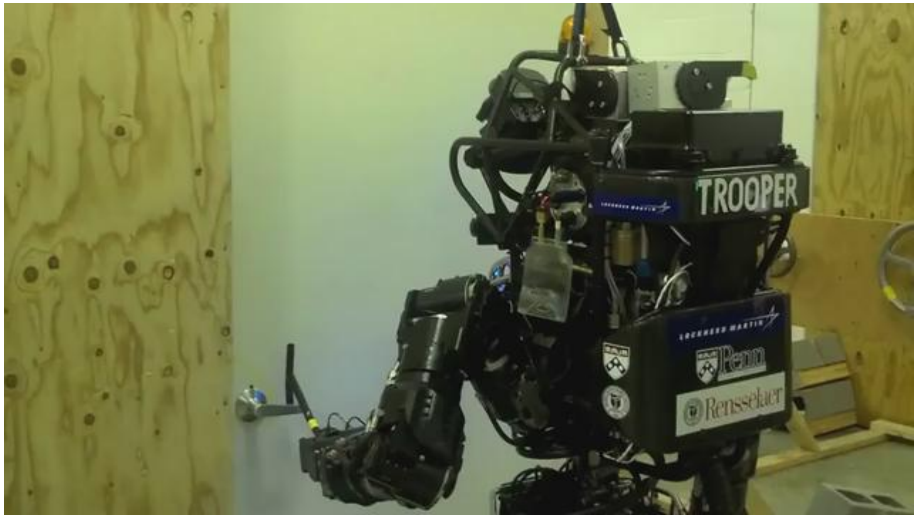
Learning Dynamic Manipulation Skills under Unknown Dynamics with Guided Policy Search

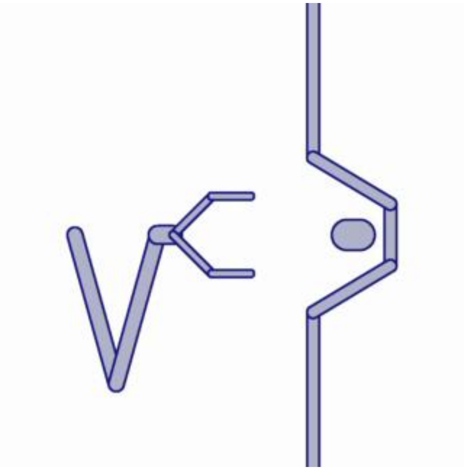
Sergey Levine
UC Berkeley

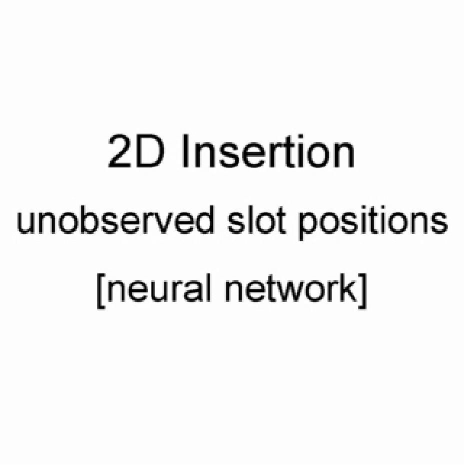
Pieter Abbeel
UC Berkeley



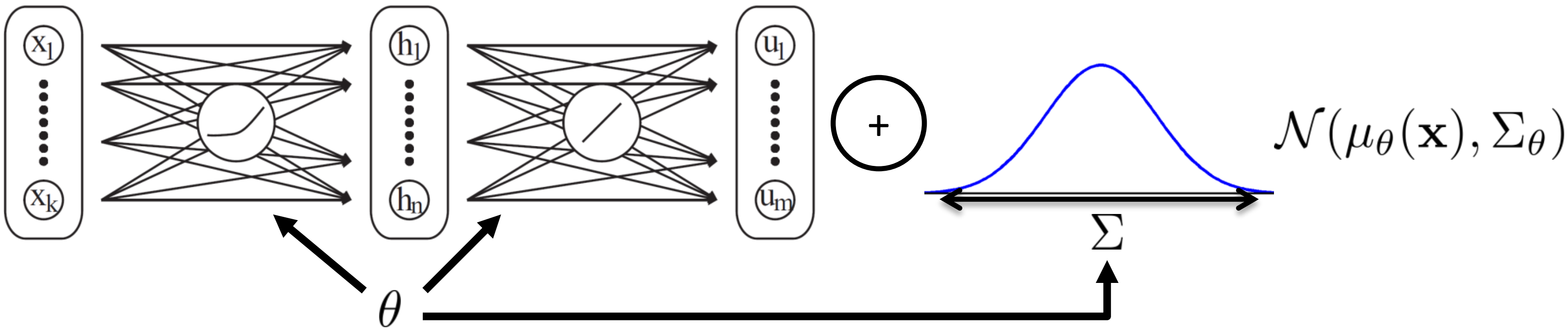
Team TROOPER: Lockheed Martin, University of Pennsylvania, Rensselaer Polytechnic Institute

Philipp Krahenbuhl, Stanford University





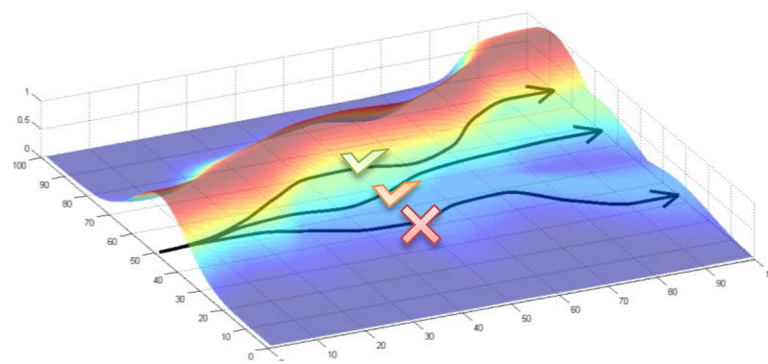
general-purpose neural network controller



