RL for Robots 🤖: Autonomous Learning

CS224R: Deep Reinforcement Learning

Reminders

- Project milestone due tonight
- HW4 due next Friday

Plan for Today

- Why aren’t robots autonomous already?
- Defining the problem: **autonomous RL**
- Developing the algorithms
    - Learning policies without human intervention
    - Single life RL

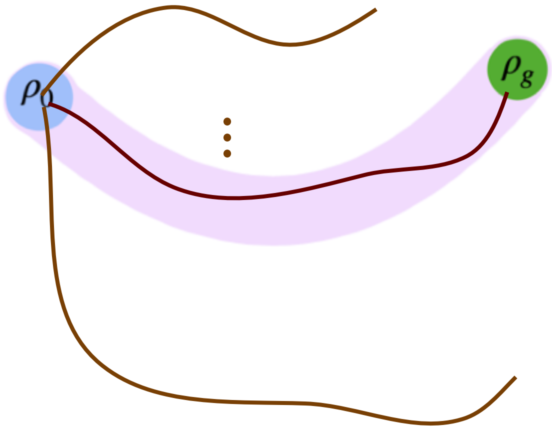
Goal: Build autonomous agents that can learn and interact in the real world

Plan for Today

- Why aren't robots autonomous already?
- Defining the problem: autonomous RL
- Developing the algorithms
    - Learning policies without human intervention
    - Single life RL

Reinforcement Learning = Trial-and-Error

Learn a policy π to go from ρ_0 to ρ_g



Repeat:

- Execute actions from the policy π
- Observe data from the environment
- Update the policy π

Slide adapted from Archit Sharma

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() # yo
  observation, reward, done, info = env.s

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

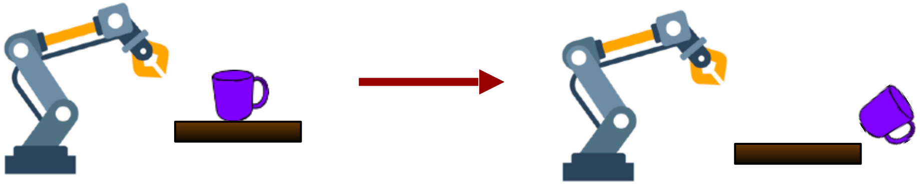
[Code snippet from https://gym.openai.com/]

Only in simulation!

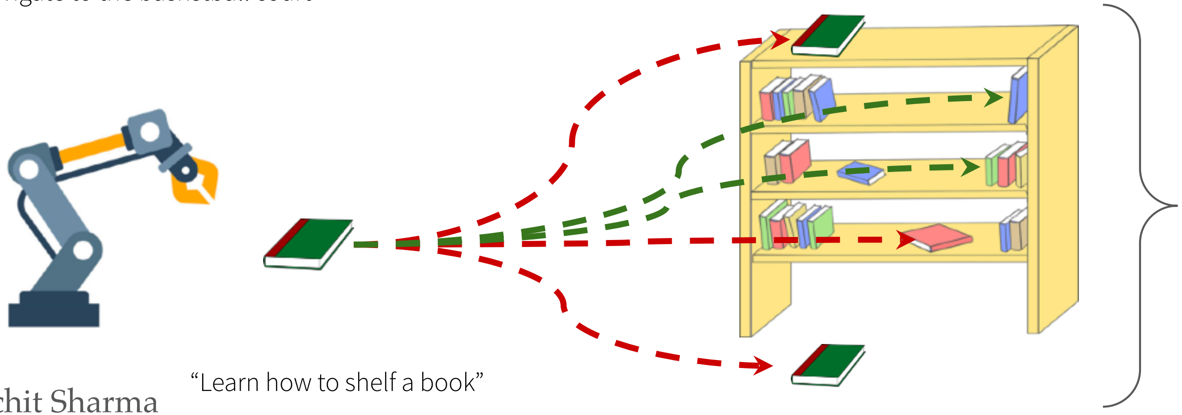
Slide adapted from Archit Sharma

The Continual Real World

"Navigate to the basketball court"



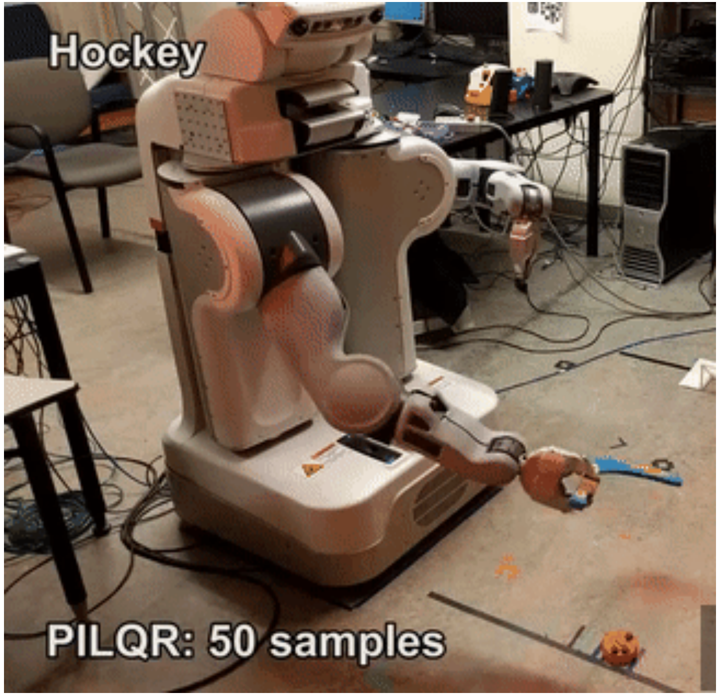
"Grasp the mug"



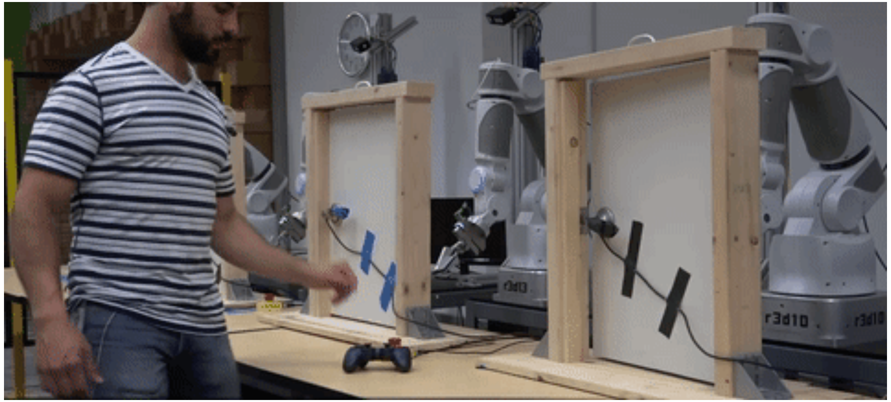
"Learn how to shelf a book"

Several thousands of trials!

