# A State-Distribution Matching Approach to Non-Episodic Reinforcement Learning

Archit Sharma*, Rehaan Ahmad*, Chelsea Finn

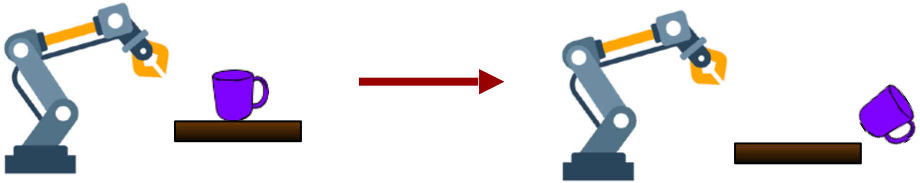
# The Continual Real World

# The Continual Real World

“Navigate to the basketball court”

# The Continual Real World

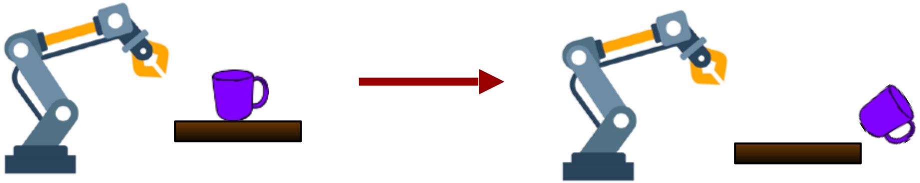
“Navigate to the basketball court”



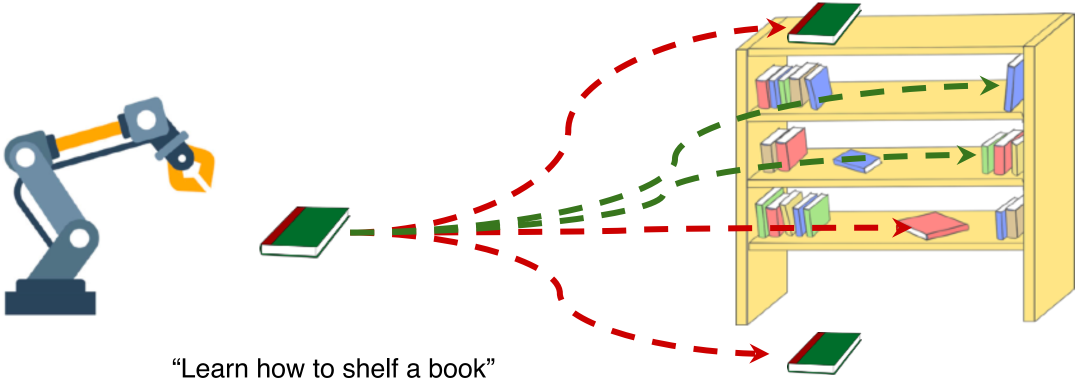
“Grasp the mug”

# The Continual Real World

“Navigate to the basketball court”



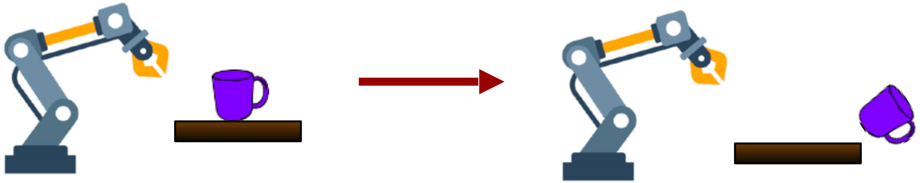
“Grasp the mug”



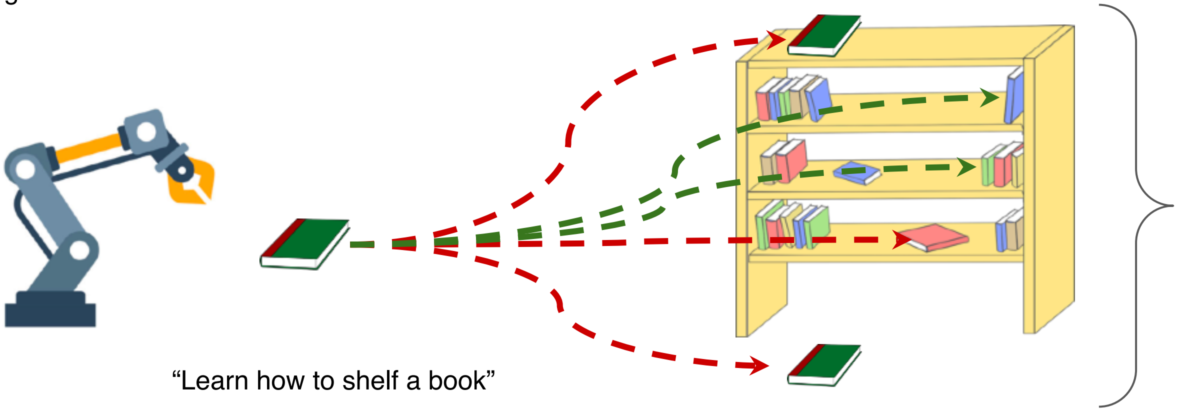
“Learn how to shelf a book”

# The Continual Real World

“Navigate to the basketball court”



“Grasp the mug”



“Learn how to shelf a book”

Several thousands of trials!

# Standard Reinforcement Learning

$$s_0, a_0, s_1, a_1 \ldots s_H$$

# Standard Reinforcement Learning

$$s_0, a_0, s_1, a_1 \ldots s_H$$

$$s'_0, a'_0, s'_1, a'_1 \ldots$$

# Standard Reinforcement Learning

$$s_0, a_0, s_1, a_1 \dots s_H$$

$$s'_0, a'_0, s'_1, a'_1 \dots$$

*How does this happen?*

# Standard Reinforcement Learning

$$s_0, a_0, s_1, a_1 \dots s_H$$

$$s'_0, a'_0, s'_1, a'_1 \dots$$

*How does this happen?*

```
import gym
env = gym.make("CartPole-v1")
observation = env.reset()
for _ in range(1000):
  env.render()
  action = env.action_space.sample() # y
  observation, reward, done, info = env.

  if done:
    observation = env.reset()
env.close()
```

[Code snippet from https://gym.openai.com/]

# Standard Reinforcement Learning

$$s_0, a_0, s_1, a_1 \ldots s_H$$

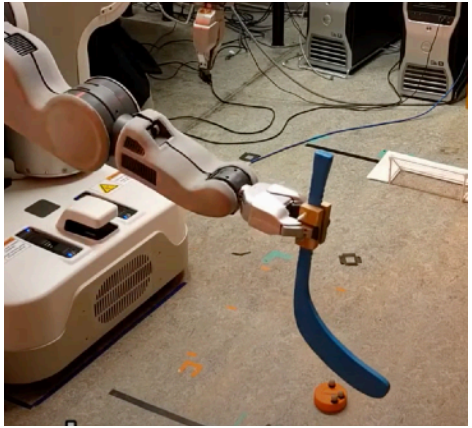
$$s'_0, a'_0, s'_1, a'_1 \ldots$$

*How does this happen?*

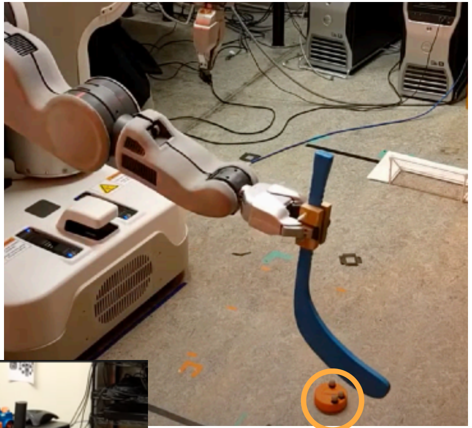
```
import gym
env = gym.make("CartPole-v1")
observation = env.reset()
for _ in range(1000):
  env.render()
  action = env.action_space.sample() # y
  observation, reward, done, info = env.

  if done:
    observation = env.reset()
env.close()
```

[Code snippet from https://gym.openai.com/]

**Only in simulation!**

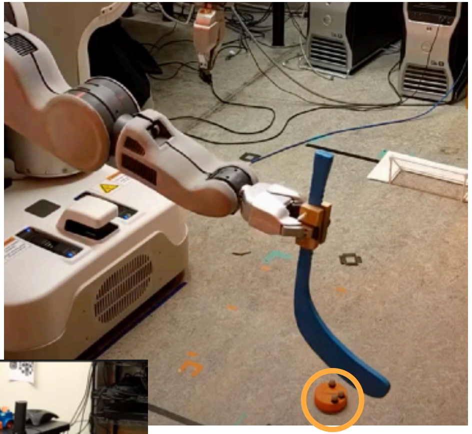


[Combining model-based and model-free updates
for trajectory-centric reinforcement learning,
Chebotar et al. 2017]



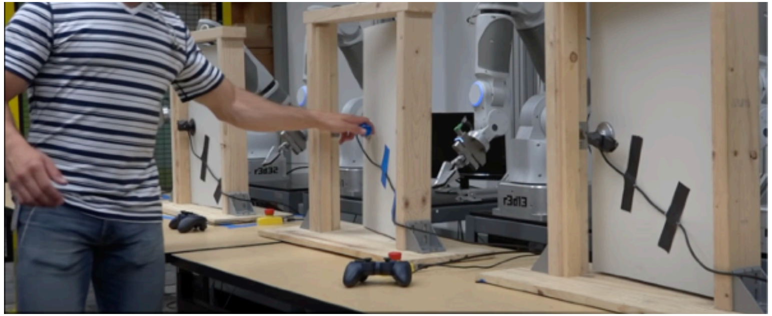

Human resets the puck
before every trial :(

[Combining model-based and model-free updates
for trajectory-centric reinforcement learning,
Chebotar et al. 2017]