$$\theta = \arg\min_\theta J(\theta)$$

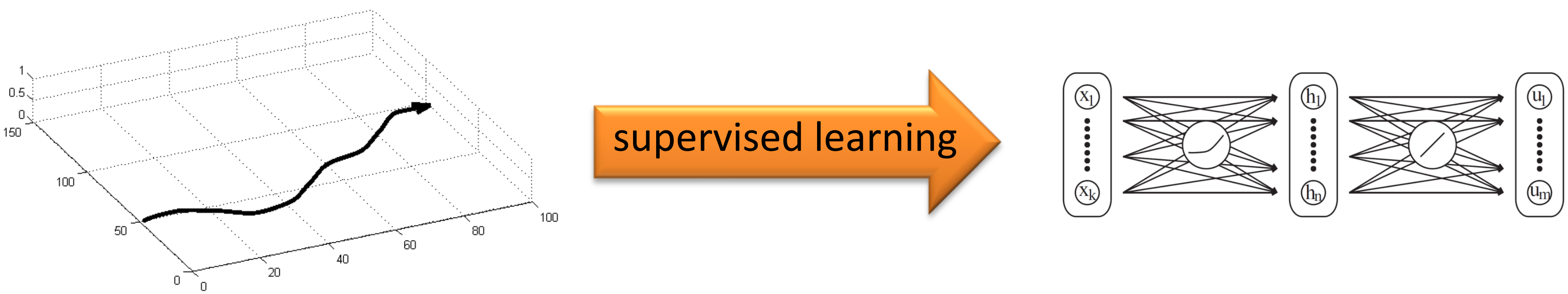
$$J(\theta) = E_{\pi_\theta}[\sum_{t=1}^{T} c(\mathbf{x}_t, \mathbf{u}_t)]$$

$$\pi_\theta(\mathbf{u}_t|\mathbf{x}_t) \text{ -- control policy}$$

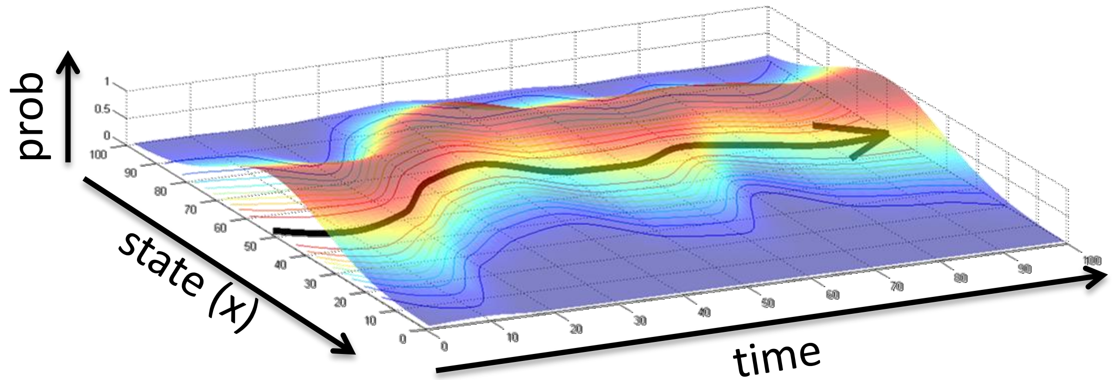


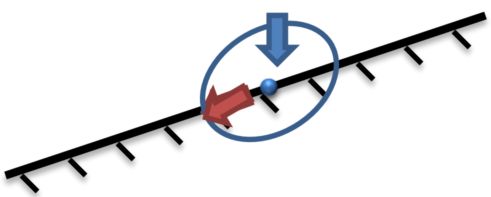
policy search (RL)	complex dynamics	complex policy	HARD
supervised learning	~~complex dynamics~~	complex policy	EASY
trajectory optimization	complex dynamics	~~complex policy~~	EASY