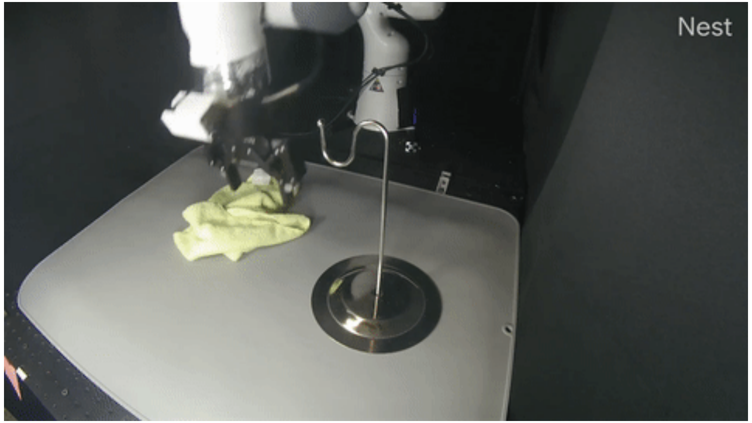
Slide adapted from Archit Sharma



[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]



[Self-Improving Robots: End-to-End Autonomous Visuomotor Reinforcement Learning, Sharma et al. 2023]

Slide adapted from Archit Sharma

Plan for Today

- Why aren't robots autonomous already?
- Defining the problem: autonomous RL
- Developing the algorithms
    - Learning policies without human intervention
    - Single life RL

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

Reset Environment

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

🚩 Problem: this requires human supervision

What if we increase *H*?

✔ Fewer environment resets, less human supervision

Autonomous RL: Definition

Agent interacts autonomously

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

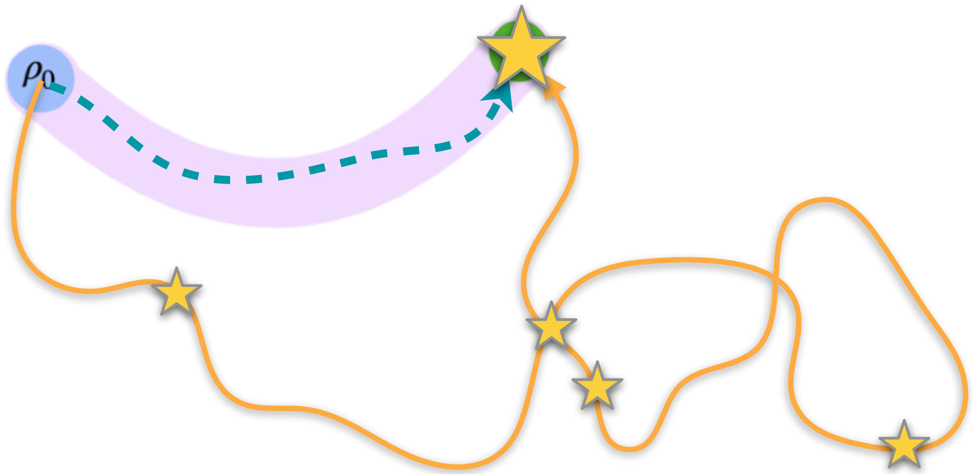
No environment resets*

Initialize *once* at the beginning

Slide adapted from Archit Sharma

*can relax this constraint to reset occasionally, or at low frequency

Autonomous RL: Evaluation



We might care about two different things:

- The amount of reward recovered over the course of its life (ex: Mars rover)
- The quality of the learned policy (ex: robot chef)

Slide adapted from Archit Sharma

Autonomous RL: Evaluation

Deployed Policy Evaluation

= Quality of policy learned

$$J(\pi) = \mathbb{E}_{\mathbf{s}_1 \sim p(\mathbf{s}_1), \mathbf{a}_t \sim \pi} \left[\sum_t \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

- Start from initial state distribution
- Take actions according to π
- Total reward over the episode

Continuing Policy Evaluation*

= Reward accumulated over lifetime

$$\lim_{h \to \infty} \mathbb{E} \left[\frac{1}{h} \sum_{t=0}^{h} r(s_t, a_t) \right]$$

Average over reward accumulated in the lifetime

We'll look at algorithms for both!

Slide adapted from Archit Sharma

*average reward RL

Why is autonomous RL important?

Robotics <-> Autonomy

- We want robots to operate with minimal human supervision
- We want robots to *train* with minimal human supervision

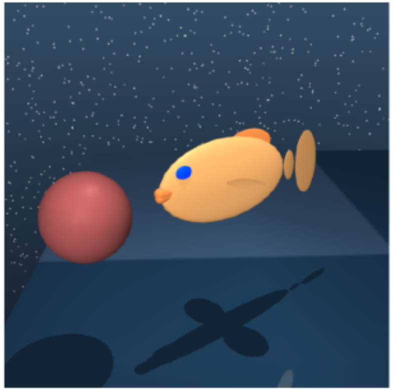
RL on real robots is promising path towards high success rate, reliability.
Autonomy unlocks more data, more data may unlock greater generalization.

