Hierarchical DRL

part 2, improvements

The problem of off-policy architectures

on-policy vs. off-policy

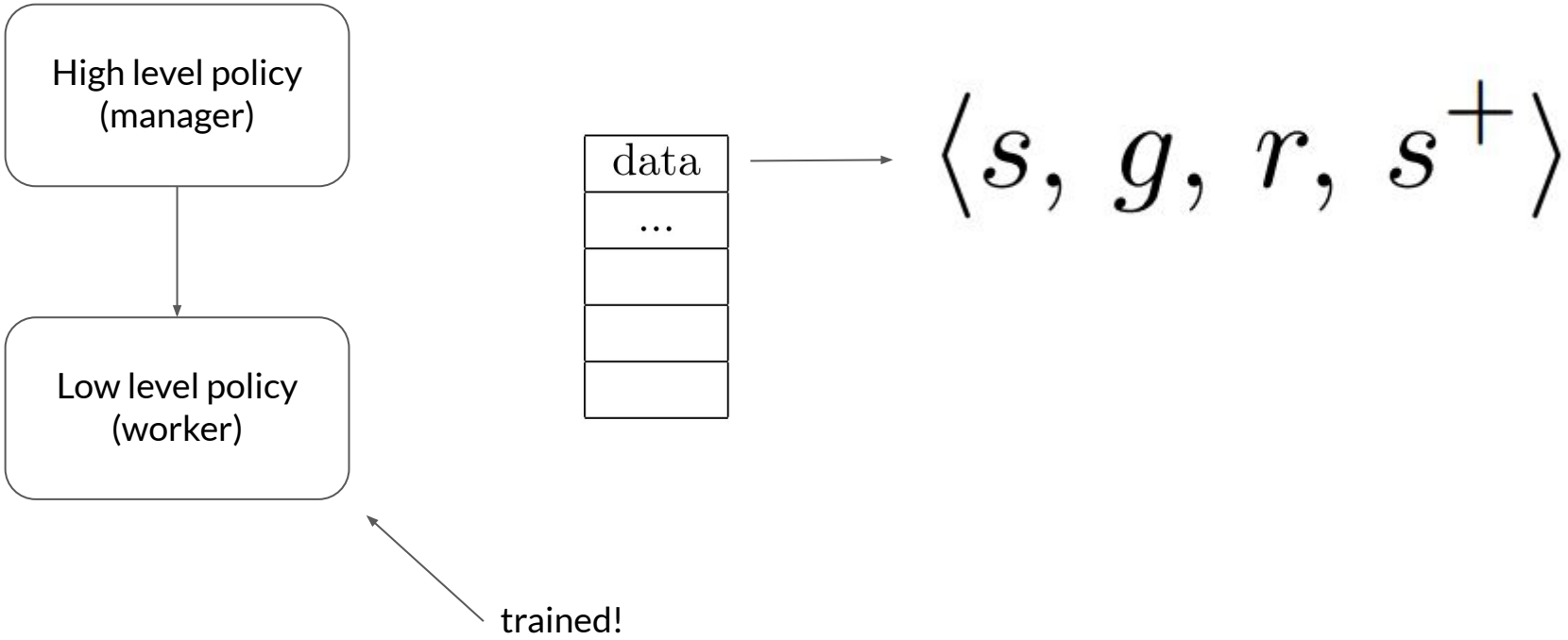
on-policy

- 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)
trained!
data
...
$\langle s, g, r, s^+ \rangle$

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

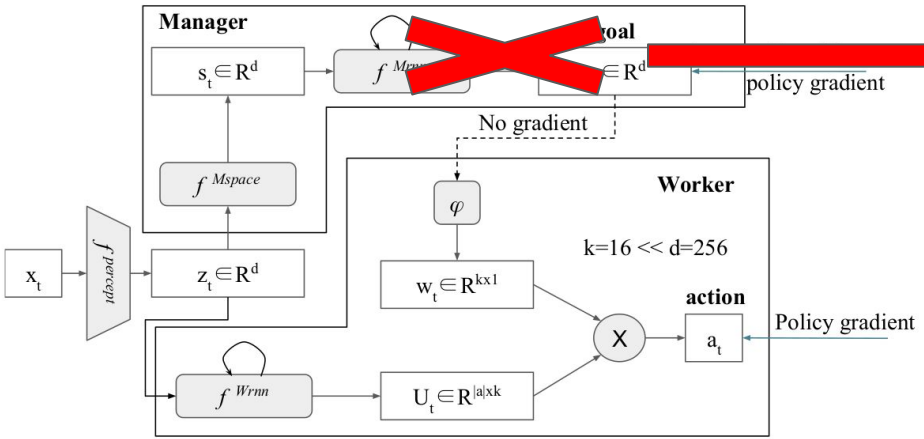
Sergey Levine†
Google Brain
slevine@google.com

