Deep Robotic Learning

Sergey Levine
University of Washington
Action (run away)
Action
(run away)
perception

Action (run away)
sensorimotor loop

Action (run away)
“When a man throws a ball high in the air and catches it again, he behaves as if he had solved a set of differential equations in predicting the trajectory of the ball ... at some subconscious level, something functionally equivalent to the mathematical calculations is going on.”

-- Richard Dawkins
“When a man throws a ball high in the air and catches it again, he behaves as if he had solved a set of differential equations in predicting the trajectory of the ball ... at some subconscious level, something functionally equivalent to the mathematical calculations is going on.”

-- Richard Dawkins

McLeod & Dienes. Do fielders know where to go to catch the ball or only how to get there? Journal of Experimental Psychology 1996, Vol. 22, No. 3, 531-543
“When a man throws a ball high in the air and catches it again, he behaves as if he had solved a set of differential equations in predicting the trajectory of the ball ... at some subconscious level, something functionally equivalent to the mathematical calculations is going on.”

-- Richard Dawkins
“When a man throws a ball high in the air and catches it again, he behaves as if he had solved a set of differential equations in predicting the trajectory of the ball ... at some subconscious level, something functionally equivalent to the mathematical calculations is going on.”

-- Richard Dawkins

McLeod & Dienes. Do fielders know where to go to catch the ball or only how to get there? Journal of Experimental Psychology 1996, Vol. 22, No. 3, 531-543
“When a man throws a ball high in the air and catches it again, he behaves as if he had solved a set of differential equations in predicting the trajectory of the ball ... at some subconscious level, something functionally equivalent to the mathematical calculations is going on.”

-- Richard Dawkins

McLeod & Dienes. Do fielders know where to go to catch the ball or only how to get there? Journal of Experimental Psychology 1996, Vol. 22, No. 3, 531-543
“When a man throws a ball high in the air and catches it again, he behaves as if he had solved a set of differential equations in predicting the trajectory of the ball ... at some subconscious level, something functionally equivalent to the mathematical calculations is going on.”

-- Richard Dawkins

McLeod & Dienes. Do fielders know where to go to catch the ball or only how to get there? Journal of Experimental Psychology 1996, Vol. 22, No. 3, 531-543
“When a man throws a ball high in the air and catches it again, he behaves as if he had solved a set of differential equations in predicting the trajectory of the ball ... at some subconscious level, something functionally equivalent to the mathematical calculations is going on.”

-- Richard Dawkins
KAIST’s DRC-HUBO opening a door

DARPA Robotics Challenge 2015
no direct supervision
no direct supervision
actions have consequences
Overview

Training visuomotor policies

Deep robotic learning at scale

Future directions
Overview

Training visuomotor policies

Deep robotic learning at scale

Future directions
general-purpose neural network policy
general-purpose neural network policy

\[ \theta = \arg \min_{\theta} E_{\pi_{\theta}} [\sum_{t=1}^{T} c(x_t, u_t)] \]

\[ \pi_{\theta}(u_t|o_t) \text{ – control policy} \]

\[ o_t \text{ – observation (may or may not be equal to } x_t) \]
general-purpose neural network policy

\[ \theta = \arg \min_{\theta} E_{\pi_{\theta}} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\[ \pi_{\theta}(u_t|o_t) - \text{control policy} \]

\[ o_t - \text{observation (may or may not be equal to } x_t) \]
general-purpose neural network policy

\[ \theta = \arg \min_{\theta} E_{\pi_{\theta}} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\[ \pi_{\theta}(u_t|o_t) \text{ – control policy} \]

\[ o_t \text{ – observation (may or may not be equal to } x_t) \]
general-purpose neural network policy

\[ \theta = \arg \min_\theta E_{\pi_\theta} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\[ \pi_\theta(u_t|o_t) \text{ – control policy} \]

\[ o_t \text{ – observation (may or may not be equal to } x_t) \]
general-purpose neural network policy

\[ \theta = \arg \min_{\theta} E_{\pi_{\theta}} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\( \pi_{\theta}(u_t|o_t) \) - control policy

\( o_t \) - observation (may or may not be equal to \( x_t \))
general-purpose neural network policy

\[ \theta = \arg\min_{\theta} E_{\pi_{\theta}} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\( \pi_{\theta}(u_t|o_t) \) – control policy

