Hierarchical DRL
part 2, improvements
The problem of off-policy architectures
on-policy vs. off-policy
on-policy

- stable and simple
- one policy to rule them all

off-policy

- data efficient
- less stable, but improvements have been done
- one policy for exploration, one policy for learning
\langle s, a, r, s^+ \rangle
High level policy (manager) 

Low level policy (worker) 

\begin{align*} 
\langle s, g, r, s^+ \rangle 
\end{align*} 

trained!
Data-Efficient Hierarchical Reinforcement Learning

Ofir Nachum  
Google Brain  
ofirnachum@google.com

Shixiang Gu*  
Google Brain  
shanegu@google.com

Honglak Lee  
Google Brain  
honglak@google.com

Sergey Levine†  
Google Brain  
slevine@google.com
\( \langle s, g, r, s^+ \rangle \)

\( \rightarrow \)

\( \langle s, \tilde{g}, r, s^+ \rangle \)

obtained from

\[
\text{arg max}_{\tilde{g}} \mu^l_{t} \left( a_{t:t+c-1} \mid s_{t:t+c-1}, \tilde{g}_{t:t+c-1} \right)
\]
state space
<table>
<thead>
<tr>
<th>Model</th>
<th>Ant Gather</th>
<th>Ant Maze</th>
<th>Ant Push</th>
<th>Ant Fall</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIRO</td>
<td>3.02 ± 1.49</td>
<td>0.99 ± 0.01</td>
<td>0.92 ± 0.04</td>
<td>0.66 ± 0.07</td>
</tr>
<tr>
<td>FuN representation</td>
<td>0.03 ± 0.01</td>
<td>0.0 ± 0.0</td>
<td>0.0 ± 0.0</td>
<td>0.0 ± 0.0</td>
</tr>
<tr>
<td>FuN transition PG</td>
<td>0.41 ± 0.06</td>
<td>0.0 ± 0.0</td>
<td>0.56 ± 0.39</td>
<td>0.01 ± 0.02</td>
</tr>
<tr>
<td>FuN cos similarity</td>
<td>0.85 ± 1.17</td>
<td>0.16 ± 0.33</td>
<td>0.06 ± 0.17</td>
<td>0.07 ± 0.22</td>
</tr>
<tr>
<td>FuN</td>
<td>0.01 ± 0.01</td>
<td>0.0 ± 0.0</td>
<td>0.0 ± 0.0</td>
<td>0.0 ± 0.0</td>
</tr>
<tr>
<td>SNN4HRL</td>
<td>1.92 ± 0.52</td>
<td>0.0 ± 0.0</td>
<td>0.02 ± 0.01</td>
<td>0.0 ± 0.0</td>
</tr>
<tr>
<td>VIME</td>
<td>1.42 ± 0.90</td>
<td>0.0 ± 0.0</td>
<td>0.02 ± 0.02</td>
<td>0.0 ± 0.0</td>
</tr>
</tbody>
</table>
An MDP reformulation
DAC: The Double Actor-Critic Architecture for Learning Options

Shangtong Zhang, Shimon Whiteson
Department of Computer Science
University of Oxford
{shangtong.zhang, shimon.whiteson}@cs.ox.ac.uk
semi MDP

MDP

MDP

actor-critic algorithm

actor-critic algorithm

a single critic!
high-MDP

\[
M^H = \{S^H, A^H, p^H, p_0^H, r^H, \gamma\}
\]

\[
S^H_t = O^+ \times S,
A^H_t = O
\]

\[
p^H(S^H_{t+1} | S^H_t, A^H_t) = p^H((O_t, S_{t+1})|(O_{t-1}, S_t), A^H_t) = \Pi_{A^H_t=O_t} p(S_{t+1}|S_t, O_t)
\]

\[
r^H(S^H_t, A^H_t) = \ldots
\pi^H(A^H_t | S^H_t) = \ldots
\]

low-MDP

\[
M^L = \{S^L, A^L, p^L, p_0^L, r^L, \gamma\}
\]

\[
S^L_t = S \times O,
A^L_t = A
\]

\[
p^L(S^L_{t+1} | S^L_t, A^L_t) = \ldots
\]

\[
r^L(S^L_t, A^L_t) = \ldots
\pi^L(A^L_t | S^L_t) = \ldots
\]
while keeping fixed 
\[ \pi^H(A_t^H | S_t^H) \quad \text{optimizing} \quad \{\pi_o\} \]

is like

while keeping fixed 
\[ \pi^L(A_t^L | S_t^L) \quad \text{optimizing} \quad \pi, \{\beta_o\} \]

\[ \{\pi_o\} \]

\[ \{\pi_o\} \]
We consider a transfer learning setting where after the first 1M training steps, we switch to a new task and train the agent for other 1M steps. The agent is not aware of the task switch. The two tasks are correlated and we expect learned options from the first task can be used to accelerate learning of the second task.
A more general auxiliary reward
Hierarchical Reinforcement Learning with Advantage-Based Auxiliary Rewards

Siyuan Li*
IIS, Tsinghua University
sy-li17@mails.tsinghua.edu.cn

Rui Wang*
Tsinghua University
rui1@stanford.edu

Minxue Tang
Tsinghua University
tangmx16@mails.tsinghua.edu.cn

Chongjie Zhang
IIS, Tsinghua University
chongjie@tsinghua.edu.cn
[slide from Andrea]
\[ r_t^I = \frac{1}{c} \sum_{i=1}^{c} s_{t-i}, g_{t-i} \]

\[ r_t^l = \frac{1}{k} A_h(s_t^h, a_t^h) \]
(a) Ant Maze
(b) Swimmer Maze
(c) Ant Gather
Composable and reusable motor primitives
MCP: Learning Composable Hierarchical Control with Multiplicative Compositional Policies

Xue Bin Peng, Michael Chang, Grace Zhang, Pieter Abbeel, Sergey Levine
Department of Electrical Engineering and Computer Science
University of California, Berkeley
{xbpeng, mbchang, grace.zhang}@berkeley.edu
pabbeel@cs.berkeley.edu
svlevine@eecs.berkeley.edu
\[ \pi(a | s, g) = \frac{1}{Z(s, g)} \prod_{i=1}^{k} \pi_i(a | s, g)^{w_i(s, g)}, \]
“walk forward”

```
...  
W    99%  
...  
...  
...  
RM   0%  
...  
...  
ex   1%  
...  
...  
```

“eat”

```
...  
W    0%  
...  
...  
...  
RM   99%  
...  
ex   1%  
...  
...  
```
\[ \mu^j(s, g) = \frac{1}{\sum_{l=1}^{k} \frac{w_i(s,g)}{\sigma_i(s,g)}} \sum_{i=1}^{k} \frac{w_i(s,g)}{\sigma_i(s,g)} \mu_i^j(s, g), \]
\[ \pi(a \mid s, g) = \frac{1}{Z(s, g)} \prod_{i=1}^{k} \pi_i(a \mid s, g) w_i(s, g), \]
Dribble: T-Rex

Finetune

Latent Space [Merel et al., 2018]

MCP (Ours)
Epilogue
Summary

- off-policy enabling
- semi MDP -> 2 MDPS
- a more general auxiliary reward
- composable and reusable motor critics
Sources

- Data Efficient Hierarchical Reinforcement Learning
- DAC: The Double Actor-Critic Architecture for Learning Options
- Hierarchical Reinforcement Learning with Advantage-Based Auxiliary Rewards
- MCP: Learning Composable Hierarchical Control with Multiplicative Compositional Policies

some additional nice transfer learning results from the Berkeley lab
- Hierarchically Decoupled Imitation for Morphological Transfer