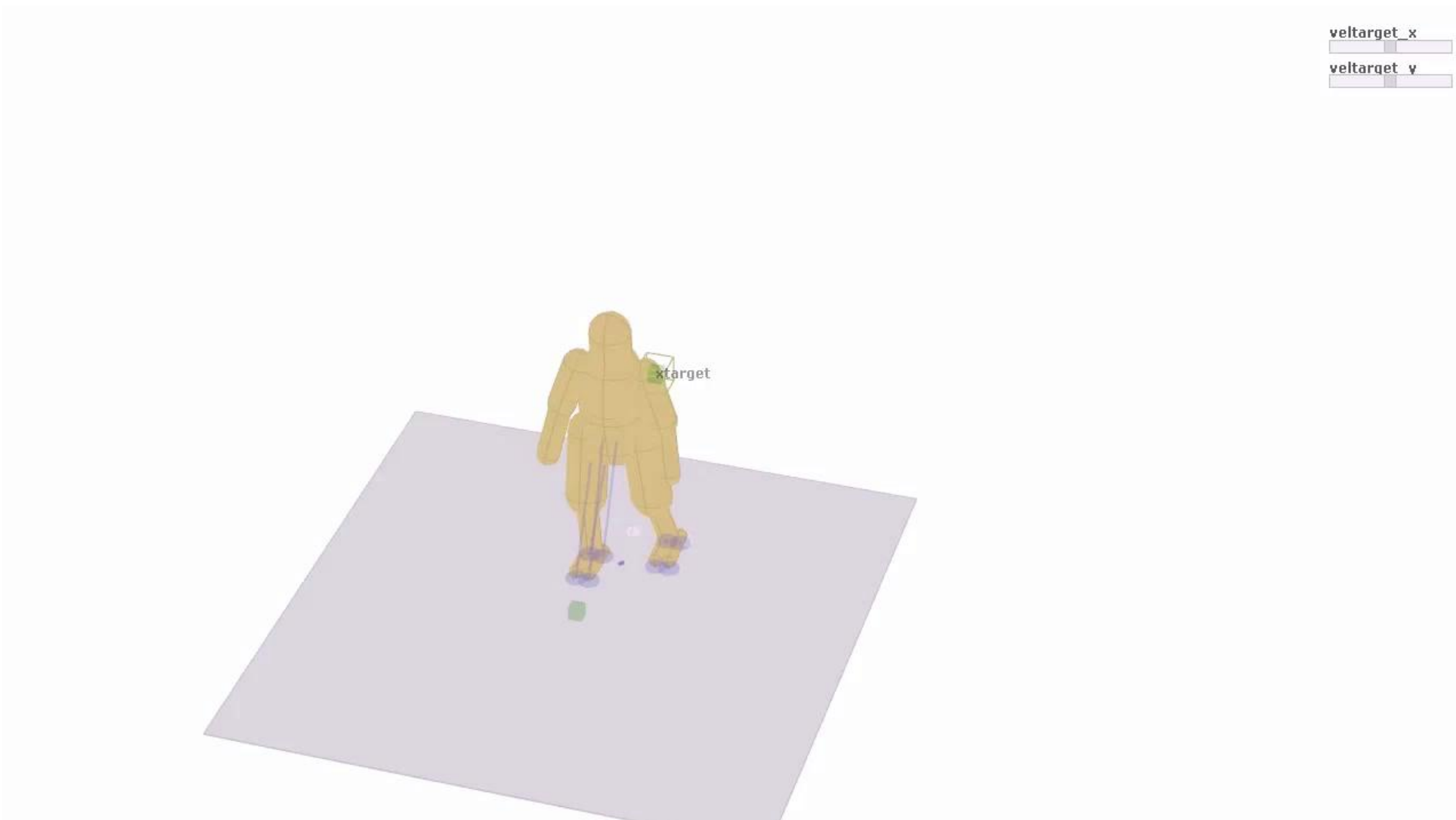
Hammer

Bottle

Locomotion

Igor Mordatch

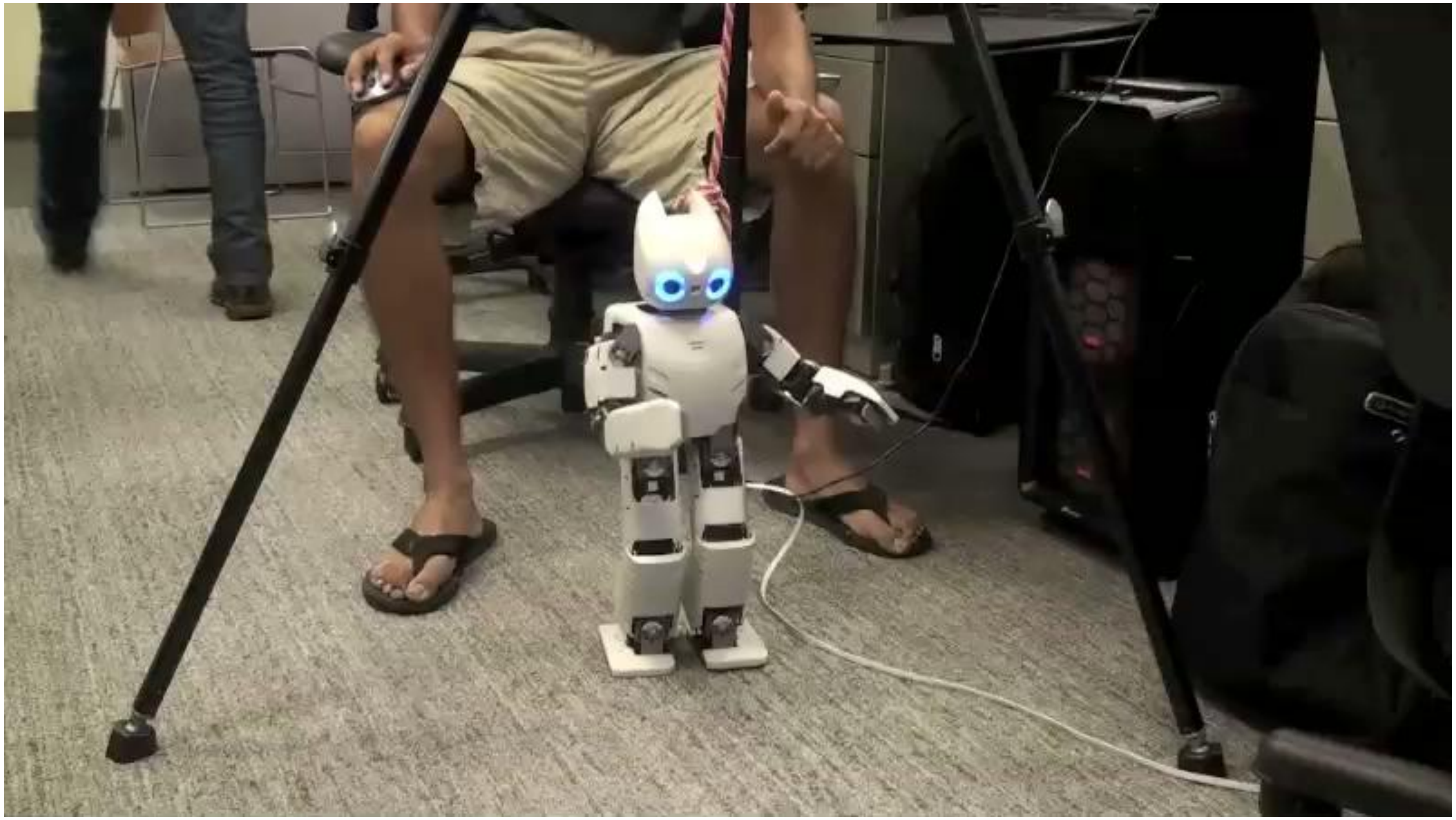
better trajectory optimization +
large scale simulation



Darwin Robot

Igor Mordatch

better trajectory optimization +
large scale simulation +
adaptation to real world dynamics