$$\langle s,\ g,\ r,\ s^{+} \rangle$$

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

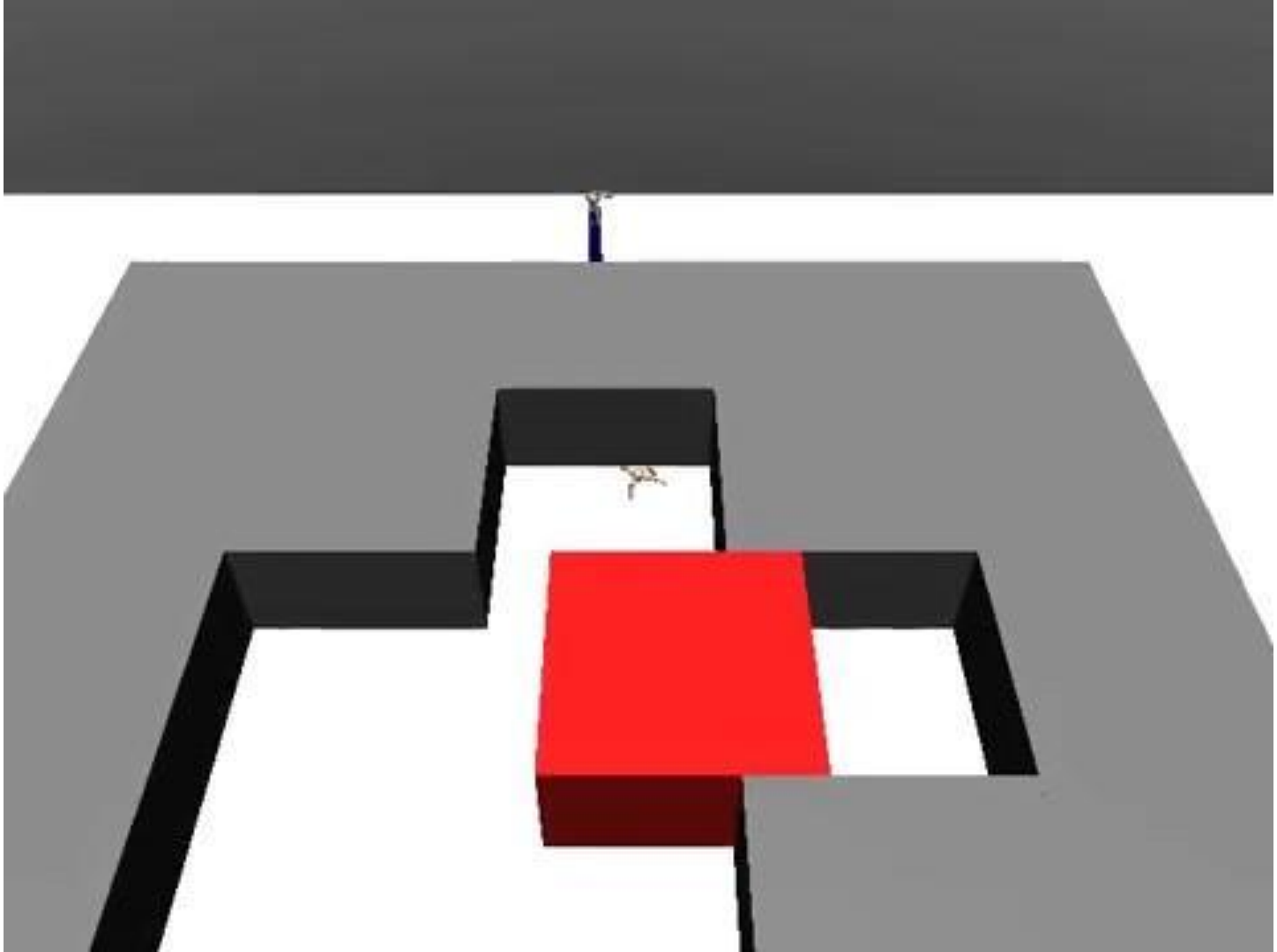
obtained from

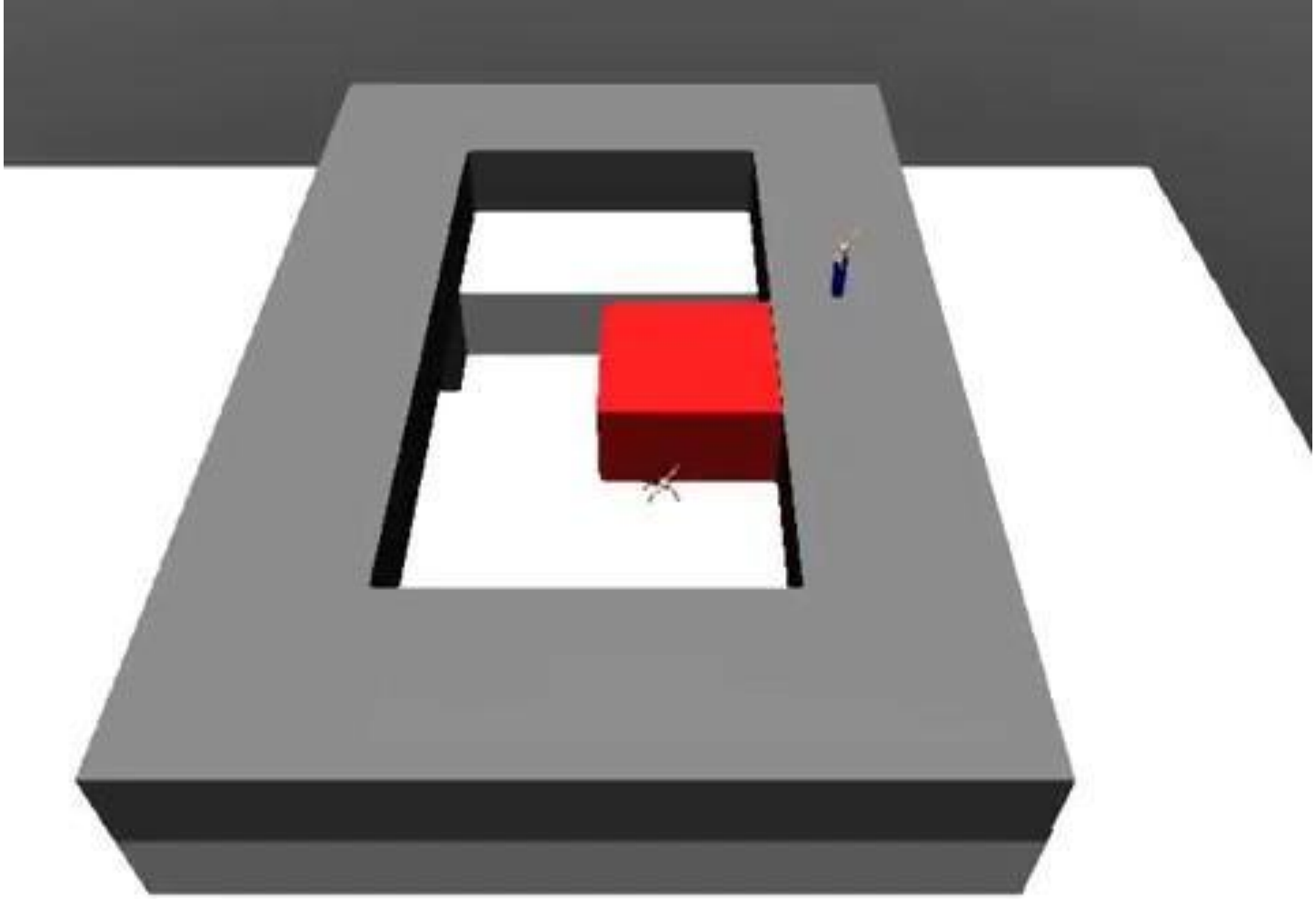
$$\arg\max_{\tilde{g}} \mu^{lo}(a_{t:t+c-1} | s_{t:t+c-1},\ \tilde{g}_{t:t+c-1})$$


Manager
$s_t \in R^d$
f^{Mrnn}
goal
$\in R^d$
policy gradient
No gradient
f^{Mspace}
Worker
φ
k=16 << d=256
$f^{percept}$
x_t
$z_t \in R^d$
$w_t \in R^{kx1}$
action
X
a_t
Policy gradient
f^{Wrnn}
$U_t \in R^{|a|xk}$

state space

	Ant Gather	**Ant Maze**	**Ant Push**	**Ant Fall**
HIRO	**3.02±1.49**	**0.99±0.01**	**0.92±0.04**	**0.66±0.07**
FuN representation	0.03 ± 0.01	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
FuN transition PG	0.41 ± 0.06	0.0 ± 0.0	0.56 ± 0.39	0.01 ± 0.02
FuN cos similarity	0.85 ± 1.17	0.16 ± 0.33	0.06 ± 0.17	0.07 ± 0.22
FuN	0.01 ± 0.01	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
SNN4HRL	1.92 ± 0.52	0.0 ± 0.0	0.02 ± 0.01	0.0 ± 0.0
VIME	1.42 ± 0.90	0.0 ± 0.0	0.02 ± 0.02	0.0 ± 0.0