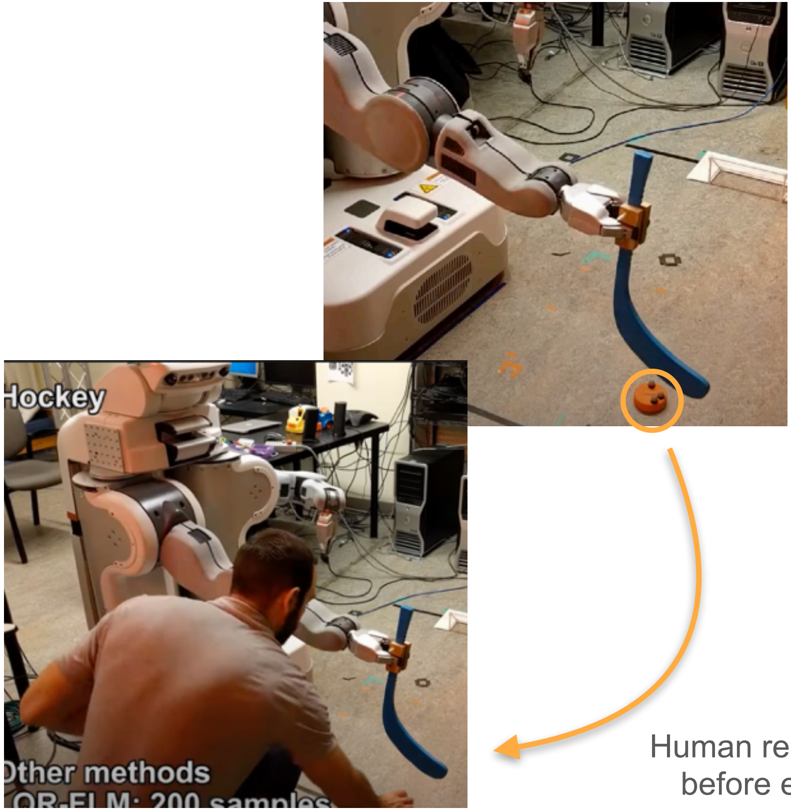

Human resets the puck
before every trial :(

[Combining model-based and model-free updates
for trajectory-centric reinforcement learning,
Chebotar et al. 2017]

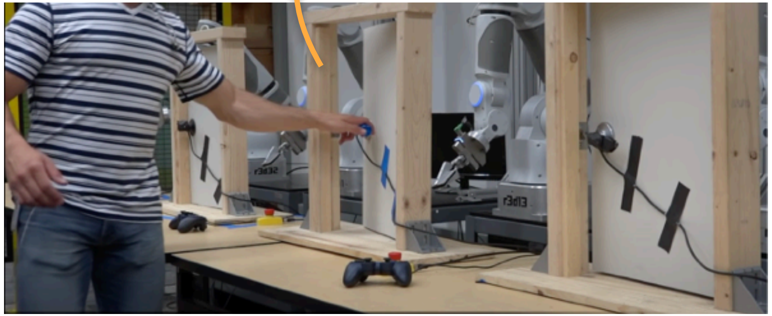


[Collective Robot Reinforcement Learning with Distributed
Asynchronous Guided Policy Search, Yahya et al. 2016]

Human closes the door
before every trial :(




Human resets the puck
before every trial :(



[Collective Robot Reinforcement Learning with Distributed Asynchronous Guided Policy Search, Yahya et al. 2016]

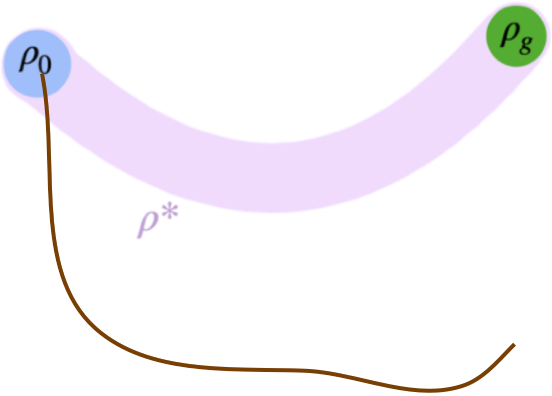
[Combining model-based and model-free updates for trajectory-centric reinforcement learning, Chebotar et al. 2017]

