Learning Exploration Strategies via Meta Reinforcement Learning

Chelsea Finn

Why are humans good at RL?

People have previous experience.

They have developed **representations** that facilitate exploration & learning.

Our RL agents start tabula rasa.

Can we allow RL agents to leverage prior experience?

Source: https://i.imgur.com/hJlVfZ5.jpg

Should we be using the same exploration algorithm for:

- Learning to navigate an environment
- Learning to make recommendations to users
- Learning a policy for computer system caching
- Learning to physically operate a new tool or machine

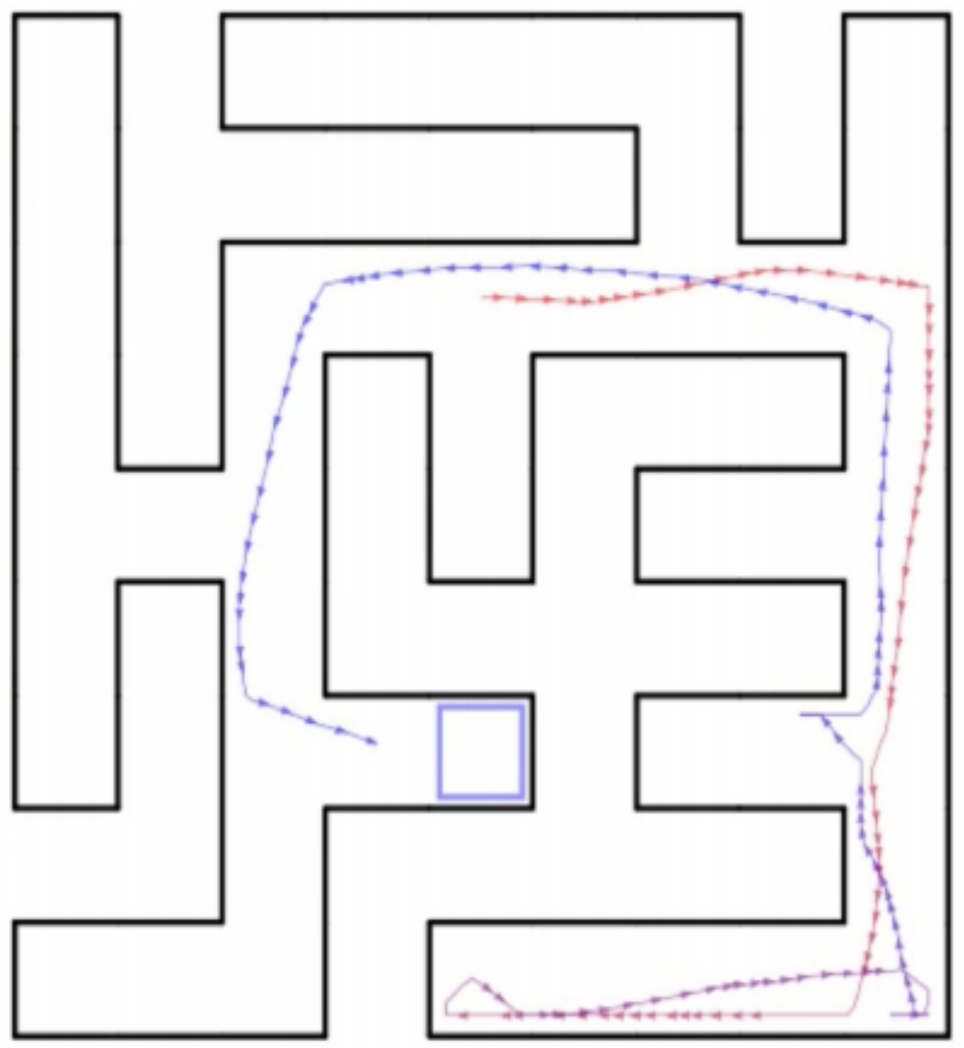
This is how we currently approach exploration.

Can we *learn exploration strategies* based on experience from other tasks in that domain?

