Demonstration-Conditioned Reinforcement Learning for Few-Shot Imitation

Théo Cachet

Julien Perez

Christopher Dance

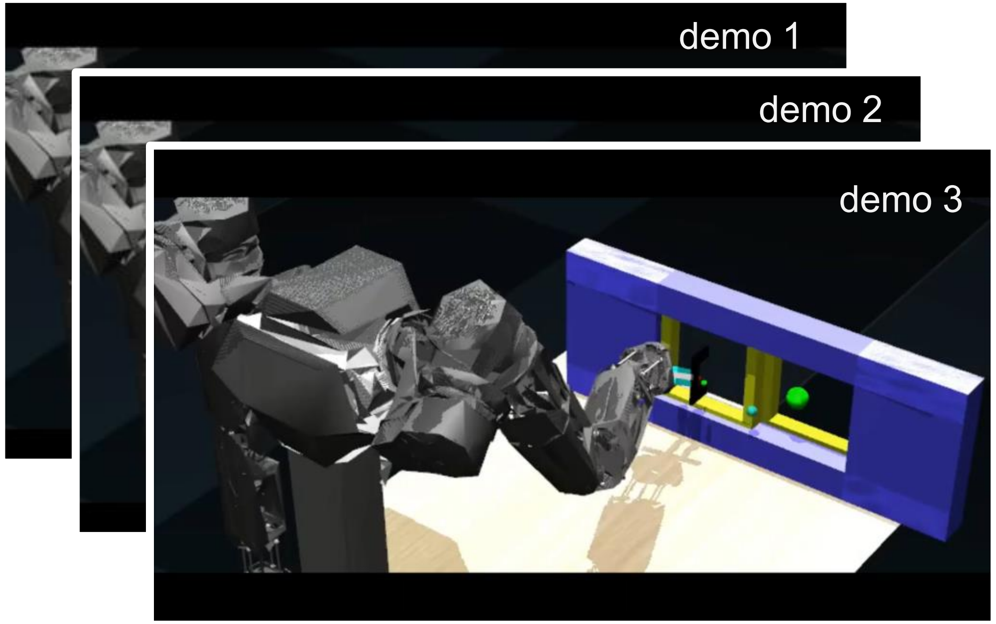
NAVER LABS

Grenoble, France

Given a few demonstrations of a new, previously unseen task → **Find** a policy which performs that task effectively.

Few-Shot Imitation

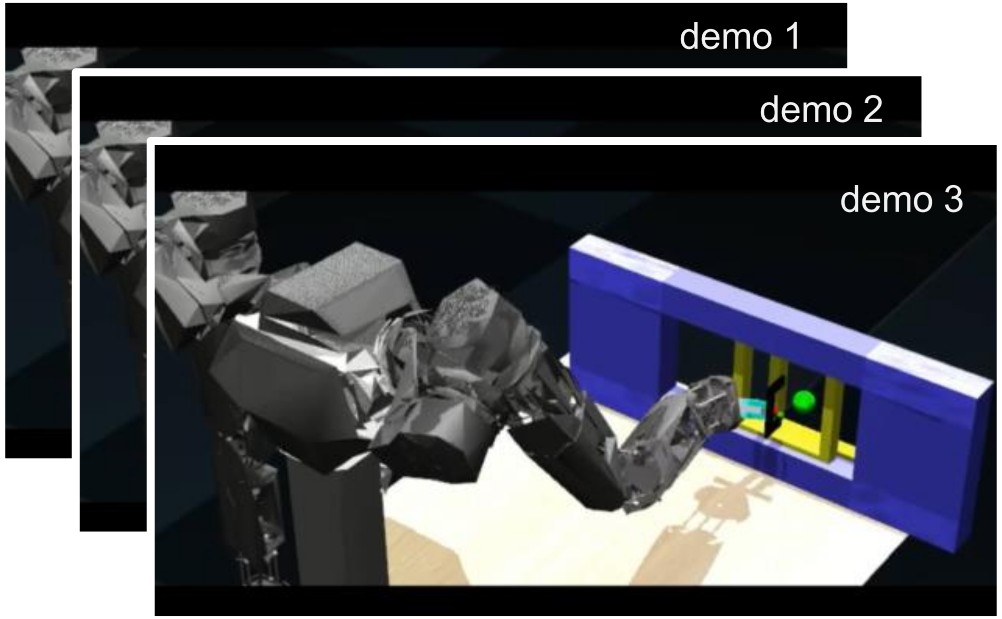
Given a few demonstrations of a new, previously unseen task → **Find** a policy which performs that task effectively.

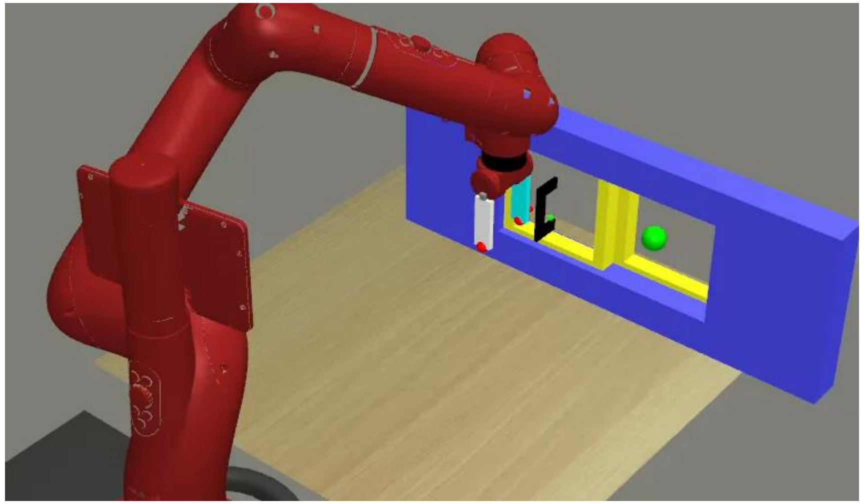


Few-Shot Imitation

Given a few demonstrations of a new, previously unseen task

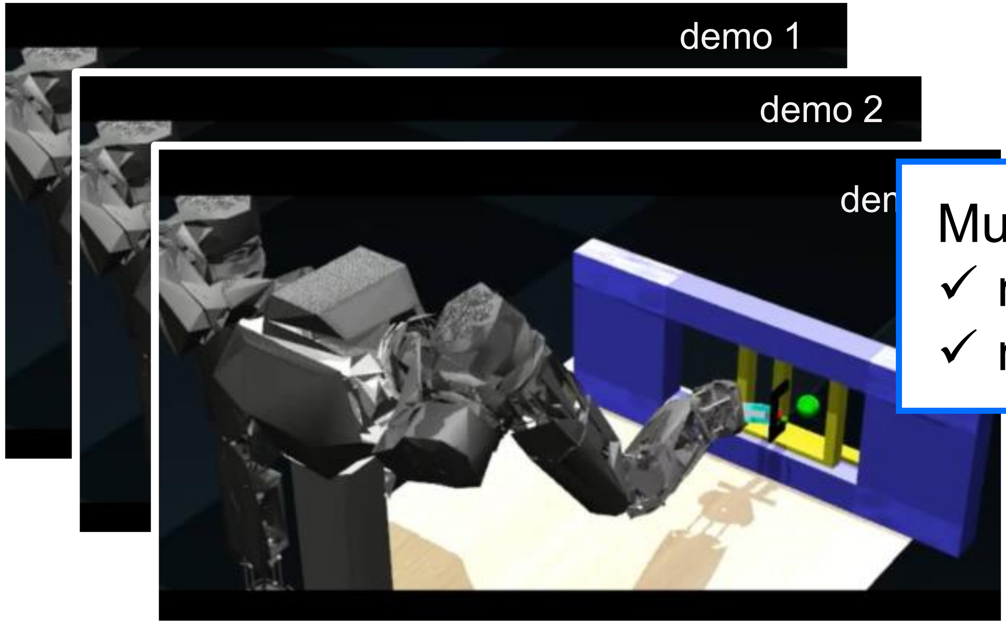
Find a policy which performs that task effectively.



