End-to-End Training of Deep Visuomotor Policies

Sergey Levine*, Chelsea Finn*, Trevor Darrell, Pieter Abbeel.

JMLR 17, 2016

(335 cites)

Visuomotor?

visuomotor

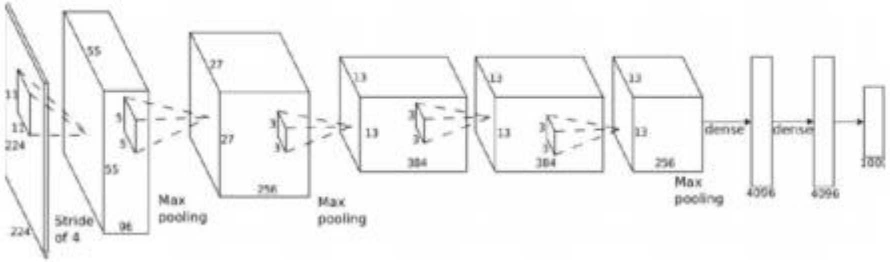
Also found in: Dictionary.

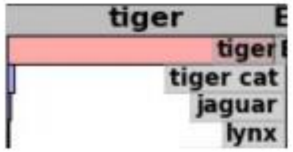
vis·u·o·mo·tor (viz'yū-ō-mō'tŏr),

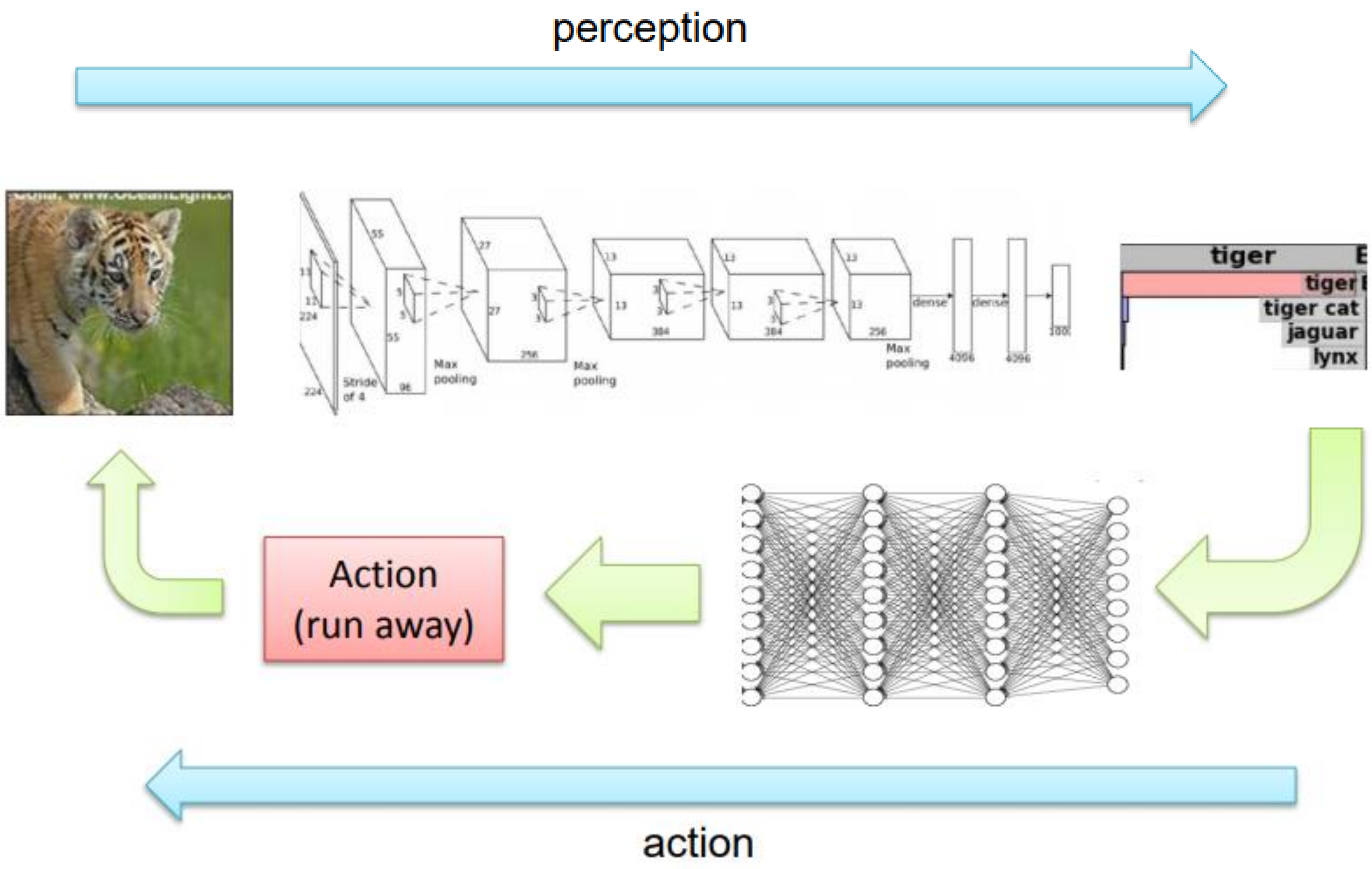
Denoting the ability to synchronize visual information with physical movement, for example, driving a car or playing a video game of skill.

"CITE" Farlex Partner Medical Dictionary © Farlex 2012

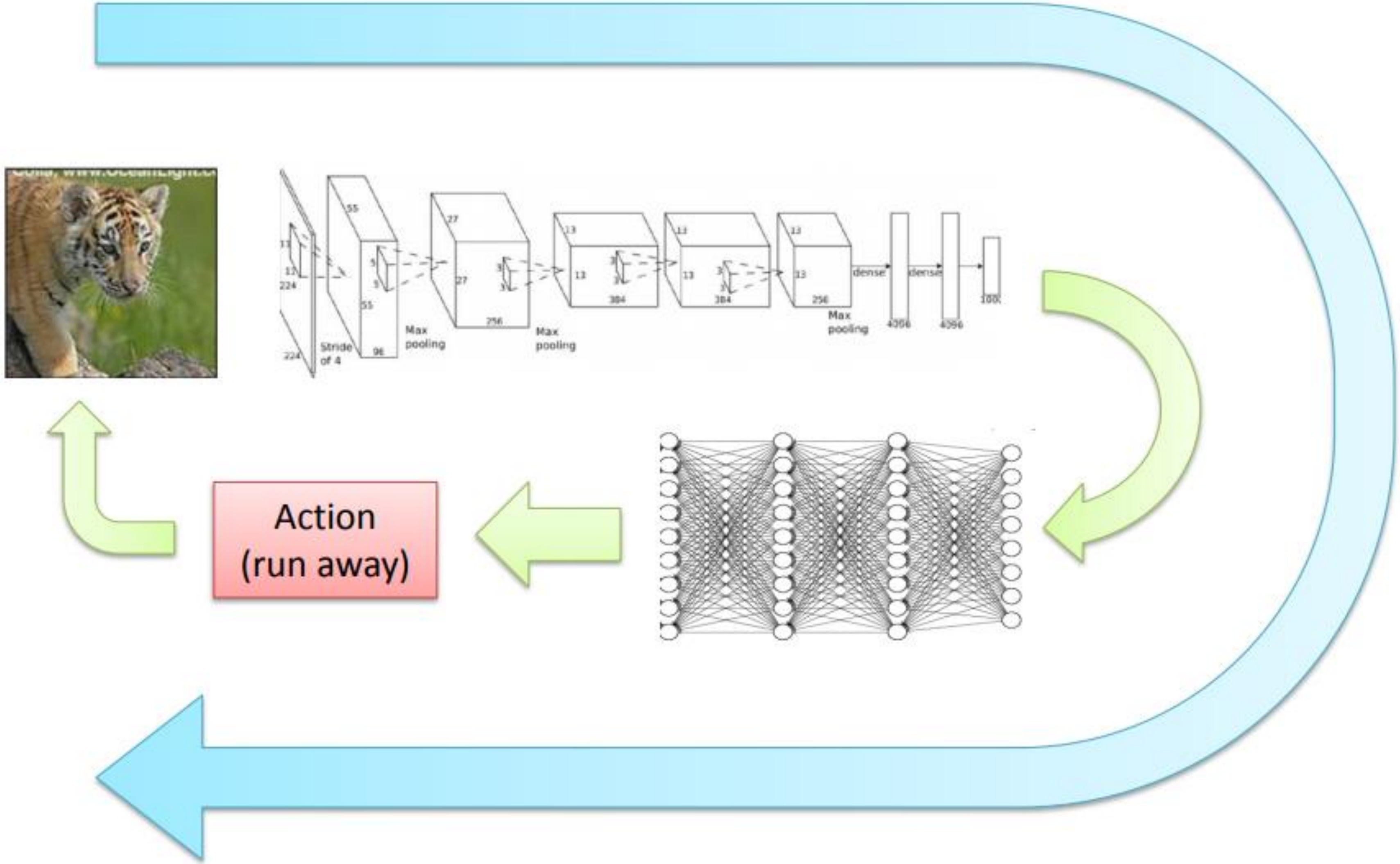
시각(카메라) 정보를 통해 신체(모터)를 움직이는 것이라고 봐도 무방할 듯

