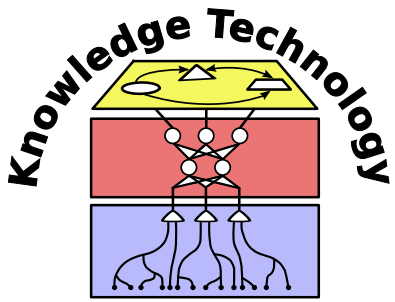
Interaction is more beneficial in complex reinforcement learning problems than in simple ones

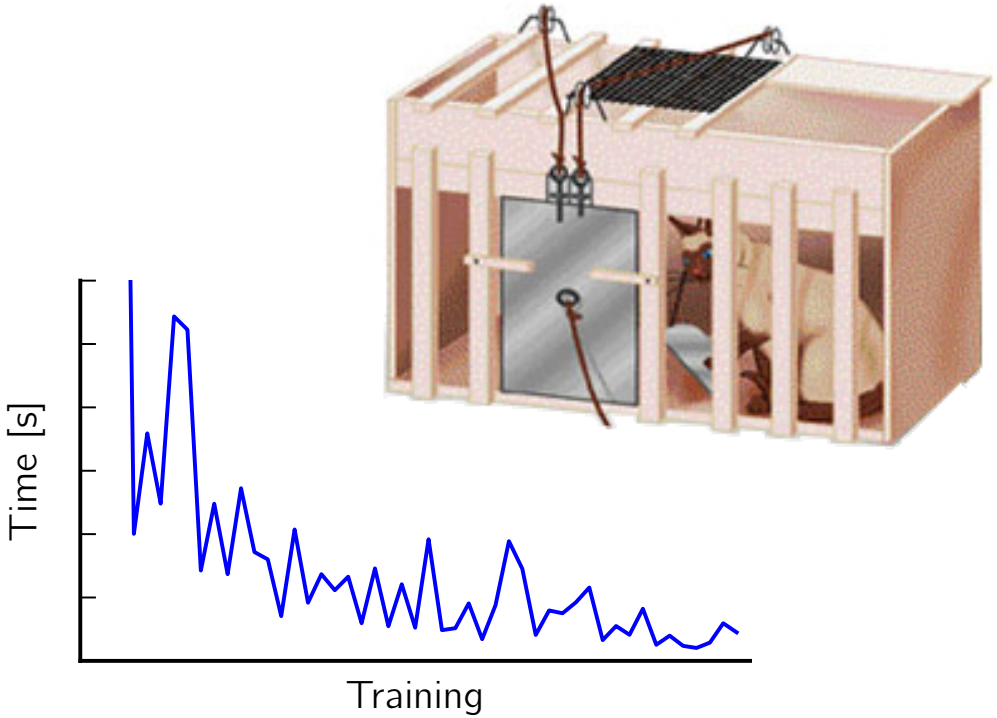
Chris Stahlhut, Nicolás Navarro-Guerrero, Cornelius Weber, Stefan Wermter
Universität Hamburg, Department Informatik



https://www.informatik.uni-hamburg.de/WTM/

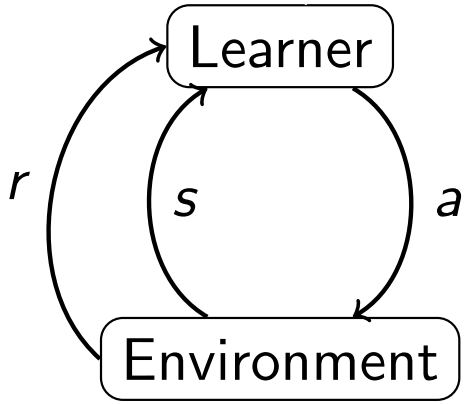
We can regard humans as social, reinforcement learning agents.

Reinforcement Learning - Inspiration



Adapted from image under public domain by Jacob Sussman
`https://commons.wikimedia.org/wiki/File:Puzzle_box.jpg`

Reinforcement Learning - Formal definition

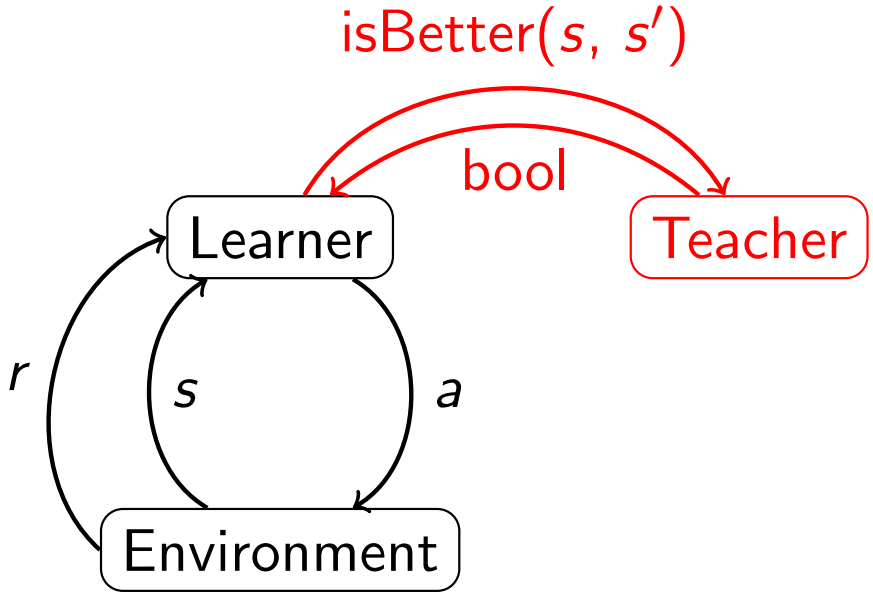


Inspired by Sutton and Barto [1998][1]

A learner senses state s, then conducts action a to maximise its reward r.

[1] Reinforcement Learning: An Introduction

Interactive Reinforcement Learning - Formal definition

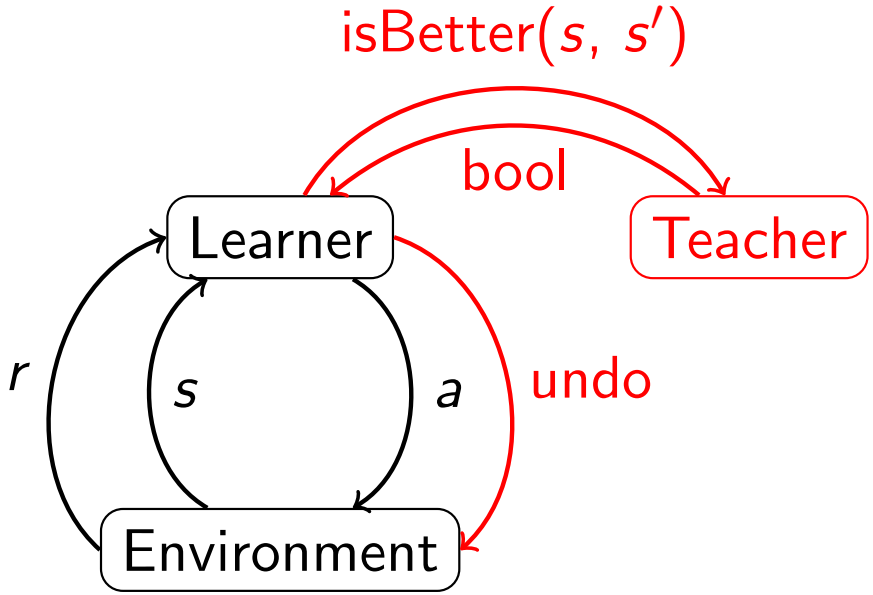


Based on the work of Thomaz and Breazeal [2008][2]

A learner asks the teacher whether the next state s' is better than the current state s, and if not, undoes the last action a.

[2] Teachable robots: Understanding human teaching behaviour to build more effective robot learners

Interactive Reinforcement Learning - Formal definition

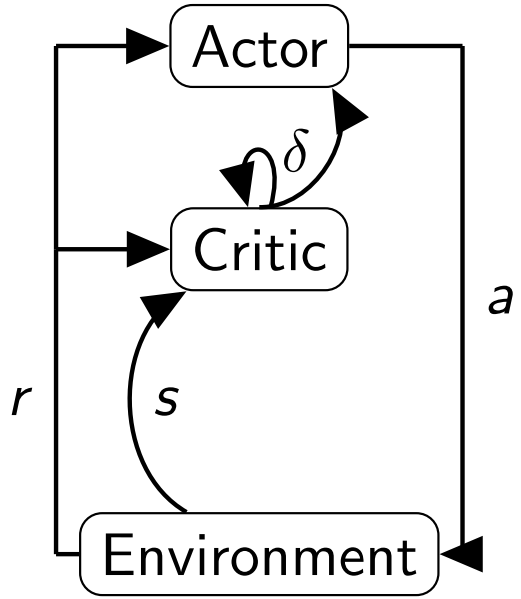


Based on the work of Thomaz and Breazeal [2008][2]

A learner asks the teacher whether the next state s' is better than the current state s, and if not, undoes the last action a.

[2] Teachable robots: Understanding human teaching behaviour to build more effective robot learners

Reinforcement Learning in continuous spaces with the Actor-critic



Inspired by Sutton and Barto [1998]

The critic learns the value-function and criticises the actor with the temporal difference error $\delta = r + \gamma \, \mathrm{V}(s') - \mathrm{V}(s)$ and the actor learns the policy π.