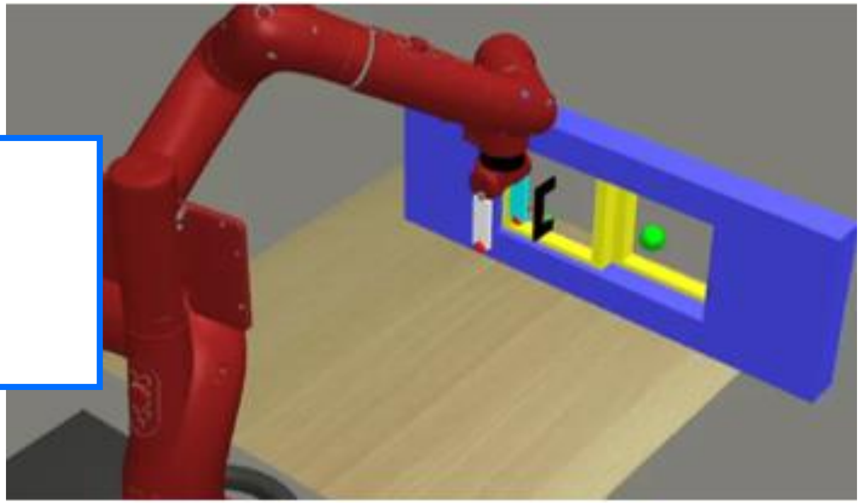


action $\sim \pi(\cdot \mid \text{history}, \text{demonstrations})$

Given a few demonstrations of a new, previously unseen task → **Find** a policy which performs that task effectively.

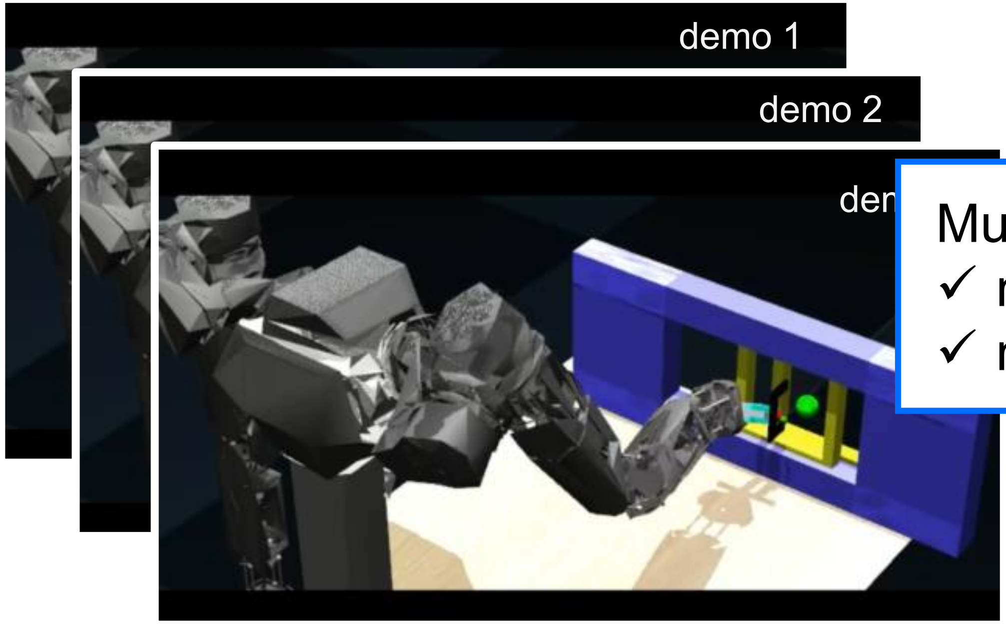


Must ***generalize*** to
- ✓ new tasks
- ✓ new states

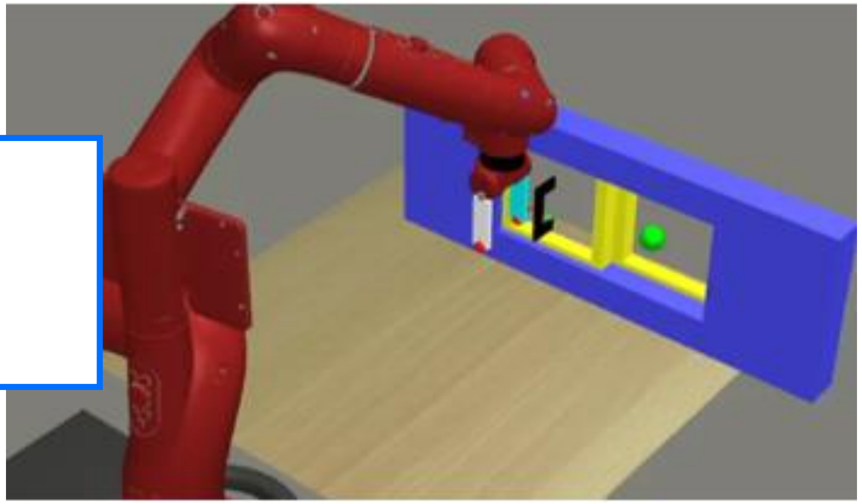


$\text{action} \sim \pi(\cdot\,|\text{history}, \text{demonstrations})$

Given a few demonstrations of a new, previously unseen task → **Find** a policy which performs that task effectively.



Must ***generalize*** to
- ✓ new tasks
- ✓ new states



action $\sim \pi(\cdot\,|\text{history}, \text{demonstrations})$

Demonstration is flexibly defined:
- ✓ noisy, incomplete, sub-optimal
- ✓ no actions
- ✓ human demonstrator + robot agent

Ingredients

η Distribution over tasks $\mu \sim \eta$

Ingredients

η	Distribution over tasks $\mu \sim \eta$
D_μ	Distribution over collections $\mathbf{d}$ of demonstrations of task μ

Few-Shot Imitation

Ingredients

η	Distribution over tasks $\mu \sim \eta$
D_μ	Distribution over collections $\mathbf{d}$ of demonstrations of task μ
$\pi(\cdot \mid h, \mathbf{d})$	*Demonstration-conditioned policy* given history h and demonstrations $\mathbf{d}$

Ingredients

η	Distribution over tasks $\mu \sim \eta$
D_μ	Distribution over collections $\mathbf{d}$ of demonstrations of task μ
$\pi(\cdot \mid h, \mathbf{d})$	*Demonstration-conditioned policy* given history h and demonstrations $\mathbf{d}$

Objective

$$\max_{\pi} \; \mathbb{E}_{\mu\sim\eta} \; \mathbb{E}_{\mathbf{d}\sim D_\mu} \; J_\mu(\pi(\cdot \mid h, \mathbf{d}))$$

demonstration-conditioned policy

tasks

collections of demonstrations

return

Each task μ is an MDP
$J_\mu(\pi)$ is the return for policy π

Ingredients

η Distribution over tasks $\mu \sim \eta$

D_μ Distribution over collections $\mathbf{d}$ of demonstrations of task μ

$\pi(\cdot\,|h,\mathbf{d})$ *Demonstration-conditioned policy* given history h and demonstrations $\mathbf{d}$

Objective

$$\max_{\pi} \; \mathbb{E}_{\mu\sim\eta} \; \mathbb{E}_{\mathbf{d}\sim D_\mu} \; J_\mu(\pi(\cdot\,|h,\mathbf{d}))$$

demonstration-conditioned policy

tasks

collections of demonstrations

return

Few-Shot Imitation

Each task μ is an MDP
$J_\mu(\pi)$ is the return for policy π

Ingredients

η Distribution over tasks $\mu \sim \eta$

D_μ Distribution over collections $\mathbf{d}$ of demonstrations of task μ

few ↔ 1 to 10

$\pi(\cdot \mid h, \mathbf{d})$ *Demonstration-conditioned policy* given history h and demonstrations $\mathbf{d}$

Objective

$$\max_{\pi} \ \mathbb{E}_{\mu\sim\eta} \ \mathbb{E}_{\mathbf{d}\sim D_\mu} \ J_\mu(\pi(\cdot \mid h, \mathbf{d}))$$

demonstration-conditioned policy

tasks

collections of demonstrations

return

Ingredients

Each task μ is an MDP
$J_\mu(\pi)$ is the return for policy π

η	Distribution over tasks $\mu \sim \eta$
D_μ	Distribution over collections $\mathbf{d}$ of demonstrations of task μ
$\pi(\cdot \mid h, \mathbf{d})$	*Demonstration-conditioned policy* given history h and demonstrations $\mathbf{d}$

few $\leftrightarrow$ 1 to 10

POMDPs

Objective

$$\max_{\pi} \; \mathbb{E}_{\mu \sim \eta} \; \mathbb{E}_{\mathbf{d} \sim D_\mu} \; J_\mu(\pi(\cdot \mid h, \mathbf{d}))$$

demonstration-conditioned policy

tasks

collections of demonstrations

return

Demonstration-conditioned reinforcement learning (DCRL)

NAVER LABS

DCRL maximizes the return of a demonstration-conditioned policy, averaged over a set of training tasks and corresponding demonstrations.