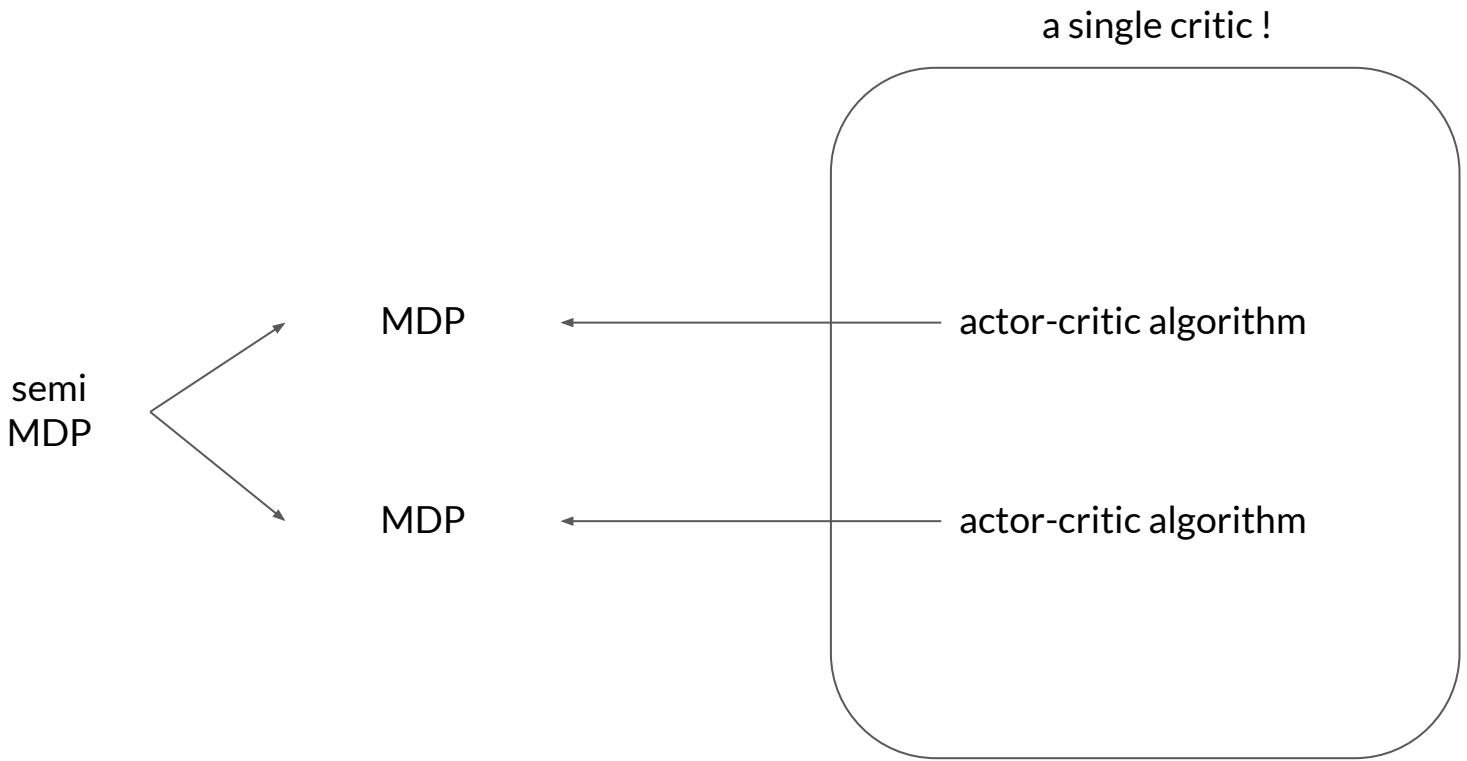




An MDP reformulation

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


a single critic !
semi
MDP
MDP
MDP
actor-critic algorithm
actor-critic algorithm

high-MDP

$$M^{\mathcal{H}} = \{\mathcal{S}^{\mathcal{H}}, \mathcal{A}^{\mathcal{H}}, p^{\mathcal{H}}, p_0^{\mathcal{H}}, r^{\mathcal{H}}, \gamma\}$$

$$\mathcal{S}_t^{\mathcal{H}} = \mathcal{O}^+ \times \mathcal{S}, \quad \mathcal{A}_t^{\mathcal{H}} = \mathcal{O}$$

$$p^{\mathcal{H}}(S_{t+1}^{\mathcal{H}} | S_t^{\mathcal{H}}, A_t^{\mathcal{H}}) = p^{\mathcal{H}}((O_t, S_{t+1}) | (O_{t-1}, S_t), A_t^{\mathcal{H}}) = \mathbb{I}_{A_t^{\mathcal{H}} = O_t} p(S_{t+1} | S_t, O_t)$$

$$r^{\mathcal{H}}(S_t^{\mathcal{H}}, A_t^{\mathcal{H}}) = \dots \quad \pi^{\mathcal{H}}(A_t^{\mathcal{H}} | S_t^{\mathcal{H}}) = \dots$$

low-MDP

$$M^{\mathcal{L}} = \{\mathcal{S}^{\mathcal{L}}, \mathcal{A}^{\mathcal{L}}, p^{\mathcal{L}}, p_0^{\mathcal{L}}, r^{\mathcal{L}}, \gamma\}$$

$$\mathcal{S}_t^{\mathcal{L}} = \mathcal{S} \times \mathcal{O}, \quad \mathcal{A}_t^{\mathcal{L}} = \mathcal{A}$$

$$p^{\mathcal{L}}(S_{t+1}^{\mathcal{L}} | S_t^{\mathcal{L}}, A_t^{\mathcal{L}}) = \dots$$

$$r^{\mathcal{L}}(S_t^{\mathcal{L}}, A_t^{\mathcal{L}}) = \dots \quad \pi^{\mathcal{L}}(A_t^{\mathcal{L}} | S_t^{\mathcal{L}}) = \dots$$