# Challenge of Non-Episodic Learning

Episodic Learning

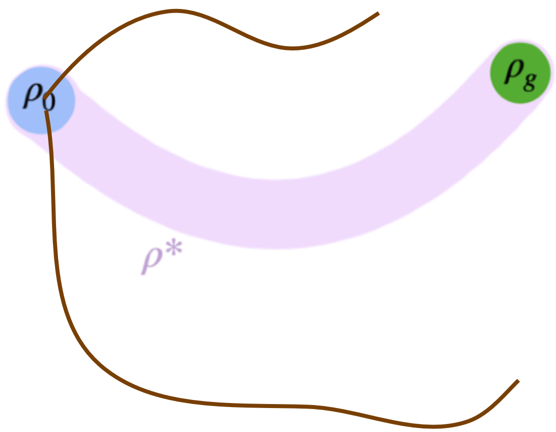
# Challenge of Non-Episodic Learning

Episodic Learning



# Challenge of Non-Episodic Learning

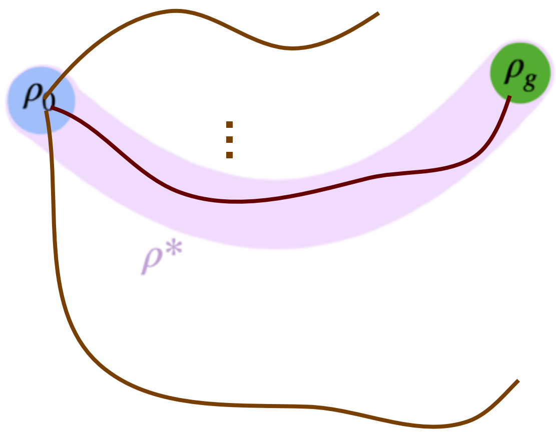
Episodic Learning




Can always retry
the task from initial
state distribution

# Challenge of Non-Episodic Learning

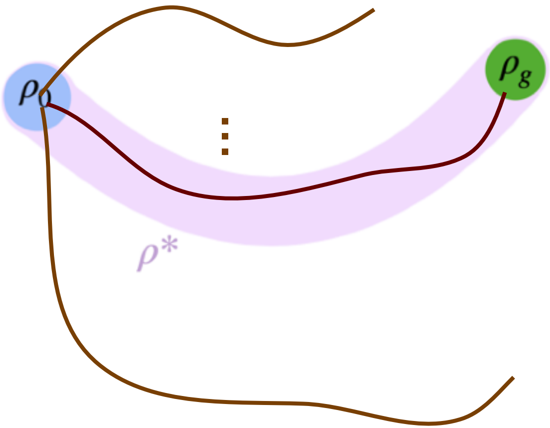
Episodic Learning




Can always retry
the task from initial
state distribution

# Challenge of Non-Episodic Learning

Episodic Learning

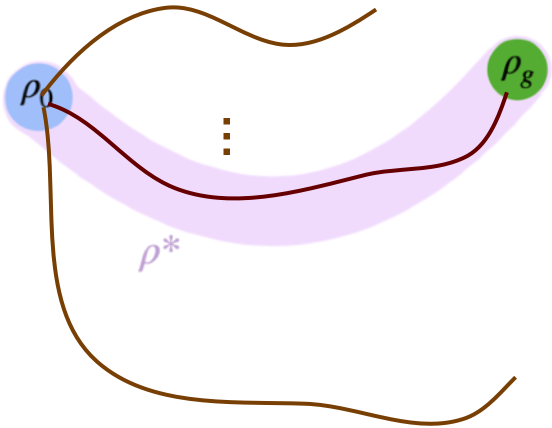



Non-Episodic Learning

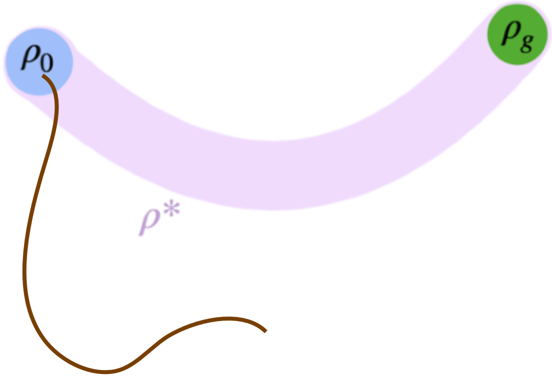

Can always retry
the task from initial
state distribution

# Challenge of Non-Episodic Learning

Episodic Learning



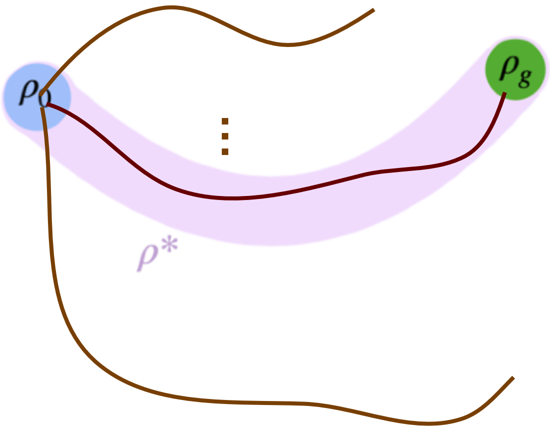

Non-Episodic Learning




Can always retry
the task from initial
state distribution

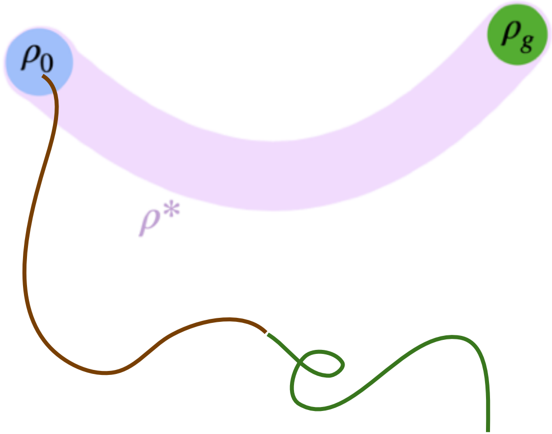
# Challenge of Non-Episodic Learning

Episodic Learning




Can always retry the task from initial state distribution

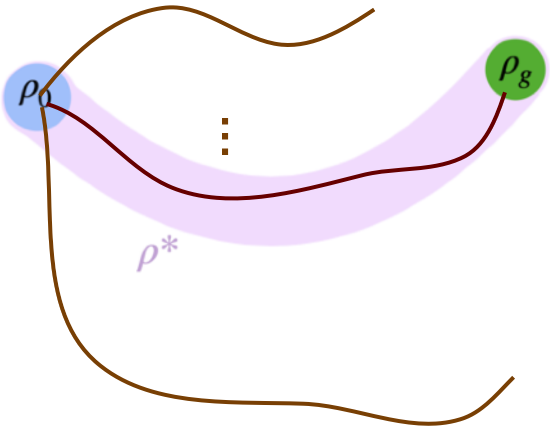
Non-Episodic Learning




Challenge 1: exploration can cause the agent to drift far away

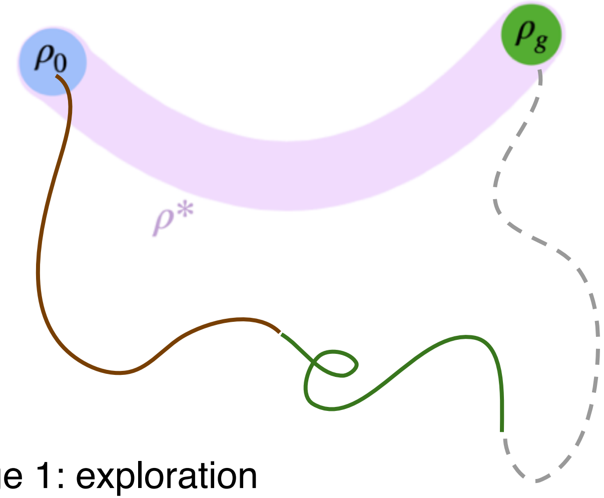
# Challenge of Non-Episodic Learning

Episodic Learning




Can always retry
the task from initial
state distribution

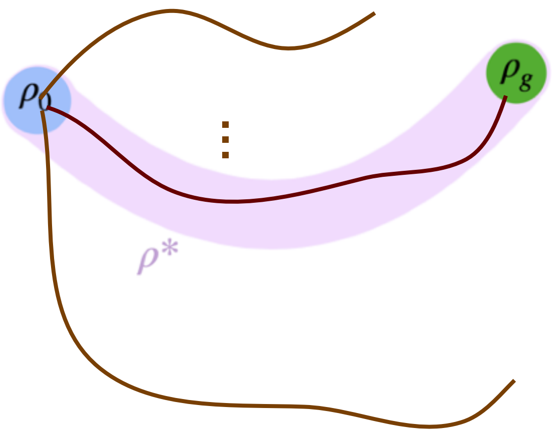
Non-Episodic Learning




Challenge 1: exploration
can cause the agent to
drift far away

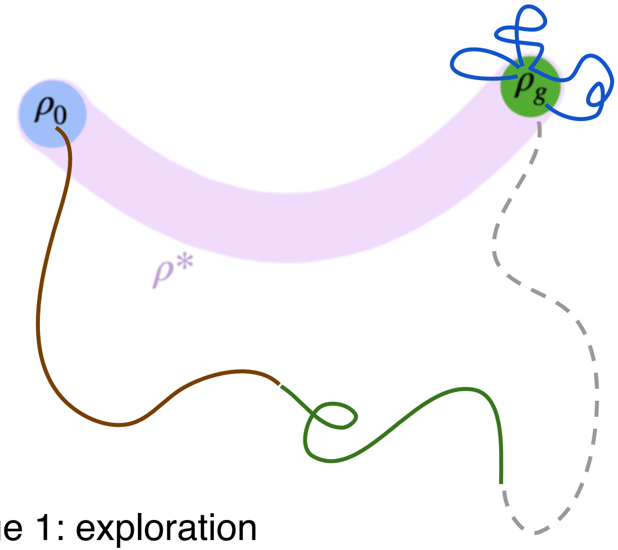
# Challenge of Non-Episodic Learning

Episodic Learning




Can always retry the task from initial state distribution

Non-Episodic Learning

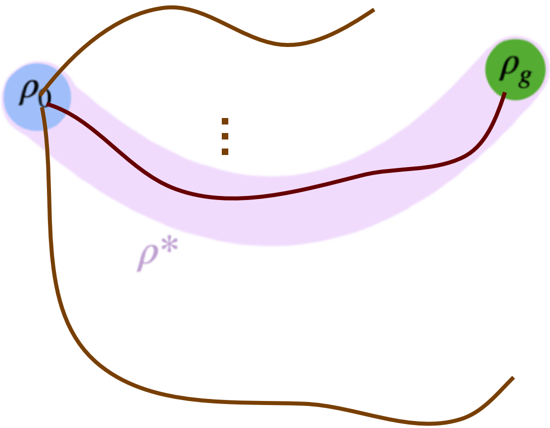



Challenge 2: state distribution collapse

Challenge 1: exploration can cause the agent to drift far away

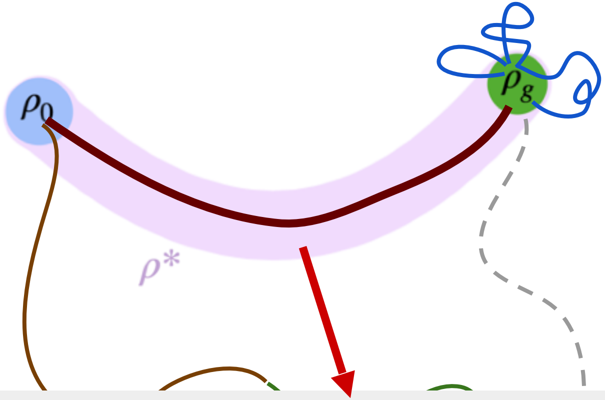
# Challenge of Non-Episodic Learning

Episodic Learning




Can always retry
the task from initial
state distribution

Non-Episodic Learning




Challenge 2: state
distribution collapse

Ch
can cause the agent to
drift far away

**The agent never learns a good policy**

# Non-Episodic Learning via MEDAL

# Non-Episodic Learning via MEDAL

**Matching Expert Distributions for Autonomous Learning**

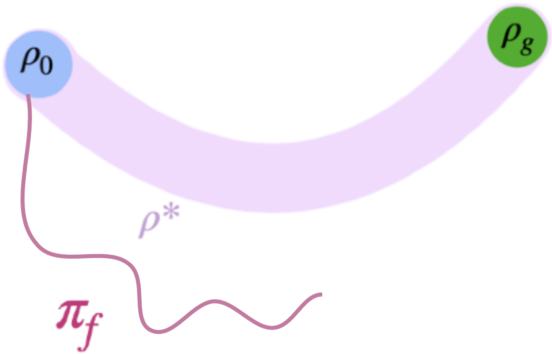
# Non-Episodic Learning via MEDAL

**Matching Expert Distributions for Autonomous Learning**


demonstrations

# Non-Episodic Learning via MEDAL

**Matching Expert Distributions for Autonomous Learning**

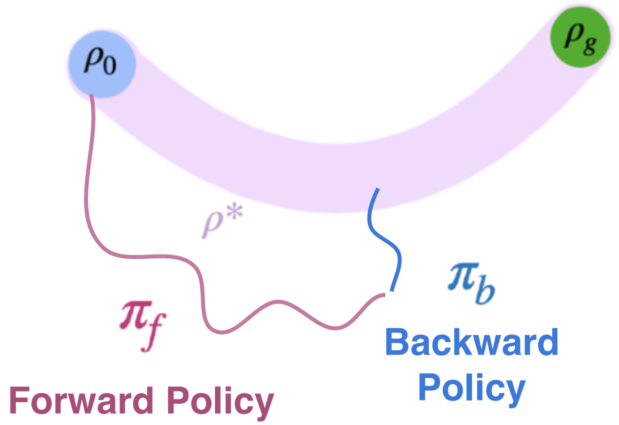



**Forward Policy**

demonstrations

# Non-Episodic Learning via MEDAL

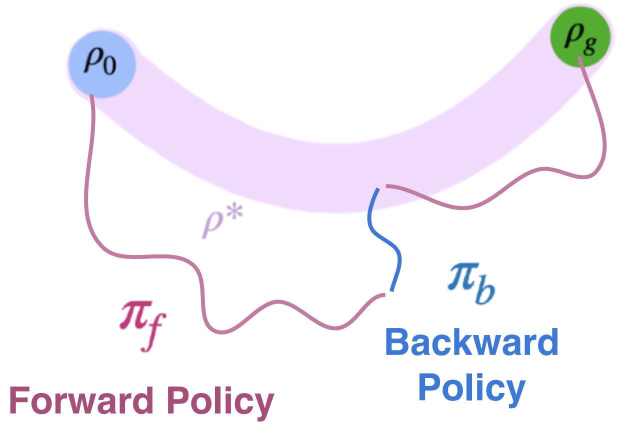
**Matching Expert Distributions for Autonomous Learning**