$\mu^0, \ldots, \mu^{N-1}$ *may not be distinct*

Train

Input Pairs $(\mathbf{d}^0, \mu^0), \ldots, (\mathbf{d}^{N-1}, \mu^{N-1})$ where $\mathbf{d}^i$ is a collection of demonstrations of task μ^i

Output A demonstration−conditioned policy $\boldsymbol{\pi}$ attaining $\max_{\pi} \sum_{i=0}^{N-1} J_{\mu^i}(\boldsymbol{\pi}(\cdot \mid \cdot, \mathbf{d}^i))$

Test

Input
- $\mathbf{d}$ Collection of demonstrations of new, previously unseen task
- $\boldsymbol{\pi}$ Demonstration−conditioned policy given by DCRL

Repeat Observe history h_t and take action $a_t \sim \boldsymbol{\pi}(\cdot \mid h_t, \mathbf{d})$

No need for reward function
No need for to explore the test env't

Behaviour cloning (BC)

Duan *et al.* '17

$$\min_{\pi} \sum_{i=0}^{N-1} \sum_{(s,a)\in \mathbf{d}^i} \ell\left(a, \pi(\cdot \mid s, \mathbf{d}^i)\right)$$

action-prediction loss

Behaviour cloning (BC)
Duan *et al.* ‘17

$$\min_{\boldsymbol{\pi}} \sum_{\boldsymbol{i}=\mathbf{0}}^{\boldsymbol{N}-\mathbf{1}} \sum_{(s,a)\in\mathbf{d}^i} \ell\left(a, \boldsymbol{\pi}(\cdot\mid s, \mathbf{d}^i)\right)$$

Meta-inverse RL
Yu *et al.* ‘19a, Goo and Niekum ‘19

Infer reward $\widehat{\boldsymbol{R}}_{\mu}$ from $\mathbf{d}$

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t} \gamma^{t}\, \widehat{\boldsymbol{R}}_{\mu}(s_t, a_t)\right]$$

NAVER LABS

Behaviour cloning (BC)
Duan *et al.* ‘17

$$\min_{\pi} \sum_{i=0}^{N-1} \sum_{(s,a)\in \mathbf{d}^i} \ell\left(a, \boldsymbol{\pi}(\cdot \mid s, \mathbf{d}^i)\right)$$

Meta-inverse RL
Yu *et al.* ‘19a, Goo and Niekum ‘19

Infer reward $\hat{\boldsymbol{R}}_{\mu}$ from $\mathbf{d}$

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t} \gamma^{t}\, \hat{\boldsymbol{R}}_{\mu}(s_t, a_t)\right]$$

Demonstration-conditioned RL
This paper

$$\max_{\pi} \sum_{i=0}^{N-1} \boldsymbol{J}_{\mu^i}\left(\boldsymbol{\pi}(\cdot \mid \cdot, \mathbf{d}^i)\right)$$

Behaviour cloning (BC)

Duan *et al.* '17

$$\min_{\pi} \sum_{i=0}^{N-1} \sum_{(s,a)\in \mathbf{d}^i} \ell(a, \boldsymbol{\pi}(\cdot \mid s, \mathbf{d}^i))$$

Needs actions in demo's

No interaction with the test env't

Compounding errors ⇒ loss is $O(H^2)$ on horizon H (Rajaraman *et al.*, 2020)

Meta-inverse RL

Yu *et al.* '19a, Goo and Niekum '19

Infer reward $\widehat{\boldsymbol{R}}_\mu$ from $\mathbf{d}$

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_t \gamma^t \widehat{\boldsymbol{R}}_\mu(s_t, a_t)\right]$$

No need for actions in demo's

Must interact with the test env't

Demonstration-conditioned RL

This paper

$$\max_{\pi} \sum_{i=0}^{N-1} \boldsymbol{J}_{\mu^i}(\boldsymbol{\pi}(\cdot \mid \cdot, \mathbf{d}^i))$$

No need for actions in demo's

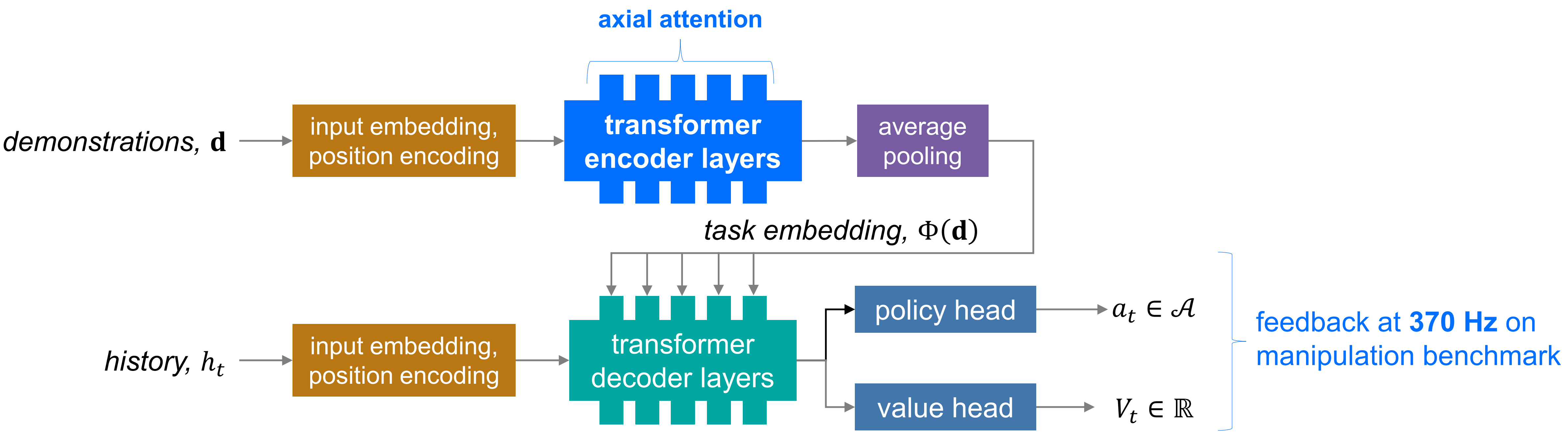
No interaction with the test env't

Improves on suboptimal demo's

Copes with demonstrator domain shift

Needs reward function for training

Transformer-based policy architecture



Literature

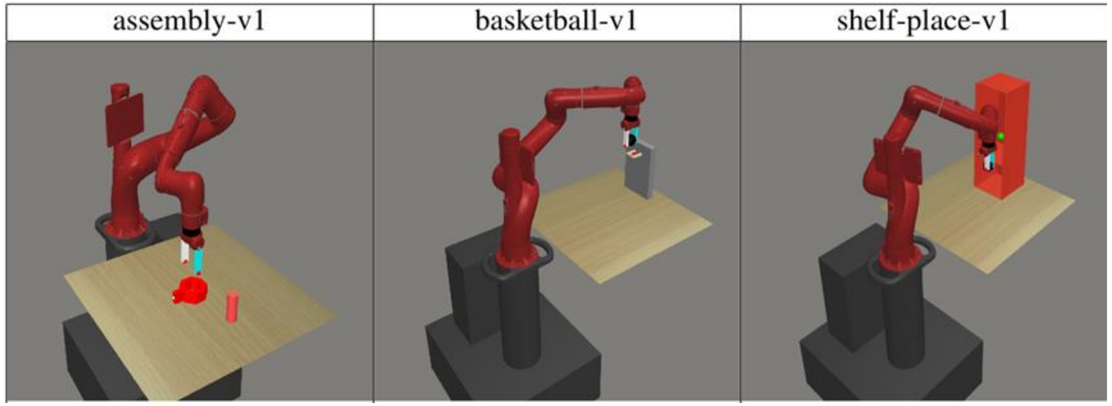
Attention and transformers already in use for few-shot imitation (Duan *et al.* '17, Mishra '18, James '18, Dasari '20)