optimizing

$\pi^{\mathcal{H}}(A_t^{\mathcal{H}}|S_t^{\mathcal{H}})$

while keeping fixed

$\{\pi_o\}$

is like

optimizing

$\pi, \{\beta_o\}$

optimizing

$\pi^{\mathcal{L}}(A_t^{\mathcal{L}}|S_t^{\mathcal{L}})$

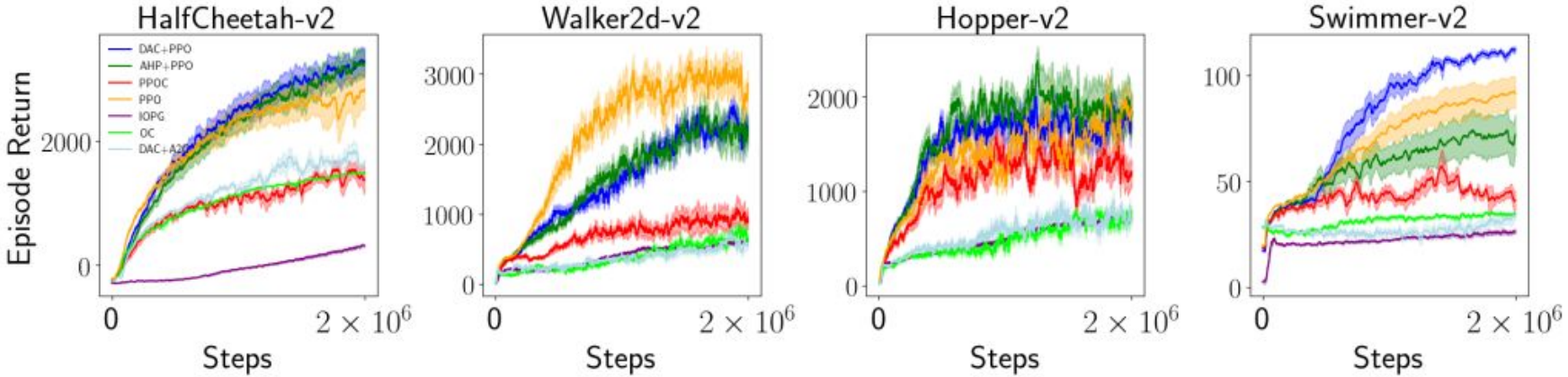
while keeping fixed

$\pi, \{\beta_o\}$

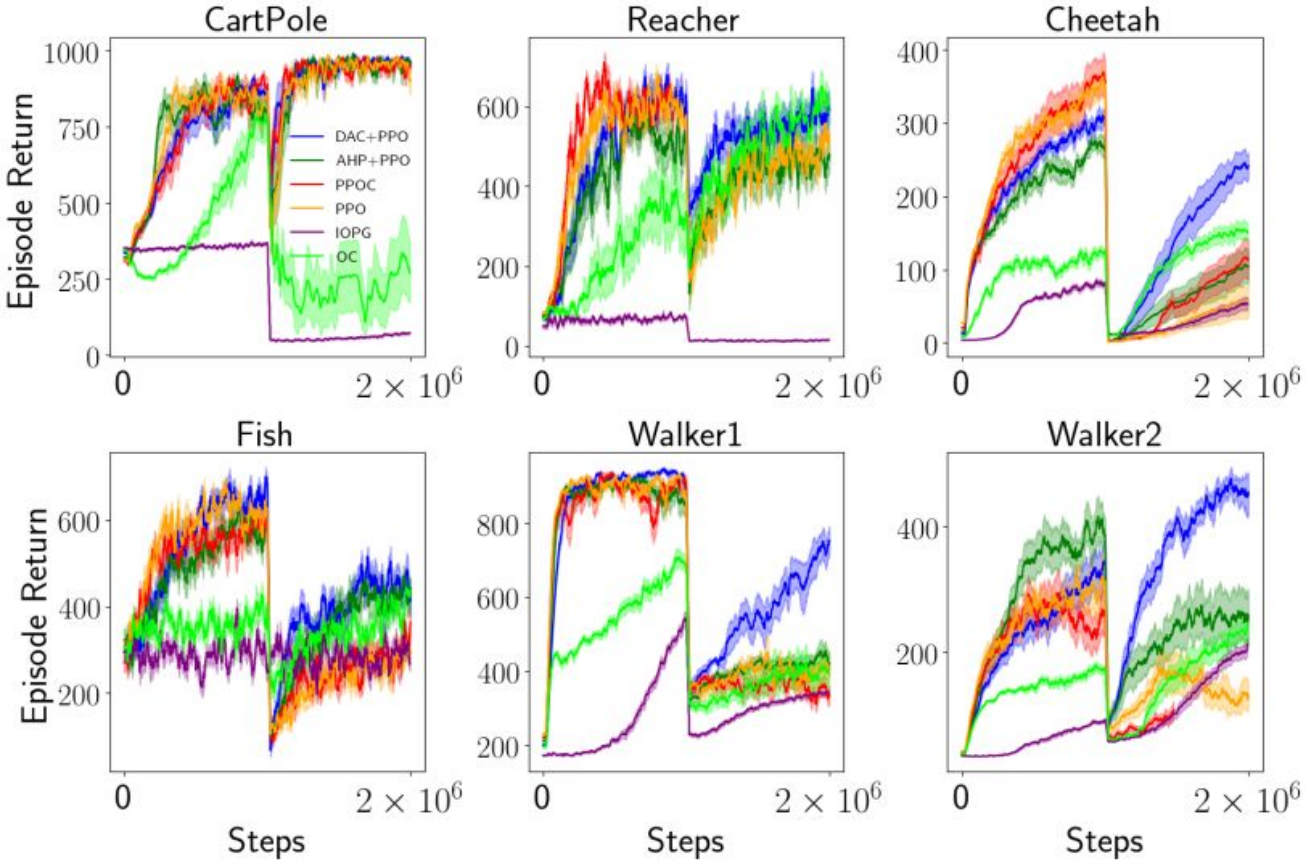
is like

optimizing

$\{\pi_o\}$


HalfCheetah-v2
Walker2d-v2
Hopper-v2
Swimmer-v2
Episode Return
Steps
DAC+PPO
AHP+PPO
PPOC
PPO
IOPG
OC
DAC+A2C
0
2000
3000
1000
100
50
2×10^6

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.