demonstrations

# Non-Episodic Learning via MEDAL

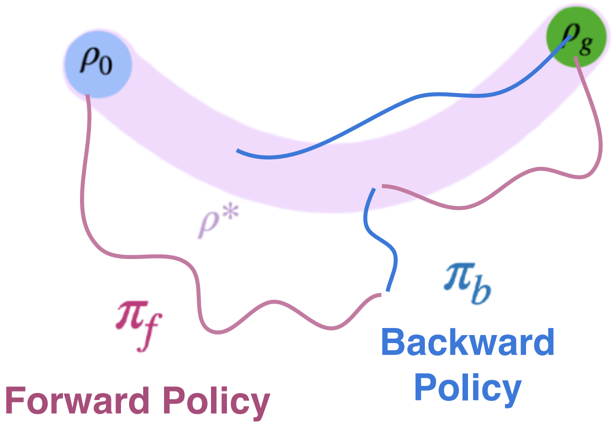
**Matching Expert Distributions for Autonomous Learning**




demonstrations

# Non-Episodic Learning via MEDAL

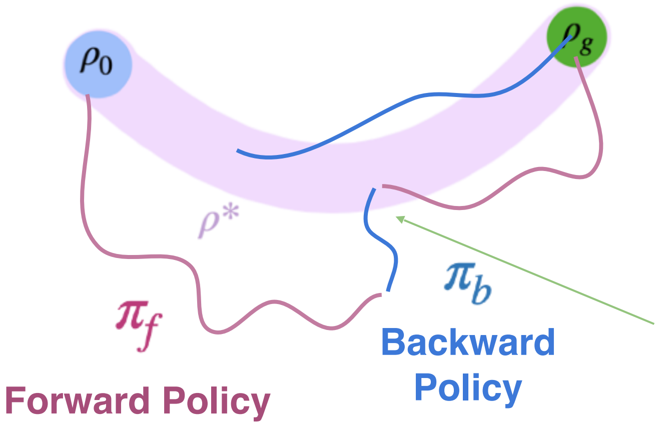
**Matching Expert Distributions for Autonomous Learning**




demonstrations

# Non-Episodic Learning via MEDAL

**Matching Expert Distributions for Autonomous Learning**



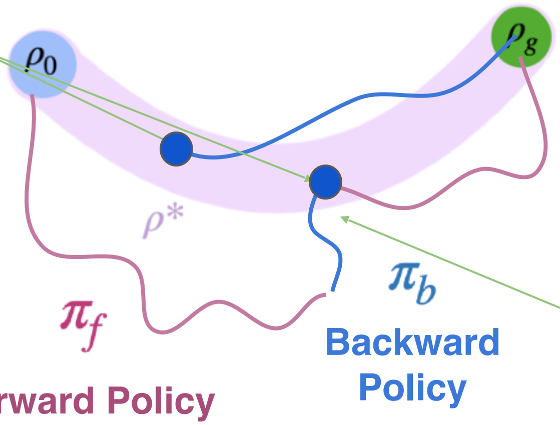

demonstrations

Addressing challenge 1: agent doesn't drift away

# Non-Episodic Learning via MEDAL

**Matching Expert Distributions for Autonomous Learning**

Addressing challenge 2: backward policy avoids collapse of state distribution



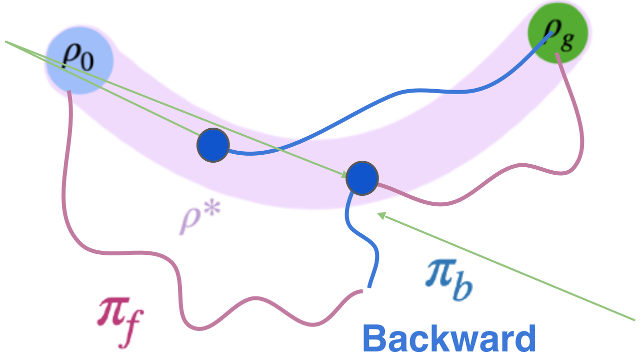

Addressing challenge 1: agent doesn't drift away

demonstrations

# Non-Episodic Learning via MEDAL

**Matching Expert Distributions for Autonomous Learning**

Addressing challenge 2: backward policy avoids collapse of state distribution




**Backward Policy**

**Forward Policy**

Addressing challenge 1: agent doesn't drift away

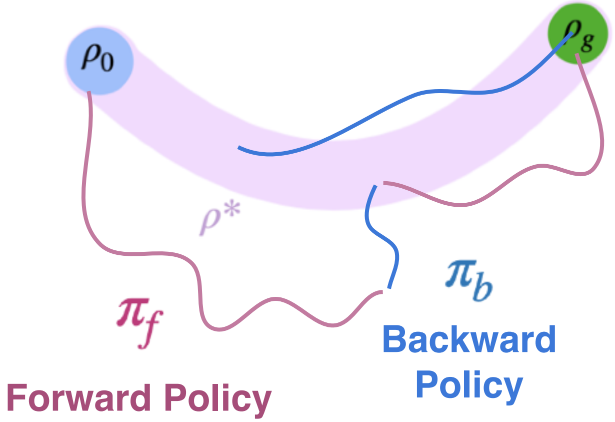
demonstrations

Pro: Forward policy tries the task from wide set of initial states, both easy and hard, improving the sample efficiency [1]

[1] Kakade & Langford. *Approximately Optimal Approximate Reinforcement Learning*. ICML 2002.

# MEDAL Overview

**Matching Expert Distributions for Autonomous Learning**

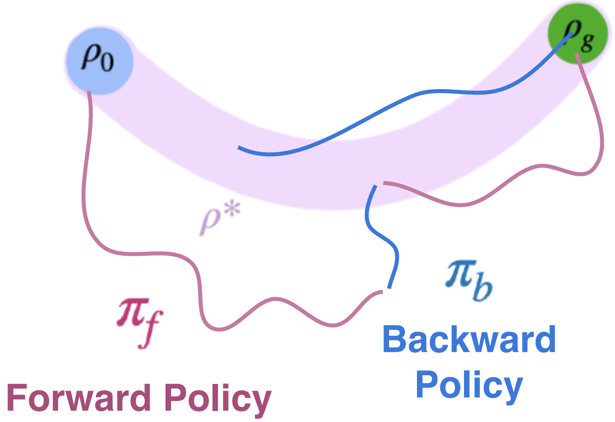


# MEDAL Overview

forward policy

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

**Matching Expert Distributions for Autonomous Learning**



# MEDAL Overview

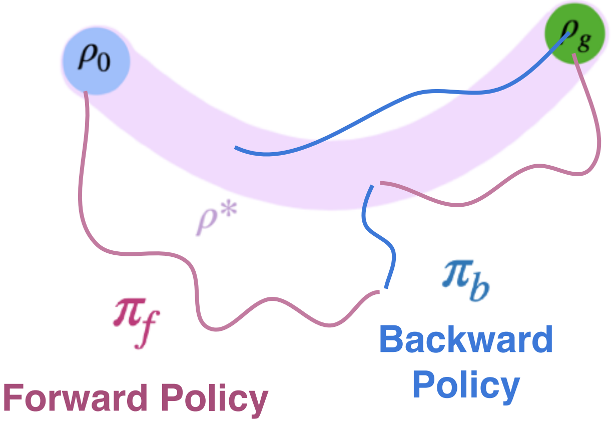
forward policy

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

backward policy

$$\mathcal{D}_{\mathrm{JS}}(\rho^{\pi_b}(s) \,||\, \rho^*(s))$$

**Matching Expert Distributions for Autonomous Learning**



# MEDAL Overview

forward policy

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

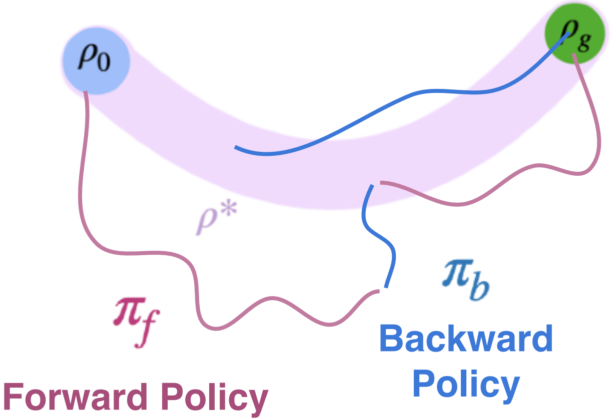
backward policy

$$\mathcal{D}_{\text{JS}}(\rho^{\pi_b}(s) \mid\mid \rho^*(s))$$

How do we minimize the $\mathcal{D}_{\text{JS}}$ ? Using the small set of demonstrations, learn a classifier $C(s)$ :

$$C(s) = \begin{cases} +1 & s \in \text{demos} \\ -1 & s \sim \rho^{\pi_b}(s) \end{cases}$$