Novelty

1. **Cross-demonstration attention.** Process multiple demonstrations *jointly*.
2. **Axial attention.** Attend to one dimension of the input at a time (Ho *et al.* '18).
   Reduces time and memory from $O(T^2n^2)$ to $O\big(Tn(T+n)\big)$ for n time series of length T

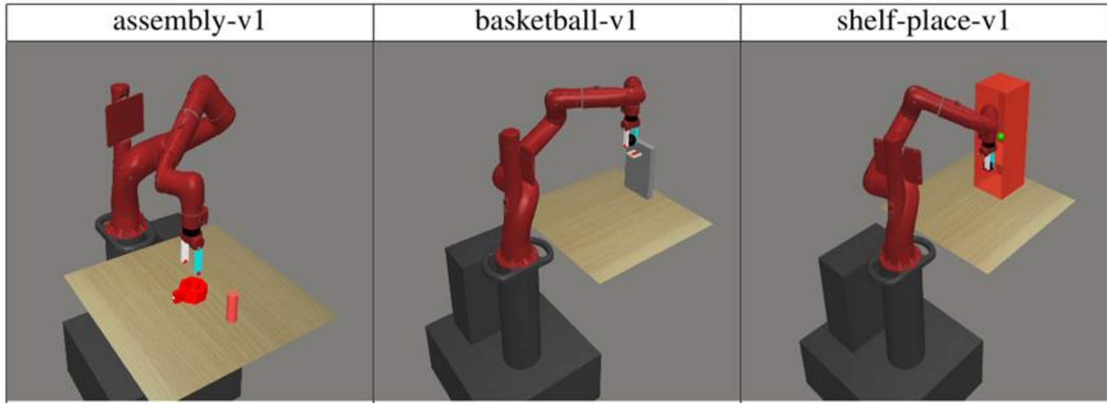
1. Meta-World Benchmark (Yu et al., ’19)

50 manipulation tasks
e.g. open-window, lock-door



1. Meta-World Benchmark (Yu et al., ’19)

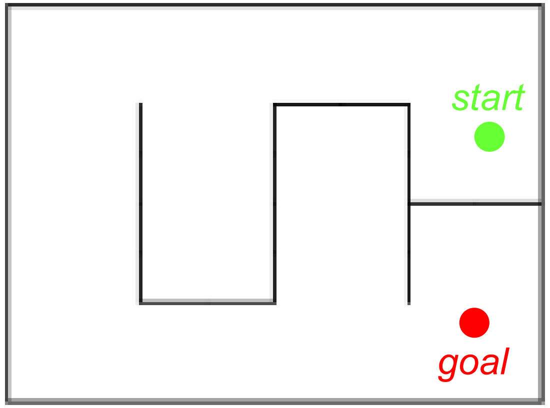
50 manipulation tasks
e.g. open-window, lock-door



2. Navigation Benchmark

60 mazes ↔ tasks

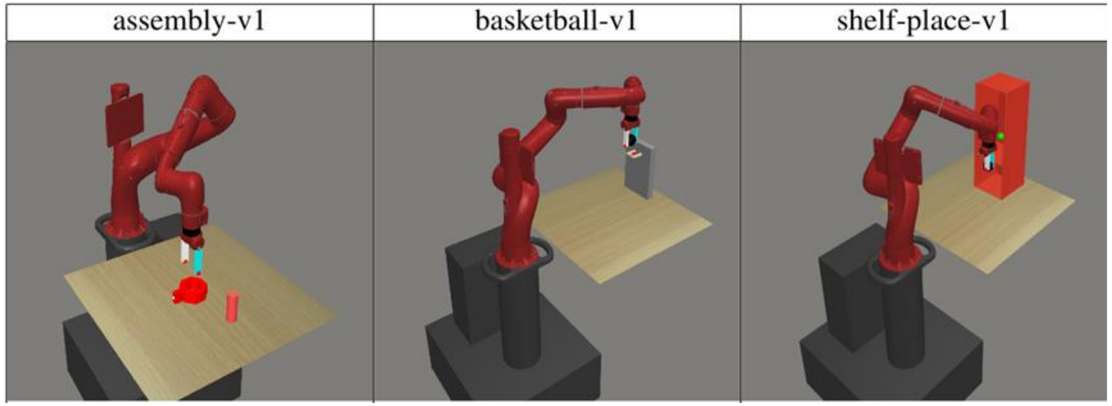
map of test maze



Aim
Get from start to goal state (randomized).

1. Meta-World Benchmark (Yu et al., ’19)

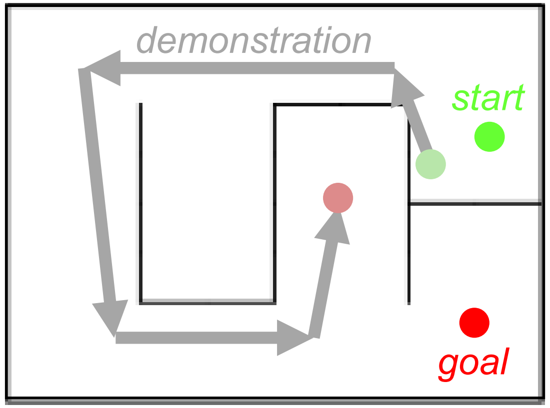
50 manipulation tasks
e.g. open-window, lock-door



2. Navigation Benchmark

60 mazes ↔ tasks

what the agent sees



Aim
Get from start to goal state (randomized).
Agent can’t see the walls, but is penalized if it hits a wall.
So, it must infer the walls from the demonstrations.

Evaluation. All experiments we will now present had distinct training and test tasks.

DCRL consistently beats behaviour-cloning methods

Definition (DCBC). *Demonstration-conditioned behavioural cloning (DCBC)* has the same architecture as DCRL, but is trained with a behaviour-cloning loss.

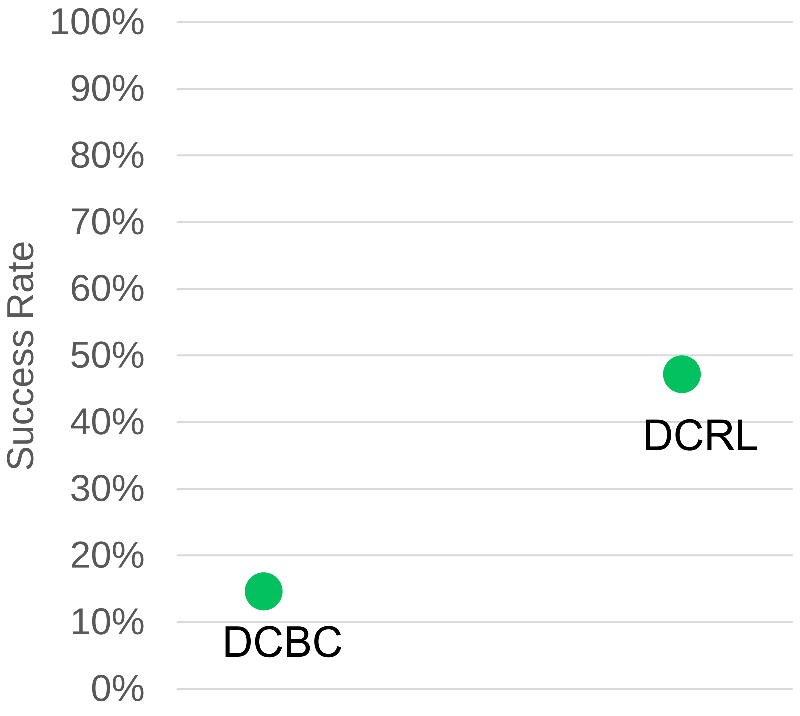
like Duan et al. (2016)