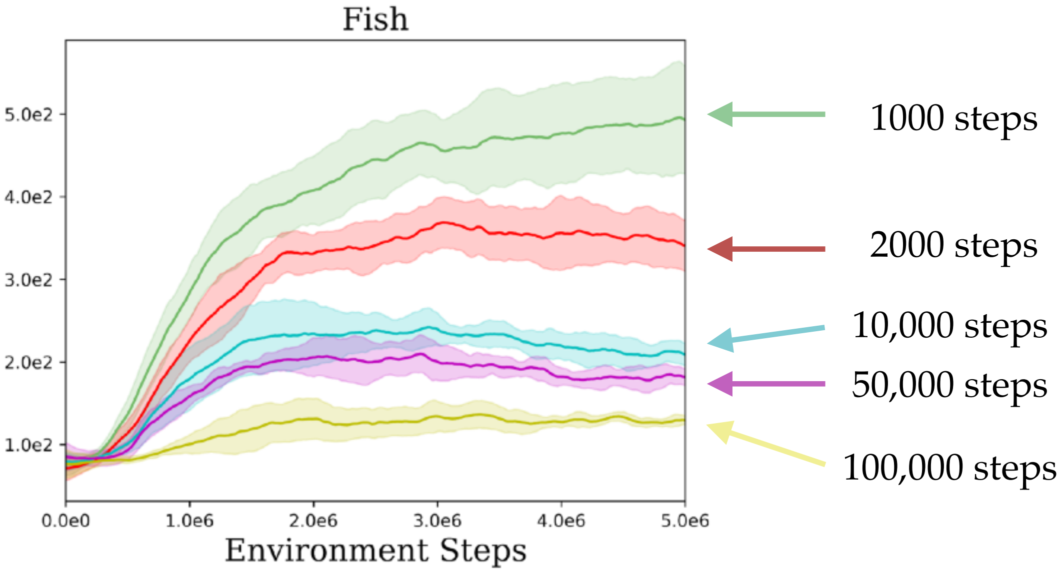
Plan for Today

- Why aren't robots autonomous already?
- Defining the problem: autonomous RL
- Developing the algorithms
    - Learning policies without human intervention
    - Single life RL

What about standard RL algorithms?

What happens when episode length increases?

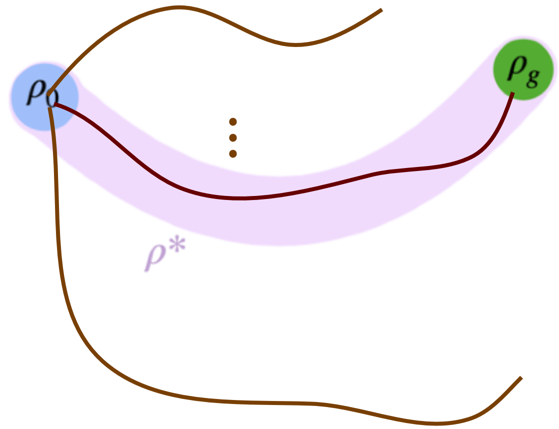




Note: this measures the **deployed policy evaluation**.

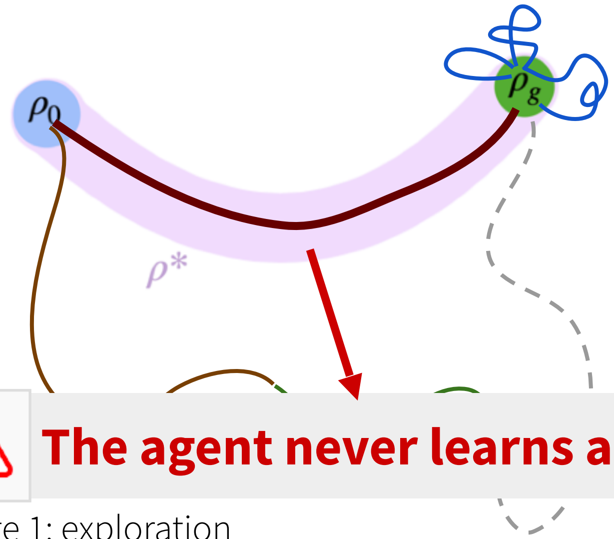
The Challenge of Learning Autonomously

Episodic Learning



Can always retry the task from initial state distribution

Non-Episodic Learning



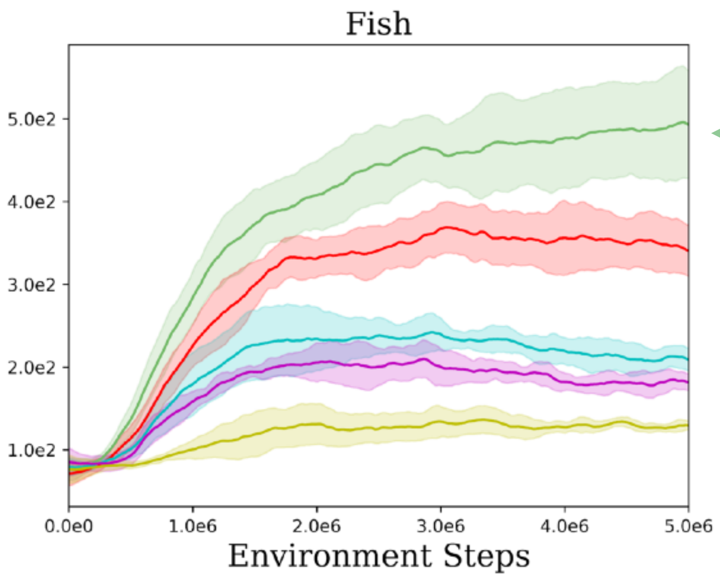
Challenge 2: state distribution collapse

⚠ **The agent never learns a good policy**

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

Slide adapted from Archit Sharma

How to autonomously learn an effective policy?

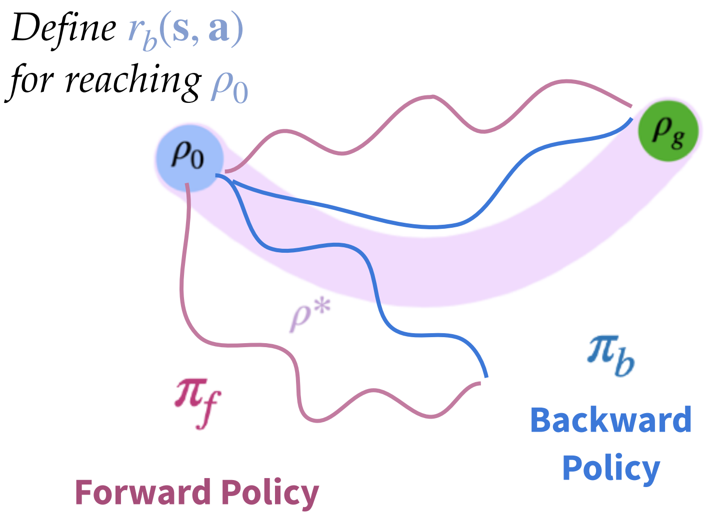


Learns successfully if allowed to frequently retry *from the initial state distribution*

Key idea: learn a policy to reset?