**Matching Expert Distributions for Autonomous Learning**



# MEDAL Overview

forward policy

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

backward policy

$$\mathcal{D}_{\mathrm{JS}}(\rho^{\pi_b}(s) \mid\mid \rho^*(s))$$

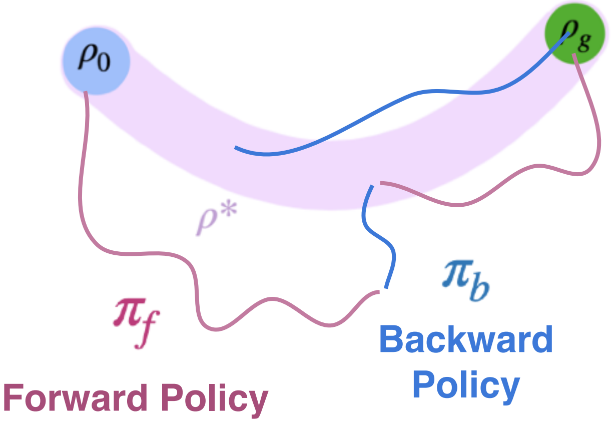
How do we minimize the $\mathcal{D}_{\mathrm{JS}}$ ? Using the small set of demonstrations, learn a classifier $C(s)$ :

$$C(s) = \begin{cases} +1 & s \in \text{demos} \\ -1 & s \sim \rho^{\pi_b}(s) \end{cases}$$

and the backward policy maximizes:

$$-\mathbb{E}\left[\sum_{t=0}^{\infty} \log(1 - C(s_{t+1}))\right]$$

**Matching Expert Distributions for Autonomous Learning**



# Results

# Results

**EARL Benchmark**

# Results

**EARL Benchmark**

**Training**: reset every 200k steps

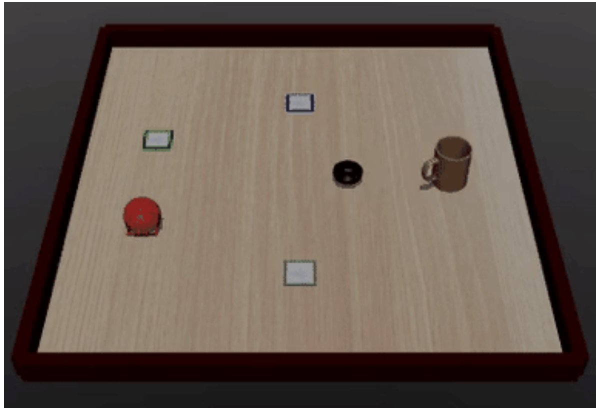
# Results

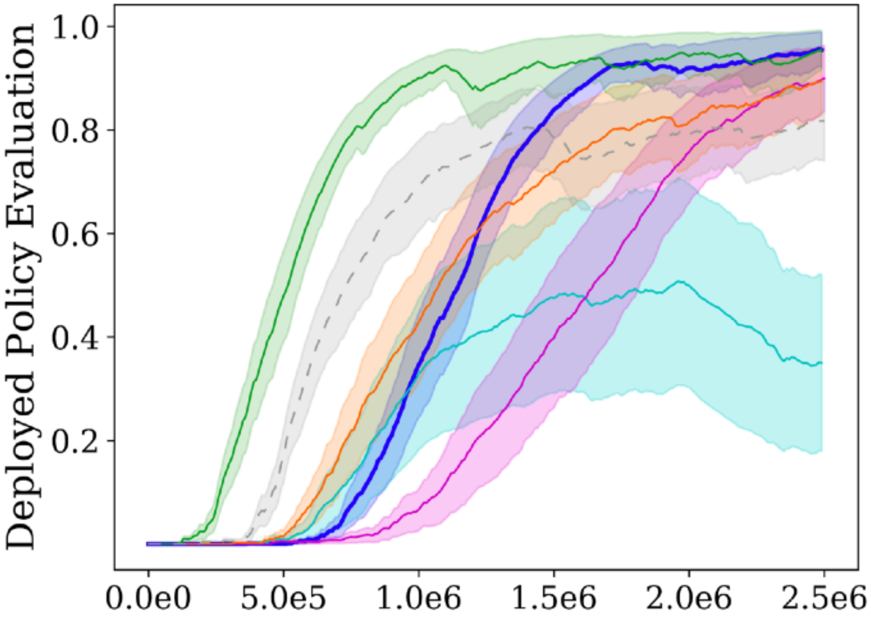
**EARL Benchmark**
**Training**: reset every 200k steps
**Evaluation**: policy performance from $\rho_0$

EARL: Sharma*, Xu* et al. Autonomous Reinforcement Learning: Formalism and Benchmarking, ICLR 2022.
VaPRL: Sharma et al. *Autonomous Reinforcement Learning via Subgoal Curricula*. NeurIPS 2021.
FBRL: Han et al. Learning Compound Multi-Step Controllers under Unknown Dynamics. IROS 2015.
R3L: Zhu et al. The Ingredients of Real-World Robotic Reinforcement Learning. ICLR 2020.

# Results

**EARL Benchmark**
**Training**: reset every 200k steps
**Evaluation**: policy performance from $\rho_0$

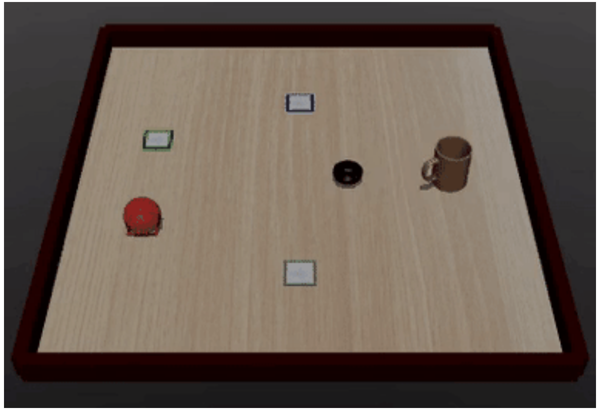


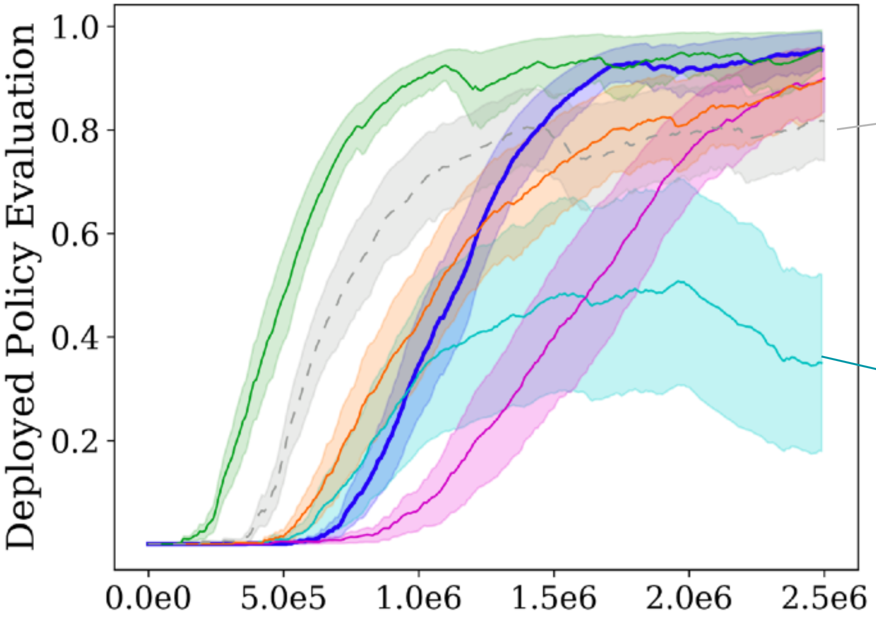



EARL: Sharma*, Xu* et al. Autonomous Reinforcement Learning: Formalism and Benchmarking, ICLR 2022.
VaPRL: Sharma et al. *Autonomous Reinforcement Learning via Subgoal Curricula*. NeurIPS 2021.
FBRL: Han et al. Learning Compound Multi-Step Controllers under Unknown Dynamics. IROS 2015.
R3L: Zhu et al. The Ingredients of Real-World Robotic Reinforcement Learning. ICLR 2020.