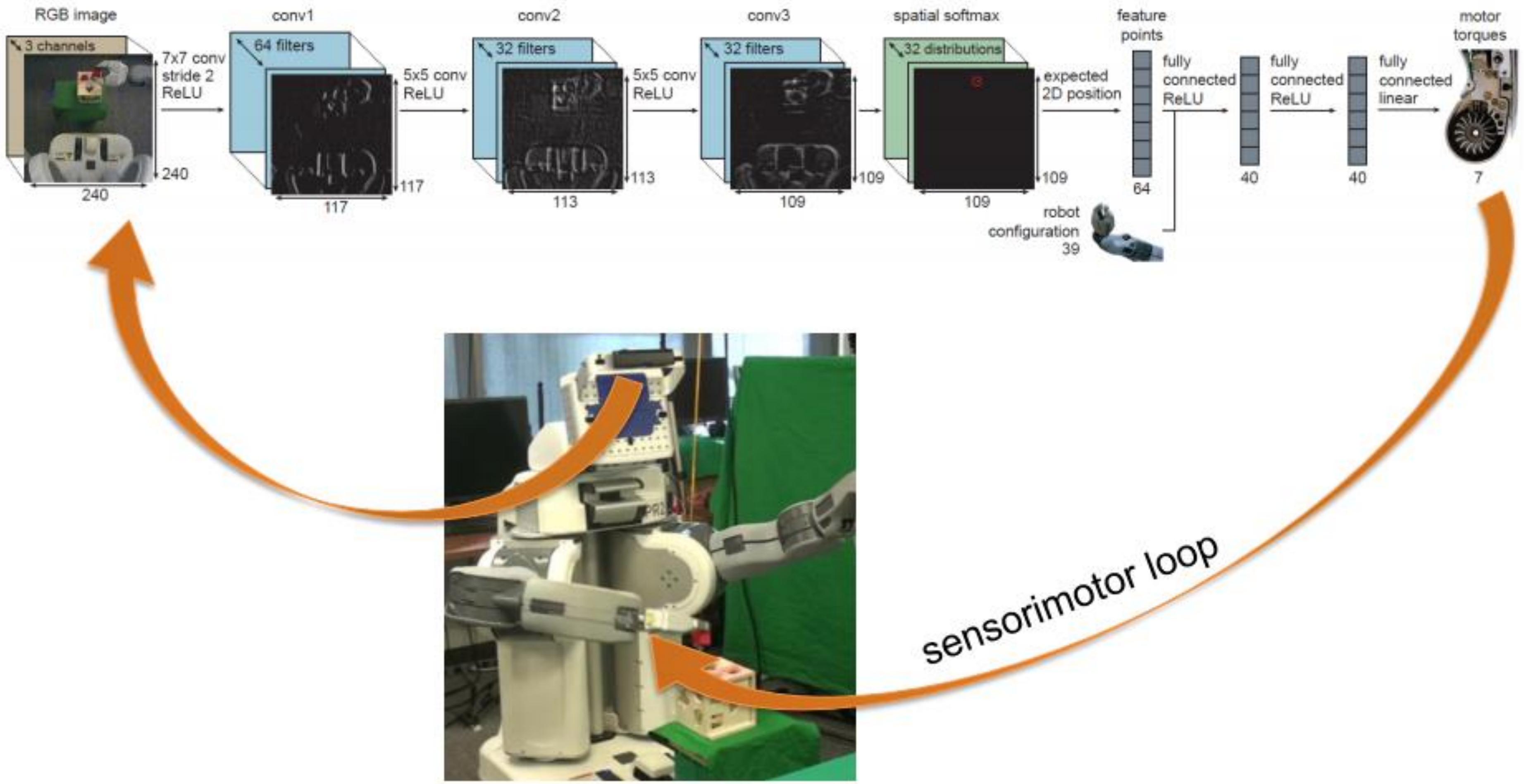

224
224
Stride of 4
55
55
96
Max pooling
27
27
256
Max pooling
13
384
13
384
13
256
Max pooling
dense
dense
4096
4096

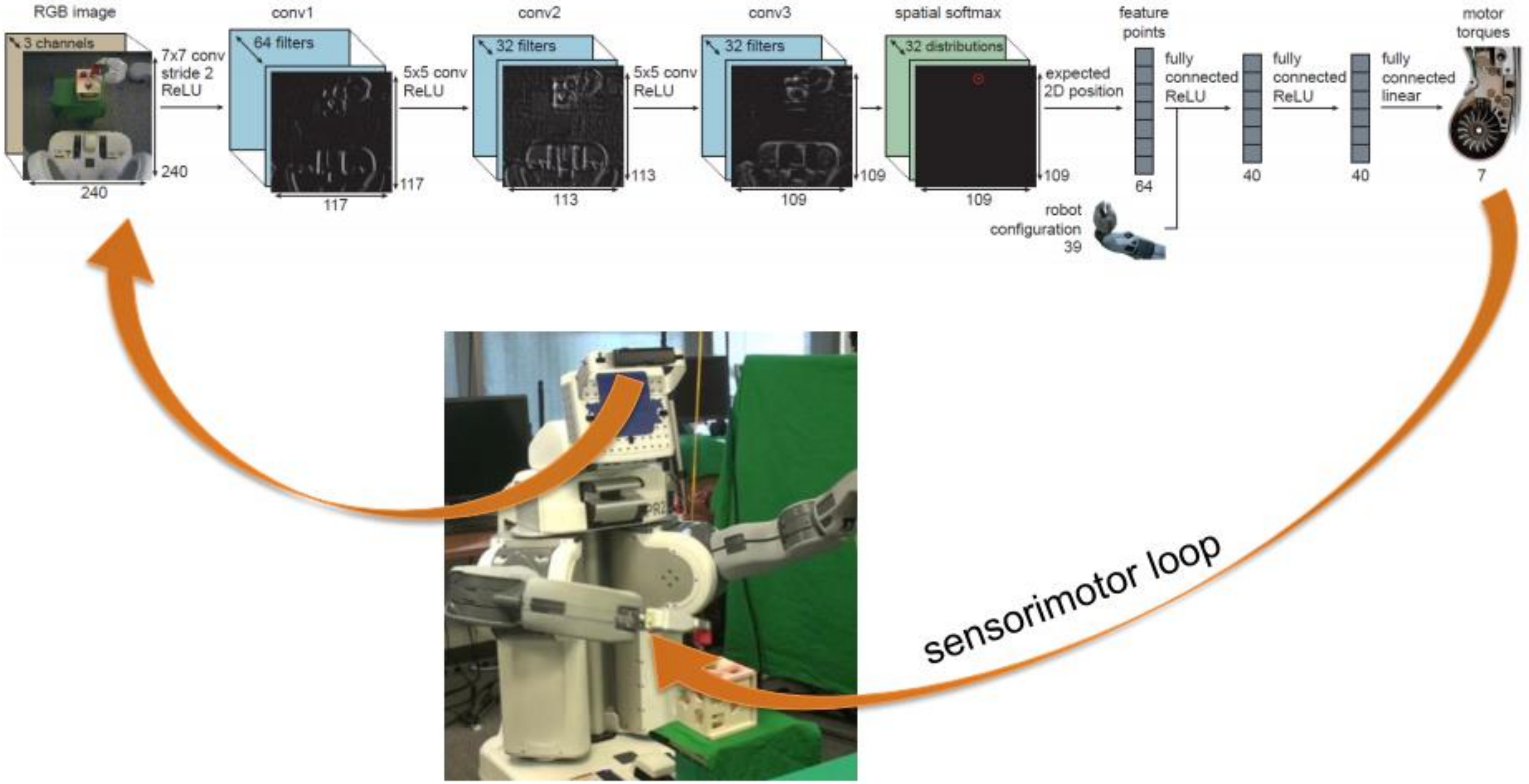

tiger
tiger
tiger cat
jaguar
lynx


perception
Stride of 4
Max pooling
Max pooling
Max pooling
dense
dense
tiger
tiger
tiger cat
jaguar
lynx
Action
(run away)
action

sensorimotor loop



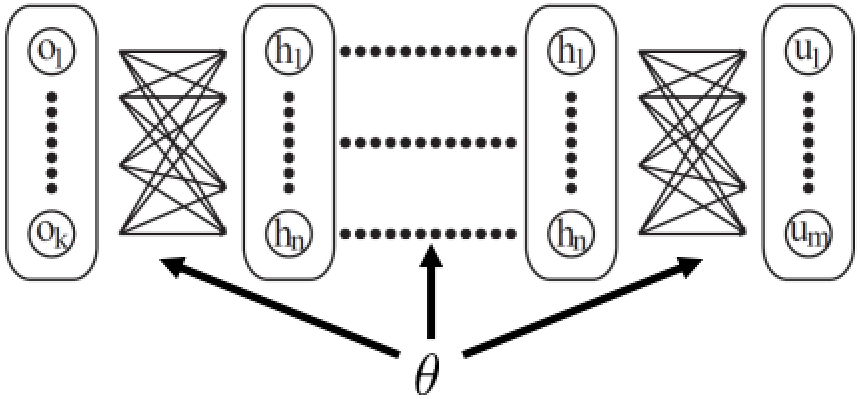

RGB image
3 channels
240
240
7x7 conv
stride 2
ReLU
conv1
64 filters
117
117
5x5 conv
ReLU
conv2
32 filters
113
113
5x5 conv
ReLU
conv3
32 filters
109
109
spatial softmax
32 distributions
109
109
expected
2D position
feature
points
64
fully
connected
ReLU
40
fully
connected
ReLU
40
fully
connected
linear
motor
torques
7
robot
configuration
39
sensorimotor loop



no direct supervision

actions have consequences

general-purpose neural network policy

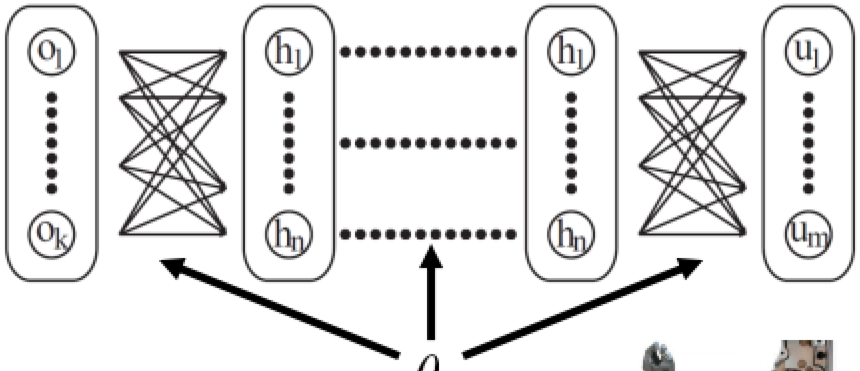


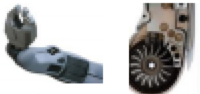
$$\theta = \arg\min_\theta E_{\pi_\theta}[\sum_{t=1}^{T} c(\mathbf{x}_t, \mathbf{u}_t)]$$