Slide adapted from Archit Sharma

Algorithm: Forward-Backward RL



Forward-Backward RL (FBRL)

1. Initialize forward policy π_f and backward policy π_b
2. Repeat:
   1. Roll out π_f for H steps
   2. Update π_f using $r_f(\mathbf{s}, \mathbf{a})$
   3. Roll out π_b for H steps
   4. Update π_b using $r_b(\mathbf{s}, \mathbf{a})$
   5. *No reset or very infrequent reset*

Test time: Discard π_b. Deploy π_f

\+ simple   - backward policy is ultimately discarded

[Han et al. Learning Compound Multi-Step Controllers under Unknown Dynamics]
[Eysenbach et al. Leave no Trace: Learning to Reset for Safe and Autonomous Reinforcement Learning]

Alternatives to Forward-Backward RL?

1. Learn to reset to expert state distribution (forward-backward with modified backward reward)

2. Practice *cycle* of different tasks

3. Set up task that can be done repetitively ("forward-forward RL"?)

Alternatives to Forward-Backward RL?

1. **Learn to reset to expert state distribution** (forward-backward with modified backward reward)

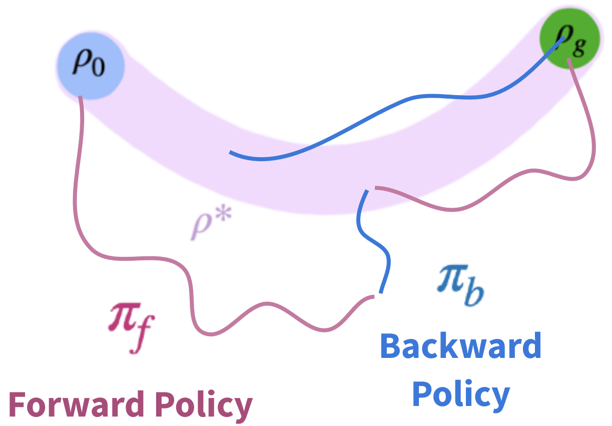
2. Practice *cycle* of different tasks

3. Set up task that can be done repetitively ("forward-forward RL"?)

Algorithm: Modified backward policy reward

Matching Expert Distributions for Autonomous Learning (MEDAL)

Modified reward $r_b(\mathbf{s}, \mathbf{a})$: learned reward for matching state distribution in demos.



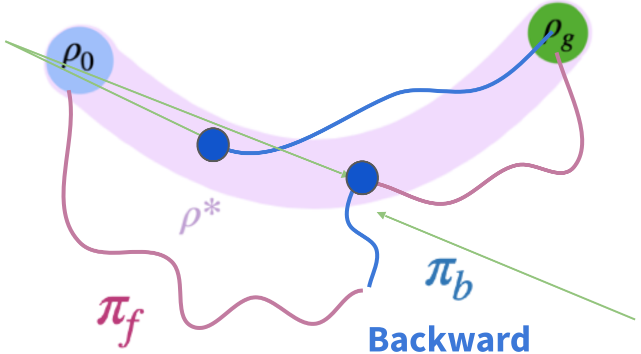
0. Given a small set of demonstrations
1. Initialize forward policy π_f and backward policy π_b
2. Repeat:
    1. Roll out π_f for H steps
    2. Update π_f using $r_f(\mathbf{s}, \mathbf{a})$
    3. Roll out π_b for H steps
    4. Update π_b using $r_b(\mathbf{s}, \mathbf{a})$
    5. *No reset or very infrequent reset*
    6. Update learned reward $r_b(\mathbf{s}, \mathbf{a})$

Test time: Discard π_b. Deploy π_f

Autonomous Learning via MEDAL

Matching Expert Distributions for Autonomous Learning

Addressing challenge 2: backward policy avoids collapse of state distribution



Forward Policy

Backward 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]

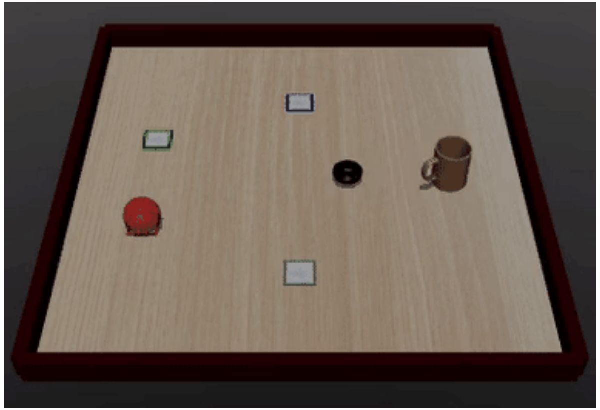
[1] Kakade & Langford. *Approximately Optimal Approximate Reinforcement Learning*. ICML 2002.