CartPole
Reacher
Cheetah
Fish
Walker1
Walker2
Episode Return
Steps
DAC+PPO
AHP+PPO
PPOC
PPO
IOPG
OC
0
2×10^6

A more general auxiliary reward

Hierarchical Reinforcement Learning with Advantage-Based Auxiliary Rewards

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

Rui Wang*
Tsinghua University
rui1@stanford.edu

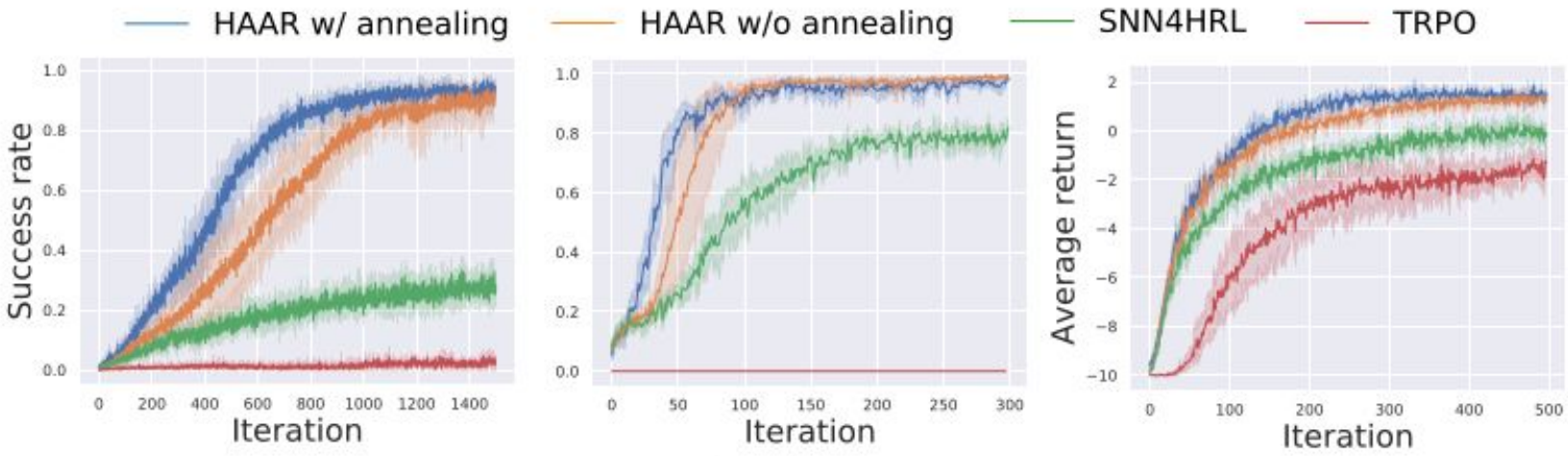
Minxue Tang
Tsinghua University
tangmx16@mails.tsinghua.edu.cn

Chongjie Zhang
IIIS, Tsinghua University
chongjie@tsinghua.edu.cn

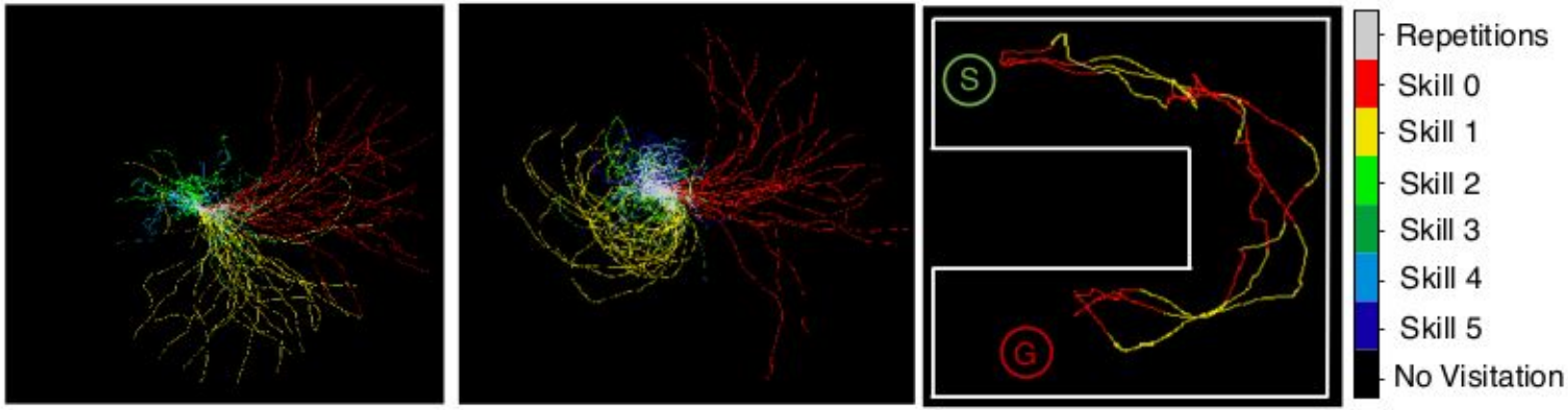
[slide from Andrea]