$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – control policy

$\mathbf{o}_t$ – observation (may or may not be equal to $\mathbf{x}_t$)

general-purpose neural network policy

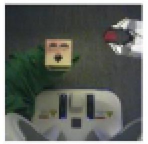




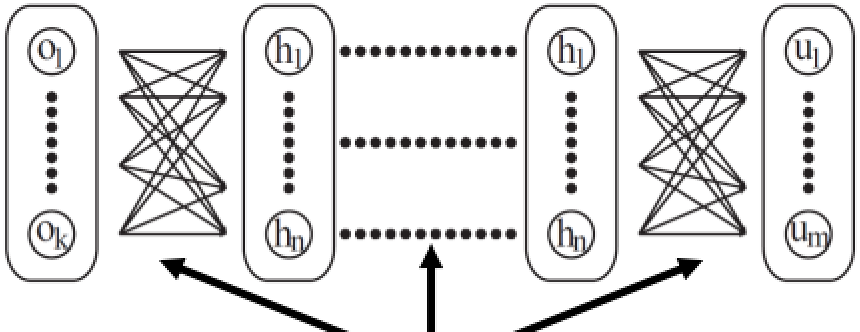
$$\theta = \arg\min_\theta E_{\pi_\theta}[\sum_{t=1}^{T} c(\mathbf{x}_t, \mathbf{u}_t)]$$

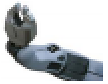
$$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t) \text{ – control policy}$$

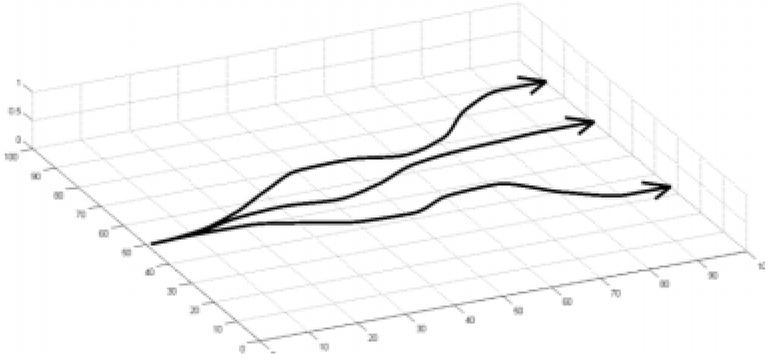
$$\mathbf{o}_t \text{ – observation (may or may not be equal to } \mathbf{x}_t)$$



general-purpose neural network policy



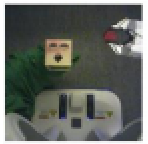




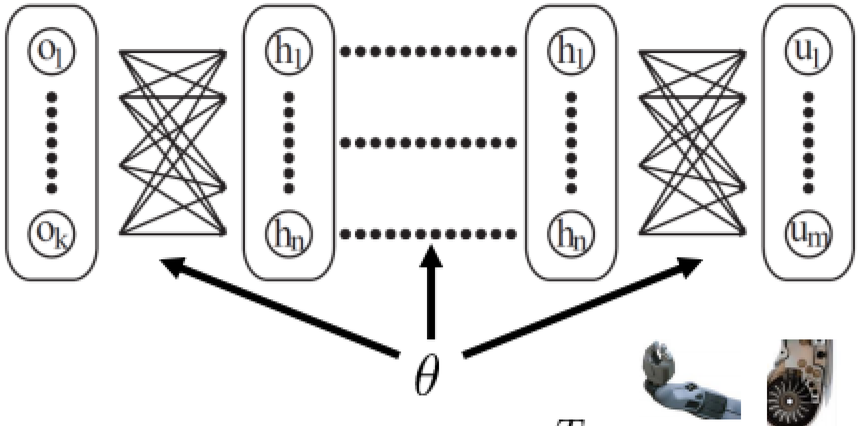
$$\theta = \arg\min_\theta E_{\pi_\theta}[\sum_{t=1}^{T} c(\mathbf{x}_t, \mathbf{u}_t)]$$

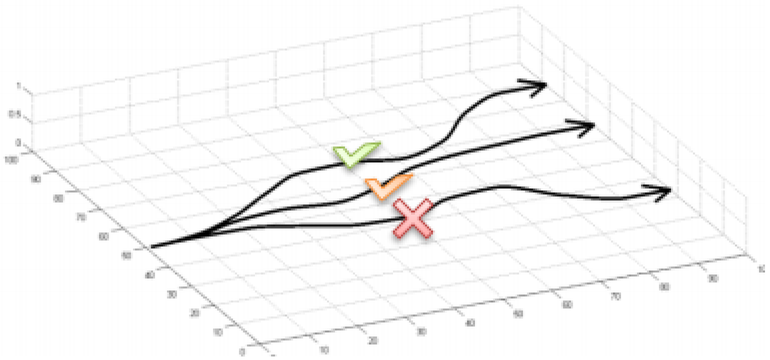
$$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t) \text{ – control policy}$$

$$\mathbf{o}_t \text{ – observation (may or may not be equal to } \mathbf{x}_t)$$



general-purpose neural network policy

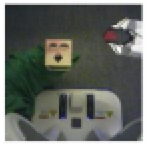




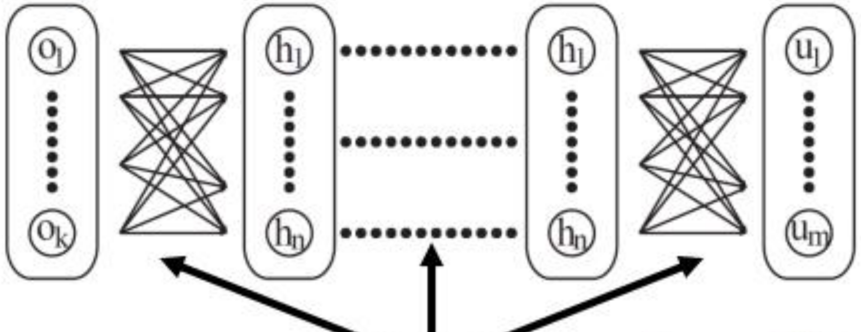
$$\theta = \arg\min_\theta E_{\pi_\theta}[\sum_{t=1}^{T} c(\mathbf{x}_t, \mathbf{u}_t)]$$

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

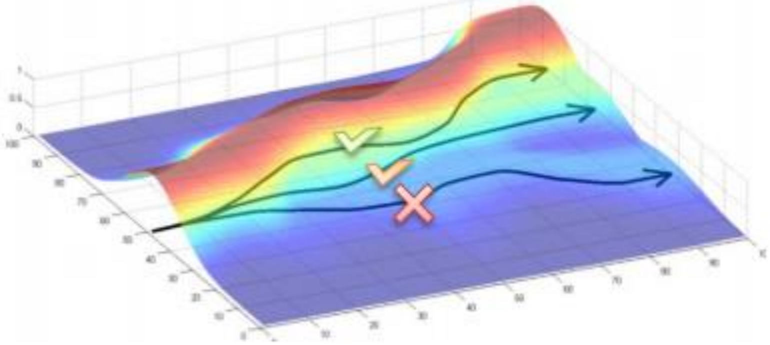
$$\mathbf{o}_t \text{ -- observation (may or may not be equal to } \mathbf{x}_t)$$



general-purpose neural network policy



θ

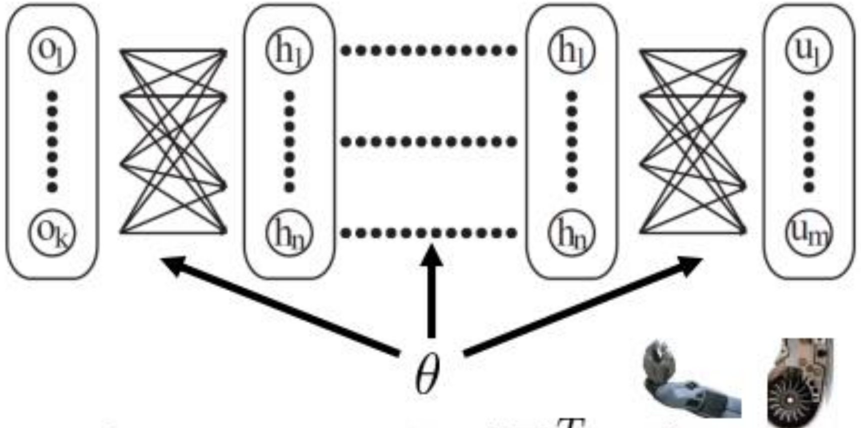


$$\theta = \arg\min_\theta E_{\pi_\theta}[\sum_{t=1}^{T} c(\mathbf{x}_t, \mathbf{u}_t)]$$

$$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t) \text{ – control policy}$$