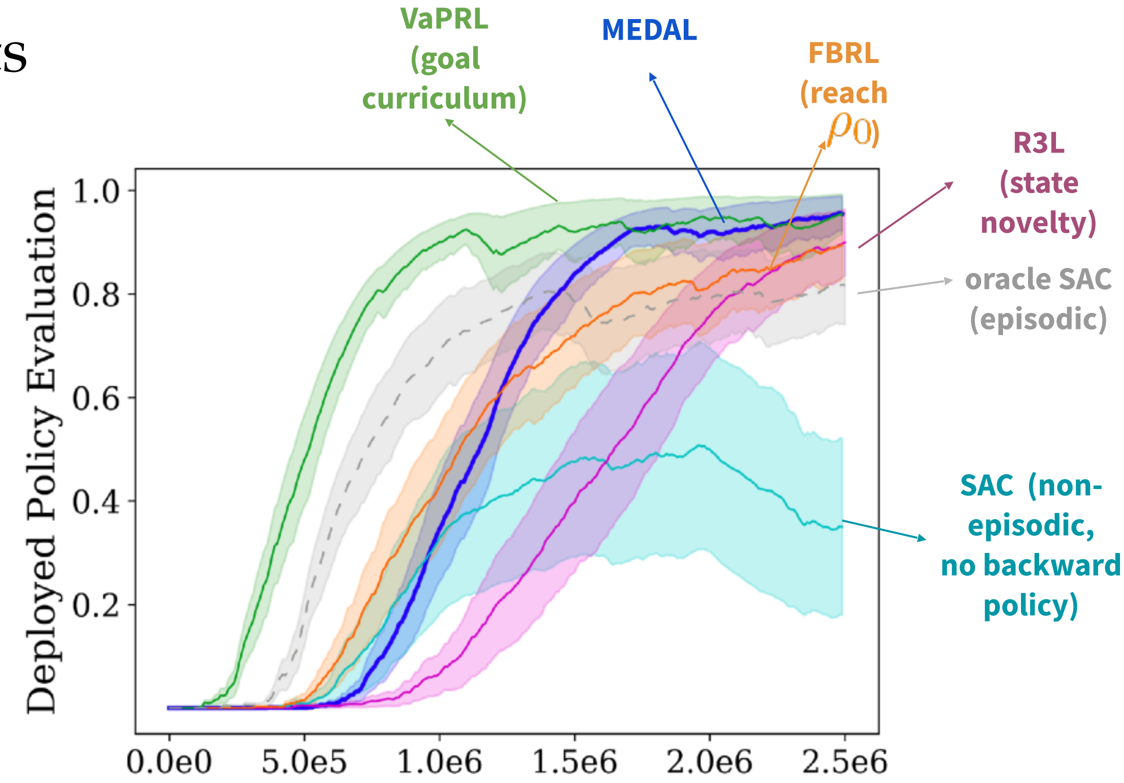
Forward backward results

EARL Benchmark

Training: reset every 200k steps

Evaluation: policy performance from ρ_0





EARL: Sharma*, Xu* et al. Autonomous Reinforcement Learning: Formalism and Benchmarking, ICLR 2022.

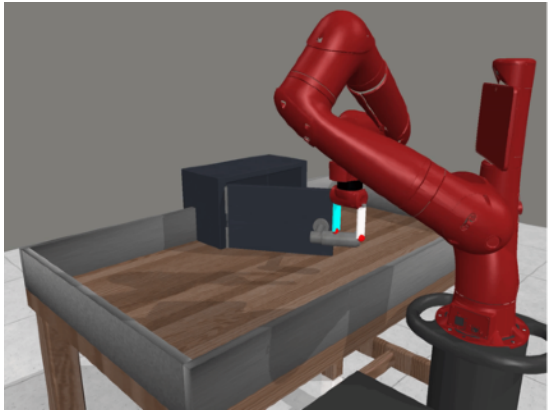
VaPRL: Sharma et al. *Autonomous Reinforcement Learning via Subgoal Curricula*. NeurIPS 2021.

FBRL: Han et al. Learning Compound Multi-Step Controllers under Unknown Dynamics. IROS 2015.

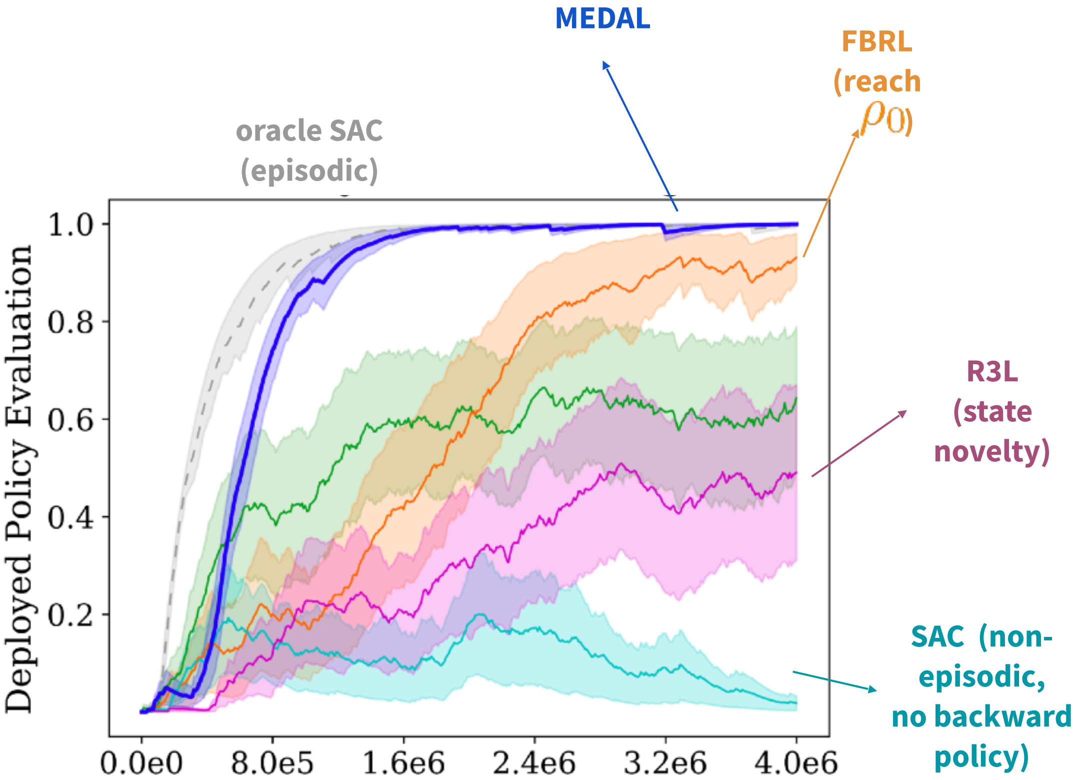
R3L: Zhu et al. The Ingredients of Real-World Robotic Reinforcement Learning. ICLR 2020.

Forward backward results

EARL Benchmark
Training: reset every 200k steps
Evaluation: policy performance from ρ_0



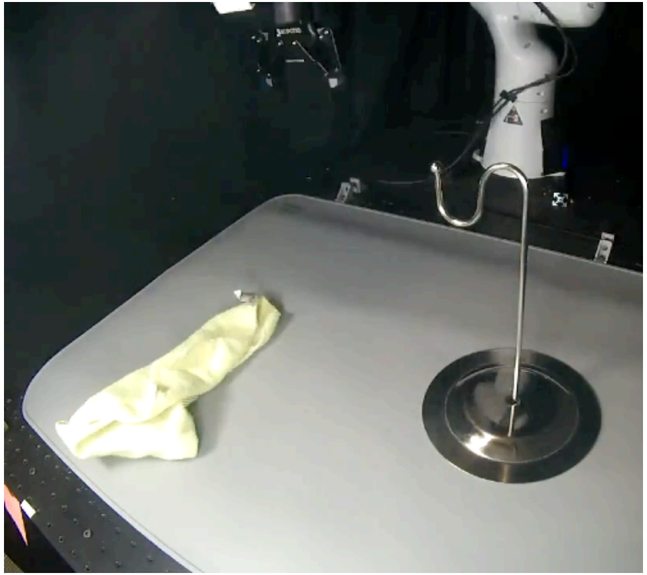
Door Closing



EARL: Sharma*, Xu* et al. Autonomous Reinforcement Learning: Formalism and Benchmarking, ICLR 2022.
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.