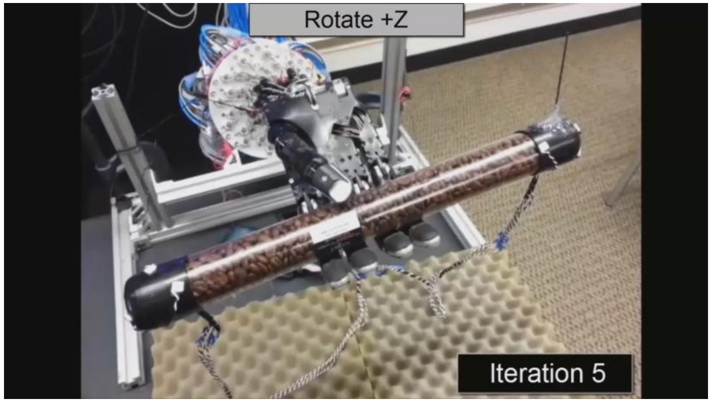
Mordatch, Mishra, Eppner, Abbeel

Guided Policy Search Applications

manipulation

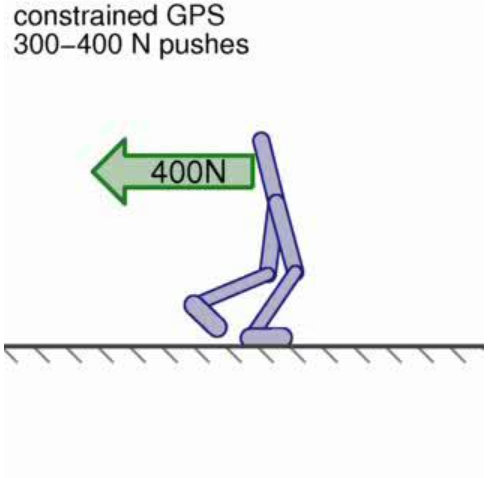
with N. Wagener and P. Abbeel

dexterous hands



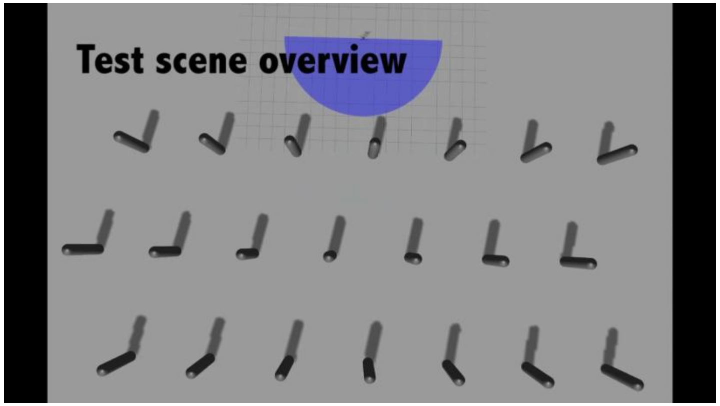
with V. Kumar and E. Todorov

locomotion



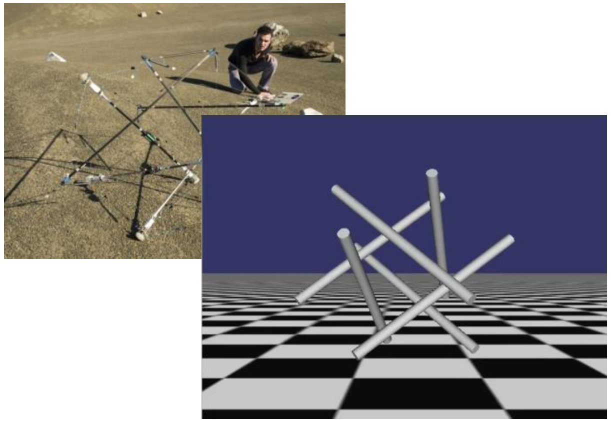
with V. Koltun

aerial vehicles



with G. Kahn, T. Zhang, P. Abbeel

tensegrity robot



with M. Zhang, K. Caluwaerts, P. Abbeel

DAGGER

A simpler way to turn policy search into supervised learning

Requires a "stronger" teacher – must give optimal action $\mathbf{u}$ in any state $\mathbf{x}$