\( o_t \) – observation (may or may not be equal to \( x_t \))
general-purpose neural network policy

\[ \theta = \arg \min_{\theta} E_{\pi_{\theta}} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\( \pi_{\theta}(u_t|o_t) \) – control policy

\( o_t \) – observation (may or may not be equal to \( x_t \)
general-purpose neural network policy

\[ \theta = \arg \min_{\theta} E_{\pi_\theta} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\( \pi_\theta(u_t|o_t) \) – control policy

\( o_t \) – observation (may or may not be equal to \( x_t \))
general-purpose neural network policy

\[ \theta = \arg \min_{\theta} \mathbb{E}_{\pi_{\theta}} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\[ \pi_{\theta}(u_t | o_t) - \text{control policy} \]

\[ o_t - \text{observation (may or may not be equal to } x_t) \]
general-purpose neural network policy

\[ \theta = \arg \min_\theta E_{\pi_\theta} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\[ \pi_\theta(u_t | o_t) \text{ – control policy} \]

\[ o_t \text{ – observation (may or may not be equal to } x_t) \]

Mnih et al. ‘13
Schulman et al. ’14 & ’15
general-purpose neural network policy

$$\theta = \arg \min_\theta \mathbb{E}_{\pi_\theta} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right]$$

$$\pi_\theta(u_t|o_t)$$ – control policy

$$o_t$$ – observation (may or may not be equal to $$x_t$$)

policy search (RL)

Mnih et al. ‘13

Schulman et al. ’14 & ’15
general-purpose neural network policy

\[ \theta = \arg \min_{\theta} \mathbb{E}_{\pi_{\theta}} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\( \pi_{\theta}(u_t \mid o_t) \) – control policy

\( o_t \) – observation (may or may not be equal to \( x_t \))

policy search (RL)

complex dynamics

Mnih et al. ‘13

Schulman et al. ’14 & ‘15
general-purpose neural network policy

\[ \theta = \arg \min_{\theta} E_{\pi_{\theta}} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\[ \pi_{\theta}(u_t|o_t) - \text{control policy} \]

\[ o_t - \text{observation (may or may not be equal to } x_t) \]

policy search (RL)  complex dynamics  complex policy
general-purpose neural network policy

$$\theta = \arg \min_{\theta} E_{\pi_{\theta}} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right]$$

$$\pi_{\theta}(u_t|o_t)$$ - control policy

$o_t$ - observation (may or may not be equal to $x_t$)

policy search (RL)  complex dynamics  complex policy  HARD
general-purpose neural network policy

\[ \theta = \arg \min_{\theta} E_{\pi_{\theta}} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\[ \pi_{\theta}(u_t|o_t) – \text{control policy} \]

\[ o_t – \text{observation (may or may not be equal to } x_t) \]

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

Mnih et al. ‘13 Schulman et al. ’14 & ’15
general-purpose neural network policy

\[ \theta = \arg \min_\theta E_{\pi_\theta} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\[ \pi_\theta(u_t|o_t) \] - control policy

\[ o_t \] - observation (may or may not be equal to \( x_t \))

policy search (RL) policy search (RL)

complex dynamics complex dynamics

supervised learning supervised learning

complex policy complex policy

HARD HARD

Mnih et al. ‘13
Schulman et al. ‘14 & ‘15
general-purpose neural network policy

\[ \theta = \underset{\theta}{\arg \min} \; E_{\pi_{\theta}} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\[ \pi_{\theta}(u_t | o_t) \text{ – control policy} \]

\[ o_t \text{ – observation (may or may not be equal to } x_t) \]

policy search (RL)  complex dynamics  complex policy  HARD

supervised learning  complex dynamics  complex policy
general-purpose neural network policy

\[
\theta = \arg\min_{\theta} E_{\pi_\theta} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right]
\]

\[\pi_\theta(u_t|o_t) \text{ – control policy}\]
\[o_t \text{ – observation (may or may not be equal to } x_t)\]

policy search (RL) complex dynamics complex policy HARD

supervised learning complex dynamics complex policy EASY
general-purpose neural network policy

\[ \theta = \arg \min_{\theta} E_{\pi_{\theta}} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\[ \pi_{\theta}(u_t | o_t) - \text{control policy} \]

\[ o_t - \text{observation (may or may not be equal to } x_t) \]

policy search (RL)

complex dynamics

complex policy

HARD

supervised learning

complex dynamics

complex policy

EASY

optimal control
A general-purpose neural network policy

\[ \theta = \arg \min_{\theta} E_{\pi_{\theta}} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\( \pi_{\theta}(u_t|o_t) \) – control policy