trajectory optimization

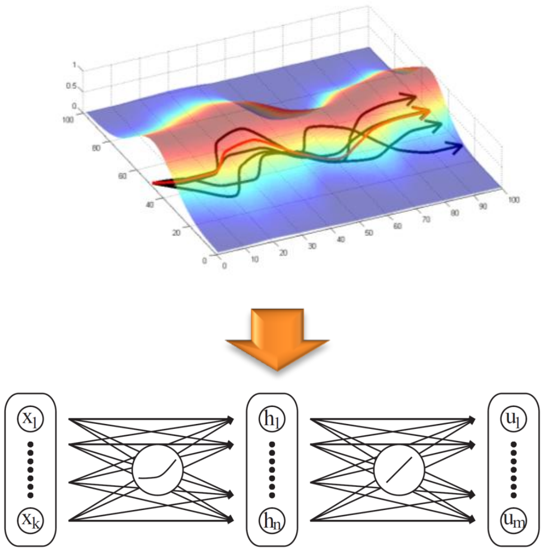


Trajectory Optimization

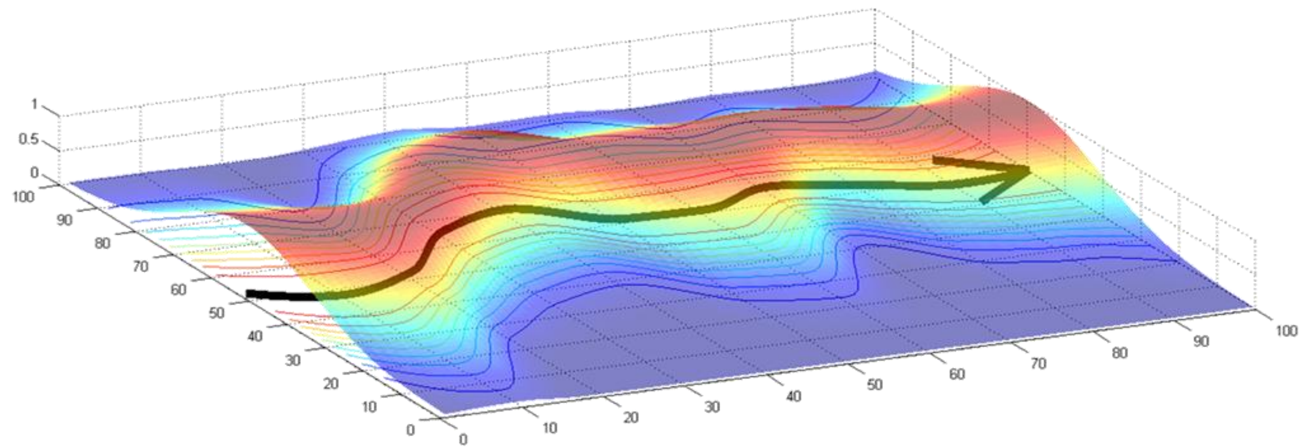




guided policy search

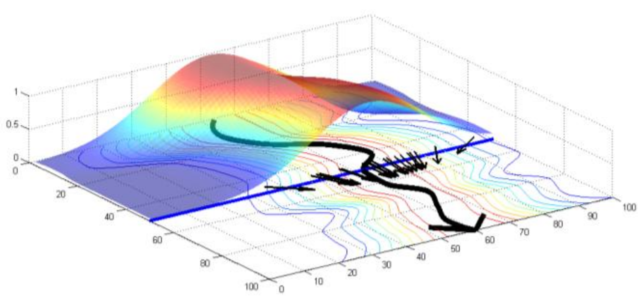


Trajectory Optimization



$$\min_{q(\tau)} E_q[c(\tau)] - \mathcal{H}(q)$$

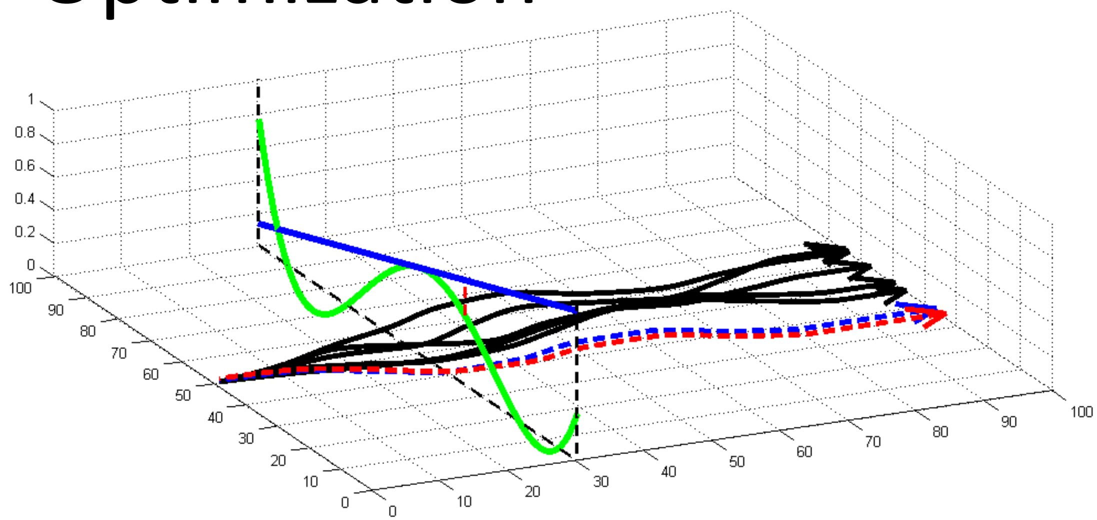
approximate solution using iterative LQR
(similar to extended Kalman filter)



- locally linear dynamics
- locally quadratic cost
- Gaussian distribution

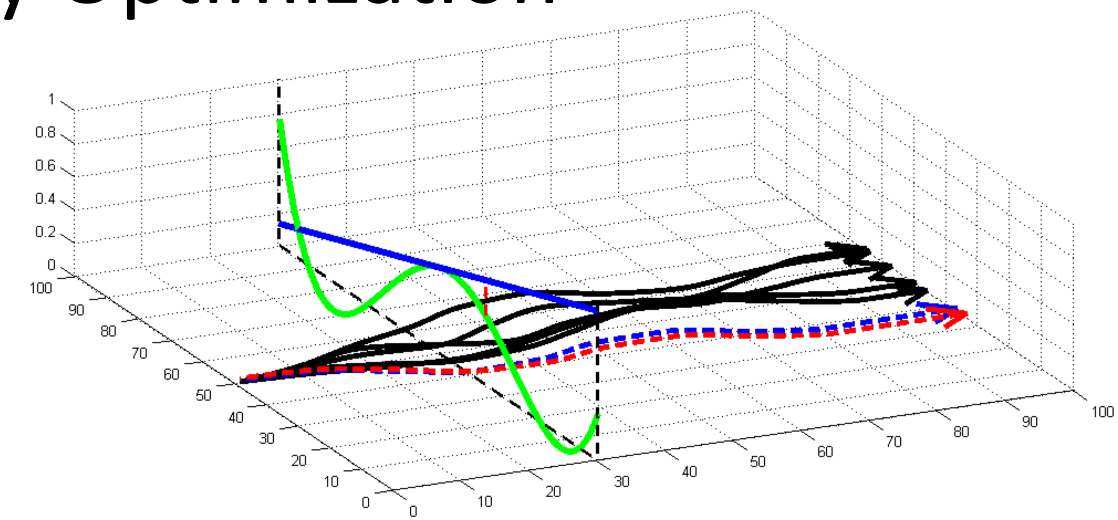
$$q(\mathbf{u}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{k}_t + \mathbf{K}_t\mathbf{x}_t, \Sigma_t)$$