Typically used for imitation learning from a human expert

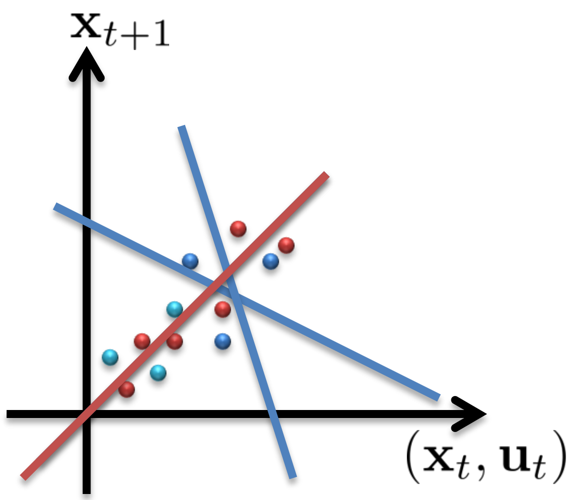
Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in Π.
for $i = 1$ **to** N **do**
  Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
  Sample T-step trajectories using π_i.
  Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by π_i and actions given by expert.
  Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
  Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
Return best $\hat{\pi}_i$ on validation.

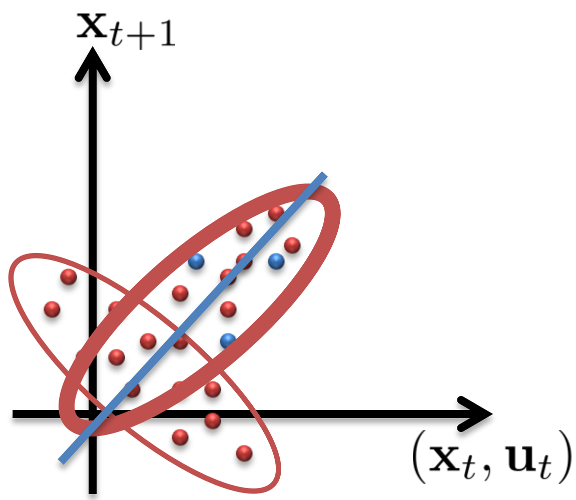
typically 0.0, except when i = 1, then 1.0

DAGGER Video

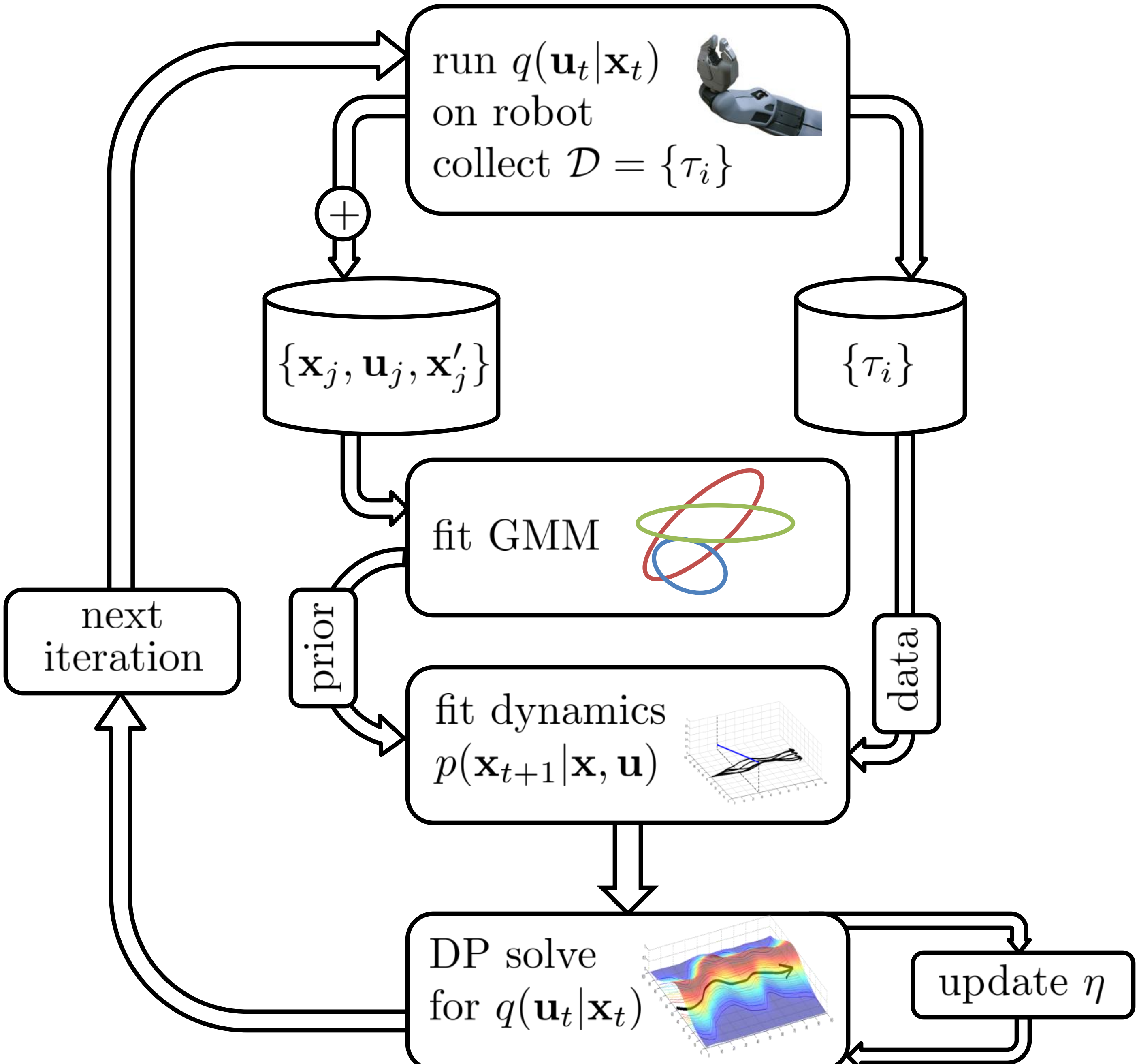
See http://videolectures.net/aistats2011_ross_reduction/