\( o_t \) – observation (may or may not be equal to \( x_t \))

Policy search (RL) complex dynamics complex policy HARD

Supervised learning complex dynamics complex policy EASY

Optimal control complex dynamics
general-purpose neural network policy

\[ \theta = \arg \min_{\theta} E_{\pi_\theta} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right] \]

\[ \pi_\theta(u_t|o_t) \text{ – control policy} \]

\[ o_t \text{ – observation (may or may not be equal to } x_t) \]

policy search (RL) complex dynamics complex policy HARD

supervised learning complex dynamics complex policy EASY

optimal control complex dynamics complex policy
A general-purpose neural network policy can be represented as:

$$\theta = \arg\min_{\theta} E_{\pi_\theta} \left[ \sum_{t=1}^{T} c(x_t, u_t) \right]$$

$$\pi_\theta(u_t | o_t)$$ - control policy

$$o_t$$ - observation (may or may not be equal to $$x_t$$)

Policy search (RL) vs. complex dynamics vs. complex policy:
- Policy search (RL) vs. complex dynamics: HARD
- Supervised learning vs. complex dynamics: EASY
- Optimal control vs. complex dynamics: EASY
1. break up the task: 
   separately solve N 
   different task instances
1. break up the task:
   separately solve N
different task instances
1. break up the task: separately solve $N$ different task instances
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
1. break up the task:
   separately solve $N$ different task instances

2. use supervised learning

   trajectory-centric RL
   (fully observed)
1. break up the task: separately solve $N$ different task instances

2. use supervised learning

trajectory-centric RL (fully observed)
1. break up the task: separately solve $N$ different task instances

2. use supervised learning

   trajectory-centric RL (fully observed)
1. break up the task: separately solve N different task instances

2. use supervised learning

trajectory-centric RL (fully observed)
1. break up the task:
   separately solve $N$ different task instances

2. use supervised learning

trajectory-centric RL
(fully observed)
1. break up the task: separately solve N different task instances

2. use supervised learning

trajectory-centric RL (fully observed)
Guided Policy Search
Guided Policy Search

trajectory-centric RL
Guided Policy Search

trajectory-centric RL
Guided Policy Search

trajectory-centric RL
Guided Policy Search

trajectory-centric RL

supervised learning
Guided Policy Search

trajectory-centric RL

supervised learning
Guided Policy Search

trajectory-centric RL

supervised learning
\[
\min_{\theta} E_{\pi_0} [c(\tau)]
\]
expectation under current policy

$$\min_\theta E_{\pi_0}[c(\tau)]$$
expectation under current policy

\[
\begin{align*}
\min_\theta E_{\pi_\theta}[c(\tau)] \\
\min_{\theta, p(\tau)} E_p[c(\tau)] \\
s.t. \quad & \pi_\theta(u_t|o(x_t)) = p(u_t|x_t) \quad \forall t, x_t, u_t
\end{align*}
\]
min_{\theta} \mathbb{E}_{\pi_{\theta}}[c(\tau)] \xrightarrow{\text{current policy}} \min_{\theta, p(\tau)} \mathbb{E}_{p}[c(\tau)] \xrightarrow{\text{trajectory distribution(s)}} s.t. \pi_{\theta}(u_t|o(x_t)) = p(u_t|x_t) \ \forall t, x_t, u_t
Minimizing the expectation under current policy:

$$\min_{\theta} \mathbb{E}_{\pi_{\theta}}[c(\tau)]$$

$$\min_{\theta, p(\tau)} \mathbb{E}_p[c(\tau)]$$

subject to: $$\pi_{\theta}(u_t|o(x_t)) = p(u_t|x_t) \quad \forall t, x_t, u_t$$
expectation under current policy