Cacla

Initialize $\alpha, \beta, \gamma, \sigma, s$
for t $\in \{0, 1, 2, \dots\}$ **do**
  Choose $a' \sim \pi(s) + \mathcal{N}(0, \sigma)$
  Perform a', observe r **and** s'
    $\delta = r + \gamma V(s') - V(s)$
    update Critic$(s, r + \gamma V(s'), \beta)$
    if $\delta > 0$ **then**
      updateActor(s, a', α)
  if s' is terminal **then**
    Reinitialize s'
  else
    $s = s'$

As described by van Hasselt and Wiering [2007][3]

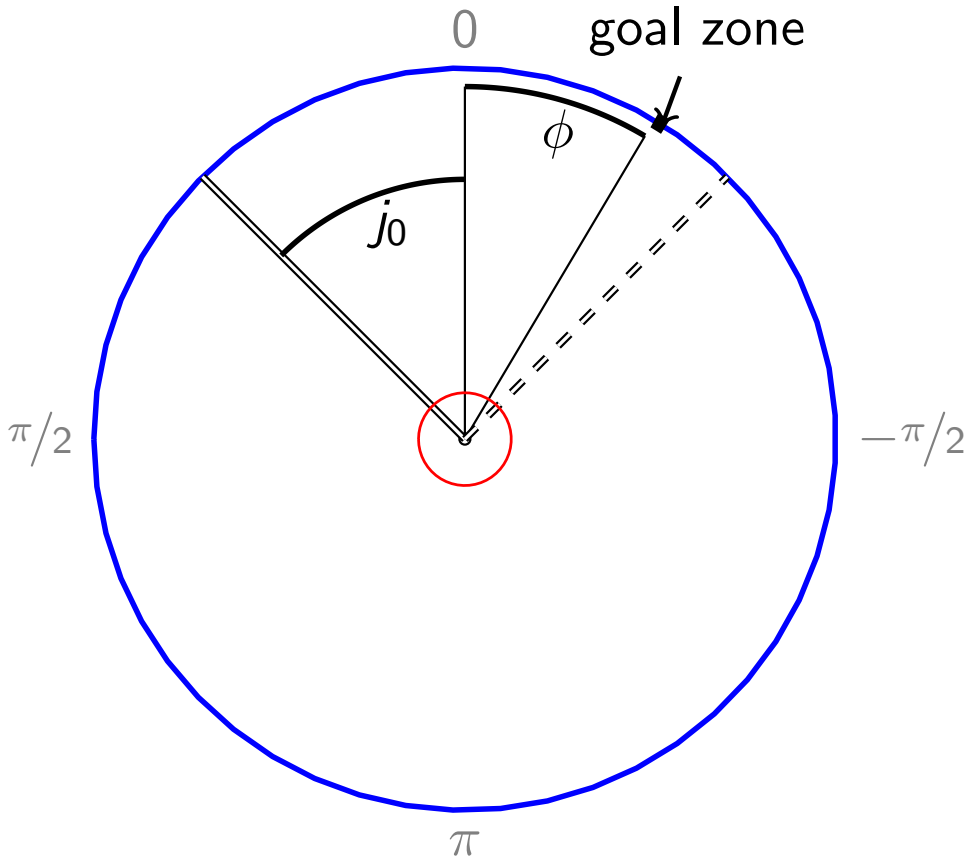
[3] Reinforcement Learning in Continuous Action Spaces

ICacla

Initialize $\alpha, \beta, \gamma, \sigma, s$
for t $\in \{0, 1, 2, \ldots\}$ **do**
    Choose $a' \sim \pi(s) + \mathcal{N}(0, \sigma)$
    Perform a', observe r **and** s'
    if $l \sim$ uniform(0, 1) $< \mathcal{L}$ then
        if isBetter(s, s') then
            $\delta = r + \gamma V(s') - V(s)$
            updateCritic$(s, r + \gamma V(s'), \beta)$
            if $\delta > 0$ **then**
                updateActor(s, a', α)
    else
        undo(a)
    if s' is terminal **then**
        Reinitialize s'
    else
        $s = s'$

Let's test Interactive Reinforcement Learning
in a continuous reaching task
with a variable goal and increasing complexity!

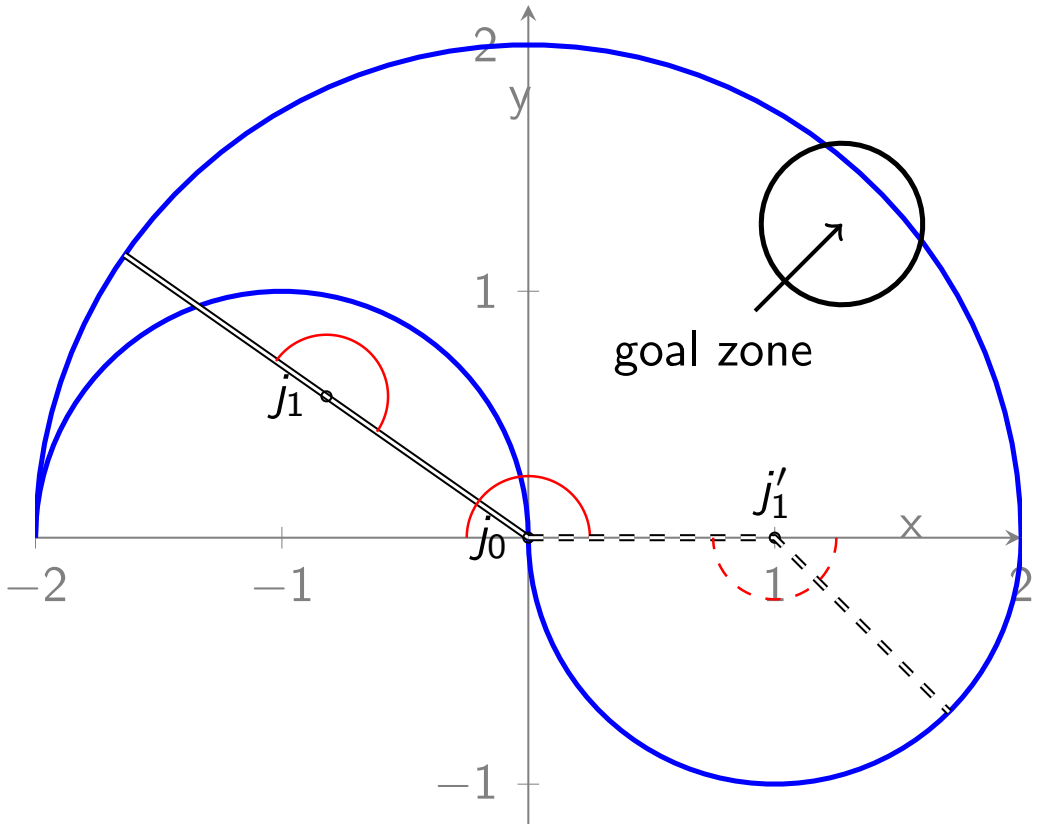
The 1-DoF arm has a circular, one dimensional workspace.



Legal actions are between 0 and $\pm 45^\circ$

$\Rightarrow$ not all targets can be reached with a single step

The 2-DoF arm has a two dimensional workspace.