# Results

**EARL Benchmark**
**Training**: reset every 200k steps
**Evaluation**: policy performance from $\rho_0$





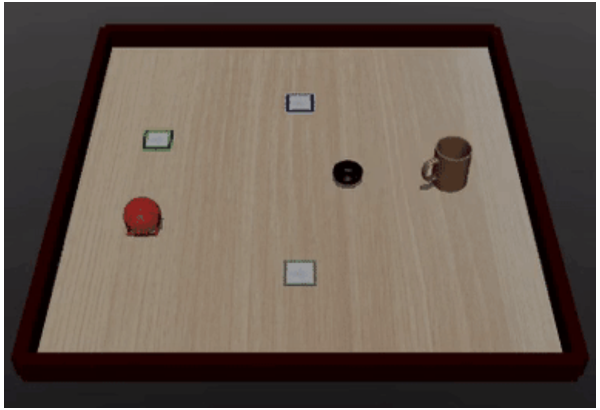

oracle SAC (episodic)

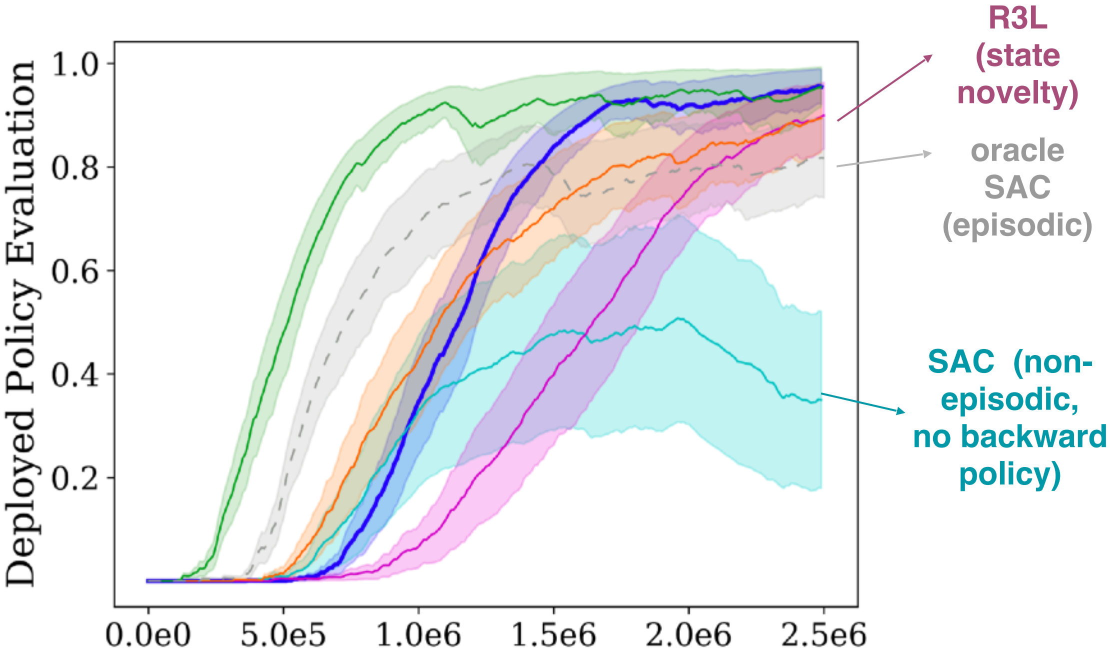
SAC (non-episodic, no backward policy)

EARL: Sharma*, Xu* et al. Autonomous Reinforcement Learning: Formalism and Benchmarking, ICLR 2022.
VaPRL: Sharma et al. *Autonomous Reinforcement Learning via Subgoal Curricula*. NeurIPS 2021.
FBRL: Han et al. Learning Compound Multi-Step Controllers under Unknown Dynamics. IROS 2015.
R3L: Zhu et al. The Ingredients of Real-World Robotic Reinforcement Learning. ICLR 2020.

# Results

**EARL Benchmark**
**Training**: reset every 200k steps
**Evaluation**: policy performance from $\rho_0$

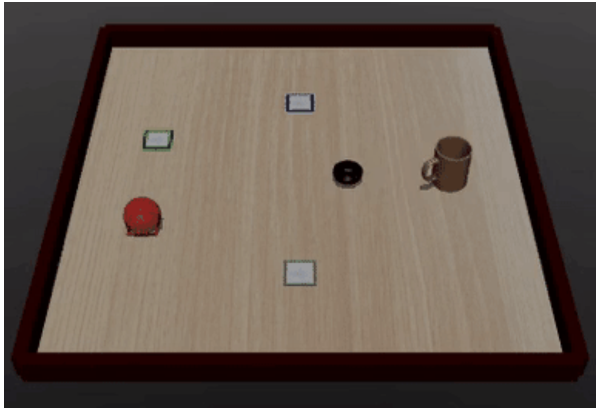


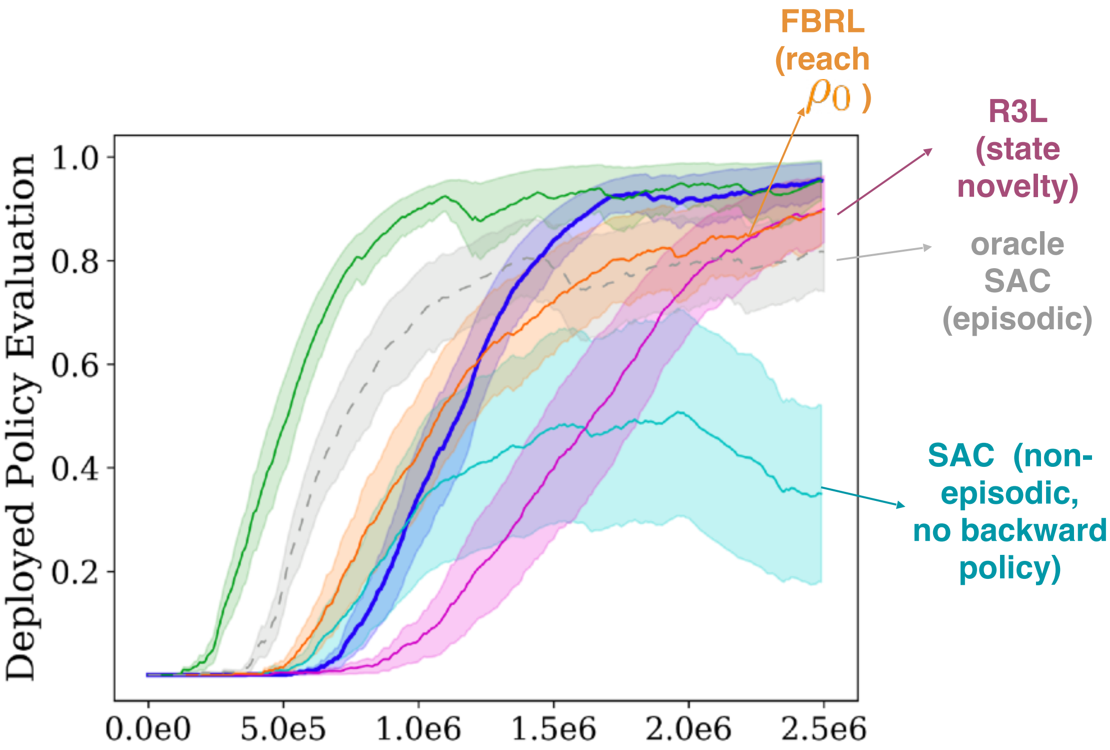



EARL: Sharma*, Xu* et al. Autonomous Reinforcement Learning: Formalism and Benchmarking, ICLR 2022.
VaPRL: Sharma et al. *Autonomous Reinforcement Learning via Subgoal Curricula*. NeurIPS 2021.
FBRL: Han et al. Learning Compound Multi-Step Controllers under Unknown Dynamics. IROS 2015.
R3L: Zhu et al. The Ingredients of Real-World Robotic Reinforcement Learning. ICLR 2020.