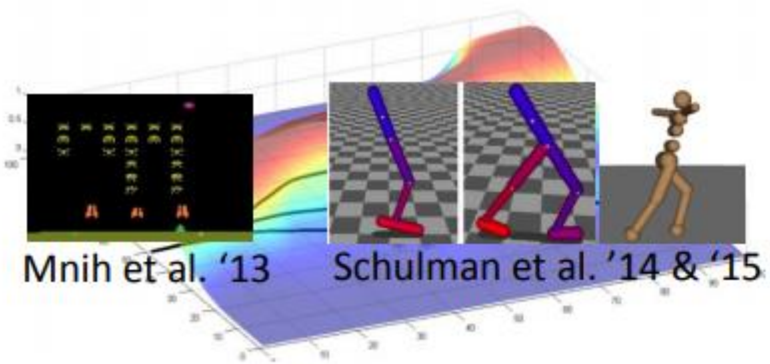
$$\mathbf{o}_t \text{ – observation (may or may not be equal to } \mathbf{x}_t)$$

general-purpose neural network policy



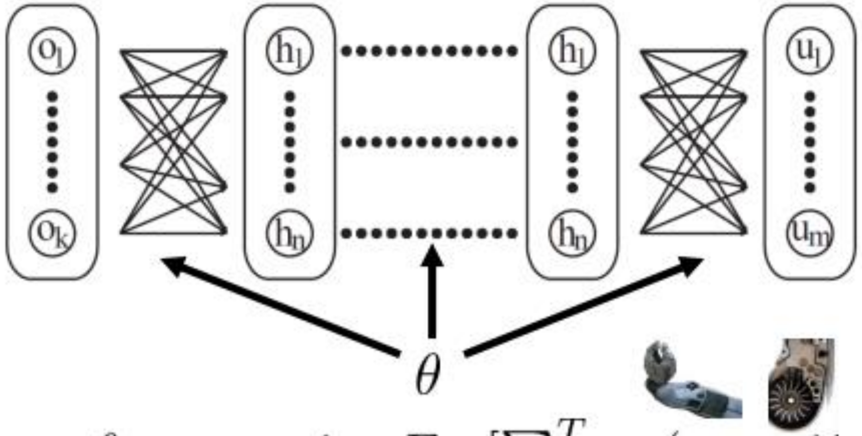
$$\theta = \arg\min_\theta E_{\pi_\theta}[\textstyle\sum_{t=1}^T c(\mathbf{x}_t, \mathbf{u}_t)]$$

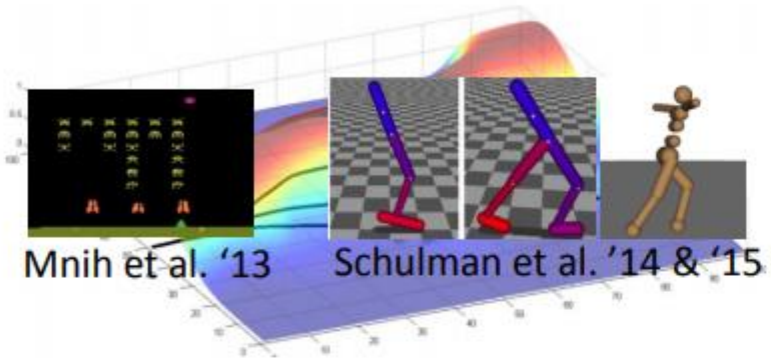
$$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t) \text{ -- control policy}$$



$\mathbf{o}_t$ – observation (may or may not be equal to $\mathbf{x}_t$)

general-purpose neural network policy





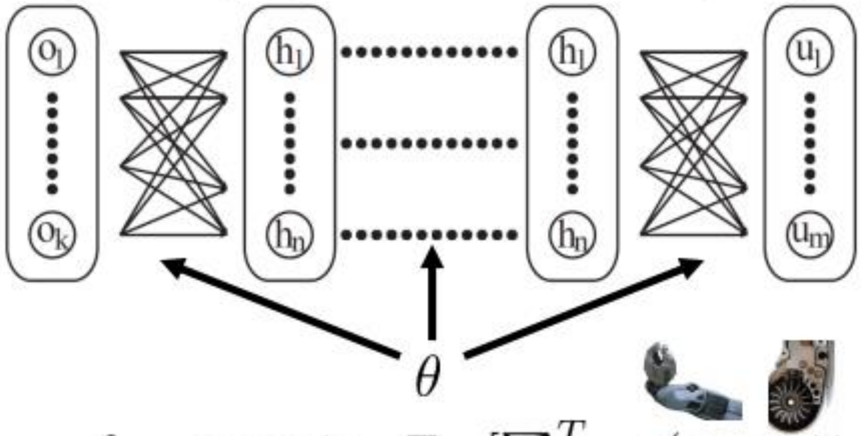
$$\theta = \arg\min_\theta E_{\pi_\theta}[\sum_{t=1}^{T} c(\mathbf{x}_t, \mathbf{u}_t)]$$

$$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t) \text{ – control policy}$$

$$\mathbf{o}_t \text{ – observation (may or may not be equal to } \mathbf{x}_t)$$

policy search (RL) complex dynamics complex policy HARD

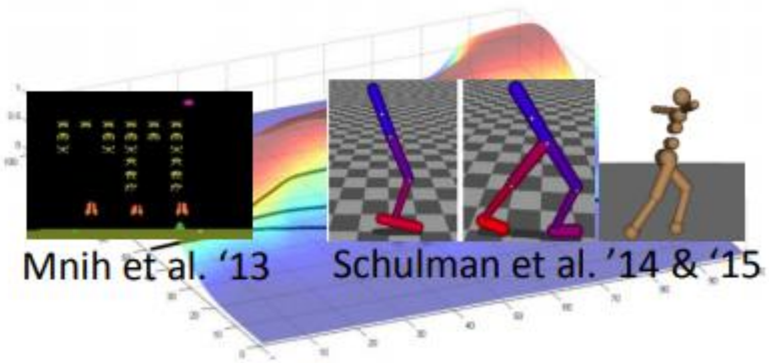
general-purpose neural network policy



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

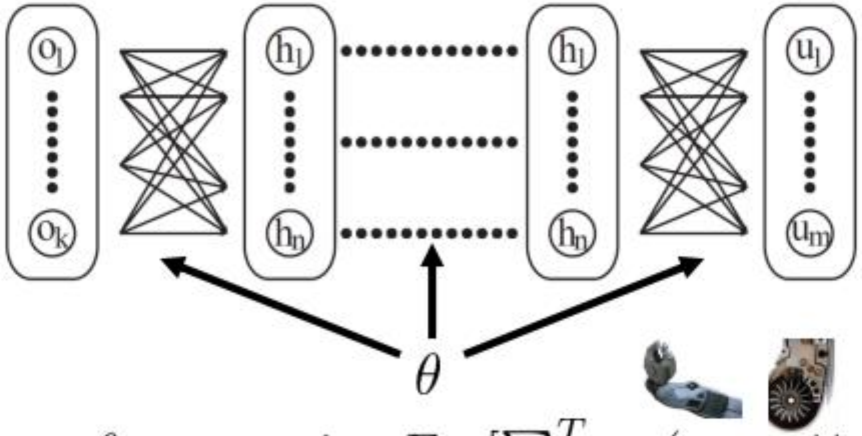
$$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t) \text{ – control policy}$$

$$\mathbf{o}_t \text{ – observation (may or may not be equal to } \mathbf{x}_t)$$



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

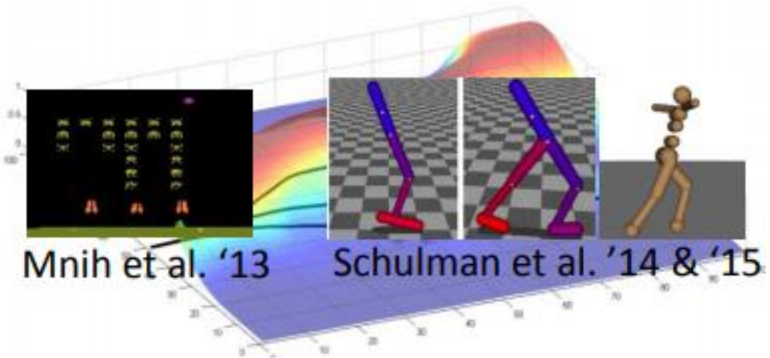
general-purpose neural network policy



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

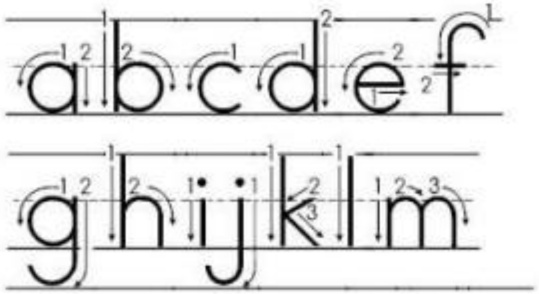
$$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t) \text{ – control policy}$$