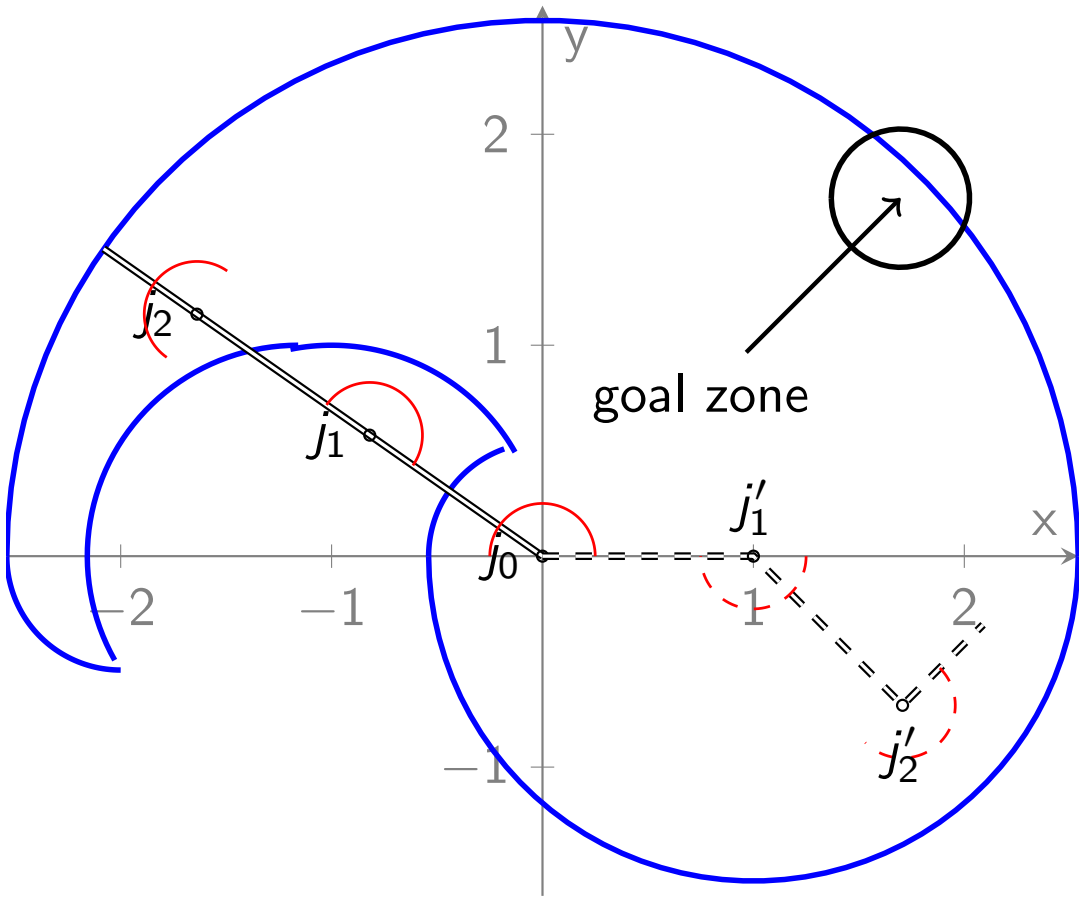


Legal actions are between 0 and $\pm 45°$

$\Rightarrow$ not all targets can be reached with a single step

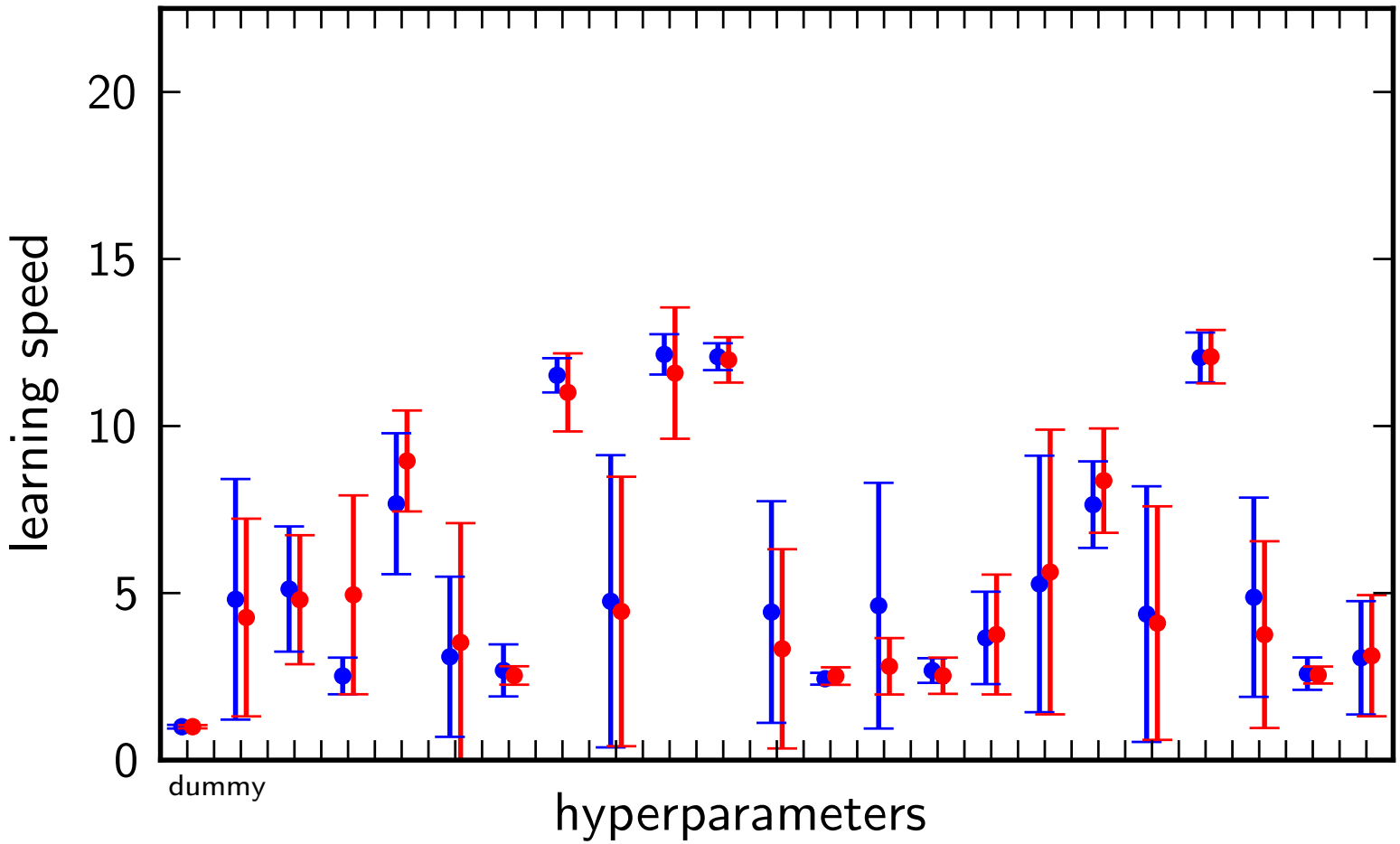
Has to occasionally increase the distance to the target

The 3-DoF arm has a two dimensional workspace.