# Results

**EARL Benchmark**
**Training**: reset every 200k steps
**Evaluation**: policy performance from $\rho_0$

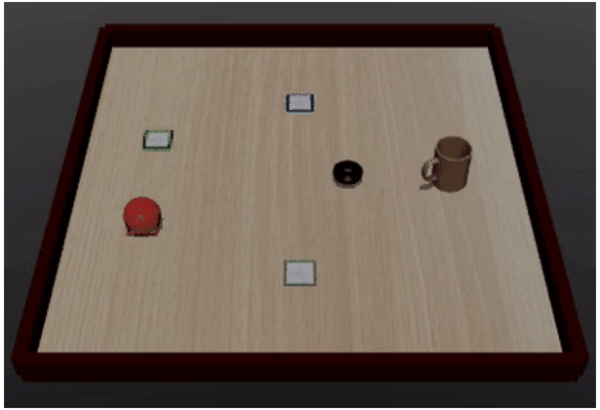


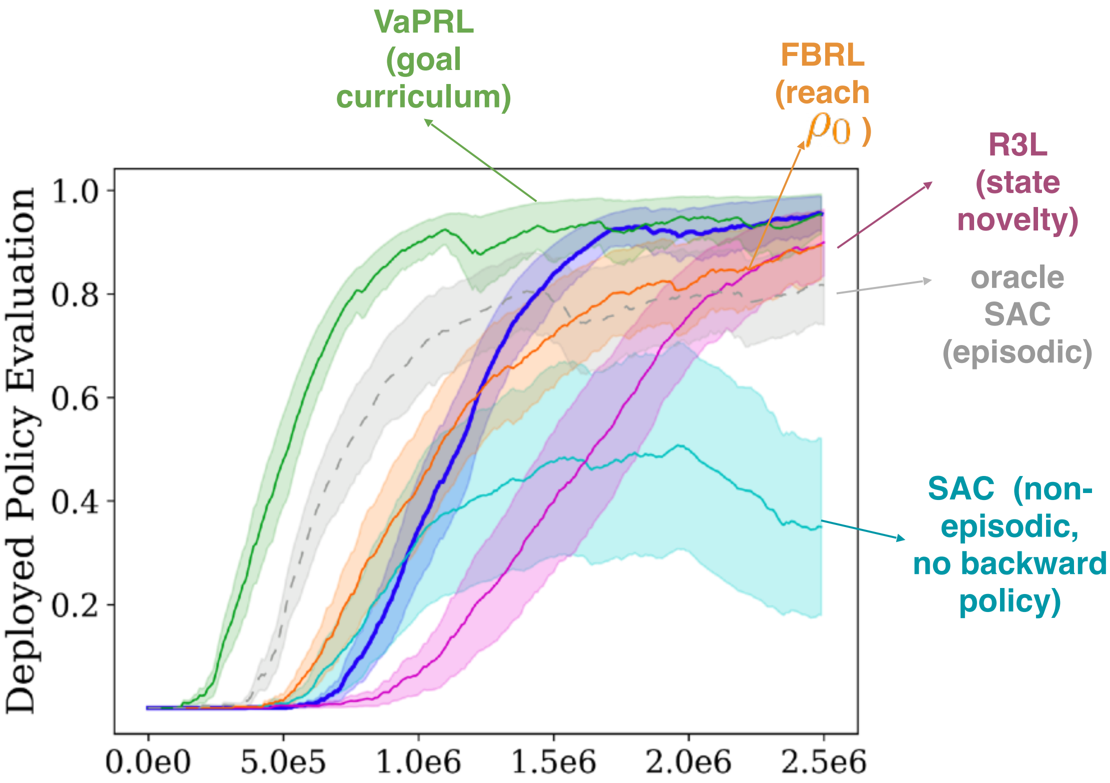



EARL: Sharma*, Xu* et al. Autonomous Reinforcement Learning: Formalism and Benchmarking, ICLR 2022.
VaPRL: Sharma et al. *Autonomous Reinforcement Learning via Subgoal Curricula*. NeurIPS 2021.
FBRL: Han et al. Learning Compound Multi-Step Controllers under Unknown Dynamics. IROS 2015.
R3L: Zhu et al. The Ingredients of Real-World Robotic Reinforcement Learning. ICLR 2020.

# Results

**EARL Benchmark**
**Training**: reset every 200k steps
**Evaluation**: policy performance from $\rho_0$

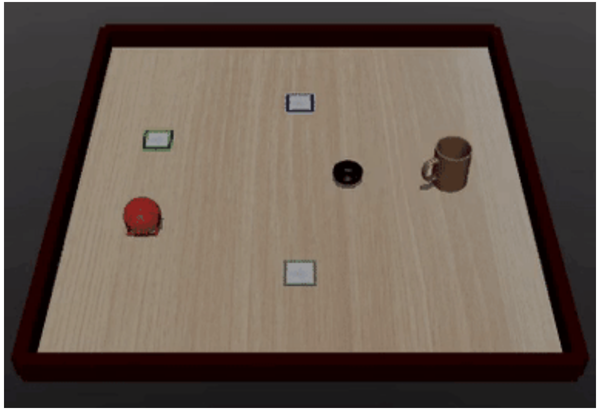


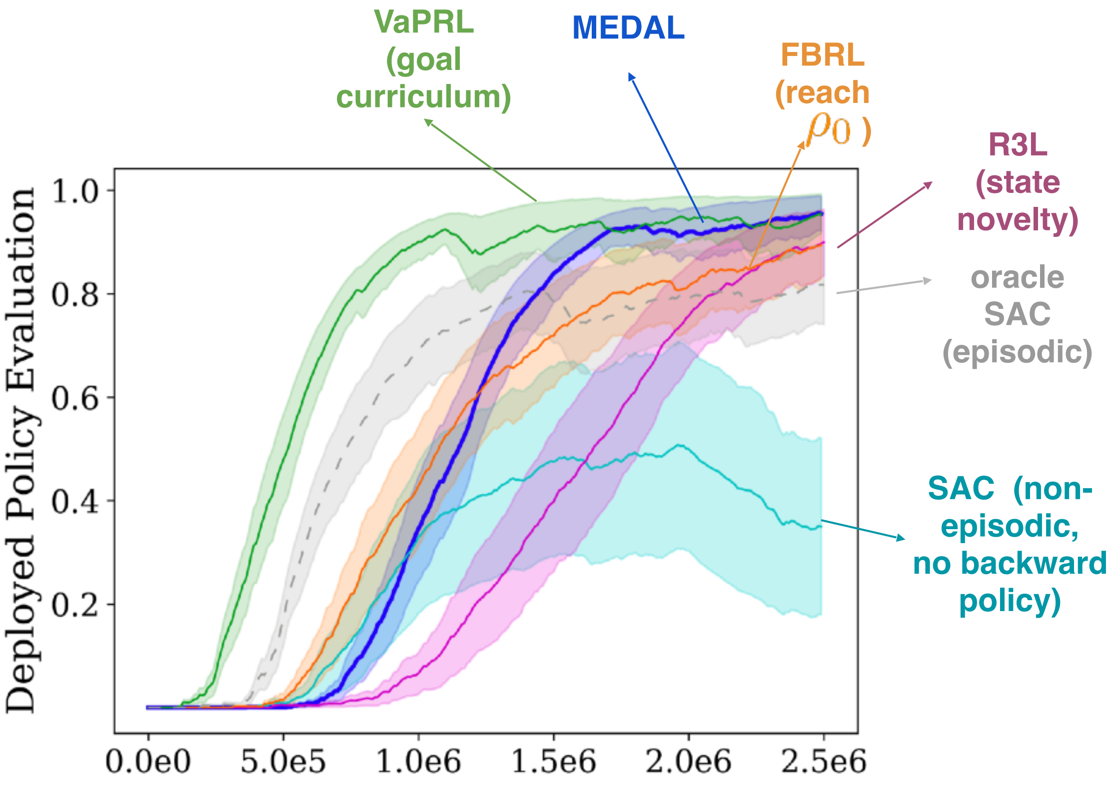



EARL: Sharma*, Xu* et al. Autonomous Reinforcement Learning: Formalism and Benchmarking, ICLR 2022.
VaPRL: Sharma et al. *Autonomous Reinforcement Learning via Subgoal Curricula*. NeurIPS 2021.
FBRL: Han et al. Learning Compound Multi-Step Controllers under Unknown Dynamics. IROS 2015.
R3L: Zhu et al. The Ingredients of Real-World Robotic Reinforcement Learning. ICLR 2020.

# Results

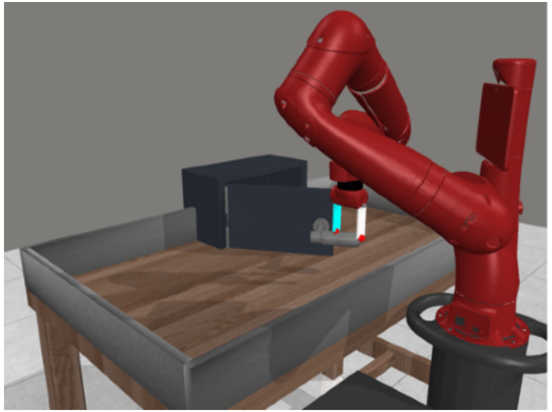
**EARL Benchmark**
**Training**: reset every 200k steps
**Evaluation**: policy performance from $\rho_0$