Trajectory Optimization



1. Run time-varying policy $q(\mathbf{u}_t|\mathbf{x}_t)$ 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 $q(\mathbf{u}_t|\mathbf{x}_t)$

Trajectory Optimization

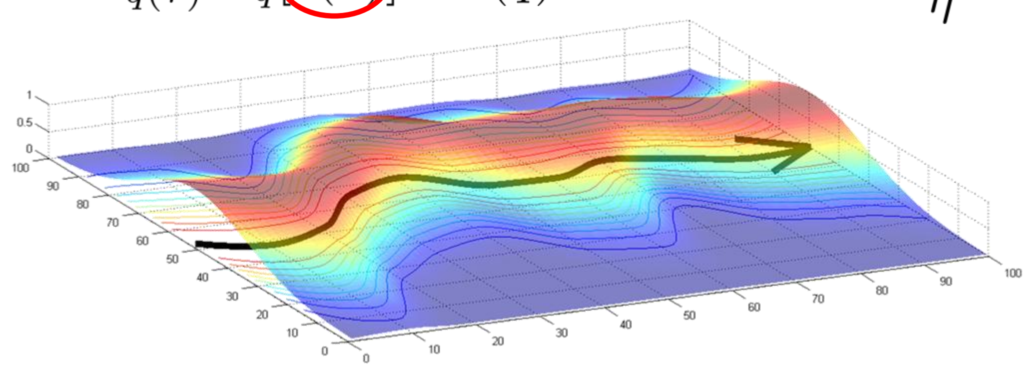


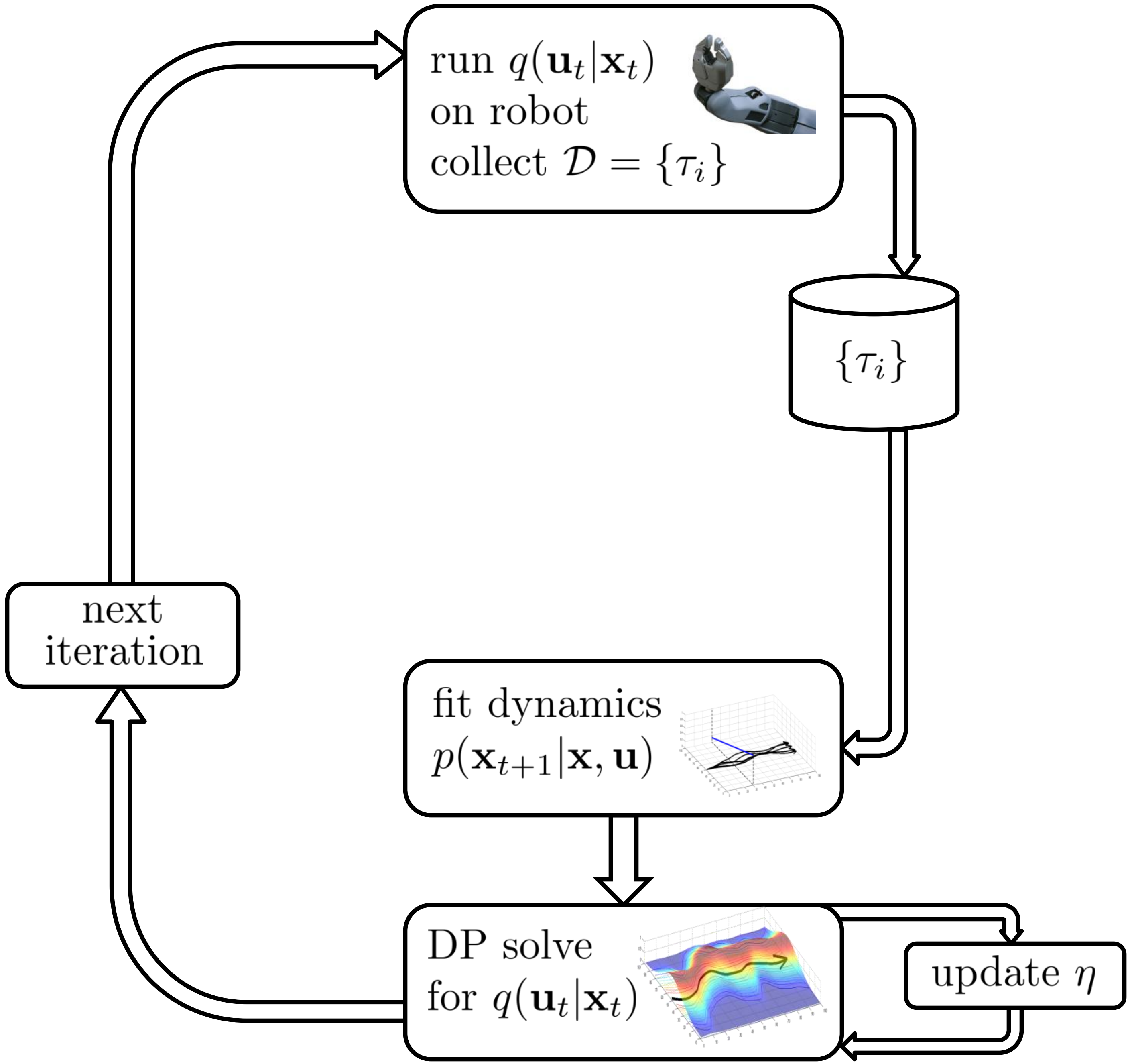
$$\min_{q(\tau)} E_q[c(\tau)] \text{ s.t. } D_{KL}(\underbrace{q(\tau)}_{\text{new}} \| \underbrace{\bar{q}(\tau)}_{\text{old}}) \le \epsilon$$