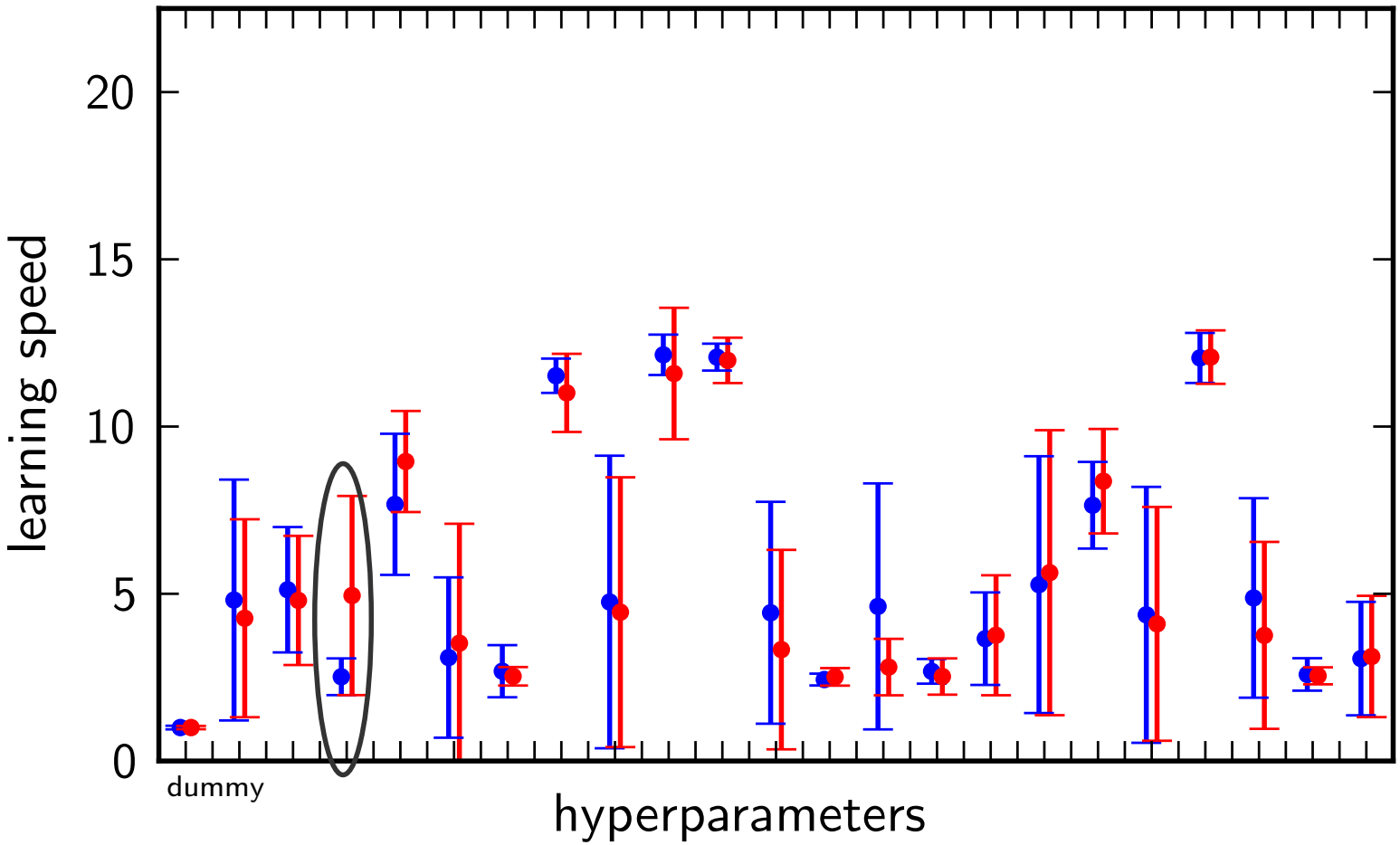


Multiple solutions for the inverse kinematics

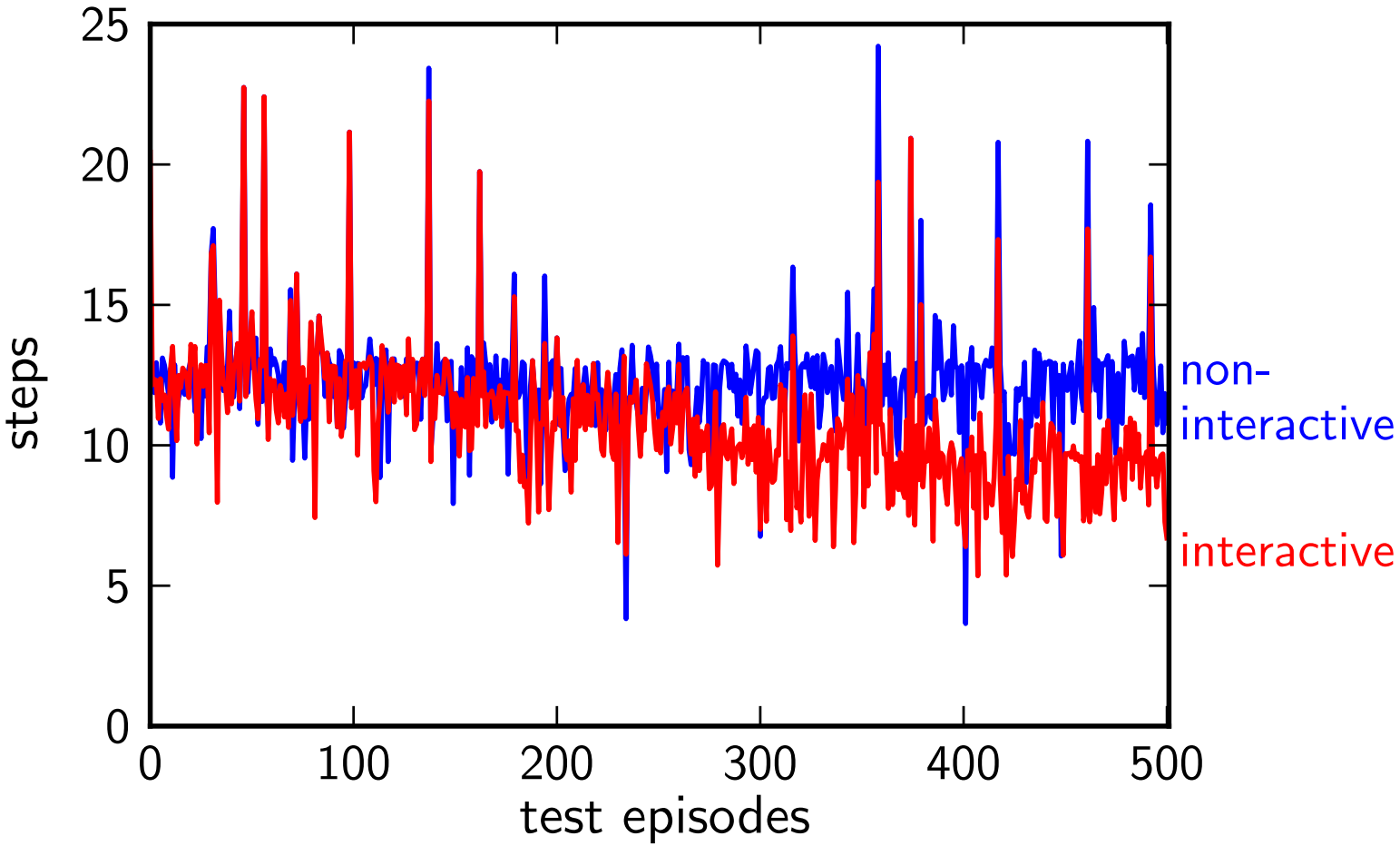
No difference between Cacla and ICacla with the 1-DoF arm



No difference between Cacla and ICacla with the 1-DoF arm



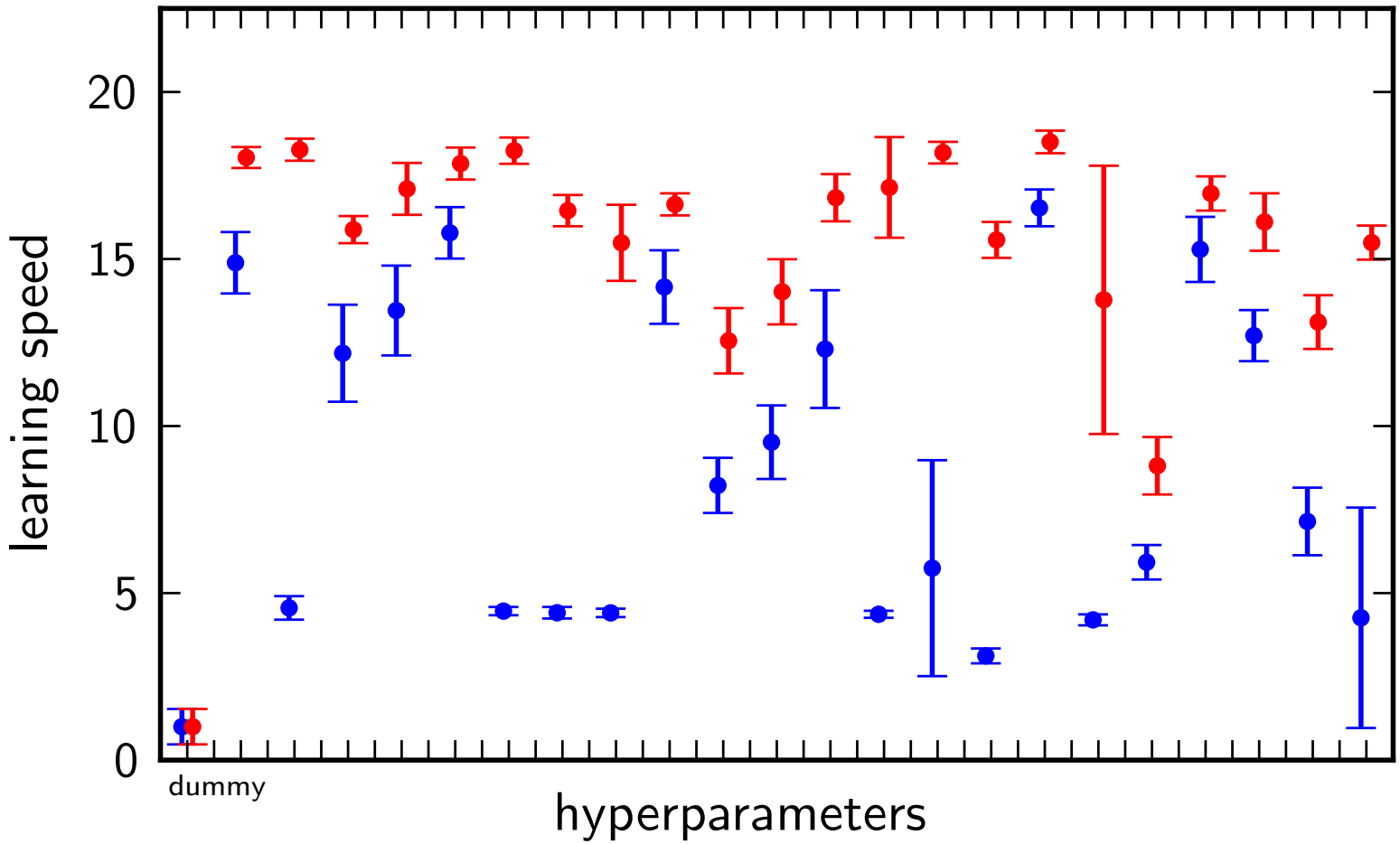
ICacla learns slightly faster
if the difference between both learners is large



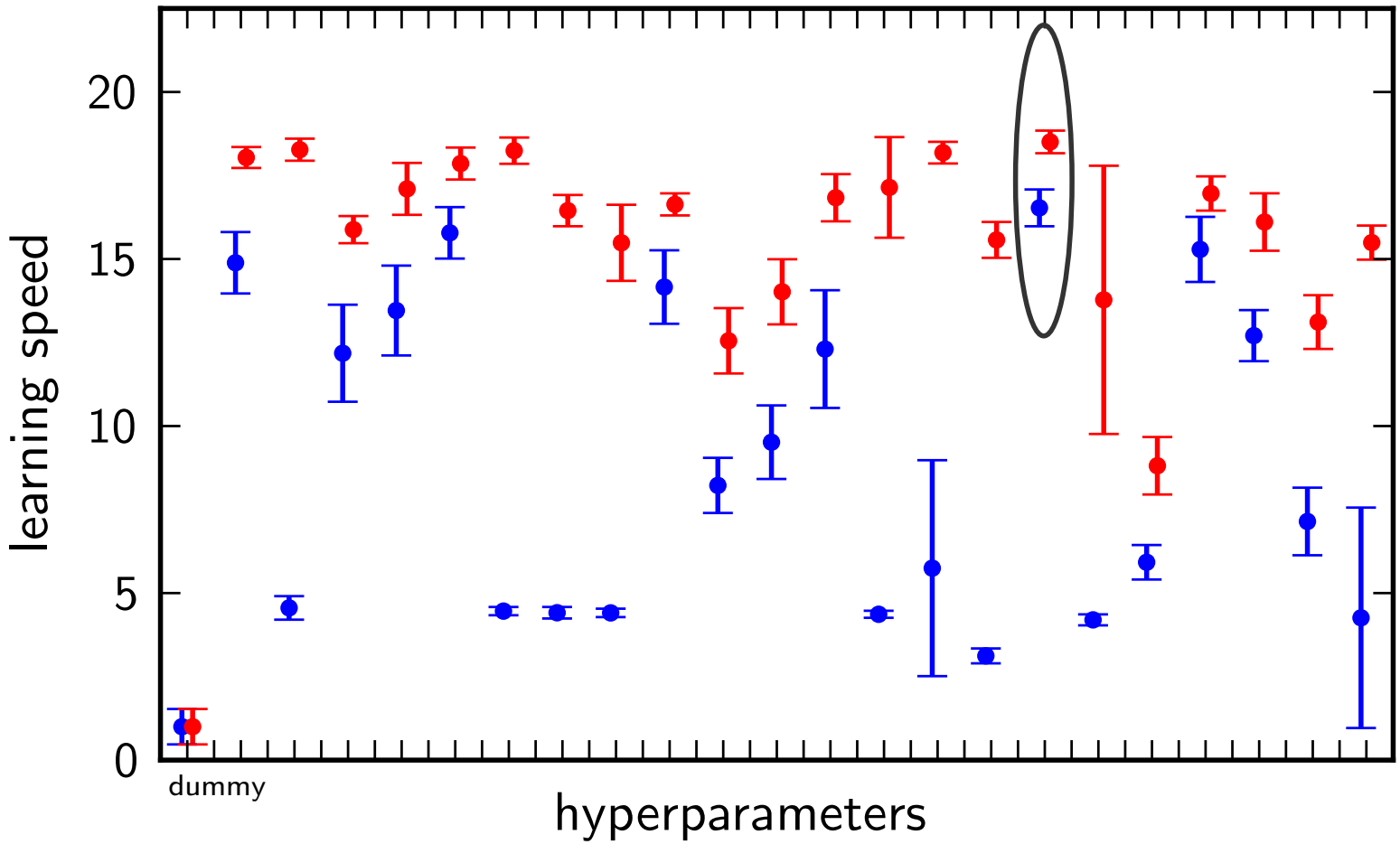
Results so far

No big difference between interactive and non-interactive learning for the simple task.