DCRL consistently beats behaviour-cloning methods

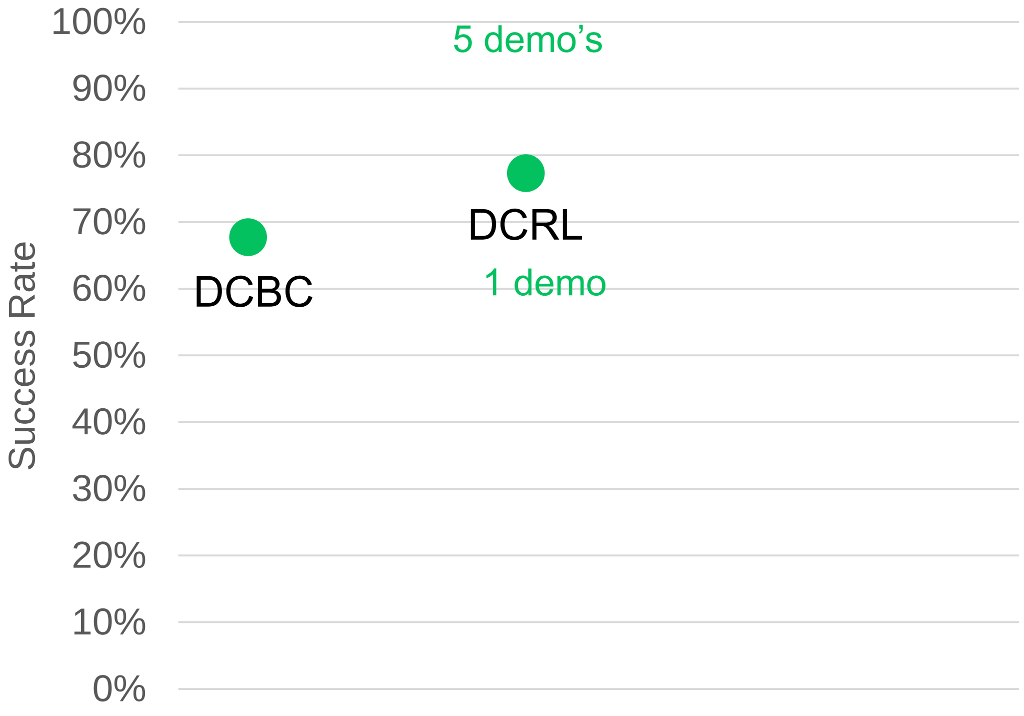
Definition (DCBC). *Demonstration-conditioned behavioural cloning (DCBC)* has the same architecture as DCRL, but is trained with a behaviour-cloning loss.

like Duan et al. (2016)

Meta-World



Navigation

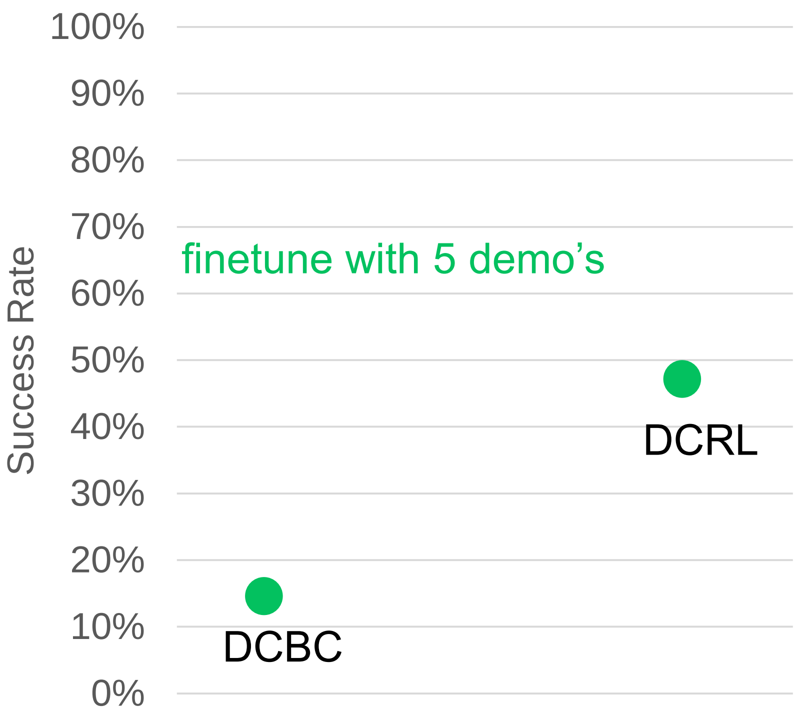


DCRL consistently beats behaviour-cloning methods

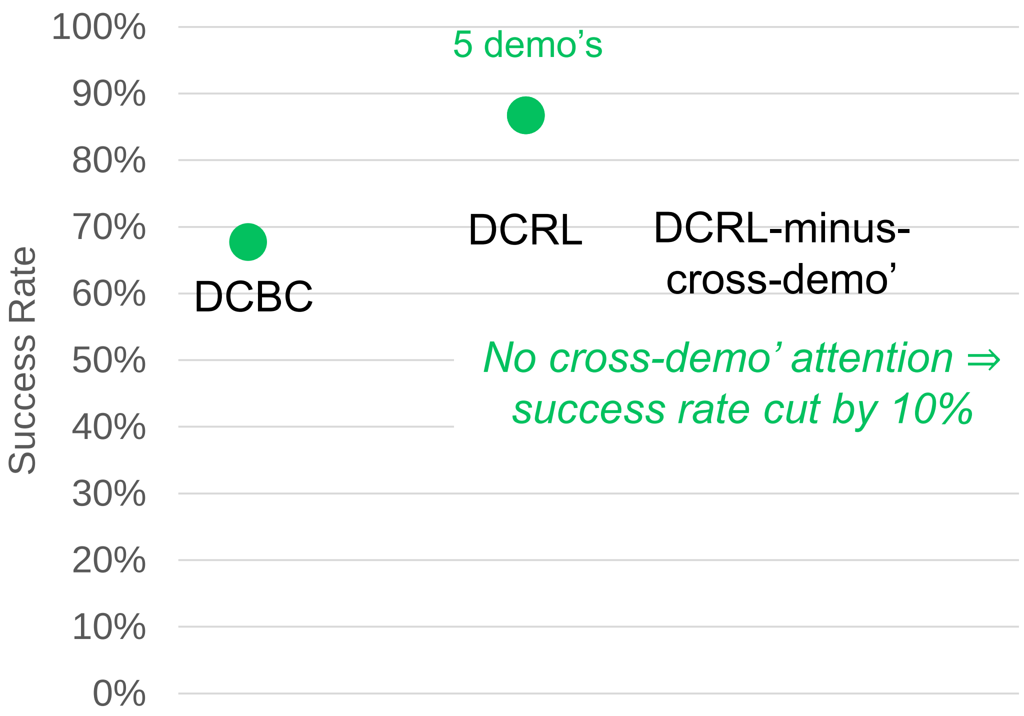
Definition (DCBC). *Demonstration-conditioned behavioural cloning (DCBC)* has the same architecture as DCRL, but is trained with a behaviour-cloning loss.

like Duan et al. (2016)