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

$$\min_{q(\tau)} E_q[c(\tau)] - \mathcal{H}(q)$$

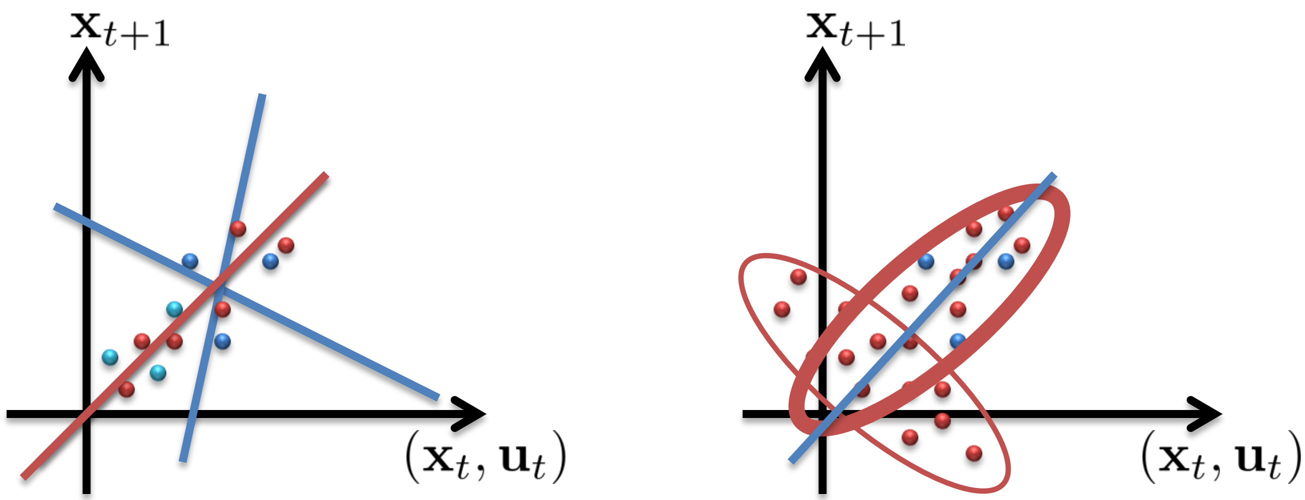
$$\tilde{c}(\tau) = \frac{1}{\eta}c(\tau) - \log \bar{q}(\tau)$$



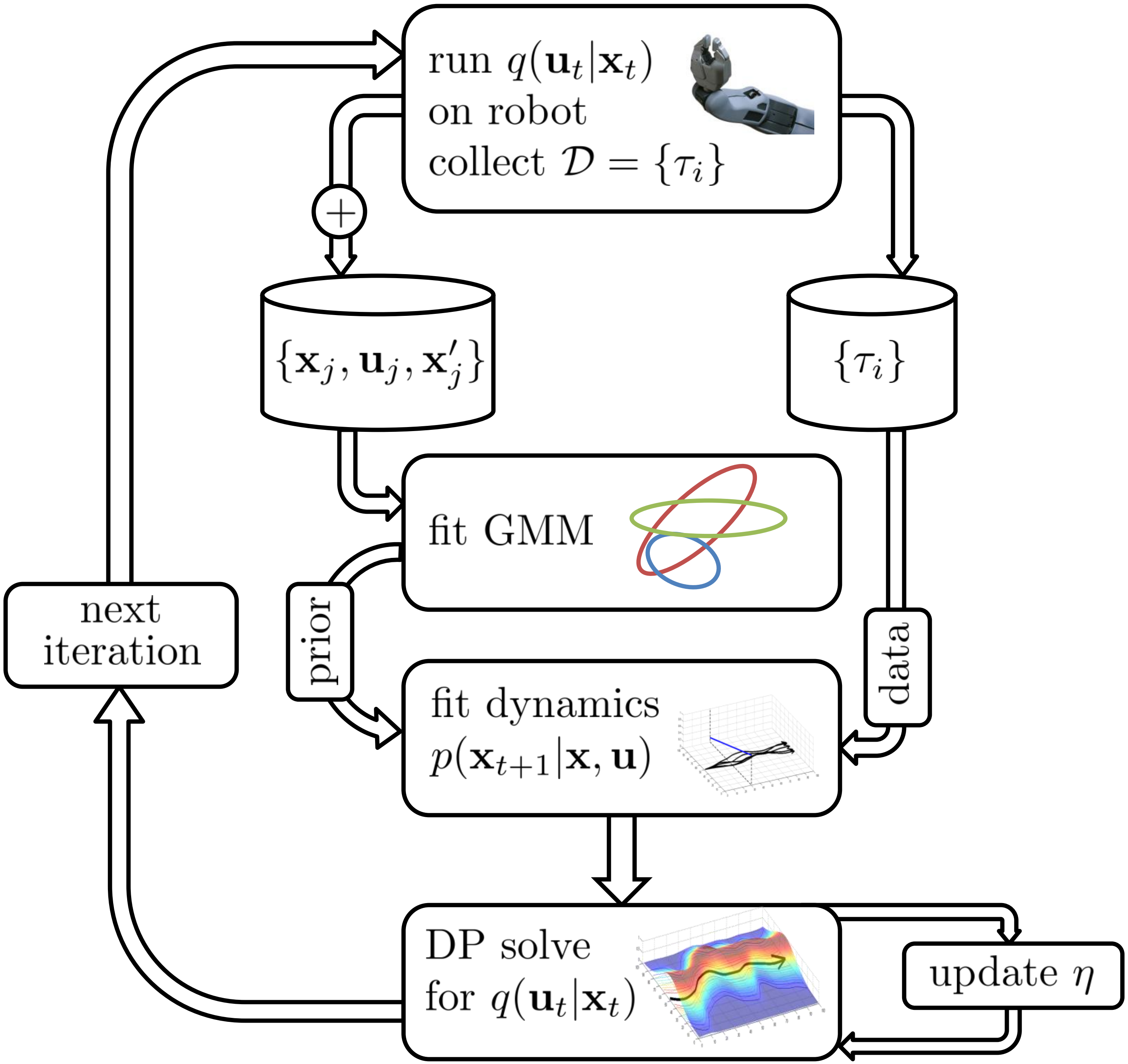


run $q(\mathbf{u}_t|\mathbf{x}_t)$
on robot
collect $\mathcal{D} = \{\tau_i\}$
$\{\tau_i\}$
fit dynamics
$p(\mathbf{x}_{t+1}|\mathbf{x}, \mathbf{u})$
DP solve
for $q(\mathbf{u}_t|\mathbf{x}_t)$
update η
next iteration