Forward-backward RL on real robots

Vanilla FBRL
Demo initialized, VLM-provided rewards



MEDAL
Demo initialized, learned rewards

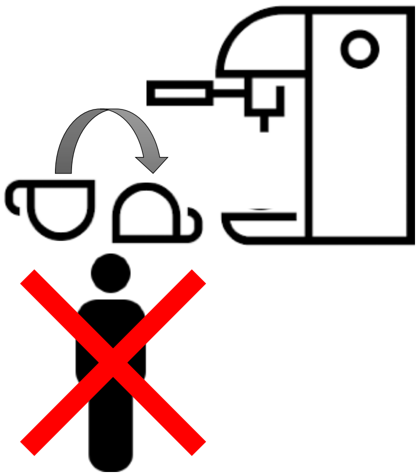
Alternatives to Forward-Backward RL?

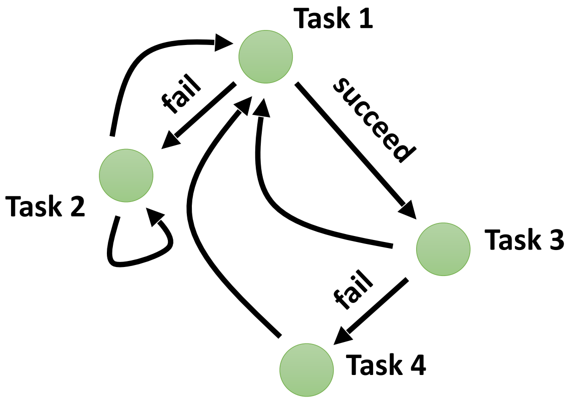
1. Learn to reset to expert state distribution (forward-backward with modified backward reward)

2. Practice *cycle* of different tasks

3. Set up task that can be done repetitively ("forward-forward RL"?)

Autonomous RL with task cycles





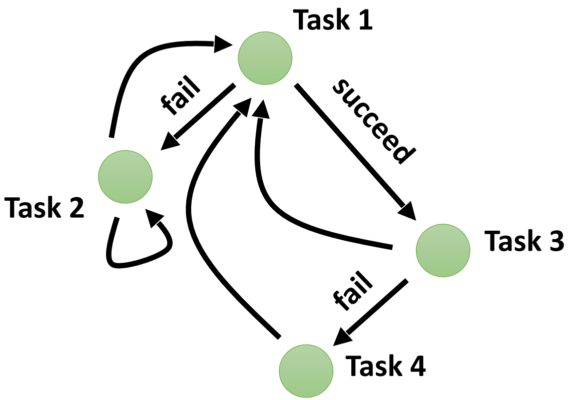
Task 1: put cup in coffee machine

Task 2: pick up cup

Task 3: replace cup

Task 4: clean up spill from cup…

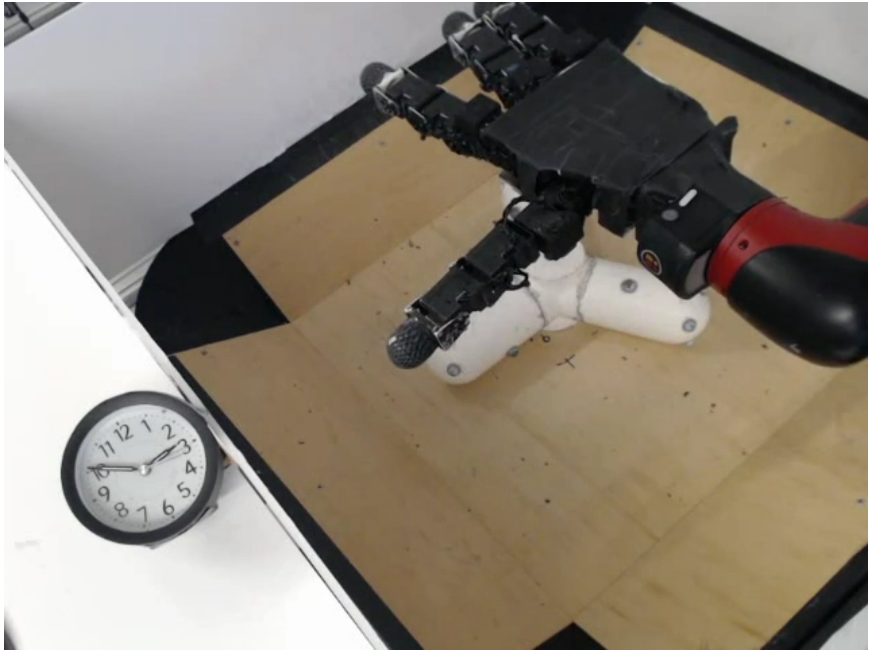
Algorithm: Autonomous RL with task cycles

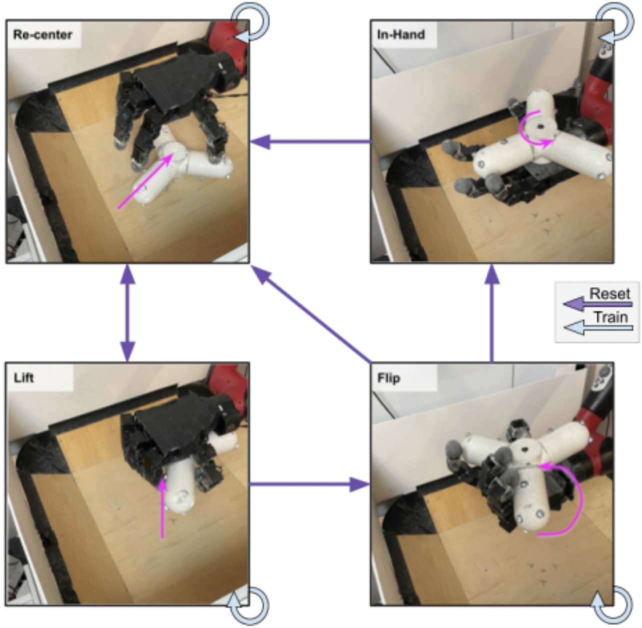


1. Initialize task-conditioned policy $\pi(\cdot \mid \mathbf{s}, \mathbf{z})$
2. Repeat:
    1. Propose task $\mathbf{z}_i$ to practice based on $\mathbf{s}$
    2. Roll out $\pi(\cdot \mid \mathbf{s}, \mathbf{z}_i)$ for H steps
    3. Update π using $r(\mathbf{s}, \mathbf{a}, \mathbf{z})$
    4. *No reset or very infrequent reset*