$$r_t^I = 1/c \sum_{i=1}^{c} (\quad = s_{t-i}, g_{t-i})$$

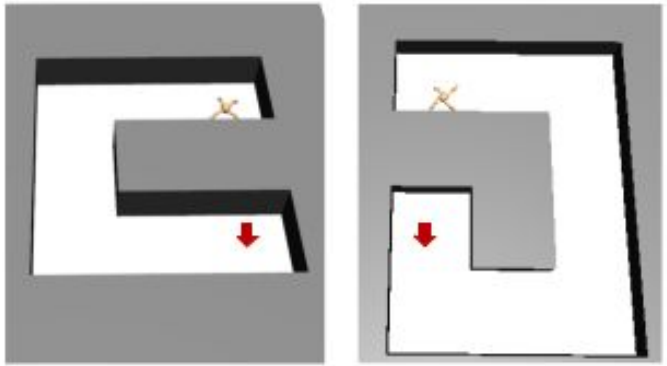
$$r_{\substack{i \\ t \leq i < t+k}}^{l} = \frac{1}{k} A_h(s_t^h, a_t^h)$$



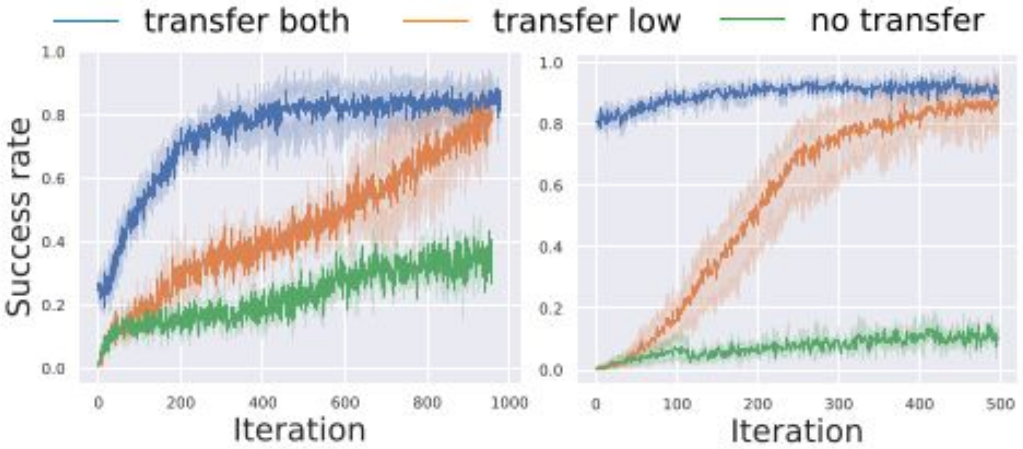
(a) Ant Maze (b) Swimmer Maze (c) Ant Gather


S
G
Repetitions
Skill 0
Skill 1
Skill 2
Skill 3
Skill 4
Skill 5
No Visitation

(a) (b) (c)



(a) (b)