Trajectory Optimization



1. Run time-varying policy $q(\mathbf{u}_t|\mathbf{x}_t)$ 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 $q(\mathbf{u}_t|\mathbf{x}_t)$


run $q(\mathbf{u}_t|\mathbf{x}_t)$
on robot
collect $\mathcal{D} = \{\tau_i\}$
$\{\mathbf{x}_j, \mathbf{u}_j, \mathbf{x}'_j\}$
$\{\tau_i\}$
fit GMM
prior
data
fit dynamics
$p(\mathbf{x}_{t+1}|\mathbf{x}, \mathbf{u})$
next
iteration
DP solve
for $q(\mathbf{u}_t|\mathbf{x}_t)$
update η

Trajectory Optimization

2D Insertion

optimized trajectory

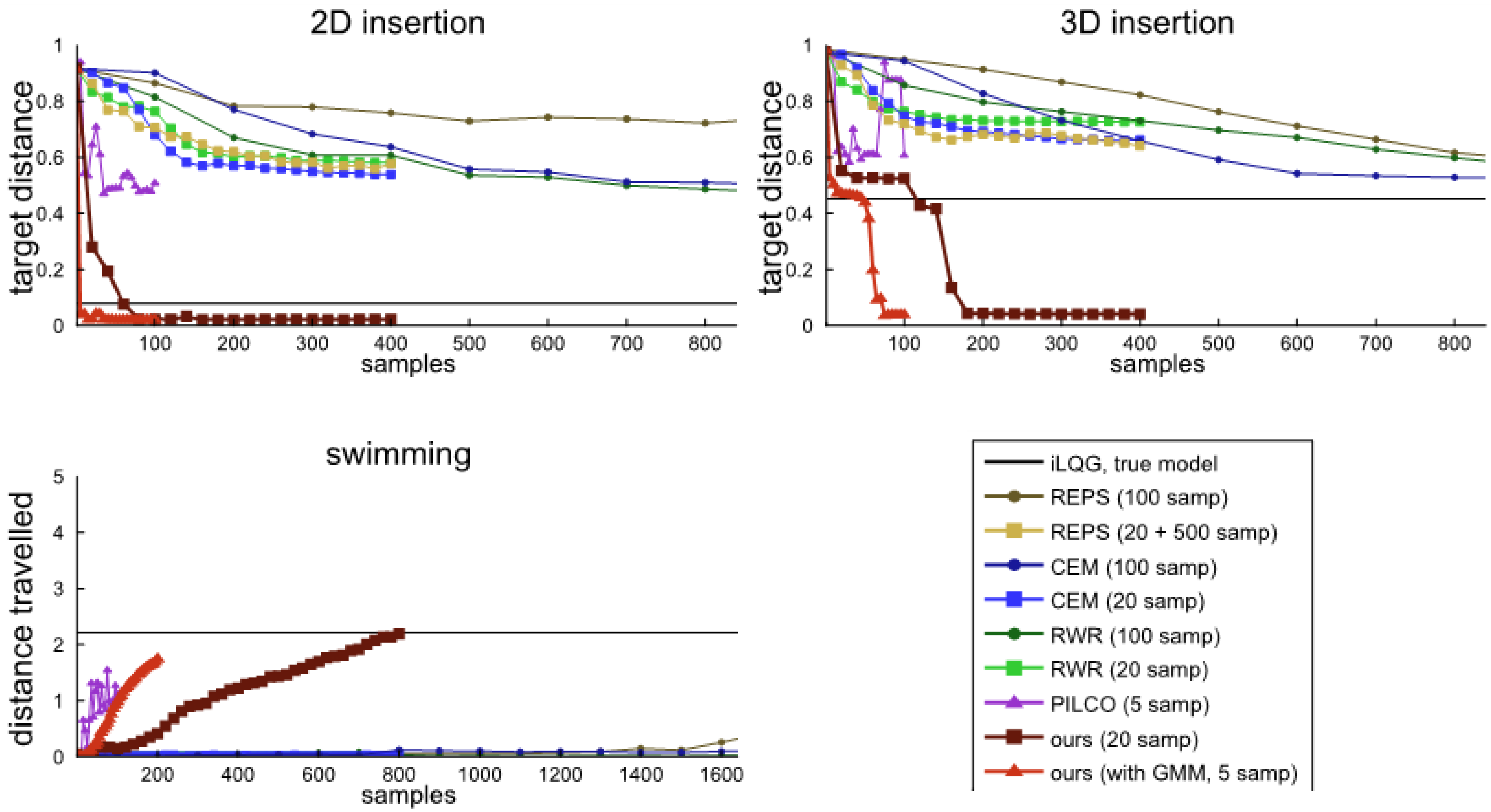
linear Gaussian control

Swimming

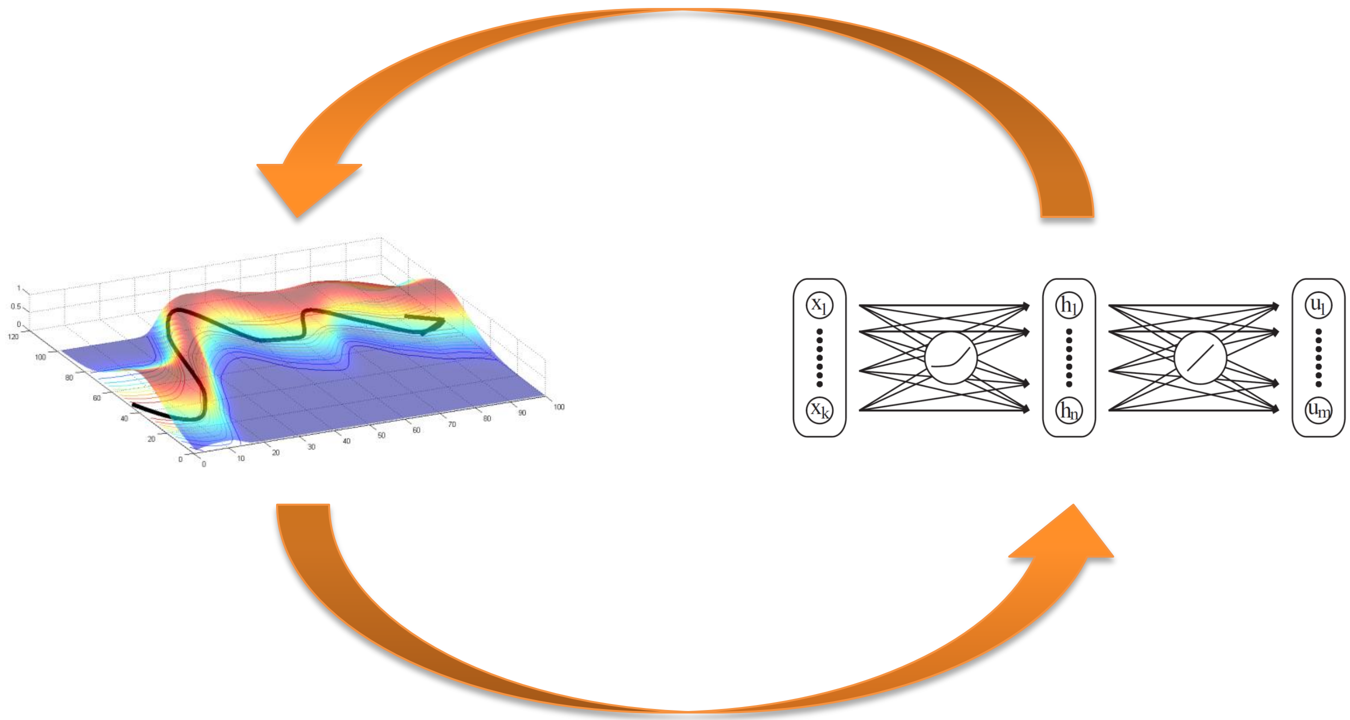
optimized trajectory

[linear Gaussian]