Trajectory Optimization – Dynamics Fitting





1. Run time-varying policy $p(\mathbf{u}_t|\mathbf{x}_t)$ on robot N times
2. Collect dataset $\mathcal{D} = \{\tau_i\}$ where $\tau_i = \{\mathbf{x}_{1i}, \mathbf{u}_{1i}, \dots, \mathbf{x}_{Ti}, \mathbf{u}_{Ti}\}$
3. For each $t \in \{0, \dots, T-1\}$, fit linear Gaussian $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$
4. Solve control problem to get new $p(\mathbf{u}_t|\mathbf{x}_t)$



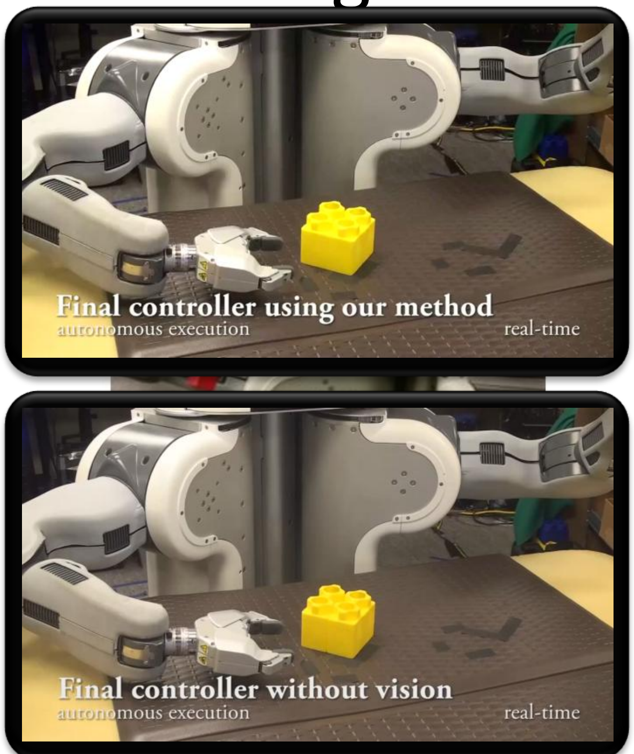
[see L. et al. NIPS '14 for details]