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

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

\[s.t. \quad \pi_\theta(u_t | o(x_t)) = p(u_t | x_t) \quad \forall t, x_t, u_t\]

solve using Bregman ADMM (BADMM), a type of dual decomposition method
expectation under current policy

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

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

s.t. $\pi_{\theta}(u_t|o(x_t)) = p(u_t|x_t) \ \forall t, x_t, u_t$

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

trajectory-centric RL

supervised learning
run $p(\mathbf{u}_t | \mathbf{x}_t)$ on robot
collect $\mathcal{D} = \{\tau_i\}$

[see L. et al. NIPS ’14 for details]
run $p(u_t | x_t)$ on robot
collect $\mathcal{D} = \{\tau_i\}$
run $p(u_t|x_t)$ on robot
collect $\mathcal{D} = \{\tau_i\}$

fit dynamics $p(x_{t+1}|x_t, u_t)$

[see L. et al. NIPS ‘14 for details]
run $p(u_t | x_t)$ on robot
collect $\mathcal{D} = \{\tau_i\}$

fit dynamics
$p(x_{t+1} | x_t, u_t)$

improve $p(u_t | x_t)$

[see L. et al. NIPS ‘14 for details]
run \( p(u_t | x_t) \) on robot
collect \( \mathcal{D} = \{\tau_i\} \)

next iteration

fit dynamics
\( p(x_{t+1} | x_t, u_t) \)

improve
\( p(u_t | x_t) \)

[see L. et al. NIPS '14 for details]
run $p(u_t | x_t)$ on robot
collect $D = \{\tau_i\}$

fit dynamics $p(x_{t+1} | x_t, u_t)$

improve $p(u_t | x_t)$

train $\pi_\theta(u_t | o_t)$

[see L. et al. NIPS '14 for details]
run $p(\mathbf{u}_t | \mathbf{x}_t)$ on robot
collect $\mathcal{D} = \{\tau_i\}$

next iteration

fit dynamics $p(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t)$

improve $p(\mathbf{u}_t | \mathbf{x}_t)$

train $\pi_\theta(\mathbf{u}_t | \mathbf{o}_t)$

[see L. et al. NIPS ’14 for details]
Learning on PR2

[L. et al. ICRA ‘15]
Learning on PR2

[L. et al. ICRA '15]
L.*, Finn*, Darrell, Abbeel ‘15
training time

\[ \mathbf{x}_t \rightarrow \mathbf{u}_t \]

test time

L.*, Finn*, Darrell, Abbeel '15
Experimental Tasks
Experimental Tasks
Experimental Tasks
Experimental Tasks
Experimental Tasks
Experimental Tasks

Learned Visuomotor Policy: Shape sorting cube
Generalization Experiments

Visual Test
Position 1
real time

autonomous execution
Comparisons

dend-to-end training
Comparisons

dend-to-end training

pose prediction
Comparisons

end-to-end training

pose prediction
Comparisons

end-to-end training

pose prediction
Comparisons

end-to-end training

pose prediction (trained on pose only)

pose features
Comparisons

end-to-end training

pose prediction

pose features
Comparisons

end-to-end training

pose prediction

pose features
Comparisons
## Comparisons

<table>
<thead>
<tr>
<th>Item</th>
<th>Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>coat hanger</td>
<td>55.6%</td>
</tr>
<tr>
<td>pose prediction</td>
<td></td>
</tr>
<tr>
<td>shape sorting cube</td>
<td>0%</td>
</tr>
<tr>
<td>pose prediction</td>
<td></td>
</tr>
<tr>
<td>toy claw hammer</td>
<td>8.9%</td>
</tr>
<tr>
<td>pose prediction</td>
<td></td>
</tr>
<tr>
<td>bottle cap</td>
<td>n/a</td>
</tr>
<tr>
<td>pose prediction</td>
<td></td>
</tr>
<tr>
<td>Object</td>
<td>Success Rate</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>coat hanger</td>
<td></td>
</tr>
<tr>
<td>shape sorting cube</td>
<td></td>
</tr>
<tr>
<td>toy claw hammer</td>
<td></td>
</tr>
<tr>
<td>bottle cap</td>
<td></td>
</tr>
</tbody>
</table>
## Comparisons