Meta-World



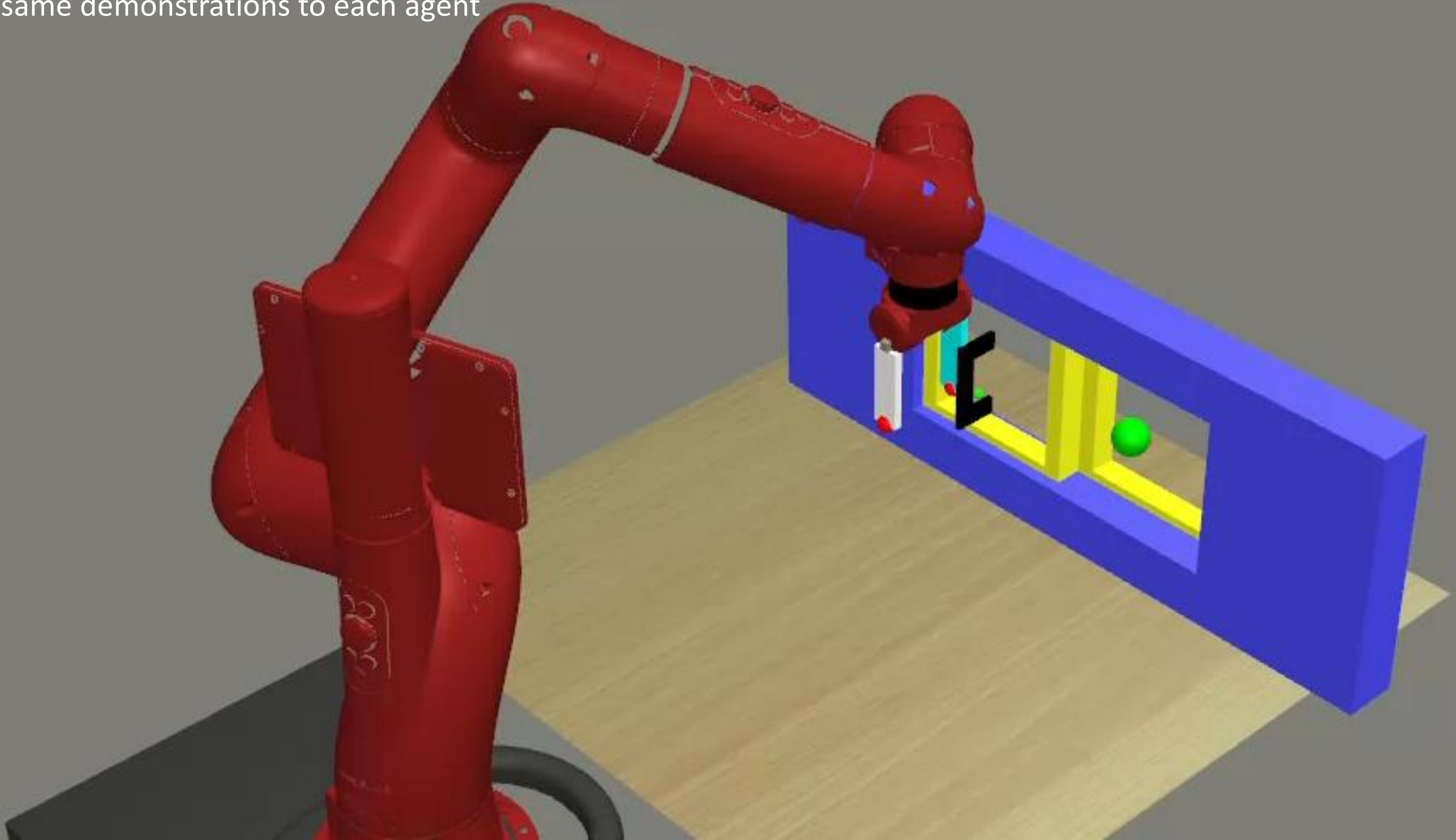
Navigation



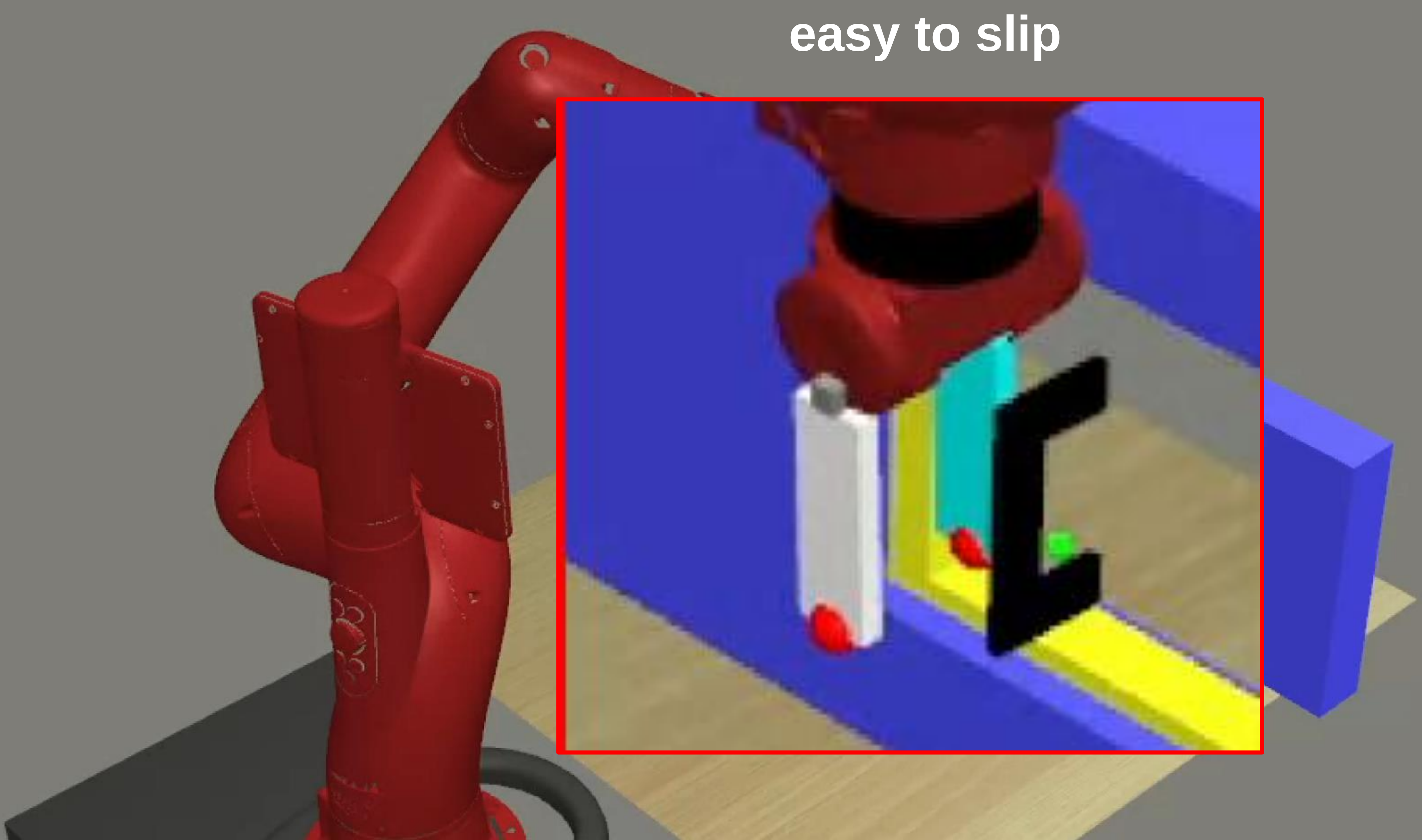
Why does DCLR outperform DCBC?