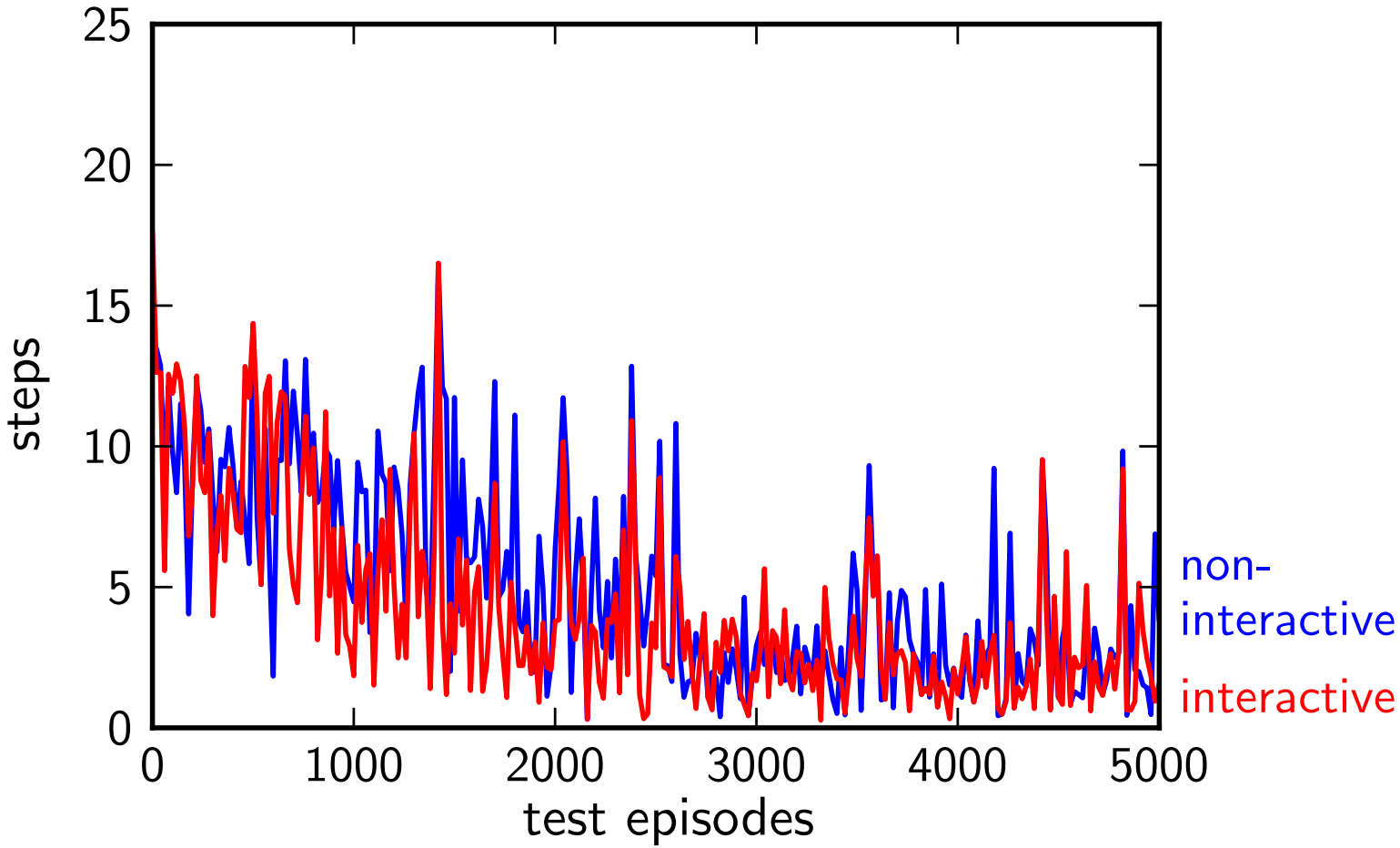
ICacla learns faster than Cacla with the 2-DoF arm



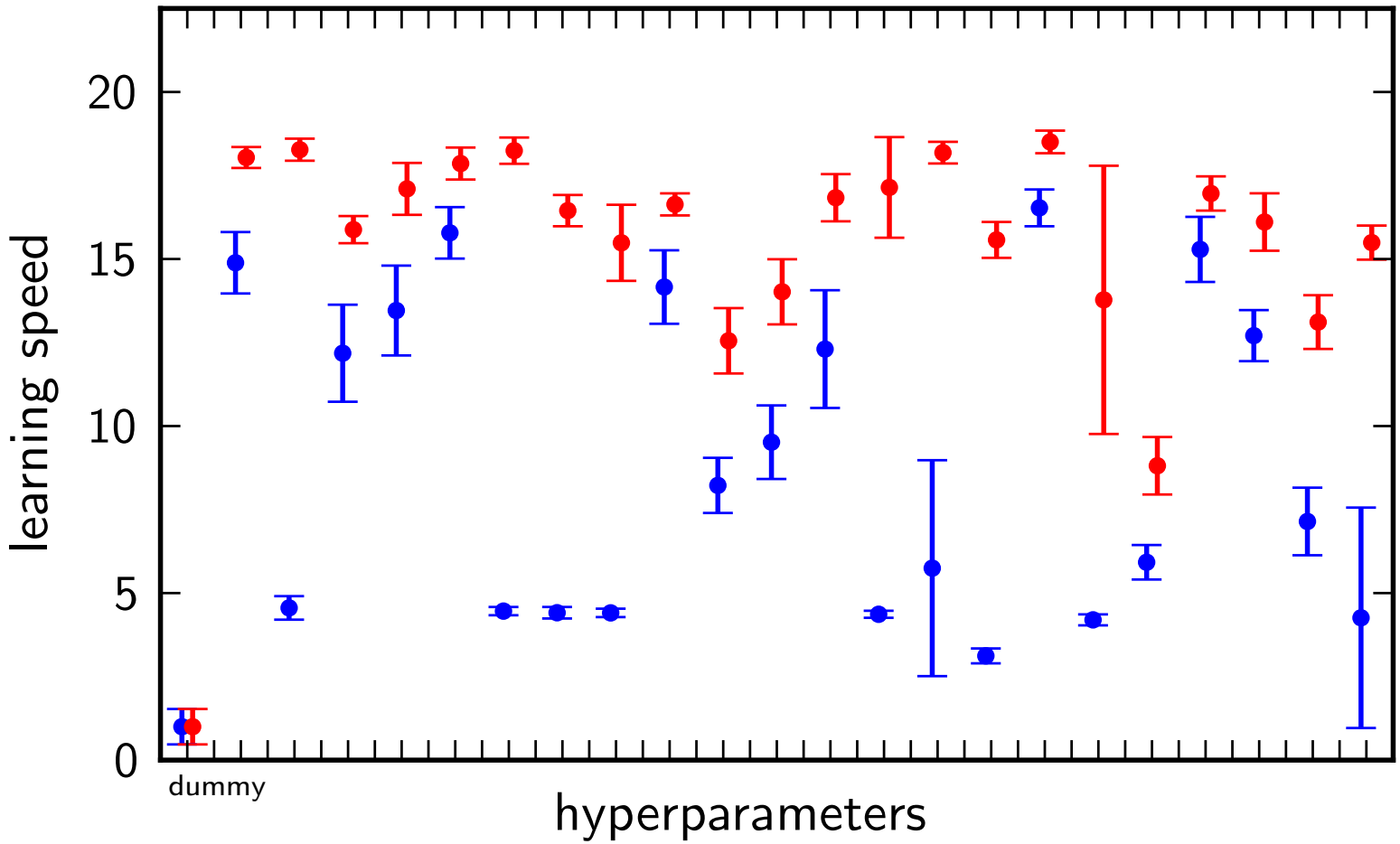
ICacla learns faster than Cacla with the 2-DoF arm