<table>
<thead>
<tr>
<th>Item</th>
<th>Success Rate</th>
<th>Pose Prediction</th>
<th>Pose Features</th>
<th>End-to-End Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coat hanger</td>
<td></td>
<td>55.6%</td>
<td>88.9%</td>
<td>100%</td>
</tr>
<tr>
<td>Pose prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose features</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End-to-end training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shape sorting cube</td>
<td></td>
<td>0%</td>
<td>70.4%</td>
<td>96.3%</td>
</tr>
<tr>
<td>Pose prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose features</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End-to-end training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toy claw hammer</td>
<td></td>
<td>8.9%</td>
<td>62.2%</td>
<td>91.1%</td>
</tr>
<tr>
<td>Pose prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose features</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End-to-end training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottle cap</td>
<td></td>
<td>n/a</td>
<td>55.6%</td>
<td>88.9%</td>
</tr>
<tr>
<td>Pose prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose features</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End-to-end training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Comparisons

<table>
<thead>
<tr>
<th>Item</th>
<th>Success Rate</th>
<th>Pose Prediction</th>
<th>Pose Features</th>
<th>End-to-End Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coat hanger</td>
<td>55.6%</td>
<td>88.9%</td>
<td></td>
<td>100%</td>
</tr>
<tr>
<td>Pose prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose features</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End-to-end training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shape sorting cube</td>
<td>0%</td>
<td></td>
<td>70.4%</td>
<td>96.3%</td>
</tr>
<tr>
<td>Pose prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose features</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End-to-end training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toy claw hammer</td>
<td>8.9%</td>
<td></td>
<td>62.2%</td>
<td>91.1%</td>
</tr>
<tr>
<td>Pose prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose features</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End-to-end training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottle cap</td>
<td>n/a</td>
<td></td>
<td>55.6%</td>
<td>88.9%</td>
</tr>
<tr>
<td>Pose prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose features</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End-to-end training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Network Architecture and Test Error (cm)

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Test Error (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softmax + feature points (ours)</td>
<td><strong>1.30 ± 0.73</strong></td>
</tr>
<tr>
<td>Softmax + fully connected layer</td>
<td>2.59 ± 1.19</td>
</tr>
<tr>
<td>Fully connected layer</td>
<td>4.75 ± 2.29</td>
</tr>
<tr>
<td>Max-pooling + fully connected</td>
<td>3.71 ± 1.73</td>
</tr>
</tbody>
</table>
## Comparisons

<table>
<thead>
<tr>
<th>Object</th>
<th>Success Rate</th>
<th>Pose Prediction</th>
<th>Pose Features</th>
<th>End-to-End Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coat hanger</td>
<td>55.6%</td>
<td>88.9%</td>
<td></td>
<td>100%</td>
</tr>
<tr>
<td>Shape sorting cube</td>
<td>0%</td>
<td>70.4%</td>
<td></td>
<td>96.3%</td>
</tr>
<tr>
<td>Toy claw hammer</td>
<td>8.9%</td>
<td>62.2%</td>
<td></td>
<td>91.1%</td>
</tr>
<tr>
<td>Bottle cap</td>
<td>n/a</td>
<td>55.6%</td>
<td></td>
<td>88.9%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network Architecture</th>
<th>Test Error (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softmax + feature points (ours)</td>
<td><strong>1.30 ± 0.73</strong></td>
</tr>
<tr>
<td>Softmax + fully connected layer</td>
<td>2.59 ± 1.19</td>
</tr>
<tr>
<td>Fully connected layer</td>
<td>4.75 ± 2.29</td>
</tr>
<tr>
<td>Max-pooling + fully connected</td>
<td>3.71 ± 1.73</td>
</tr>
</tbody>
</table>
## Comparisons