DEMONSTRATIONS

Present the same demonstrations to each agent

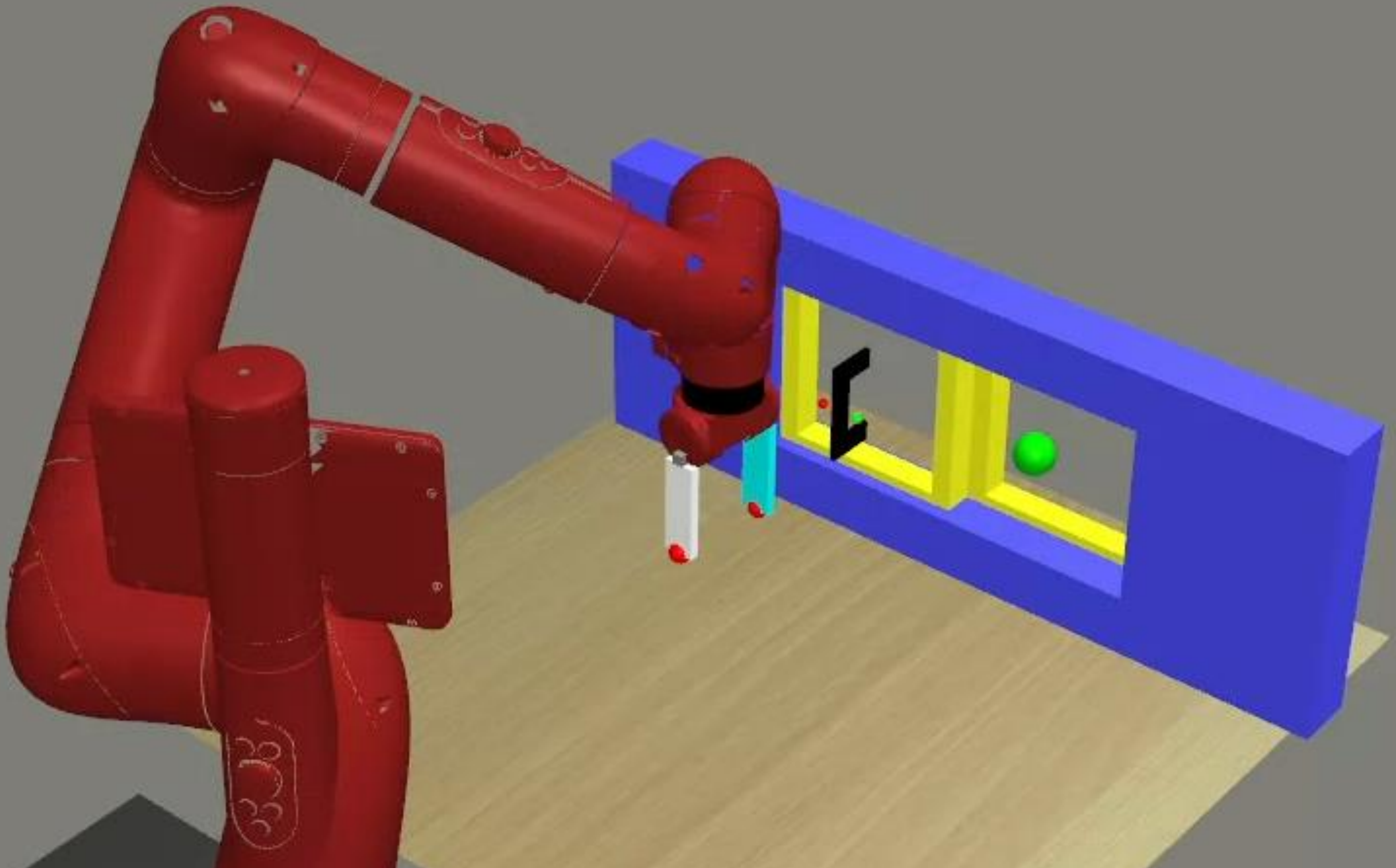


Why does DCLR outperform DCBC?

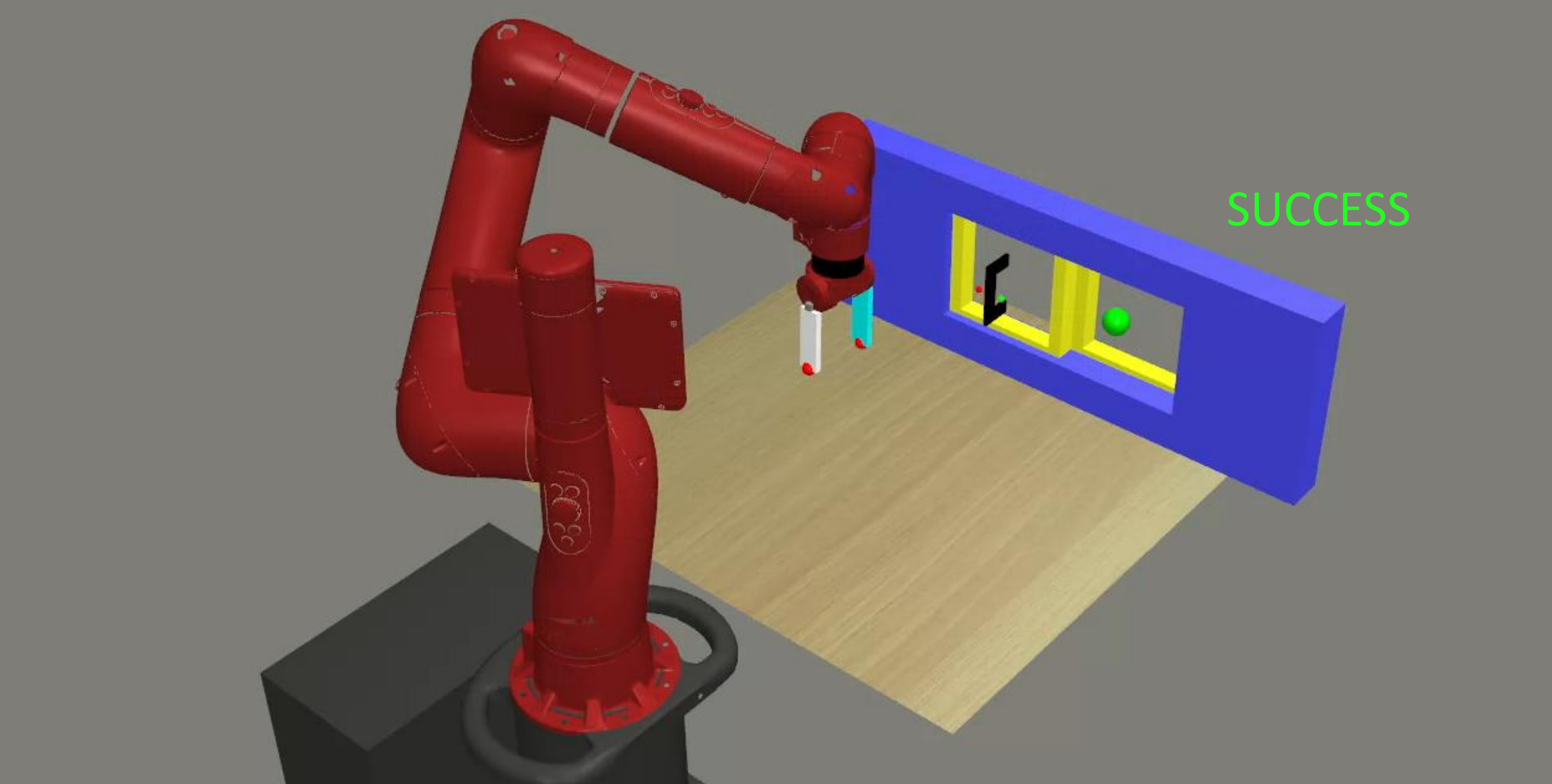


Why does DCLR outperform DCBC?