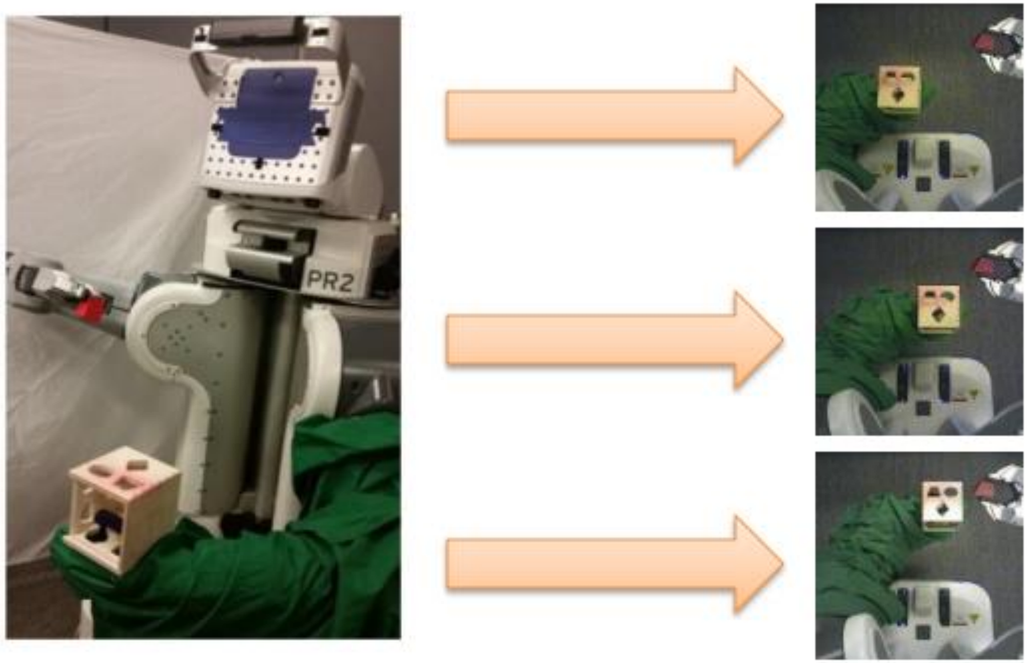
$$\mathbf{o}_t \text{ – observation (may or may not be equal to } \mathbf{x}_t)$$



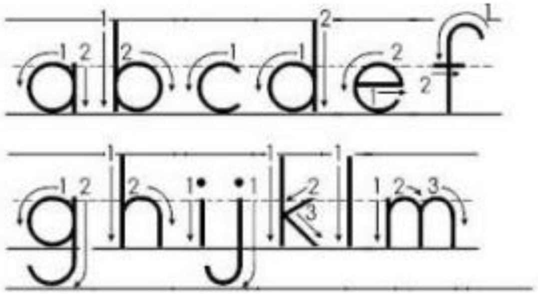
policy search (RL)	complex dynamics	complex policy	HARD
supervised learning	~~complex dynamics~~	complex policy	EASY
optimal control	complex dynamics	~~complex policy~~	EASY

1. break up the task: separately solve N different task instances



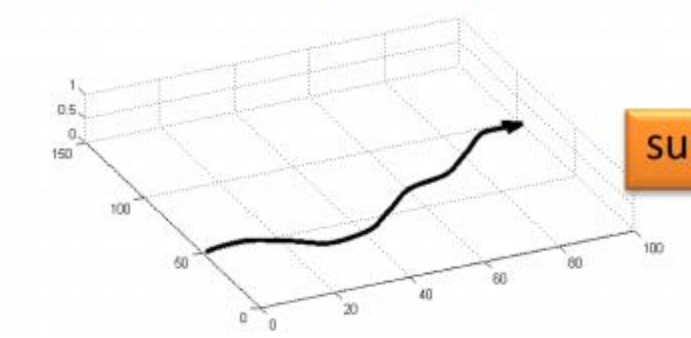


1. break up the task: separately solve N different task instances



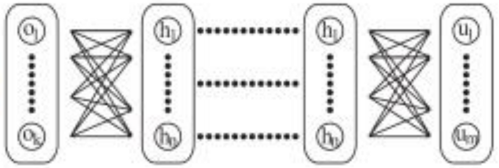
2. use supervised learning

trajectory-centric RL
(fully observed)



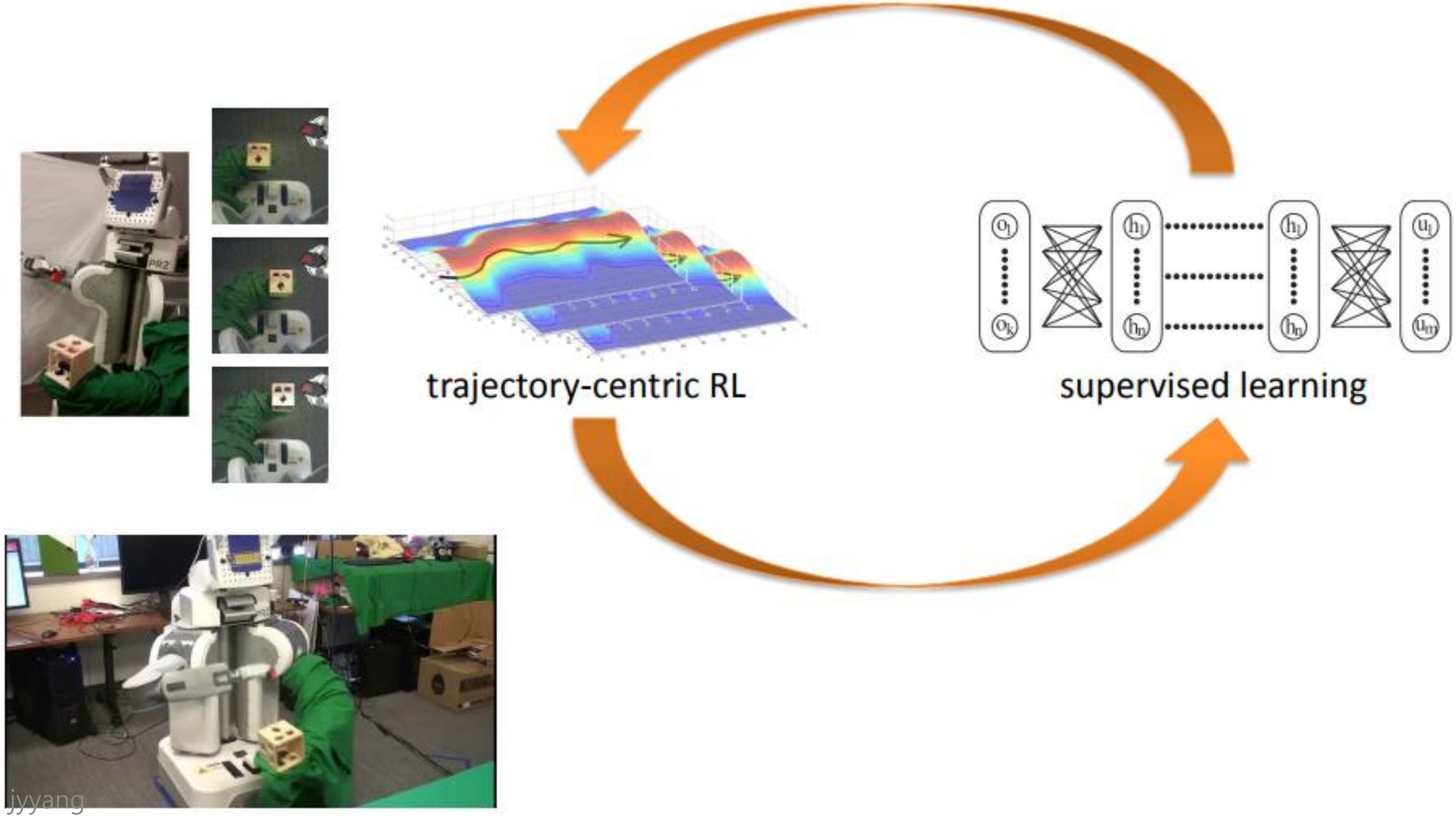
state to **action**

supervised learning



observation to **action**

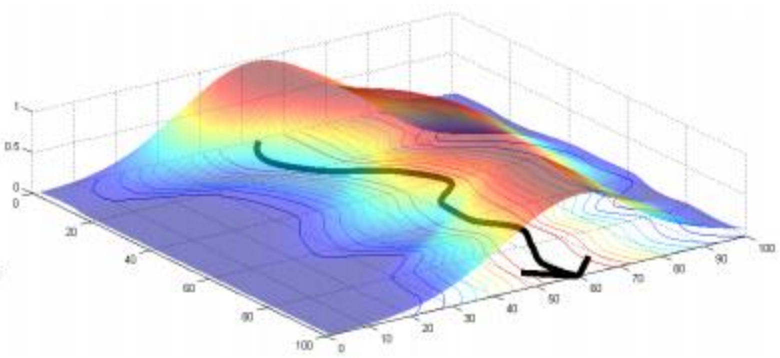
Guided Policy Search



$$\min_{\theta} \overbrace{E_{\pi_\theta}}^{\text{expectation under current policy}}[c(\tau)]$$

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

trajectory distribution(s)