<table>
<thead>
<tr>
<th>Object</th>
<th>Success Rate</th>
<th>Pose Prediction</th>
<th>Pose Features</th>
<th>End-to-End Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coat hanger</td>
<td>55.6%</td>
<td>88.9%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Shape sorting cube</td>
<td>0%</td>
<td>70.4%</td>
<td>96.3%</td>
<td></td>
</tr>
<tr>
<td>Toy claw hammer</td>
<td>8.9%</td>
<td>62.2%</td>
<td>91.1%</td>
<td></td>
</tr>
<tr>
<td>Bottle cap</td>
<td>n/a</td>
<td>55.6%</td>
<td>88.9%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network Architecture</th>
<th>Test Error (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softmax + feature points (ours)</td>
<td>1.30 ± 0.73</td>
</tr>
<tr>
<td>Softmax + fully connected layer</td>
<td>2.59 ± 1.19</td>
</tr>
<tr>
<td>Fully connected layer</td>
<td>4.75 ± 2.29</td>
</tr>
<tr>
<td>Max-pooling + fully connected</td>
<td>3.71 ± 1.73</td>
</tr>
</tbody>
</table>

Meeussen et al. (Willow Garage)
## Comparisons

<table>
<thead>
<tr>
<th>Object</th>
<th>Success Rate</th>
<th>Pose Prediction</th>
<th>Pose Features</th>
<th>End-to-End Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>coat hanger</td>
<td>55.6%</td>
<td>88.9%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>shape sorting cube</td>
<td>0%</td>
<td>70.4%</td>
<td>96.3%</td>
<td></td>
</tr>
<tr>
<td>toy claw hammer</td>
<td>8.9%</td>
<td>62.2%</td>
<td>91.1%</td>
<td></td>
</tr>
<tr>
<td>bottle cap</td>
<td>n/a</td>
<td>55.6%</td>
<td>88.9%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network Architecture</th>
<th>Test Error (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>softmax + feature points (ours)</td>
<td>1.30 ± 0.73</td>
</tr>
<tr>
<td>softmax + fully connected layer</td>
<td>2.59 ± 1.19</td>
</tr>
<tr>
<td>fully connected layer</td>
<td>4.75 ± 2.29</td>
</tr>
<tr>
<td>max-pooling + fully connected</td>
<td>3.71 ± 1.73</td>
</tr>
</tbody>
</table>

Meeussen et al. (Willow Garage)

2 cm
## Comparisons

<table>
<thead>
<tr>
<th>Object</th>
<th>Success Rate</th>
<th>Pose Prediction</th>
<th>Pose Features</th>
<th>End-to-End Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>coat hanger</td>
<td>55.6%</td>
<td>88.9%</td>
<td></td>
<td>100%</td>
</tr>
<tr>
<td>shape sorting cube</td>
<td>0%</td>
<td>70.4%</td>
<td></td>
<td>96.3%</td>
</tr>
<tr>
<td>toy claw hammer</td>
<td>8.9%</td>
<td>62.2%</td>
<td></td>
<td>91.1%</td>
</tr>
<tr>
<td>bottle cap</td>
<td>n/a</td>
<td>55.6%</td>
<td></td>
<td>88.9%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network Architecture</th>
<th>Test Error (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>softmax + feature points (ours)</td>
<td>1.30 ± 0.73</td>
</tr>
<tr>
<td>softmax + fully connected layer</td>
<td>2.59 ± 1.19</td>
</tr>
<tr>
<td>fully connected layer</td>
<td>4.75 ± 2.29</td>
</tr>
<tr>
<td>max-pooling + fully connected</td>
<td>3.71 ± 1.73</td>
</tr>
</tbody>
</table>

Meeussen et al. (Willow Garage)

2 cm
## Comparisons

<table>
<thead>
<tr>
<th>Object</th>
<th>Success Rate</th>
<th>Pose Prediction</th>
<th>Pose Features</th>
<th>End-to-End Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coat hanger</td>
<td>55.6%</td>
<td>88.9%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Pose prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose features</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End-to-end training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shape sorting cube</td>
<td>0%</td>
<td>70.4%</td>
<td>96.3%</td>
<td></td>
</tr>
<tr>
<td>Pose prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose features</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End-to-end training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toy claw hammer</td>
<td>8.9%</td>
<td>62.2%</td>
<td>91.1%</td>
<td></td>
</tr>
<tr>
<td>Pose prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose features</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End-to-end training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottle cap</td>
<td>n/a</td>
<td>55.6%</td>
<td>88.9%</td>
<td></td>
</tr>
<tr>
<td>Pose prediction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose features</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End-to-end training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network Architecture</th>
<th>Test Error (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softmax + feature points (ours)</td>
<td>1.30 ± 0.73</td>
</tr>
<tr>
<td>Softmax + fully connected layer</td>
<td>2.59 ± 1.19</td>
</tr>
<tr>
<td>Fully connected layer</td>
<td>4.75 ± 2.29</td>
</tr>
<tr>
<td>Max-pooling + fully connected</td>
<td>3.71 ± 1.73</td>
</tr>
</tbody>
</table>