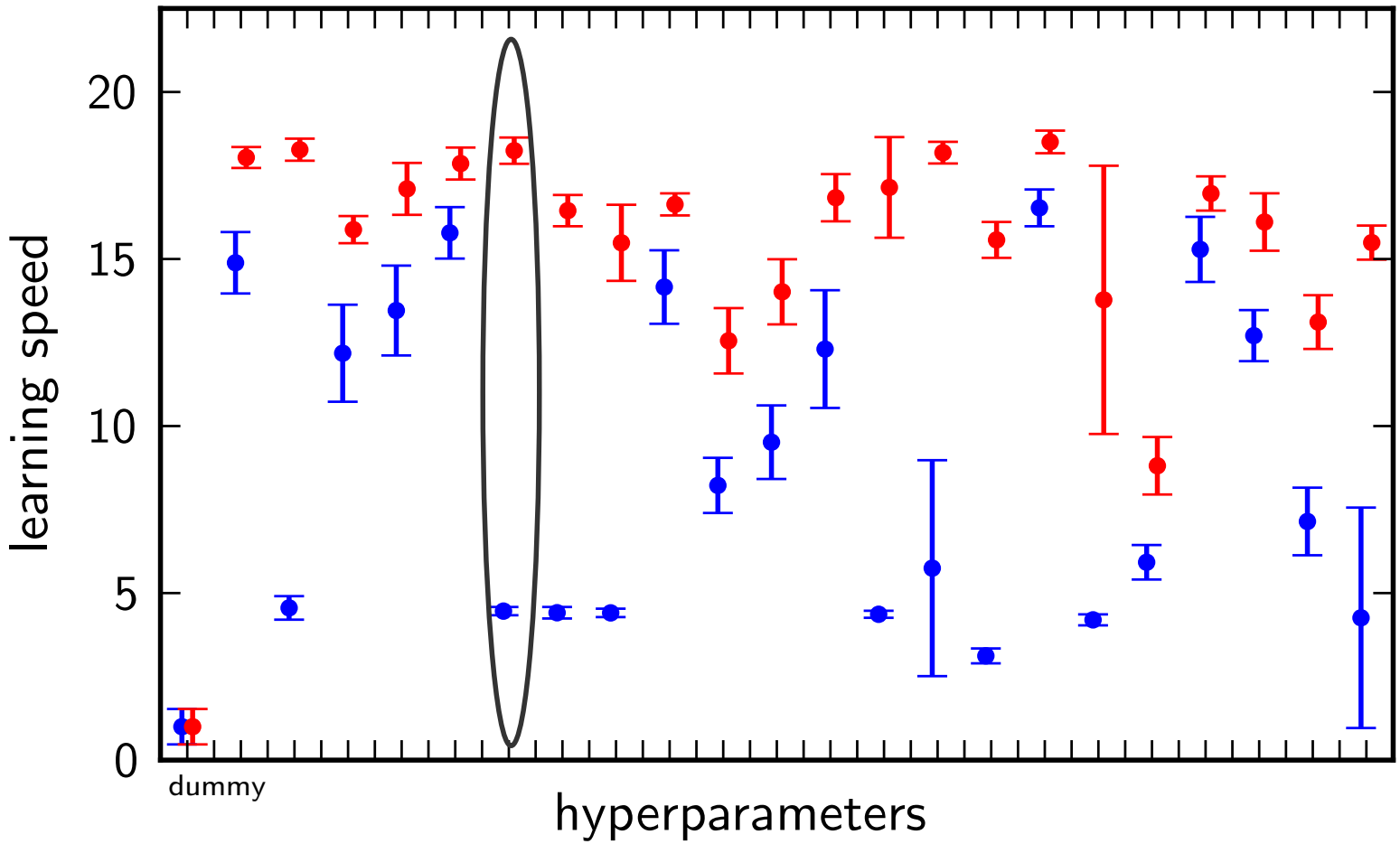
Both learners learn equally well if the hyperparameters are well set



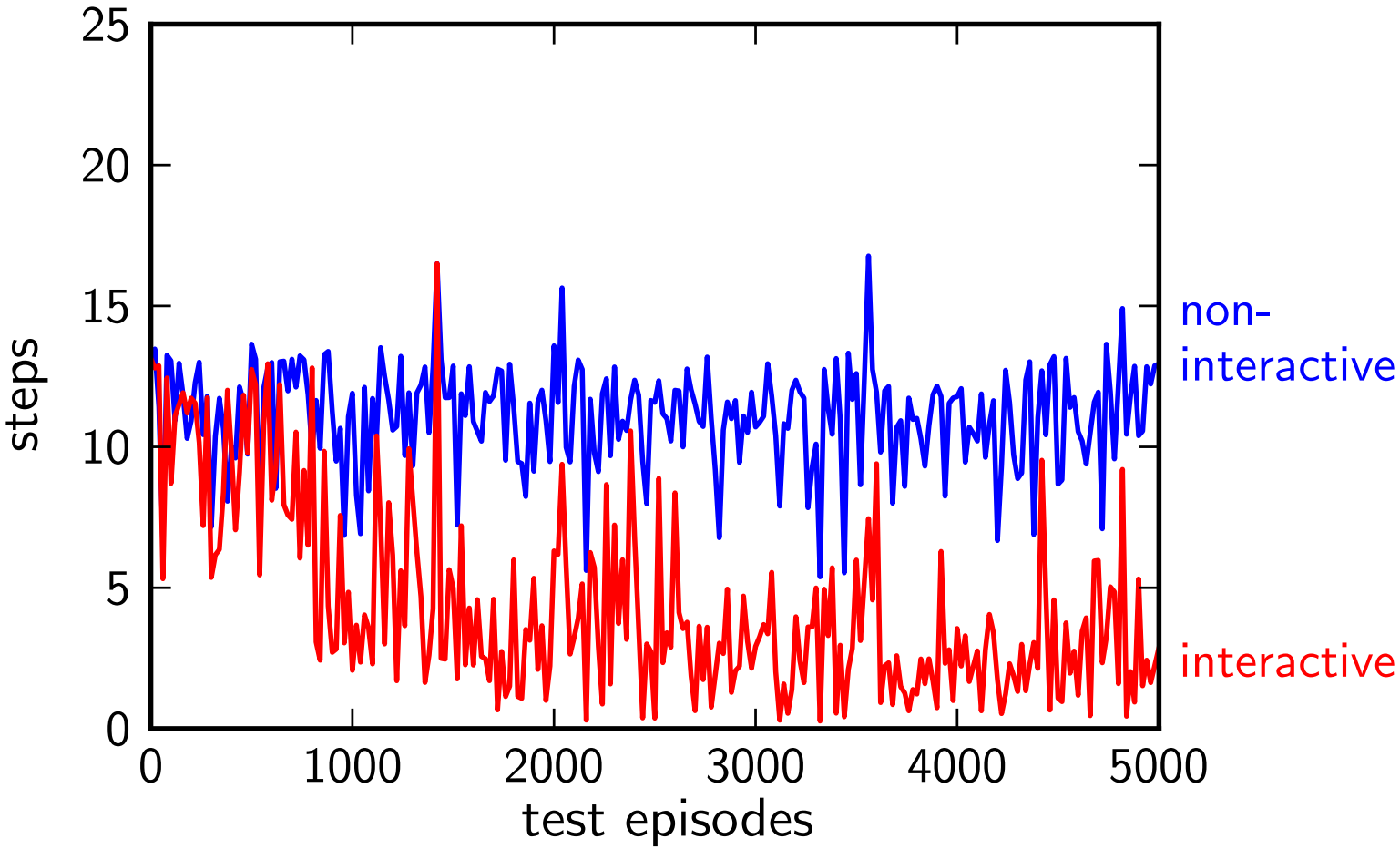
ICacla learns faster than Cacla with the 2-DoF arm



ICacla learns faster than Cacla with the 2-DoF arm



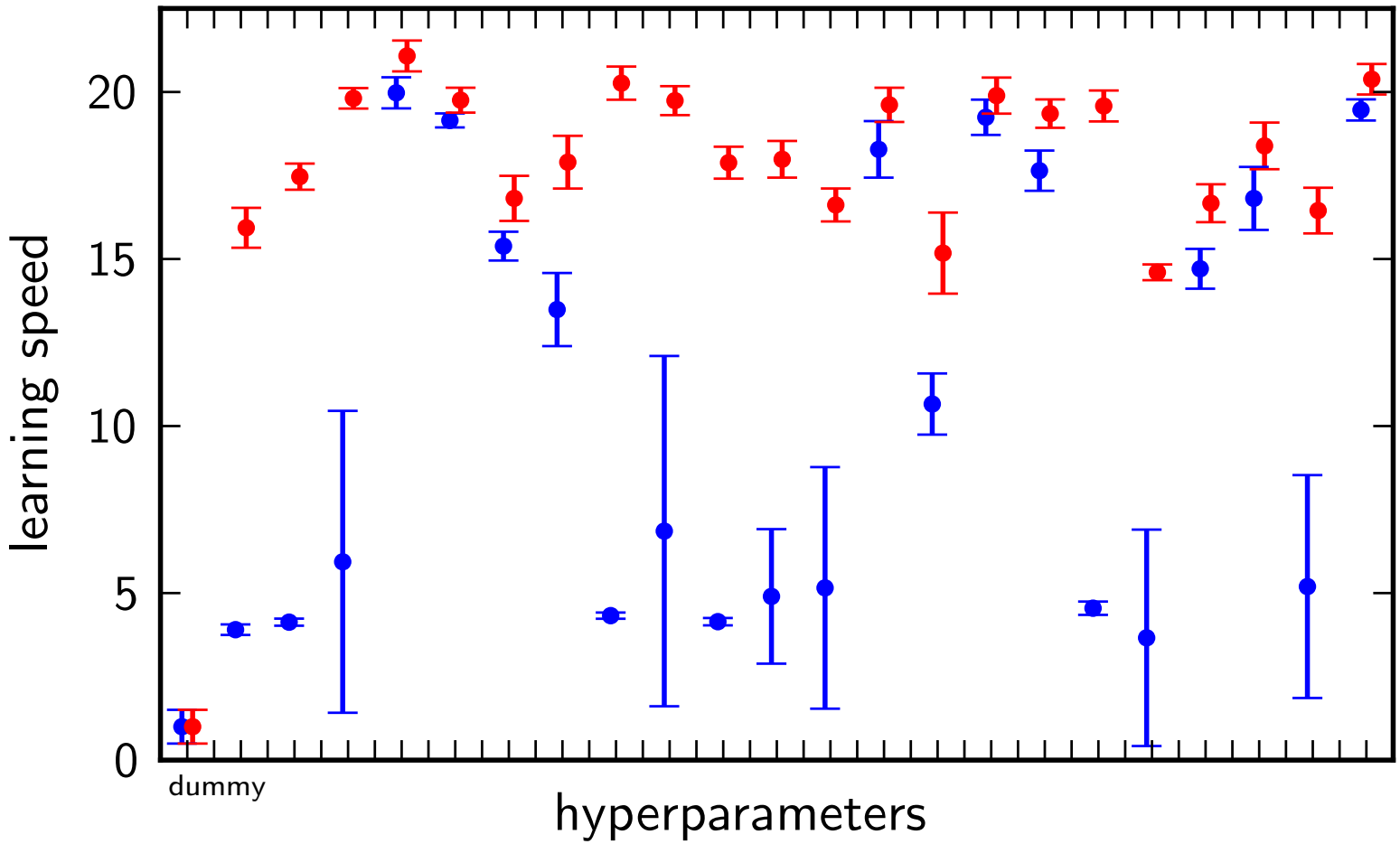
ICacla learns much faster
if the hyperparameters as poorly set