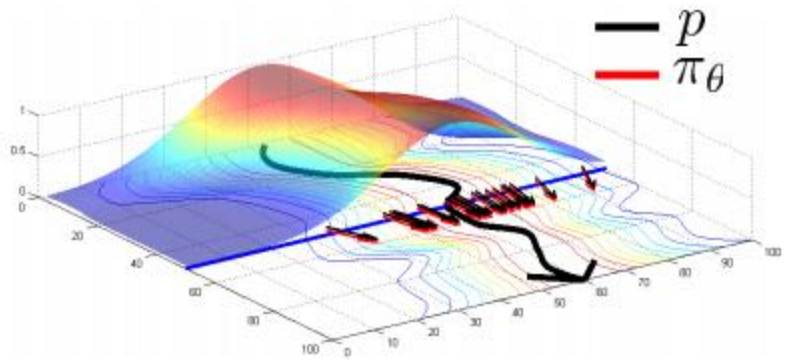
$$s.t. \;\; \pi_\theta(\mathbf{u}_t|\mathbf{o}(\mathbf{x}_t)) = p(\mathbf{u}_t|\mathbf{x}_t) \;\; \forall t, \mathbf{x}_t, \mathbf{u}_t$$

$$\min_{\theta} \overbrace{E_{\pi_\theta}}^{\text{expectation under current policy}}[c(\tau)]$$

$$\Updownarrow$$

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

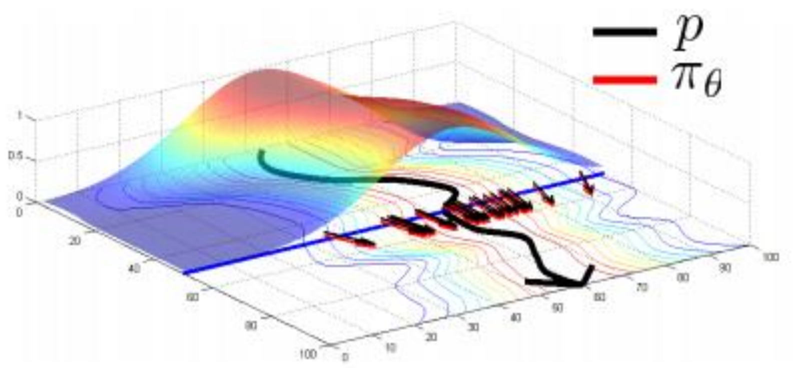
$$s.t. \;\; \pi_\theta(\mathbf{u}_t|\mathbf{o}(\mathbf{x}_t)) = p(\mathbf{u}_t|\mathbf{x}_t) \;\; \forall t, \mathbf{x}_t, \mathbf{u}_t$$



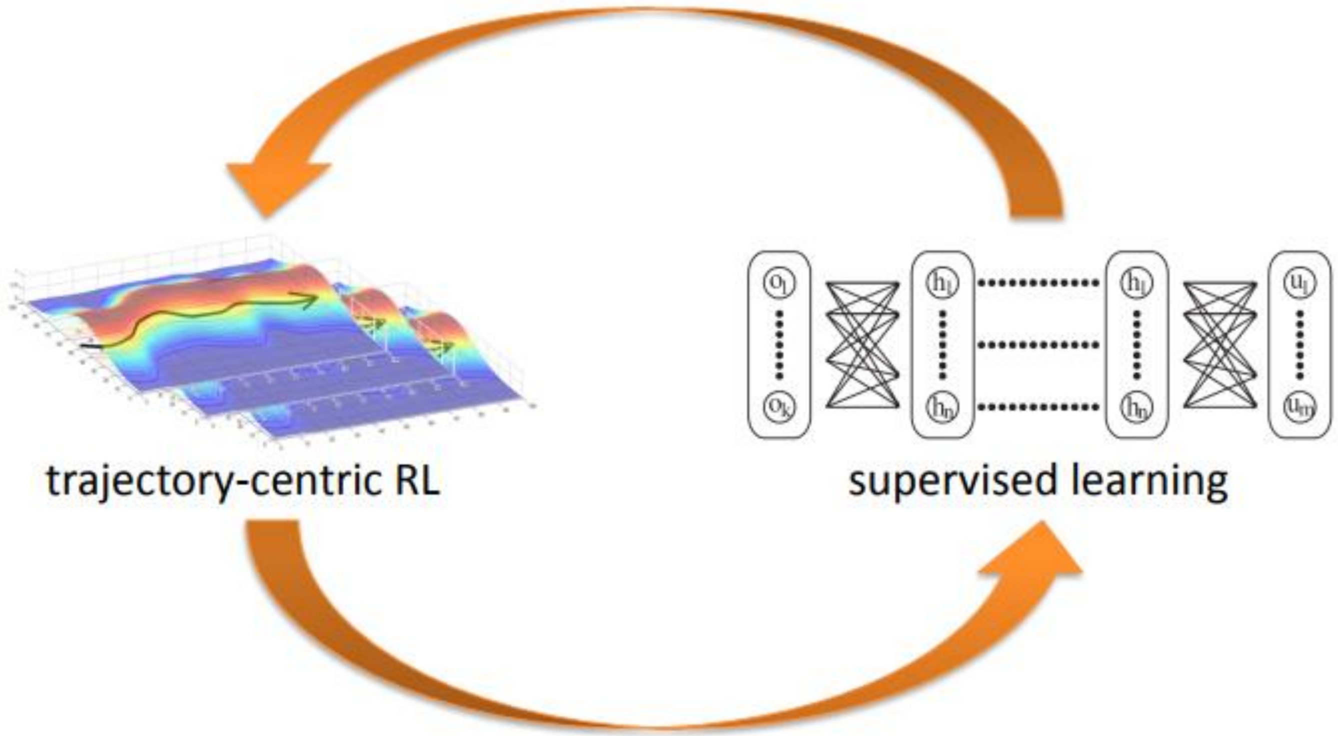
$$\min_{\theta} \overbrace{E_{\pi_\theta}}^{\text{expectation under current policy}}[c(\tau)]$$

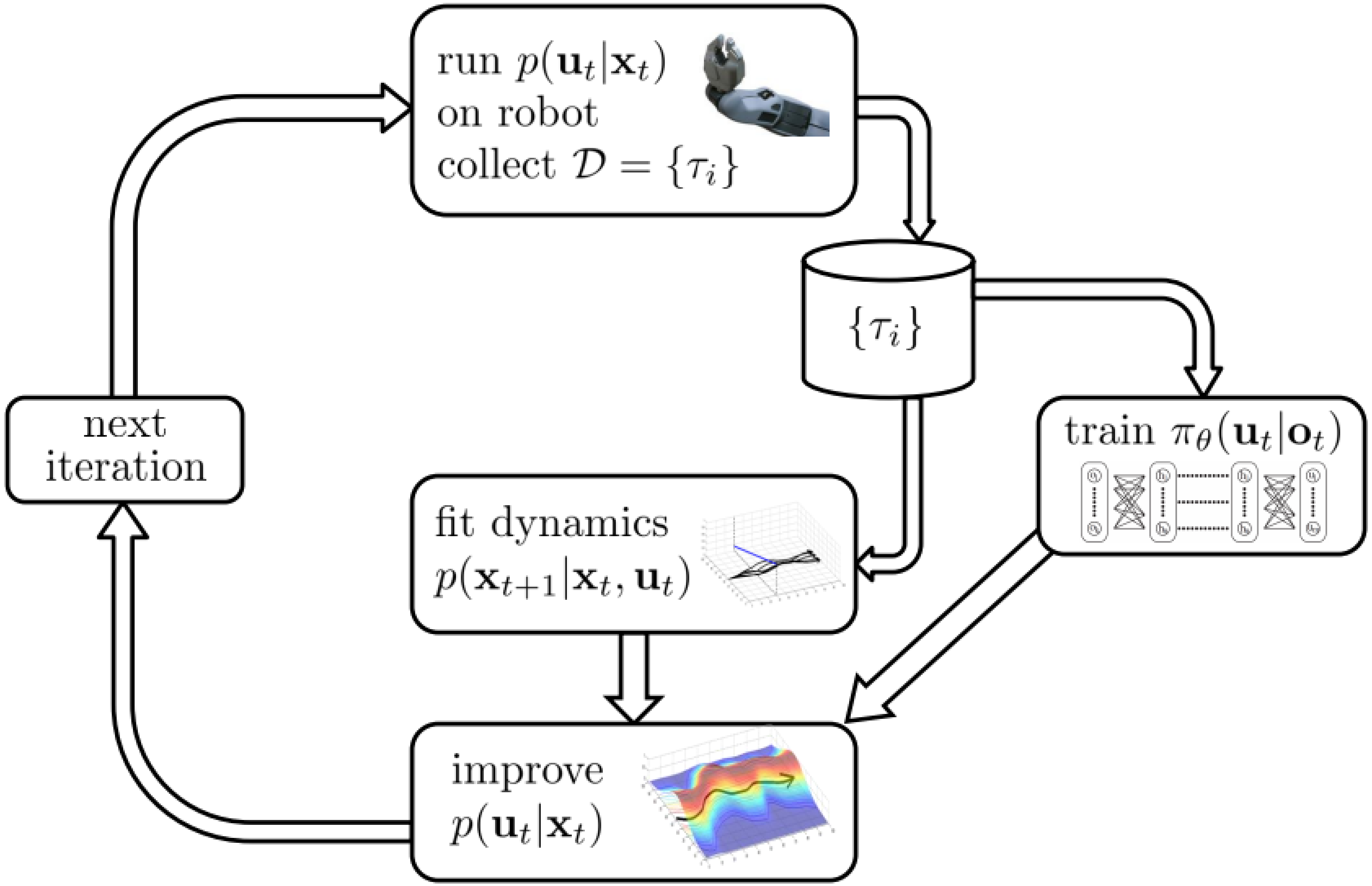
$$\min_{\theta, p(\tau)} \; E_p[c(\tau)]$$