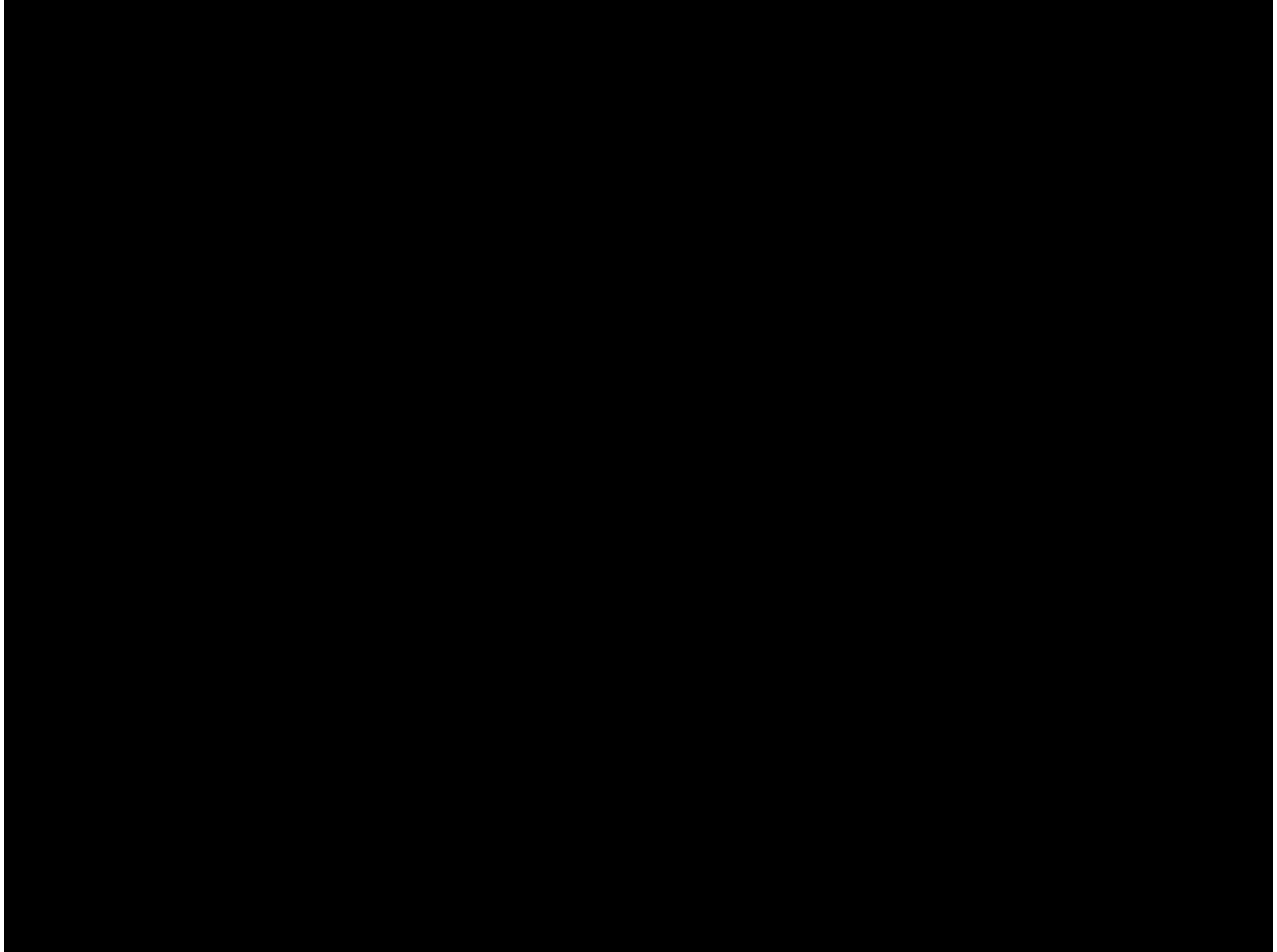

transfer both
transfer low
no transfer
Success rate
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000
Iteration
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
500
Iteration

(c) (d)

Composable and reusable motor primitives

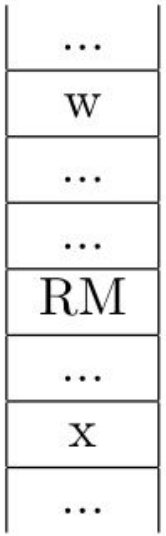
MCP: Learning Composable Hierarchical Control with Multiplicative Compositional Policies

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”

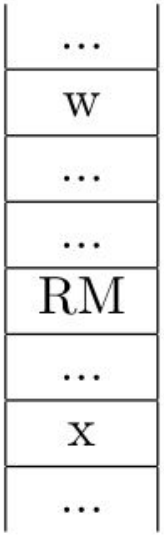


99%

0%

1%

“eat”



0%

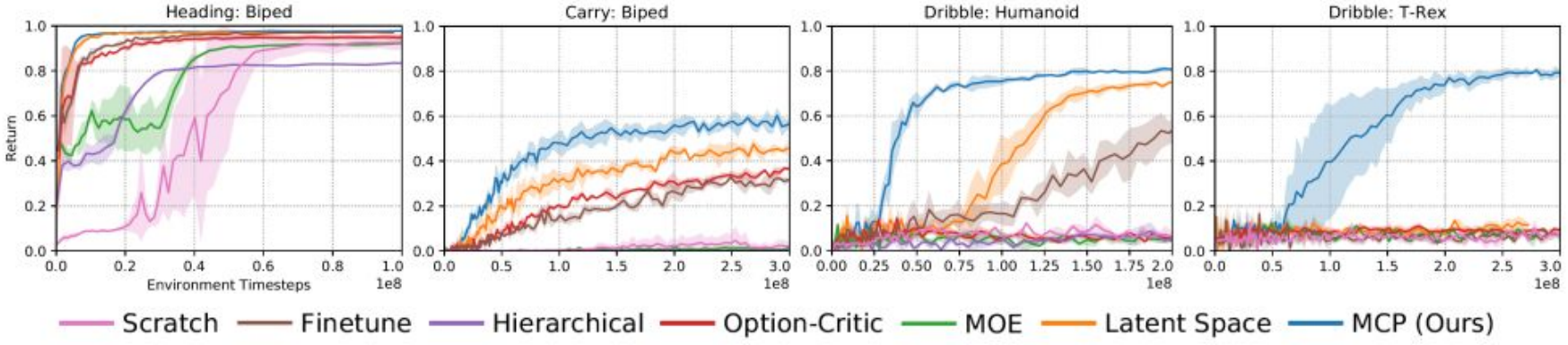
99%

1%

$$\mu^j(s,g) = \frac{1}{\sum_{l=1}^{k} \frac{w_l(s,g)}{\sigma_l^j(s,g)}} \sum_{i=1}^{k} \frac{w_i(s,g)}{\sigma_i^j(s,g)} \mu_i^j(s,g),$$

$$\pi(a|s,g) = \frac{1}{Z(s,g)} \prod_{i=1}^{k} \pi_i(a|s,g)^{w_i(s,g)},$$

$$\pi(a|s,g) = \frac{1}{Z(s,g)} \prod_{i=1}^{k} \pi_i(a|s)^{w_i(s,g)},$$


Heading: Biped
Carry: Biped
Dribble: Humanoid
Dribble: T-Rex
Return
Environment Timesteps
1e8
Scratch
Finetune
Hierarchical
Option-Critic
MOE
Latent Space
MCP (Ours)

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