Meeussen et al. (Willow Garage)

![2 cm measurement](image)
Guided Policy Search Applications
Guided Policy Search Applications

manipulation

with N. Wagener and P. Abbeel
Guided Policy Search Applications

manipulation

with N. Wagener and P. Abbeel

locomotion

constrained GPS
300–400 N pushes

with V. Koltun
Guided Policy Search Applications

**manipulation**
with N. Wagener and P. Abbeel

**dexterous hands**
with V. Kumar and E. Todorov

**locomotion**
constrained GPS
300–400 N pushes
with V. Koltun
Guided Policy Search Applications

**manipulation**
with N. Wagener and P. Abbeel

**dexterous hands**
with V. Kumar and E. Todorov

**soft hands**
with A. Gupta, C. Eppner, P. Abbeel

**locomotion**
with V. Koltun
Guided Policy Search Applications

**Manipulation**
- with N. Wagener and P. Abbeel

**Dexterous Hands**
- with V. Kumar and E. Todorov

**Soft Hands**
- with A. Gupta, C. Eppner, P. Abbeel

**Locomotion**
- constrained GPS
- 300–400 N pushes
  - with V. Koltun

**Aerial Vehicles**
- MPC-guided policy search (our method)
  - with G. Kahn, T. Zhang, P. Abbeel
Overview

Training visuomotor policies

Deep robotic learning at scale

Future directions
ingredients for success in learning:

supervised learning:
ingredients for success in learning:

supervised learning:

✓ computation
ingredients for success in learning:

supervised learning:

✓ computation
✓ algorithms
ingredients for success in learning:

supervised learning:

✅ computation
✅ algorithms
✅ data
ingredients for success in learning:

- supervised learning:
- learning sensorimotor skills:
  - computation
  - algorithms
  - data
ingredients for success in learning:

- supervised learning:
  - computation
  - algorithms
  - data

- learning sensorimotor skills:
  - computation
ingredients for success in learning:

supervised learning:

- computation
- algorithms
- data

learning sensorimotor skills:

- computation
- algorithms
ingredients for success in learning:

supervised learning:
- ✔ computation
- ✔ algorithms
- ✔ data

learning sensorimotor skills:
- ✔ computation
- ~ algorithms
- ? data
ingredients for success in learning:

supervised learning:
- ✔ computation
- ✔ algorithms
- ✔ data

learning sensorimotor skills:
- ✔ computation
- ~ algorithms
- ? data

L., Pastor, Krizhevsky, Quillen ‘16
Grasping with Learned Hand-Eye Coordination

- 800,000 grasp attempts for training (3,000 robot-hours)
- monocular camera (no depth)
- 2-5 Hz update
- no prior knowledge

L., Pastor, Krizhevsky, Quillen ’16
Using Grasp Success Prediction
Using Grasp Success Prediction

L., Pastor, Krizhevsky, Quillen '16
Using Grasp Success Prediction
Using Grasp Success Prediction

L., Pastor, Krizhevsky, Quillen ‘16
Using Grasp Success Prediction

L., Pastor, Krizhevsky, Quillen ‘16
Using Grasp Success Prediction

L., Pastor, Krizhevsky, Quillen ‘16
Using Grasp Success Prediction

L., Pastor, Krizhevsky, Quillen ’16
Using Grasp Success Prediction

L., Pastor, Krizhevsky, Quillen ’16
Using Grasp Success Prediction
Using Grasp Success Prediction