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

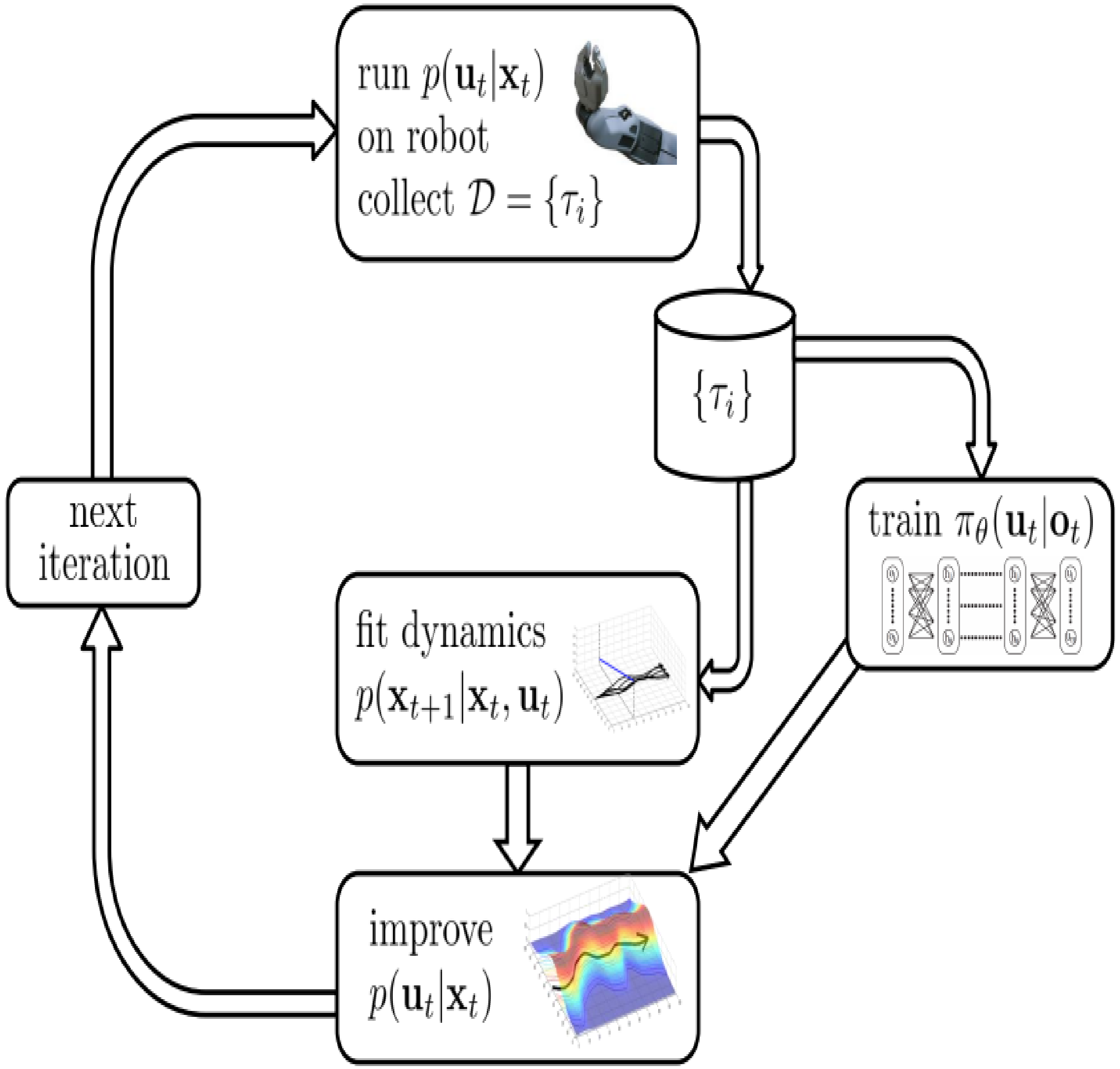


solve using Bregman ADMM (BADMM), a type of dual decomposition method



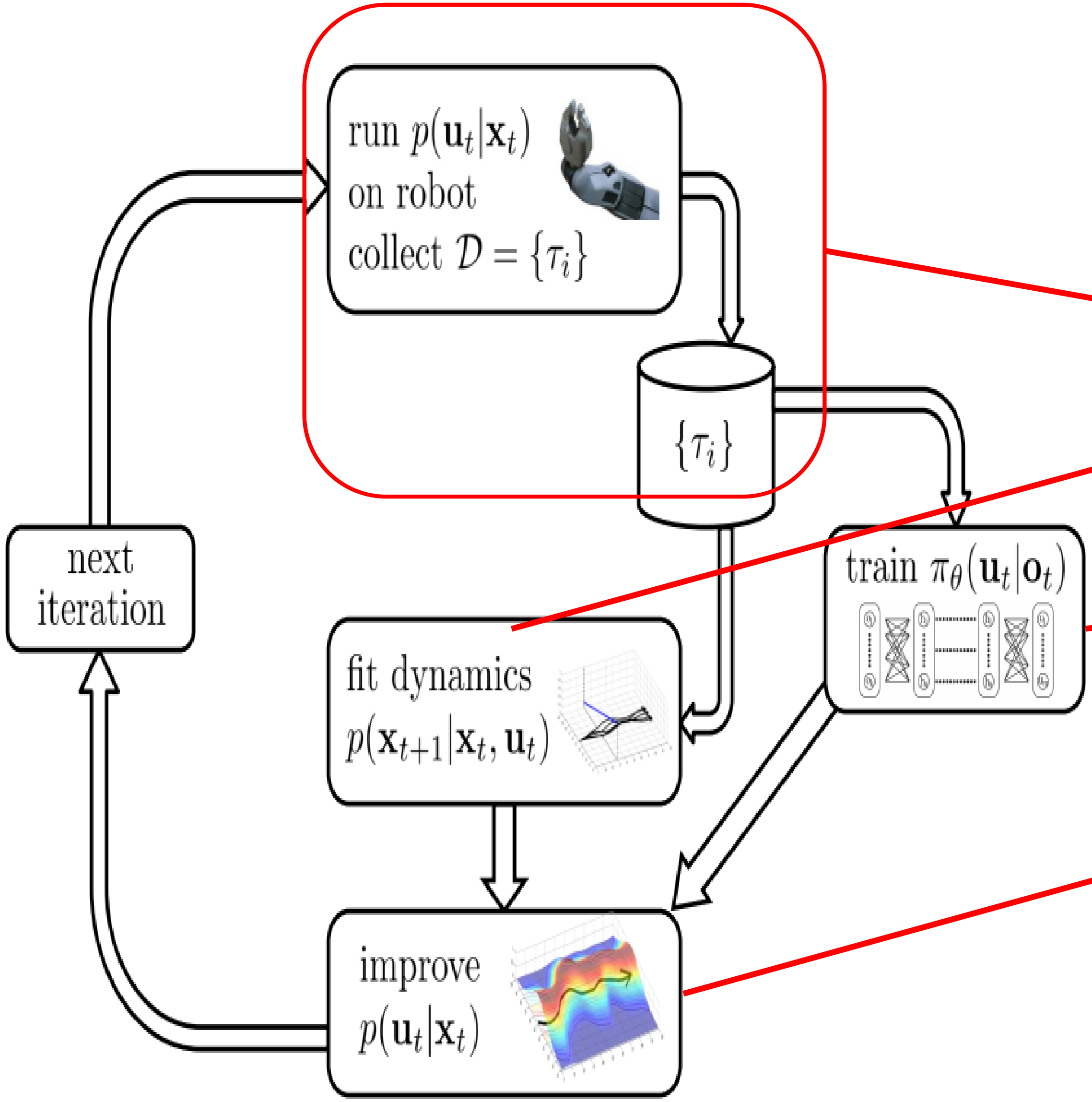


run $p(\mathbf{u}_t|\mathbf{x}_t)$
on robot
collect $\mathcal{D} = \{\tau_i\}$
$\{\tau_i\}$
train $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$
next
iteration
fit dynamics
$p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$
improve
$p(\mathbf{u}_t|\mathbf{x}_t)$



Algorithm 1 Guided Policy Search

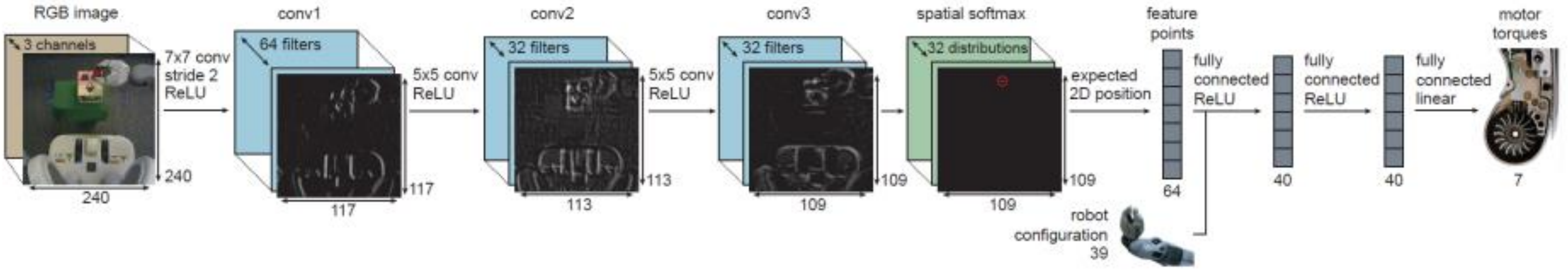
Generate DDP solutions $\pi_{\mathcal{G}_1}, \ldots, \pi_{\mathcal{G}_n}$
Sample $\zeta_1, \ldots, \zeta_m$ from $q(\zeta) = \frac{1}{n} \sum_i \pi_{\mathcal{G}_i}(\zeta)$
Initialize $\theta^\star \leftarrow \arg\max_\theta \sum_i \log \pi_{\theta^\star}(\zeta_i)$
Build initial sample set $\mathcal{S}$ from $\pi_{\mathcal{G}_1}, \ldots, \pi_{\mathcal{G}_n}, \pi_{\theta^\star}$
for iteration $k = 1$ to K **do**
  Choose current sample set $\mathcal{S}_k \subset \mathcal{S}$
  Optimize $\theta_k \leftarrow \arg\max_\theta \Phi_{\mathcal{S}_k}(\theta)$
  Append samples from π_{θ_k} to $\mathcal{S}_k$ and $\mathcal{S}$
  Optionally generate adaptive guiding samples
  Estimate the values of π_{θ_k} and $\pi_{\theta^\star}$ using $\mathcal{S}_k$
  if π_{θ_k} is better than $\pi_{\theta^\star}$ **then**
    Set $\theta^\star \leftarrow \theta_k$
    Decrease w_r
  else
    Increase w_r
    Optionally, resample from $\pi_{\theta^\star}$
  end if
end for
Return the best policy $\pi_{\theta^\star}$