Test time: Deploy π

Example: autonomous RL with task cycles

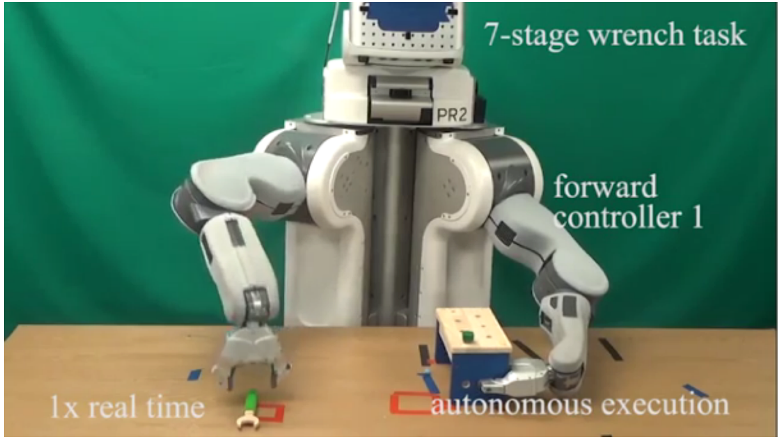




Gupta, Yu, Zhao, Kumar, Rovinsky, Xu, Devlin, Levine. **Reset-Free Reinforcement Learning via Multi-Task Learning: Learning Dexterous Manipulation Behaviors without Human Intervention**. 2021.

Example: autonomous RL with task cycles

Laura Smith, J. Chase Kew, Xue Bin Peng, Sehoon Ha, Jie Tan, Sergey Levine. **Legged Robots that Keep on Learning: Fine-Tuning Locomotion Policies in the Real World**. 2021.

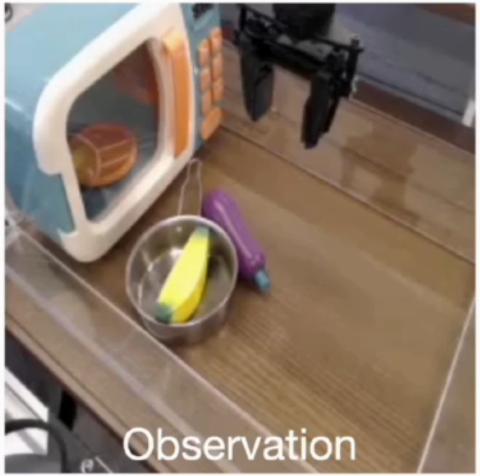


Han, Levine, Abbeel. **Learning Compound Multi-Step Controllers under Unknown Dynamics.**

How to identify which task to collect data for?

In general: an open research question

One idea: Ask a vision language model.



initial frame of a trajectory.

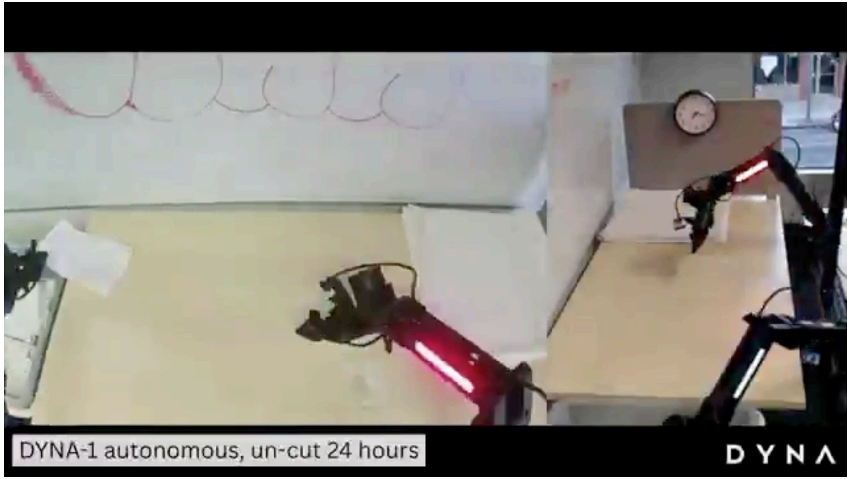
Zhou*, Atreya*, Lee, Walke, Mees, Levine. **SOAR: Autonomous Improvement of Instruction Following Skills via Foundation Models.**

Alternatives to Forward-Backward RL?

1. Learn to reset to expert state distribution (forward-backward with modified backward reward)

2. Practice cycle of different tasks

3. **Set up task that can be done repetitively (“forward-forward RL”?)**



Source: x.com/JasonMa2020/status/1917255026751553629

Plan for Today

- Why aren't robots autonomous already?
- Defining the problem: autonomous RL
- Developing the algorithms
    - Learning policies without human intervention
    - Single life RL

So far…

We've looked at learning policies without resets:
- Can improve the quality of the policy through *autonomous practice*

What happens after we deploy the policy?

Obviously, everything works perfectly and nothing ever goes wrong. ✓

So far…

We've looked at learning policies without resets:
- Can improve the quality of the policy through *autonomous practice*

What happens after we deploy the policy?

~~*Obviously*, everything works perfectly and nothing ever goes wrong.~~

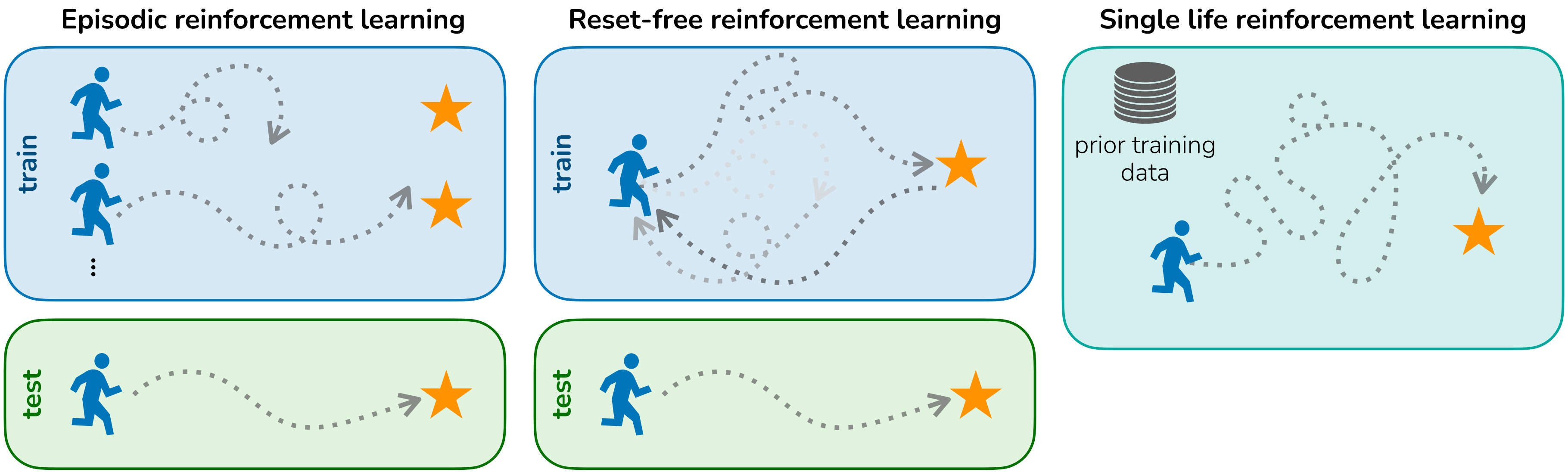
The natural world is complex, and something likely will go wrong, despite preparation.