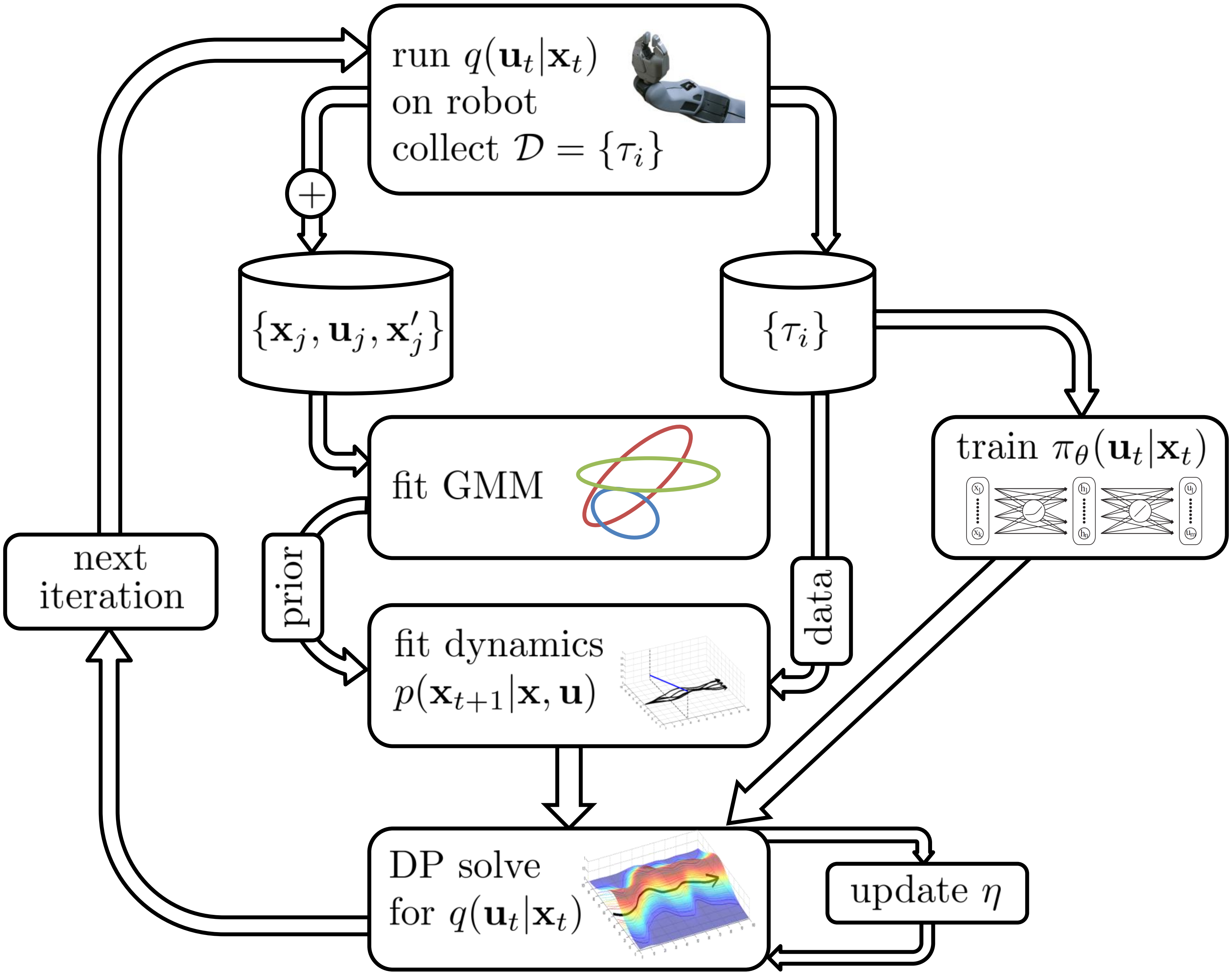
Trajectory Optimization



Guided Policy Search



see Levine & Koltun, ICML 2014



run $q(\mathbf{u}_t|\mathbf{x}_t)$
on robot
collect $\mathcal{D} = \{\tau_i\}$
$\{\mathbf{x}_j, \mathbf{u}_j, \mathbf{x}'_j\}$
$\{\tau_i\}$
fit GMM
train $\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)$
next iteration
prior
data
fit dynamics
$p(\mathbf{x}_{t+1}|\mathbf{x}, \mathbf{u})$
DP solve
for $q(\mathbf{u}_t|\mathbf{x}_t)$
update η

2D Insertion

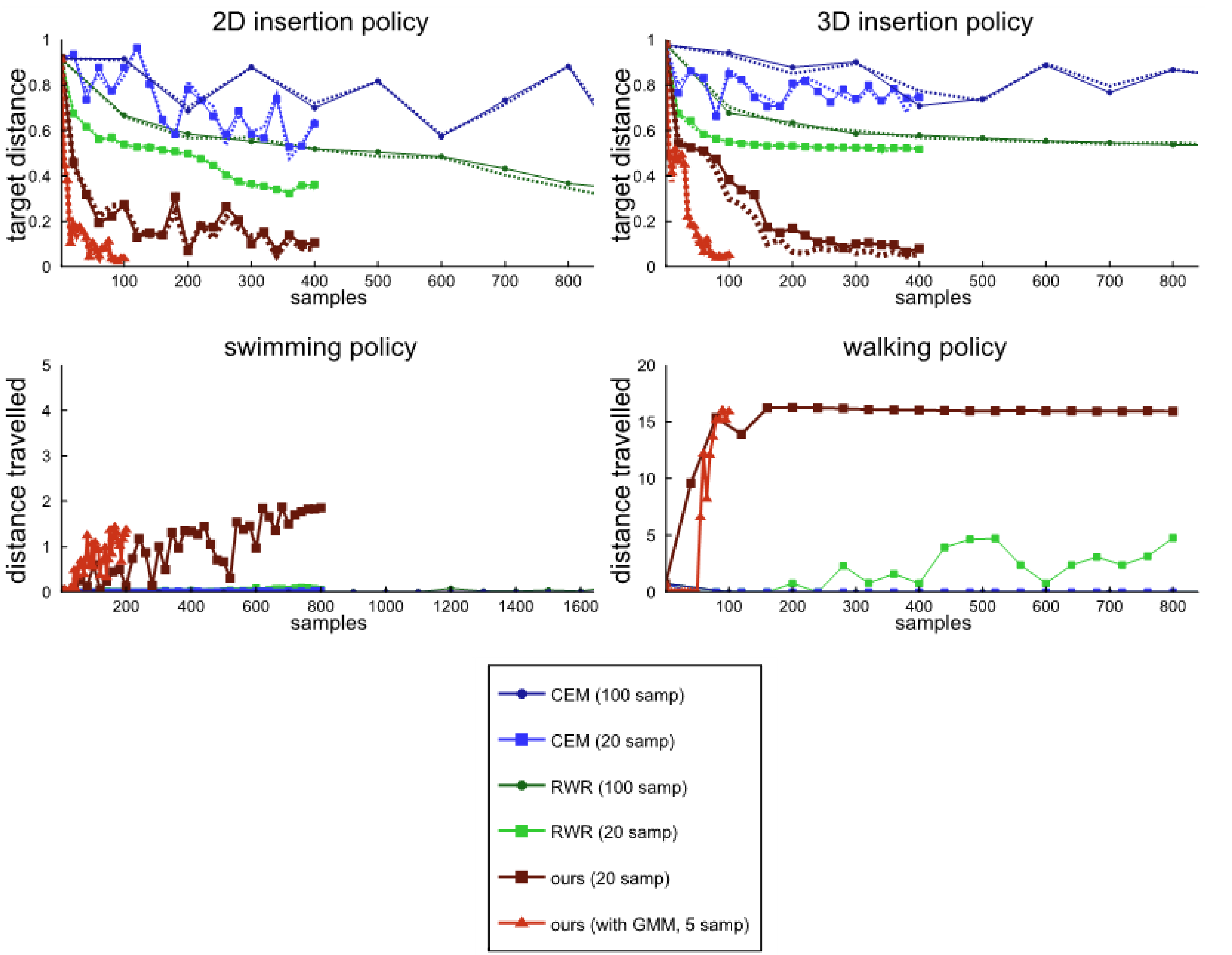
unobserved slot positions

[neural network]

Swimming

learned policy

[neural network]


2D insertion policy
target distance
0
0.2
0.4
0.6
0.8
1
100
200
300
400
500
600
700
800
samples
3D insertion policy
target distance
0
0.2
0.4
0.6
0.8
1
100
200
300
400
500
600
700
800
samples
swimming policy
distance travelled
0
1
2
3
4
5
200
400
600
800
1000
1200
1400
1600
samples
walking policy
distance travelled
0
5
10
15
20
100
200
300
400
500
600
700
800
samples
CEM (100 samp)
CEM (20 samp)
RWR (100 samp)
RWR (20 samp)
ours (20 samp)
ours (with GMM, 5 samp)

Concluding Comments

- simple <u>linear</u> dynamics model
- fast, simple, standard LQR solver
- can handle contacts despite linear model
- fit very complex policies with guided policy search