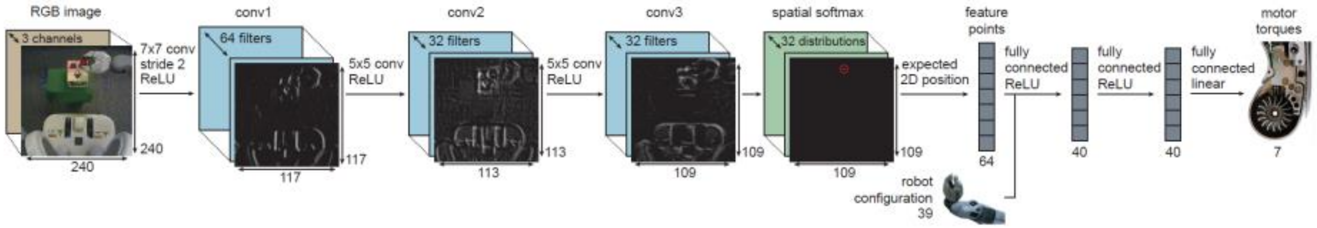


Algorithm 1 Guided Policy Search

```
Generate DDP solutions π_{G_1}, ..., π_{G_n}
Sample ζ_1, ..., ζ_m from q(ζ) = (1/n) Σ_i π_{G_i}(ζ)
Initialize θ* ← arg max_θ Σ_i log π_{θ*}(ζ_i)
Build initial sample set S from π_{G_1}, ..., π_{G_n}, π_{θ*}
for iteration k = 1 to K do
    Choose current sample set S_k ⊂ S
    Optimize θ_k ← arg max_θ Φ_{S_k}(θ)
    Append samples from π_{θ_k} to S_k and S
    Optionally generate adaptive guiding samples
    Estimate the values of π_{θ_k} and π_{θ*} using S_k
    if π_{θ_k} is better than π_{θ*} then
        Set θ* ← θ_k
        Decrease w_r
    else
        Increase w_r
        Optionally, resample from π_{θ*}
    end if
end for
Return the best policy π_{θ*}
```

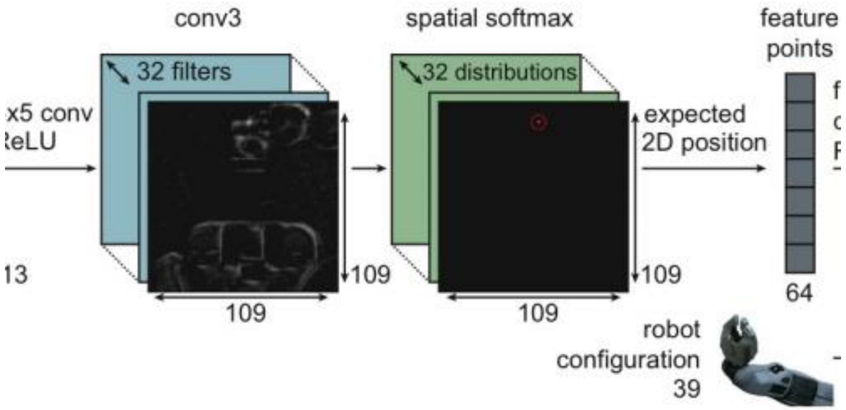
Network architecture





spatial softmax ~~라고 읽고 spatial soft-argmax라 부른다~~

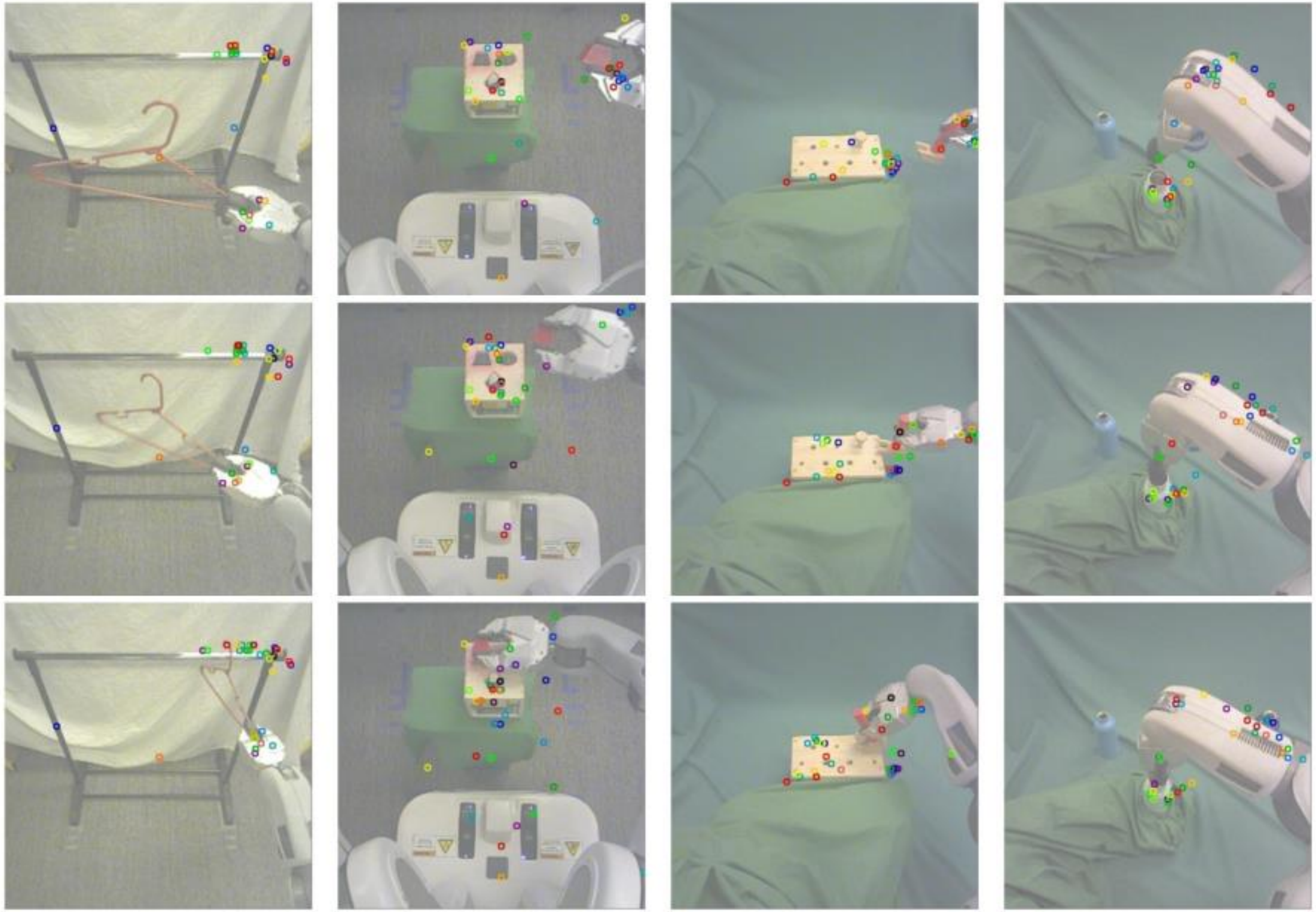
1. 채널별로 softmax를 구함
2. 채널별로 구한 softmax 결과와 x, y 좌표 값을 곱한 후 더한다.



$$s_{cij} = e^{a_{cij}} / \sum_{i'j'} e^{a_{ci'j'}}$$

$$f_{cx} = \sum_{ij} s_{cij} x_{ij} \text{ and } f_{cy} = \sum_{ij} s_{cij} y_{ij}$$

$$\mathrm{softargmax}(x) = \sum_i \frac{e^{\beta x_i}}{\sum_j e^{\beta x_j}} i$$



(a) hanger (b) cube (c) hammer (d) bottle

Figure 10: Feature points tracked by the policy during task execution for each of the four tasks. Each feature point is displayed in a different random color, with consistent coloring across images. The policy finds features on the target object and the robot gripper and arm. In the bottle cap task, note that the policy correctly ignores the distractor bottle in the background, even though it was not present during training.


Position 2
real time
autonomous execution

Training

- SL + RL
- Cost function