Even humans make mistakes and adapt

Even humans make mistakes and adapt

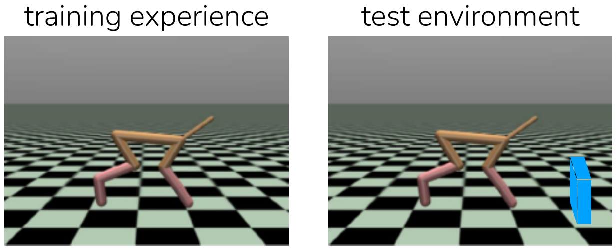
Even humans make mistakes and adapt

Single-Life Reinforcement Learning (SLRL)



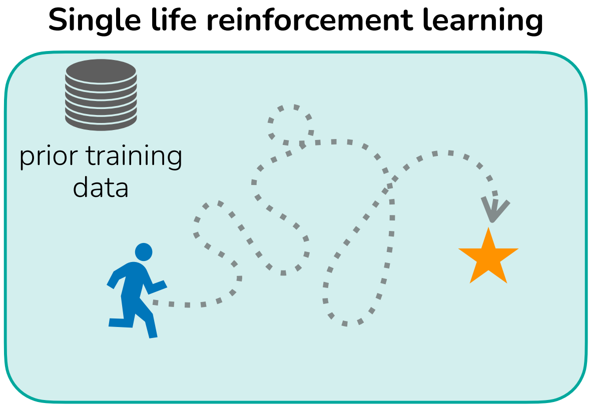
Given prior data in train env, agent has one “life” to autonomously complete task in novel scenario.

Can we run RL in a single episode at test time?

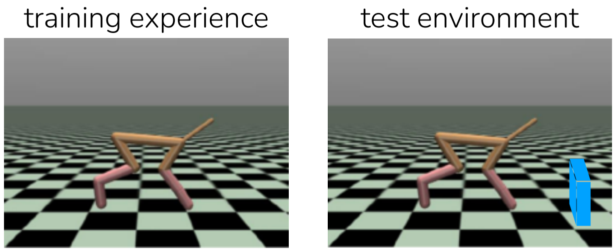


Chen, Sharma, Levine, Finn. *You Only Live Once: Single-Life Reinforcement Learning*, NeurIPS ‘22

Single-Life Reinforcement Learning (SLRL)



Given prior data in train env(s), agent has one "life" to autonomously complete task in **novel, OOD scenario**.



Can we fine-tune with RL in a single episode at test time?

How to approach single-life RL?

Can we fine-tune with vanilla reinforcement learning, but at test time?

Agent fails to handle or recover from out-of-distribution states without interventions.

Chen, Sharma, Levine, Finn. *You Only Live Once: Single-Life Reinforcement Learning*, NeurIPS '22

How to approach single-life RL?

A few useful ideas:

- Guide the learning process towards good, familiar states to avoid getting stuck
- Leverage pre-training
    - Pre-trained skills can allow for adaptation in *skill* space rather than in action space
    - Pre-trained LLMs may be able to reason through how to adapt

Example of guiding policy towards good states in prior data:

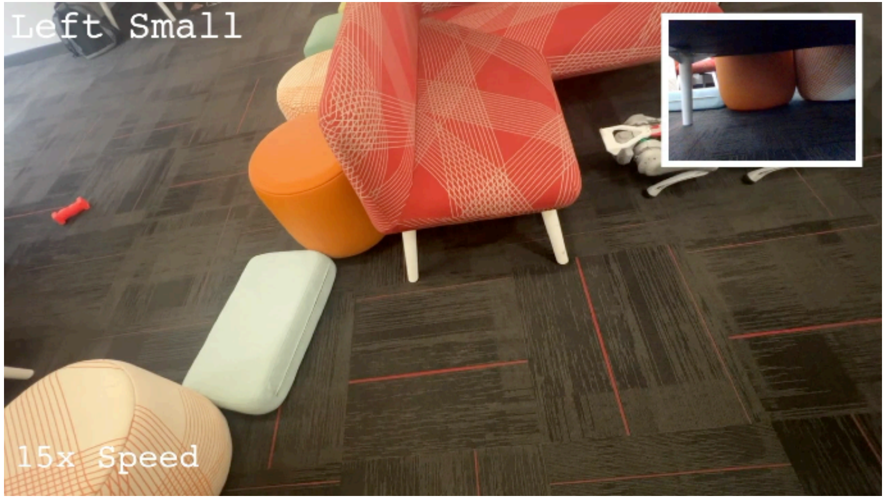
Chen, Sharma, Levine, Finn. *You Only Live Once: Single-Life Reinforcement Learning*, NeurIPS ‘22

How to approach single-life RL?

Adaptation in low-level action space

Kumar, Fu, Pathak, Malik. RMA: Rapid Motor Adaptation for Legged Robots. RSS 2021

Adaptation in high-level skill space



Chen, Lessing, Tang, Chada, Smith, Levine, Finn. Commonsense Reasoning for Legged Robot Adaptation with Vision-Language Models. ICRA 2025

Summary

- Why aren't robots autonomous already?
- Defining the problem: autonomous RL
- Developing the algorithms
    - Learning policies without human intervention
    - Single life RL

Reminders

Project milestone due tonight
Homework 4 due next Friday

Next time

Guest lecture from Ashish Kumar
(Tesla Optimus)