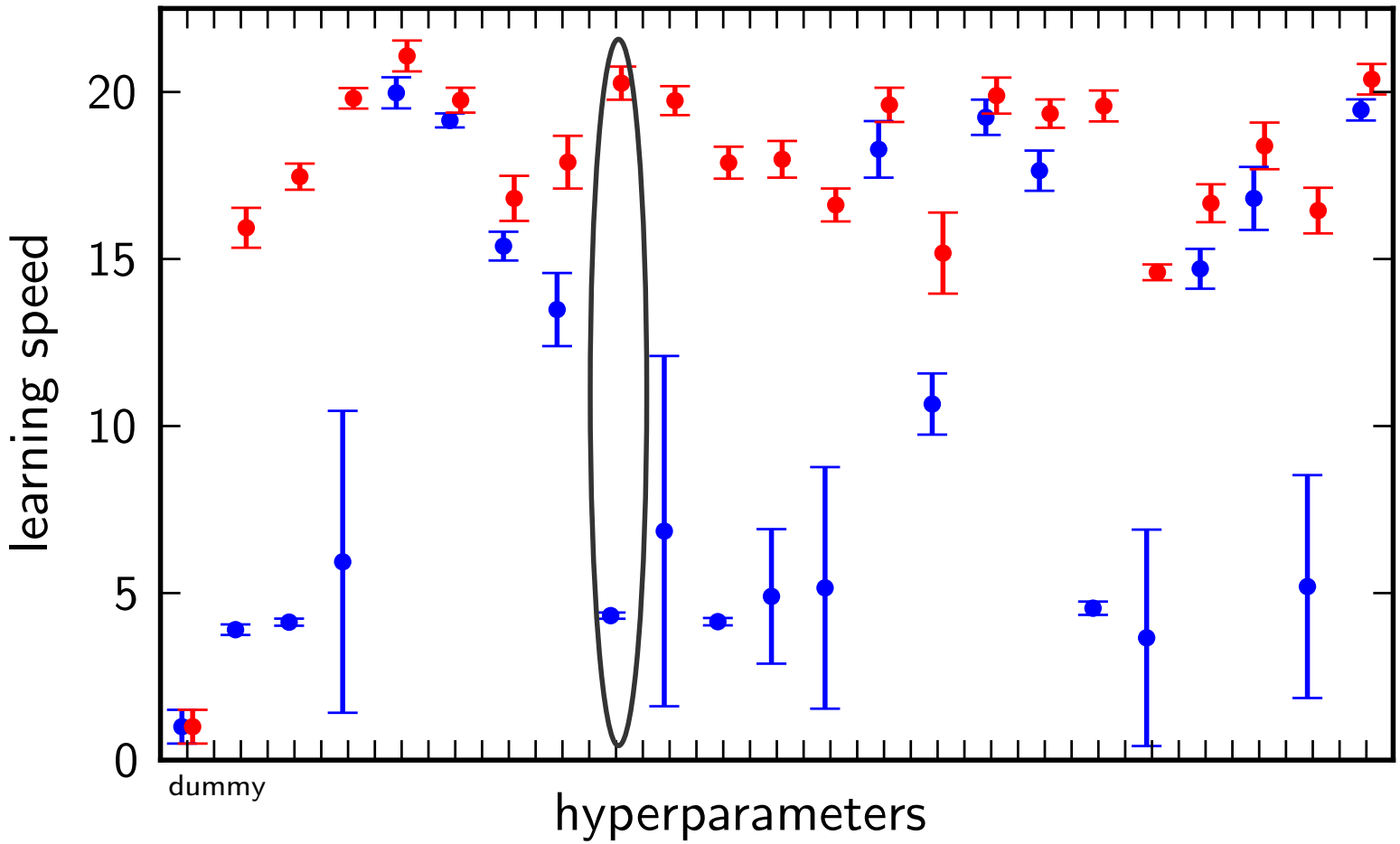
Results so far

ICacla learns well while Cacla doesn't learn much.

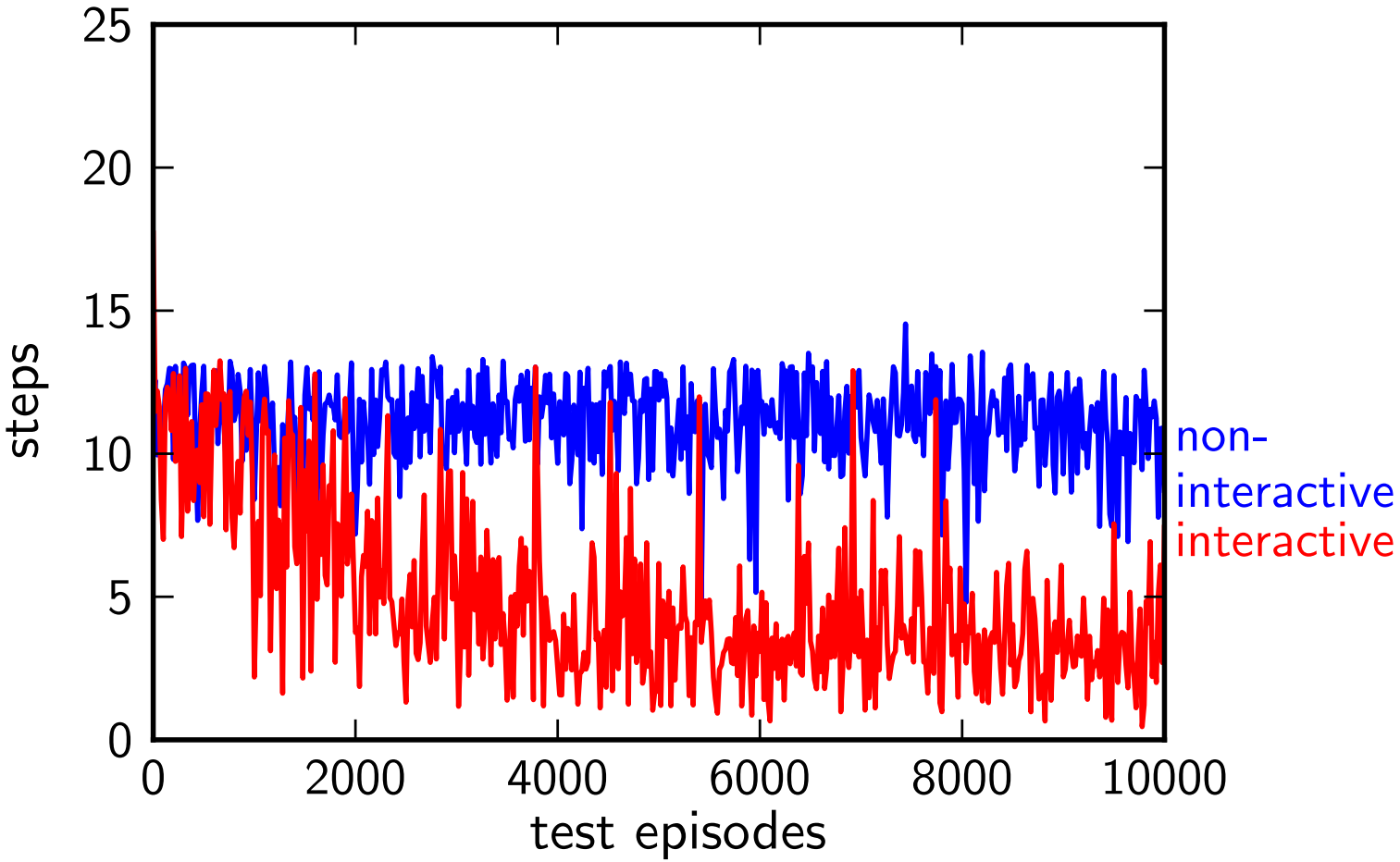
ICacla learns faster than Cacla with the 3-DoF arm