L., Pastor, Krizhevsky, Quillen ‘16
Using Grasp Success Prediction

L., Pastor, Krizhevsky, Quillen ‘16
Using Grasp Success Prediction

L., Pastor, Krizhevsky, Quillen '16
Using Grasp Success Prediction

L., Pastor, Krizhevsky, Quillen '16
Using Grasp Success Prediction

L., Pastor, Krizhevsky, Quillen ‘16
Open-Loop vs. Closed-Loop Grasping

open-loop grasping

closed-loop grasping
Open-Loop vs. Closed-Loop Grasping

open-loop grasping

closed-loop grasping

L., Pastor, Krizhevsky, Quillen ‘16
Open-Loop vs. Closed-Loop Grasping

open-loop grasping

closed-loop grasping

Pinto & Gupta, 2015

L., Pastor, Krizhevsky, Quillen ‘16
Open-Loop vs. Closed-Loop Grasping

open-loop grasping

closed-loop grasping

L., Pastor, Krizhevsky, Quillen ’16
Open-Loop vs. Closed-Loop Grasping

open-loop grasping

failure rate: 33.7%

L., Pastor, Krizhevsky, Quillen ‘16
Open-Loop vs. Closed-Loop Grasping

open-loop grasping

failure rate: 33.7%

closed-loop grasping

failure rate: 17.5%

L., Pastor, Krizhevsky, Quillen ‘16
Open-Loop vs. Closed-Loop Grasping

open-loop grasping

failure rate: 33.7%

depth + segmentation failure rate: 35%

closed-loop grasping

our method 1x real time

failure rate: 17.5%

L., Pastor, Krizhevsky, Quillen ‘16
Open-Loop vs. Closed-Loop Grasping

open-loop grasping

failure rate: 33.7%

closed-loop grasping

depth + segmentation

failure rate: 17.5%

failure rate: 35%

L., Pastor, Krizhevsky, Quillen ‘16
Grasping Experiments
Overview

Training visuomotor policies

Deep robotic learning at scale

Future directions
<table>
<thead>
<tr>
<th>Ingredients for Success in Learning:</th>
<th>Supervised Learning:</th>
<th>Learning Sensorimotor Skills:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- computation</td>
<td>✔ computation</td>
<td>✔ computation</td>
</tr>
<tr>
<td>- algorithms</td>
<td>✔ algorithms</td>
<td>✏ algorithms</td>
</tr>
<tr>
<td>- data</td>
<td>✔ data</td>
<td>❔ data</td>
</tr>
</tbody>
</table>
ingredients for success in learning:

supervised learning:

✔ computation
✔ algorithms
✔ data

learning sensorimotor skills:

✔ computation
~ algorithms
? data
ingredients for success in learning:

supervised learning:

- ✔ computation
- ✔ algorithms
- ✔ data

learning sensorimotor skills:

- ✔ computation
- ~ algorithms
- ~ data
Learning what Success Means
Learning what Success Means

\[ c(x, u) = \]
\[ w_1 f_{\text{target}}(x) + \]
\[ w_2 f_{\text{torque}}(u) \]
Learning what Success Means

\[ c(x, u) = w_1 f_{\text{target}}(x) + w_2 f_{\text{torque}}(u) \]
Learning what Success Means

can we \textit{learn} the cost with visual features?

\begin{align*}
  c(x, u) &= w_1 f_{\text{target}}(x) + w_2 f_{\text{torque}}(u)
\end{align*}

Finn, L., Abbeel ‘16
Learning what Success Means

can we learn the cost with visual features?

\[
c(x, u) = w_1 f_{\text{target}}(x) + w_2 f_{\text{torque}}(u)
\]

Finn, L., Abbeel ‘16
Learning what Success Means

can we learn the cost
with visual features?

\[ c(x, u) = w_1 f_{\text{target}}(x) + w_2 f_{\text{torque}}(u) \]
Learning what Success Means

can we learn the cost with visual features?

\[ c(x, u) = w_1 f_{\text{target}}(x) + w_2 f_{\text{torque}}(u) \]

Finn, L., Abbeel ‘16
Broader implications: end-to-end learning for decision making in the real world
Broader implications: end-to-end learning for decision making in the real world
Broader implications: end-to-end learning for decision making in the real world
Broader implications: end-to-end learning for decision making in the real world
Acknowledgements

BRETT

Chelsea Finn
Trevor Darrell
Pieter Abbeel

r3d10

Peter Pastor
Alex Krizhevsky
Deirdre Quillen
Questions?

Bibliography:

website: http://homes.cs.washington.edu/~svlevine/