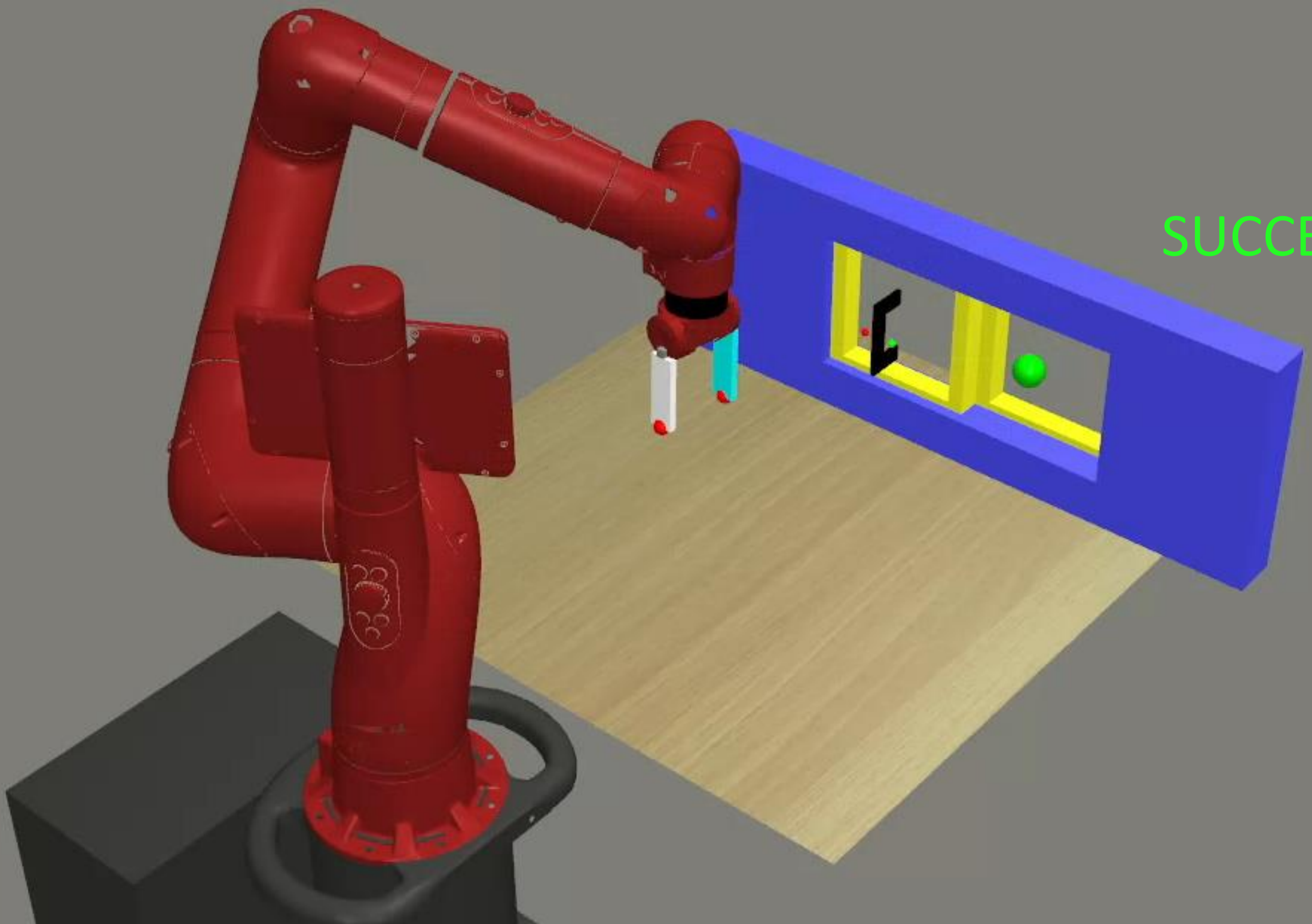
DCBC FAILURE



Why does DCLR outperform DCBC?



Why does DCLR outperform DCBC?

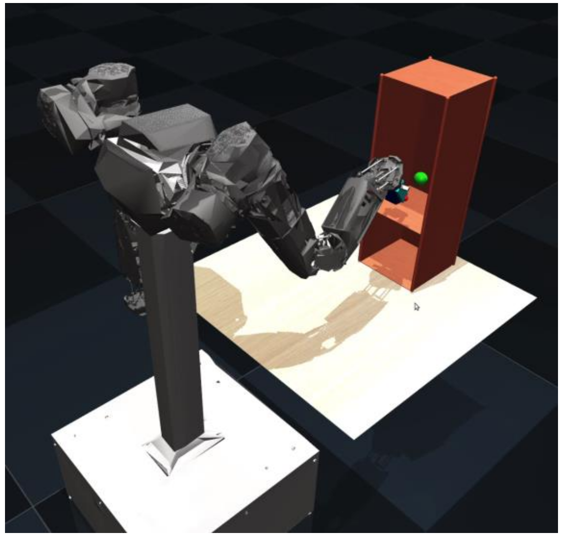


DCRL addresses demonstrator domain shift

Motivation. Control a robot given human demonstrations.

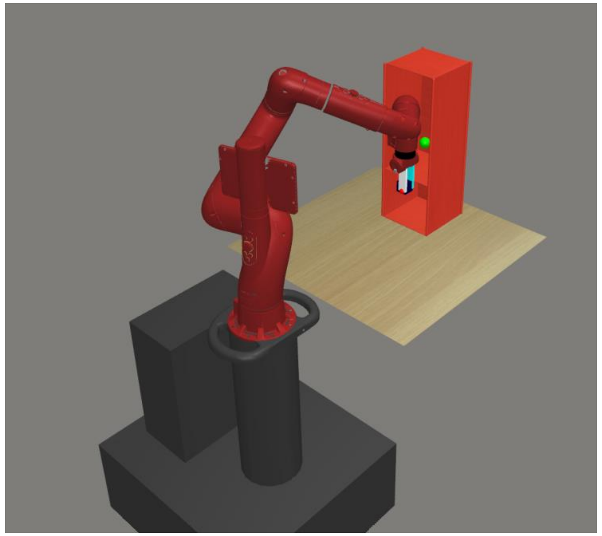
Experiment. Control Sawyer robot given demo's from an AMBIDEX robot.

AMBIDEX demo's



→ DCRL agent →

Sawyer robot doing shelf-place task



Results.

Demo's from	# demo's	1	5	1	5
	Sawyer	**51%**	**51%**	**316**	323
	AMBIDEX	45%	48%	308	**329**
		success rates		average returns	

DCRL improves on suboptimal demonstrations

Motivation. Demonstrations are often suboptimal

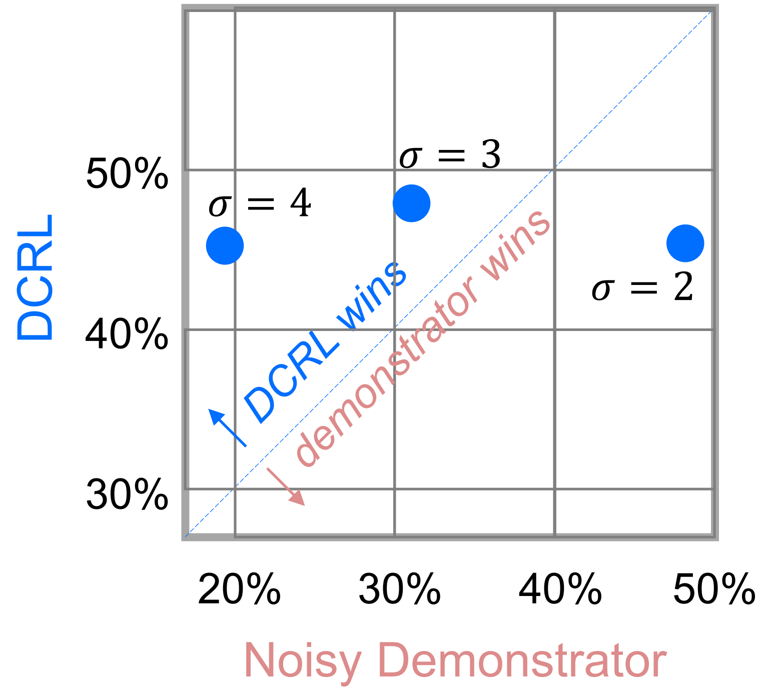
Experiment. Add noise $\sim \mathcal{N}(0, \sigma^2 I_{4\times4})$ to demonstrator actions (only at test)

DCRL improves on suboptimal demonstrations

Motivation. Demonstrations are often suboptimal

Experiment. Add noise $\sim \mathcal{N}(0, \sigma^2 I_{4\times4})$ to demonstrator actions (only at test)

Success Rates on Meta-World



For $\sigma > 2$ DCRL outperforms the noisy demonstrator

Conclusion

DCRL is a new, third family of approaches to few-shot imitation

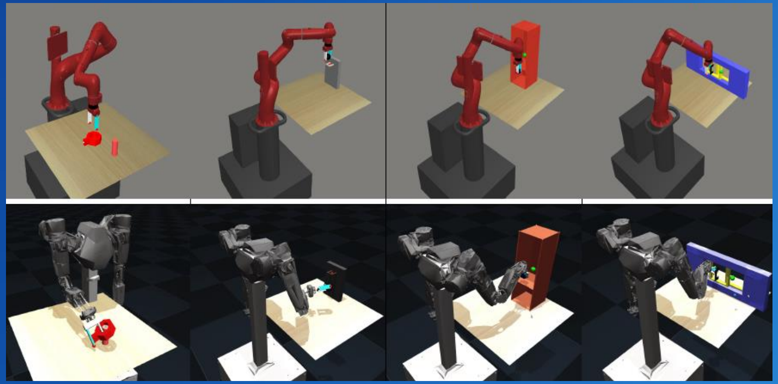
$\{\text{ IRL, BC }\} \cup \{\text{ DCRL }\}$

Pros	Cons	Future
+ suboptimal demo's	- needs training rewards	☼ more training tasks
+ demonstrator domain shift		☼ better actor-critic training
+ no test-time exploration		☼ video of humans + real robot

Thank you

See the full paper for more experiments, t-SNE plots of the task embeddings, …

See europe.naverlabs.com for videos of task execution



NAVER LABS