ICacla learns faster than Cacla with the 3-DoF arm



ICacla learns much faster
if the hyperparameters as poorly set



Short reminder of the results so far

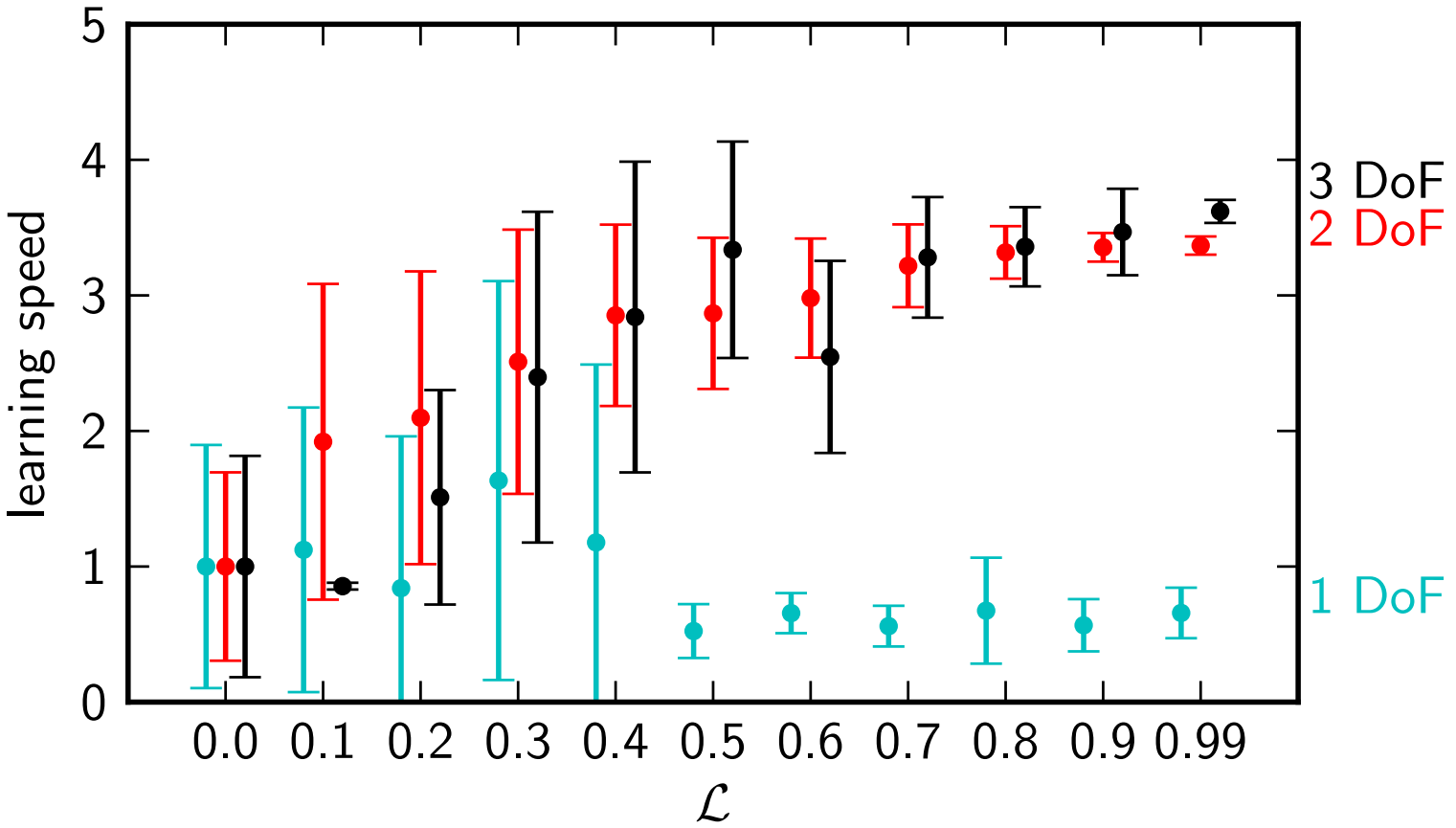
- A small benefit for fully interactive learning in the simplest task
- Only a small difference between fully interactive and non-interactive learning with well chosen hyperparameters
- A substantial advantage for fully interactive learning in both complex tasks if the hyperparameters are poorly chosen

Steps between non-interactive and fully interactive learning

We asked ourself

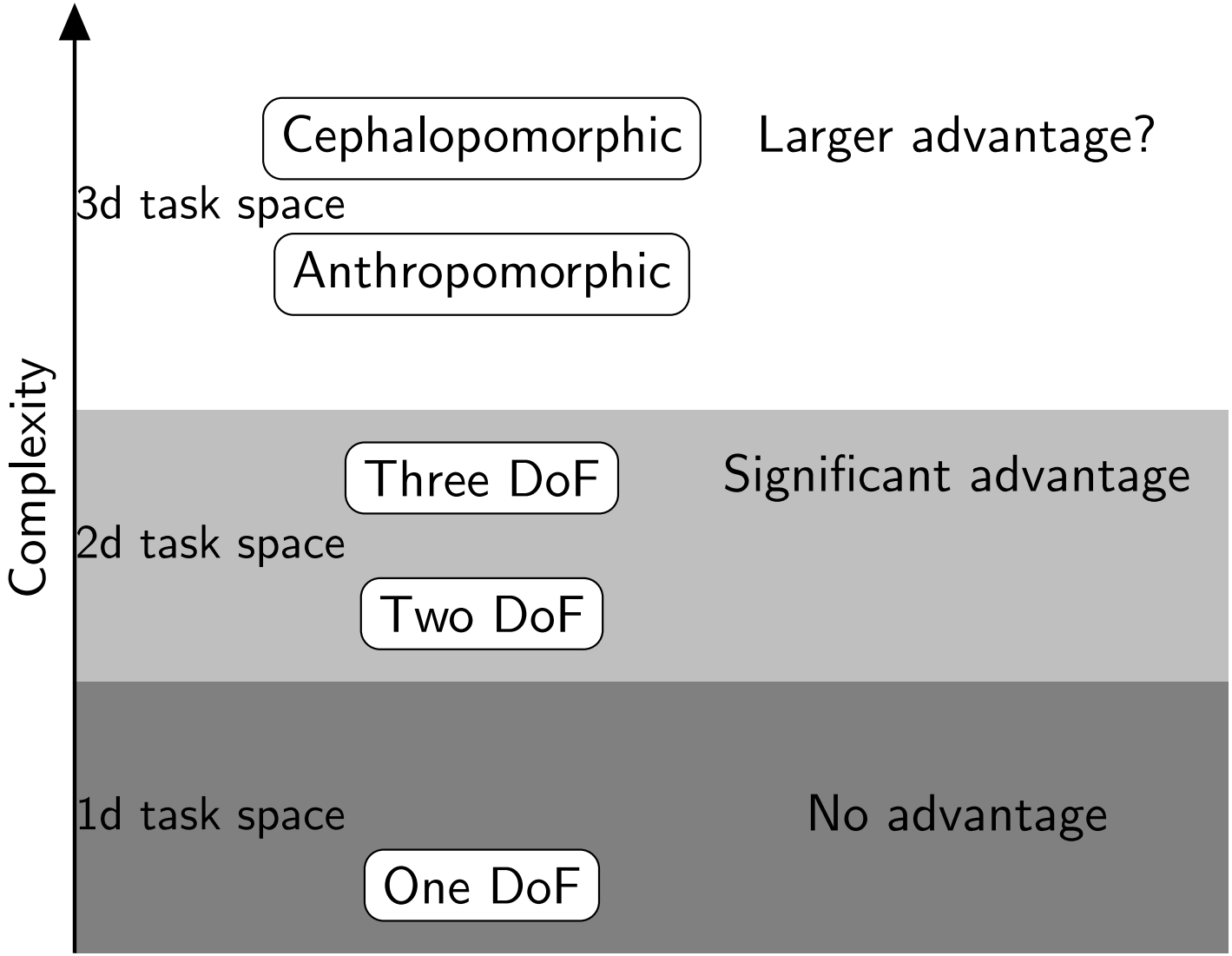
What happens if we take the hyperparameters with the large difference between fully interactive and non-interactive learning and increase the likelihood of asking gradually from 0 to 1?

Interactive learning is only beneficial in complex tasks



Fine, but what do these results mean and what does the future hold?

Interesting directions for the future



Conclusion

- Lower bound for the advantage of interactive learning

Conclusion

- Lower bound for the advantage of interactive learning
- Interactive learning is faster than non-interactive learning in complex problems

Conclusion

- Lower bound for the advantage of interactive learning
- Interactive learning is faster than non-interactive learning in complex problems
- No large difference between the task with 2 and 3-DoF

Conclusion

- Lower bound for the advantage of interactive learning
- Interactive learning is faster than non-interactive learning in complex problems
- No large difference between the task with 2 and 3-DoF
- Interactive learning is more robust against badly chosen hyperparameters

Any questions?

Bibliography

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Adaptive computation and machine learning. A Bradford Book/The MIT Press, Cambridge, MA, USA, first edition, 1998. ISBN 0262193981.

Andrea L. Thomaz and Cynthia Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6-7):716–737, 2008. ISSN 0004-3702. doi: 10.1016/j.artint.2007.09.009.

Hado van Hasselt and Marco A. Wiering. Reinforcement learning in continuous action spaces. In *IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 272–279, Honolulu, HI, USA, 2007. IEEE. ISBN 1-4244-0706-0. doi: 10.1109/ADPRL.2007.368199.

Metrics

Learning speed $\mathcal{LS} = \left|(\tau'_i \,|\tau'_i < 4, \tau'_i \in \mathcal{T}')_{i=0}^{\infty}\right|$

of normalized test runs

$$\mathcal{T}' = \left(\frac{\tau_i}{\lceil d_i \rceil} \,|\tau_i \in \mathcal{T}, d_i \in \mathcal{D}\right)_{i=0}^{\infty}$$

whereas test runs $\mathcal{T}$ and initial distances $\mathcal{D}$