Learned Motion Skills

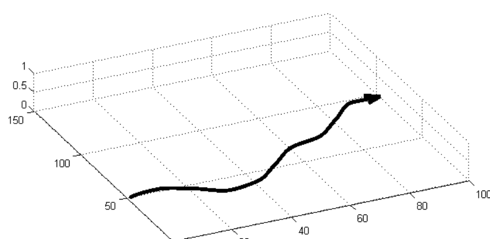
More Visuomotor Experiments

Beyond Instrumented Training

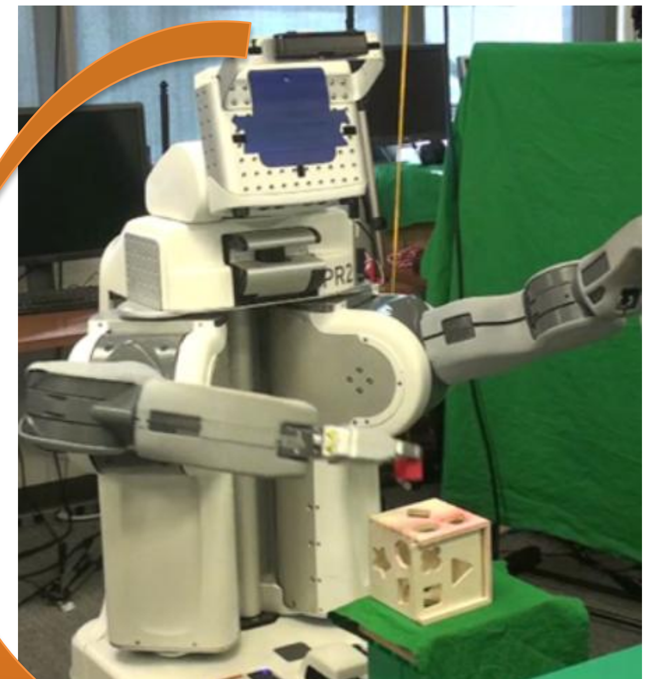
training time



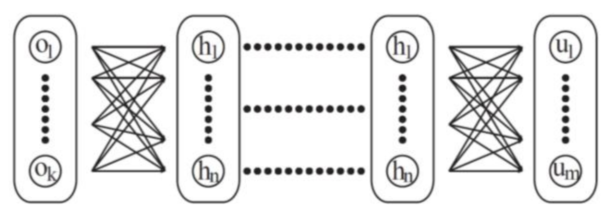
$\mathbf{x}_t \rightarrow \mathbf{u}_t$



test time

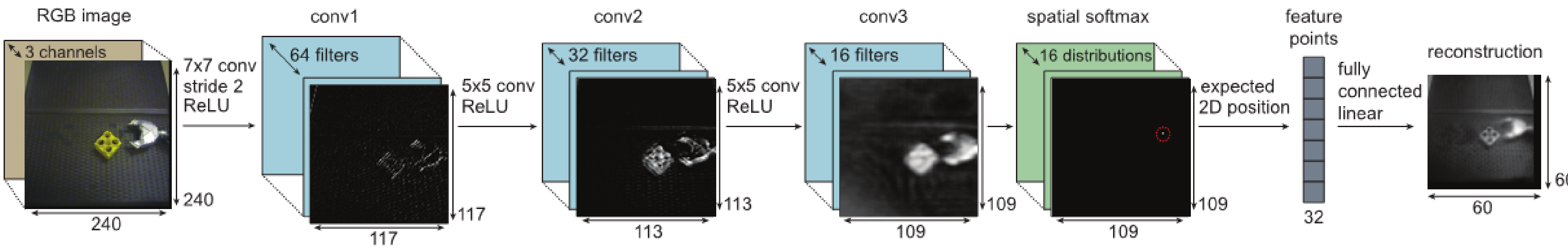


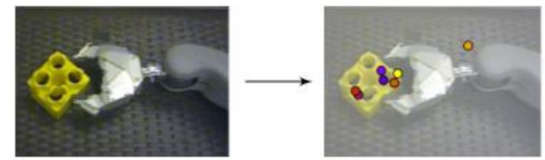
$\mathbf{o}_t \rightarrow \mathbf{u}_t$



Finn, Tan, Duan, Darrell, L., Abbeel ‘15

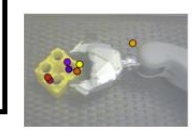
Learning Visual State Spaces





$\mathbf{f}_t$

$$\tilde{\mathbf{x}}_t = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{f}_t \end{bmatrix}$$



Visual State Space Experiments