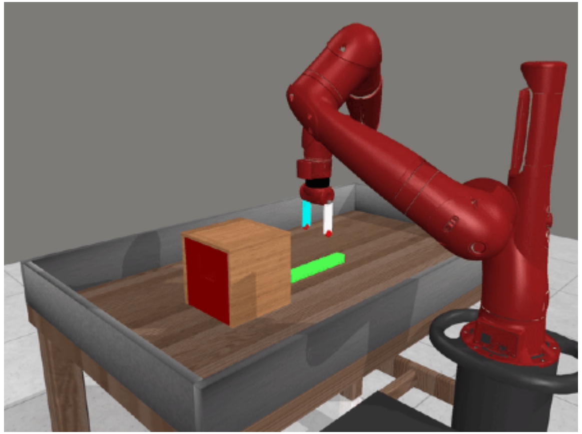




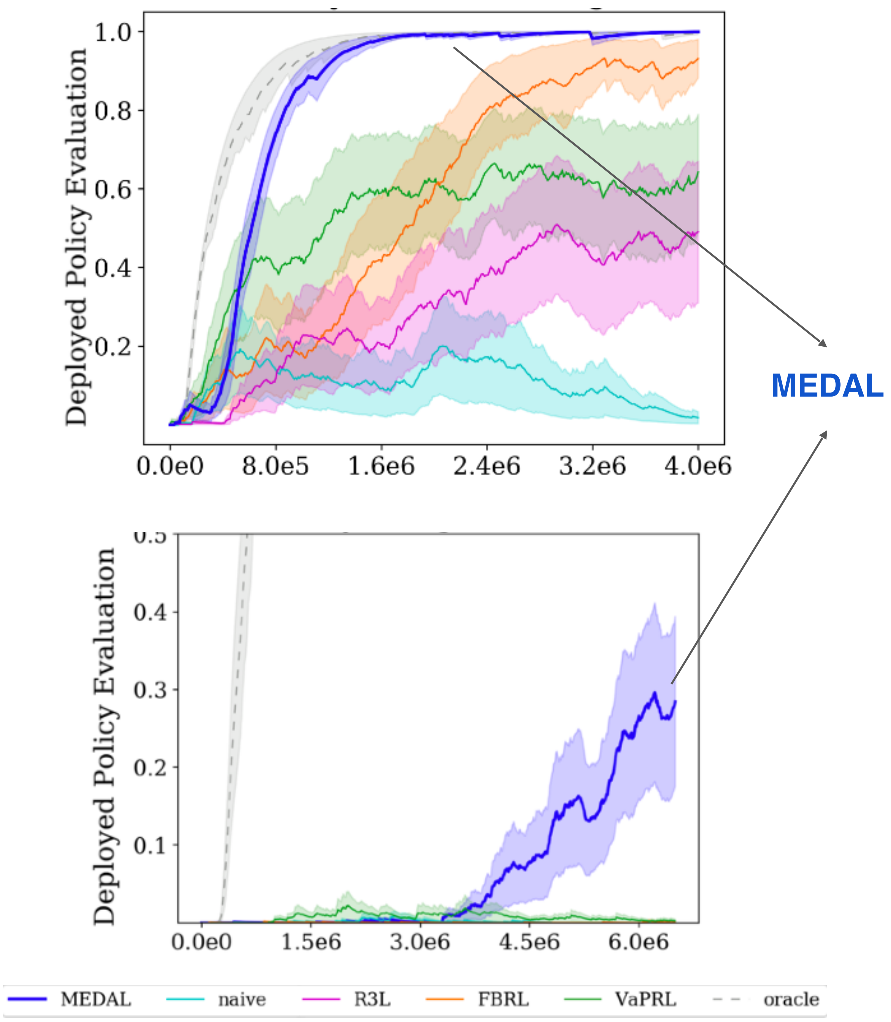
EARL: Sharma*, Xu* et al. Autonomous Reinforcement Learning: Formalism and Benchmarking, ICLR 2022.
VaPRL: Sharma et al. *Autonomous Reinforcement Learning via Subgoal Curricula*. NeurIPS 2021.
FBRL: Han et al. Learning Compound Multi-Step Controllers under Unknown Dynamics. IROS 2015.
R3L: Zhu et al. The Ingredients of Real-World Robotic Reinforcement Learning. ICLR 2020.



Door Closing



Peg Insertion



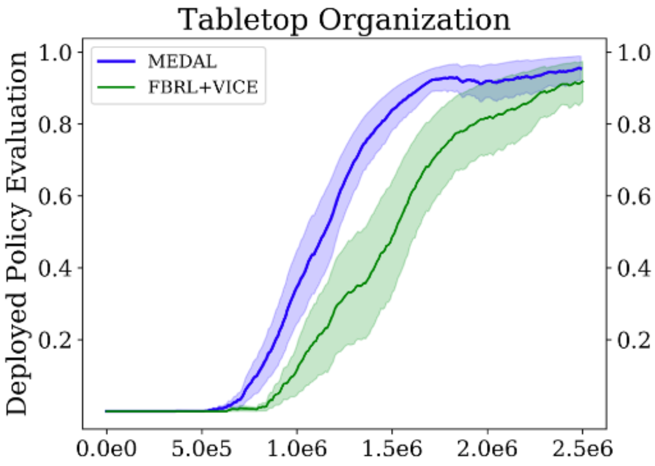
How important is it to match the expert distribution in MEDAL?

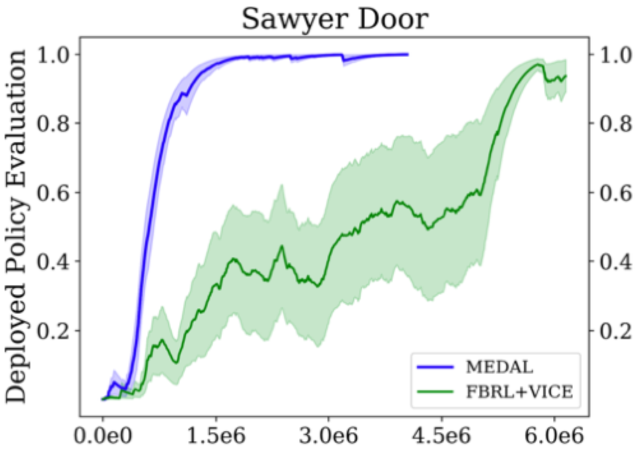
How important is it to match the expert distribution in MEDAL?

Ablation: match the **initial state distribution**

How important is it to match the expert distribution in MEDAL?

Ablation: match the **initial state distribution**

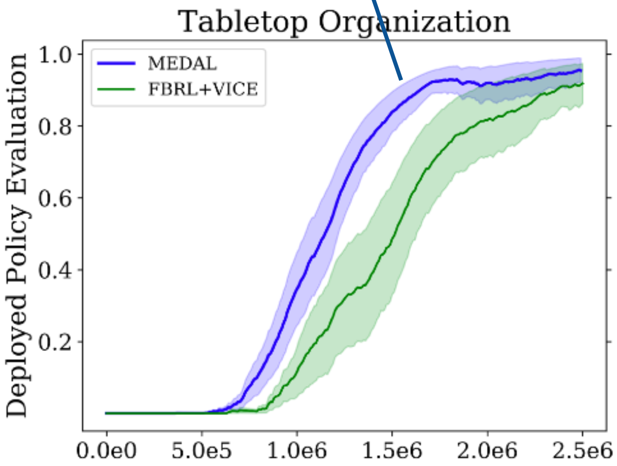


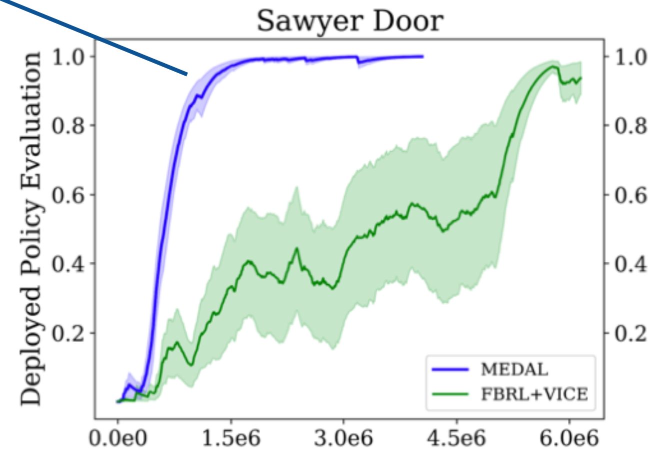



How important is it to match the expert distribution in MEDAL?

Ablation: match the **initial state distribution**

$$\mathcal{D}_{\mathrm{JS}}(\rho^{\pi_b}(s) \mid\mid \rho^*(s))$$

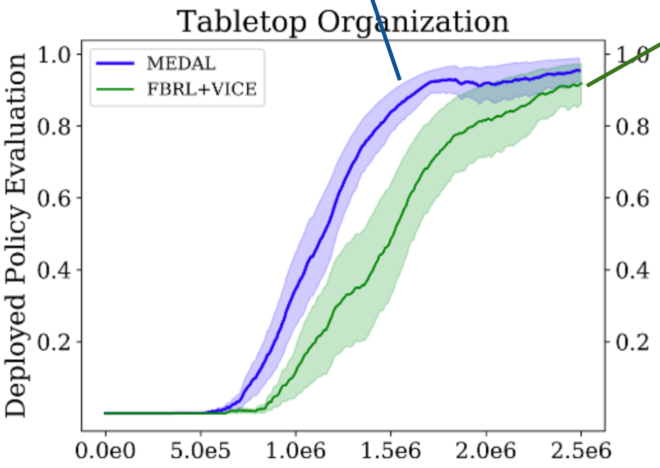


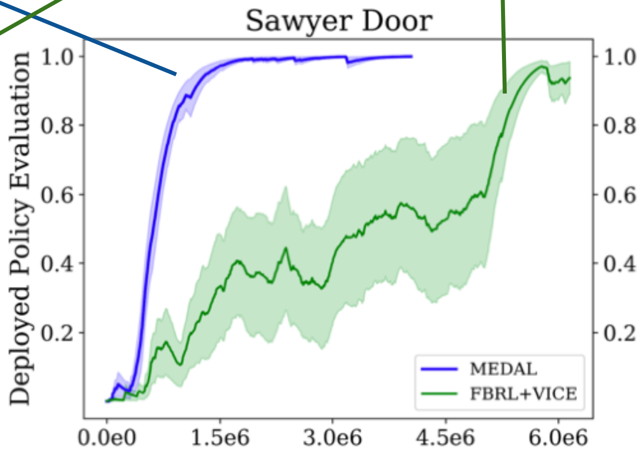



How important is it to match the expert distribution in MEDAL?

Ablation: match the **initial state distribution**

$$\mathcal{D}_{\mathrm{JS}}(\rho^{\pi_b}(s) \,||\, \rho^*(s)) \qquad \mathcal{D}_{\mathrm{JS}}(\rho^{\pi_b}(s) \,||\, \rho_0(s))$$



# Conclusion

# Conclusion

- Proposed MEDAL, a simple and efficient autonomous RL algorithm

# Conclusion

- Proposed MEDAL, a simple and efficient autonomous RL algorithm
    - Encourages the agent to stay close to the expert state distribution

# Conclusion

- Proposed MEDAL, a simple and efficient autonomous RL algorithm
    - Encourages the agent to stay close to the expert state distribution
    - Wider initial state distribution enables sample efficient learning

# Conclusion

- Proposed MEDAL, a simple and efficient autonomous RL algorithm
  - Encourages the agent to stay close to the expert state distribution
  - Wider initial state distribution enables sample efficient learning

Website: https://sites.google.com/view/medal-arl/home
Code: https://github.com/architsharma97/medal

Archit Sharma

Rehaan Ahmad

Chelsea Finn