A brief primer on meta-reinforcement learning

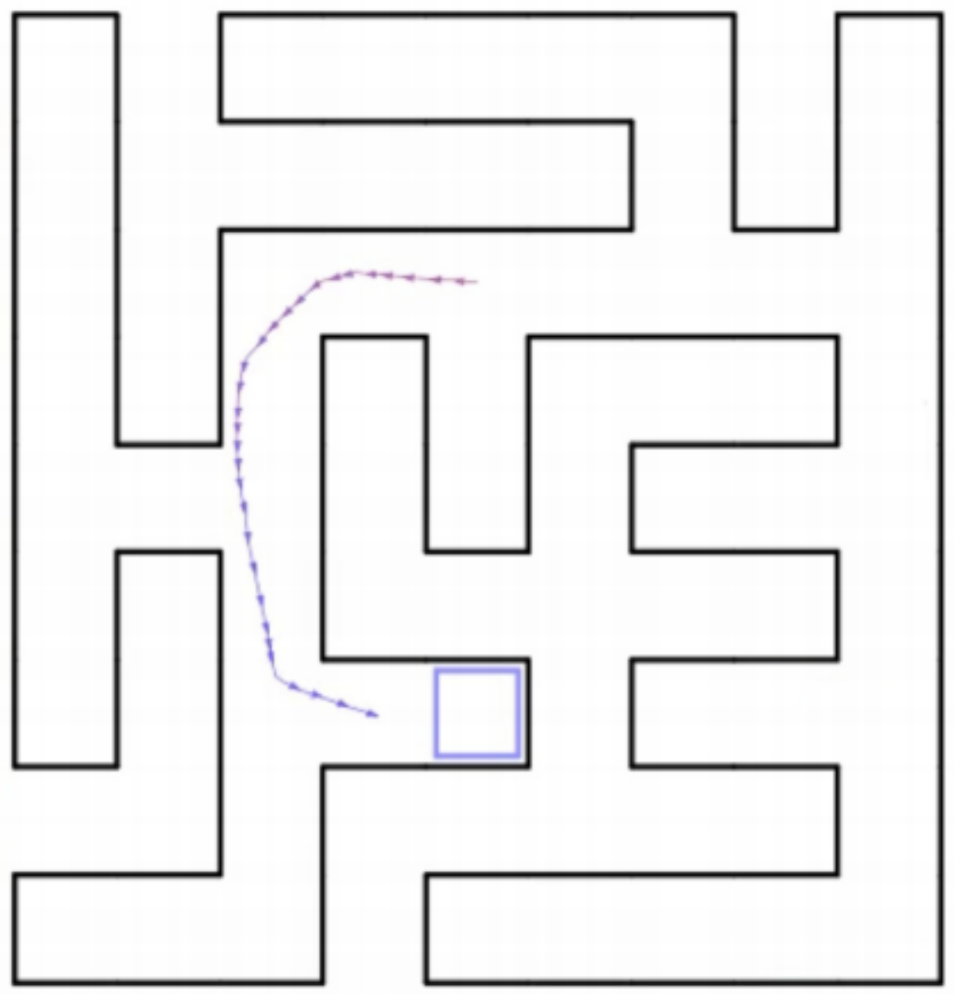
Goal:

Collect small amount of experience in new MDP



Collect $\mathscr{D}_{\mathrm{tr}} \sim \pi^{\mathrm{exp}}$

Learn policy that solves that MDP



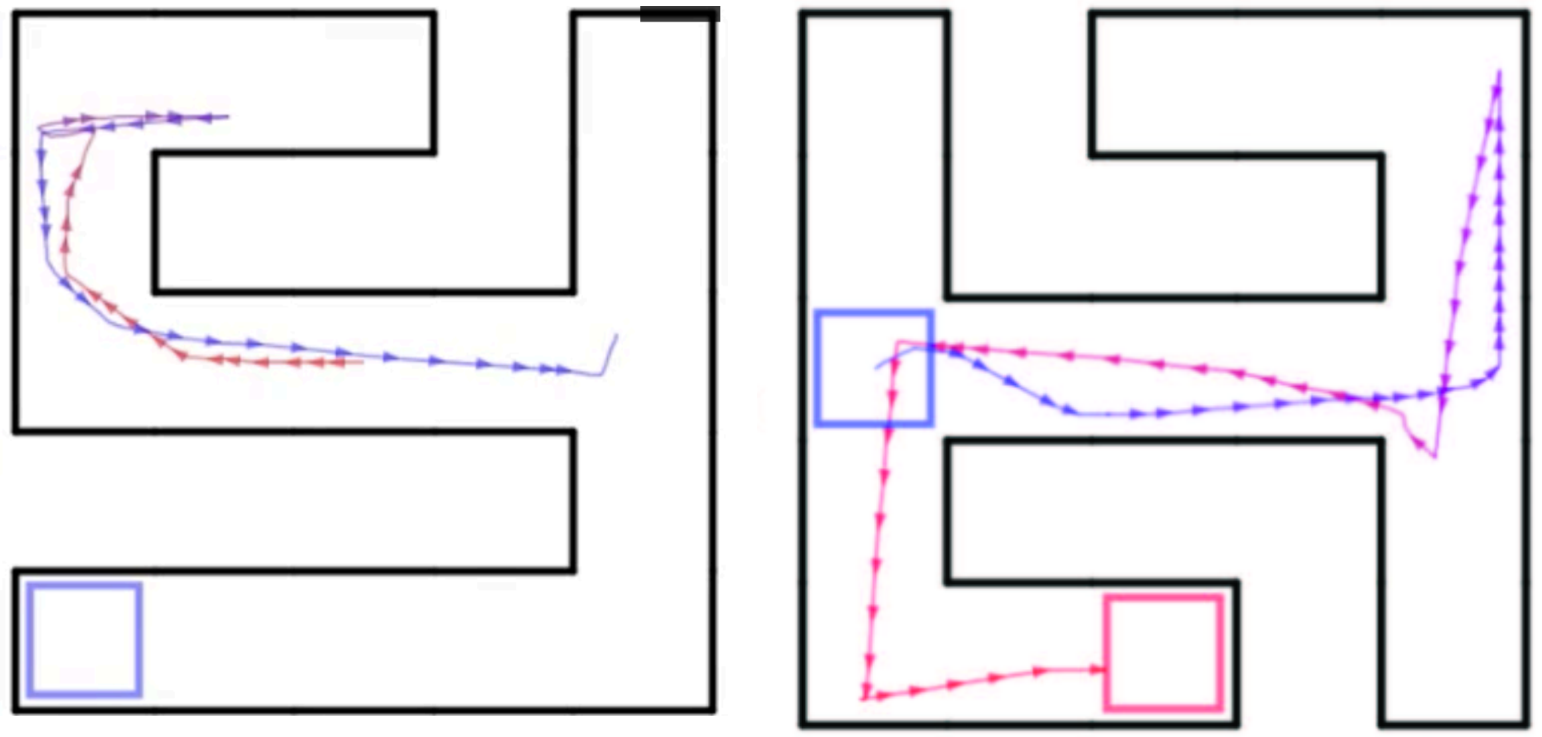
$\mathscr{D}_{\mathrm{tr}} \rightarrow \pi^{\mathrm{task}}$

diagram adapted from Duan et al. ‘17

A brief primer on meta-reinforcement learning

Meta-Train Time:

Learn how to efficiently explore & solve many MDPs:

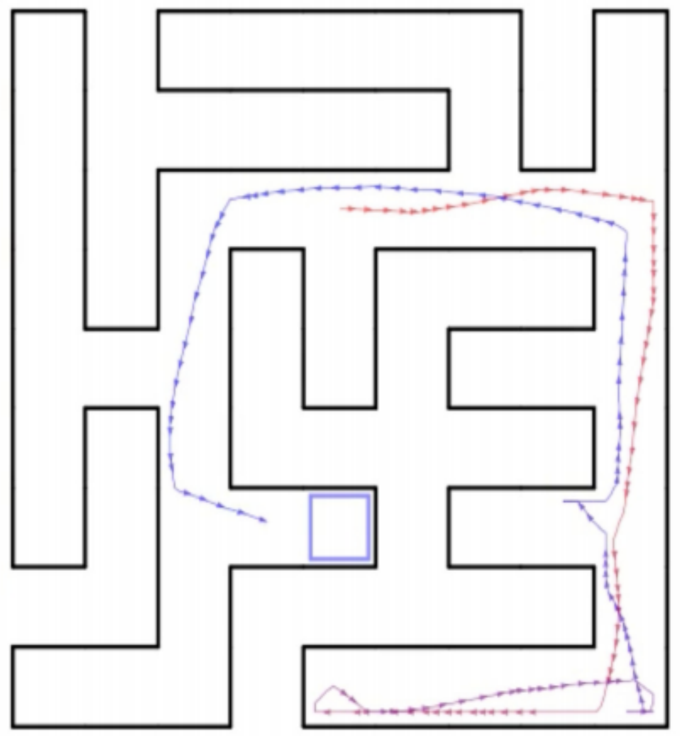


. . . meta-training tasks

Meta-train $\pi^{\text{exp}}, \pi^{\text{task}}$

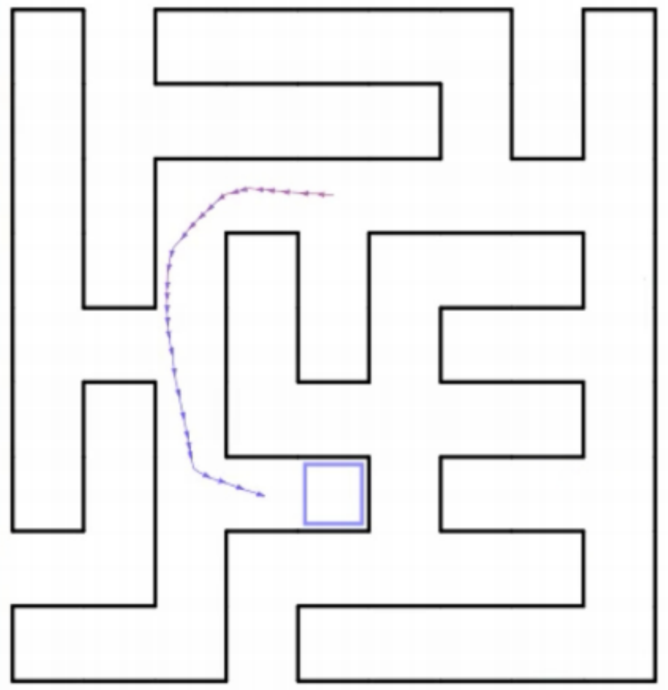
Meta-Test Time:

Collect small amount of experience in new MDP



Collect $\mathscr{D}_{\text{tr}} \sim \pi^{\text{exp}}$

Learn policy that solves that MDP



$\mathscr{D}_{\text{tr}} \rightarrow \pi^{\text{task}}$

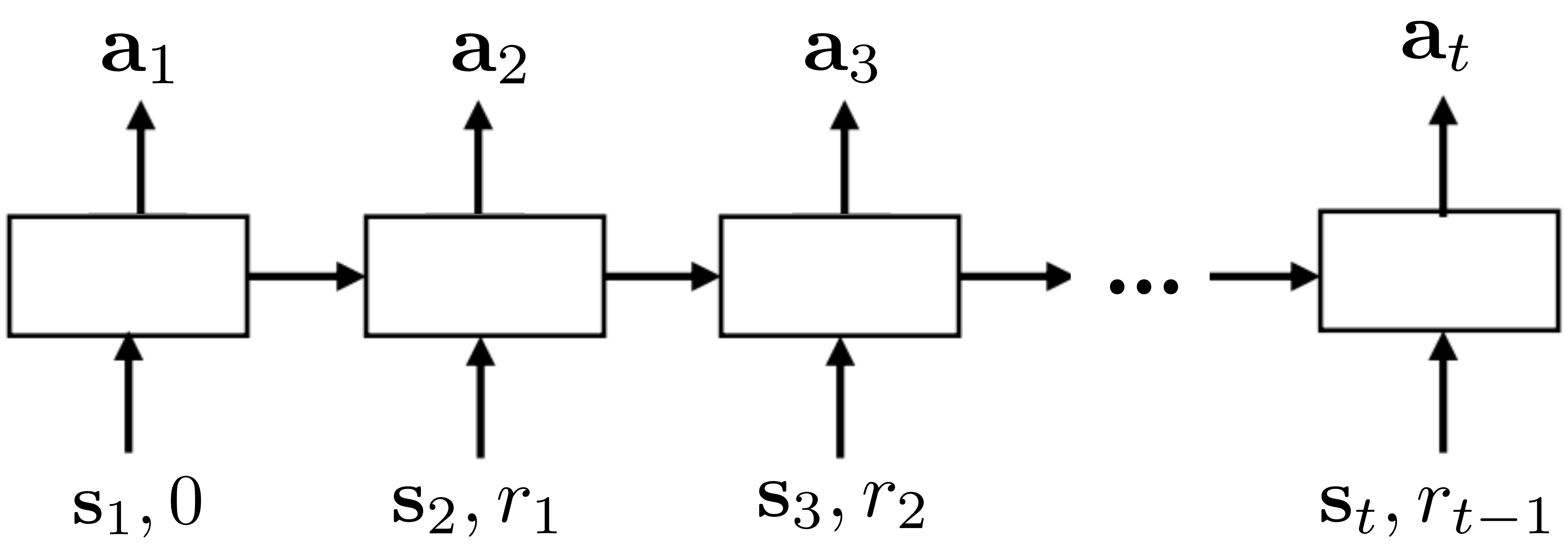
Key assumption: Meta-training & meta-testing MDPs come from same distribution.

(so that we can expect generalization)

diagram adapted from Duan et al. ‘17

A brief primer on meta-reinforcement learning

Common approach: Implement the learning procedure with a recurrent network.



Is this just a recurrent policy?

Hidden state maintained *across episodes* within a task!

Trained across a *family of MDPs* with varying dynamics, rewards.

Wang et al. Learning to Reinforcement Learn. 2017; Duan et al. RL^2. 2017

How Do We Learn to Explore?

Solution #1: Optimize for Exploration & Exploitation *End-to-End* w.r.t. Reward

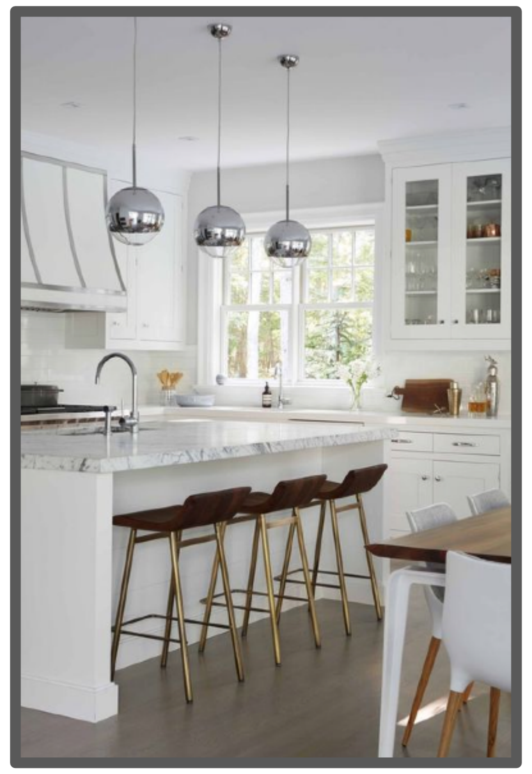
(Duan et al., 2016, Wang et al., 2016, Mishra et al., 2017, Stadie et al., 2018, Zintgraf et al., 2019, Kamienny et al., 2020)

+ simple
+ leads to optimal strategy in principle

-- challenging optimization when exploration is hard

Example of a Hard Exploration Meta-RL Problem

Learned cooking tasks in previous kitchens



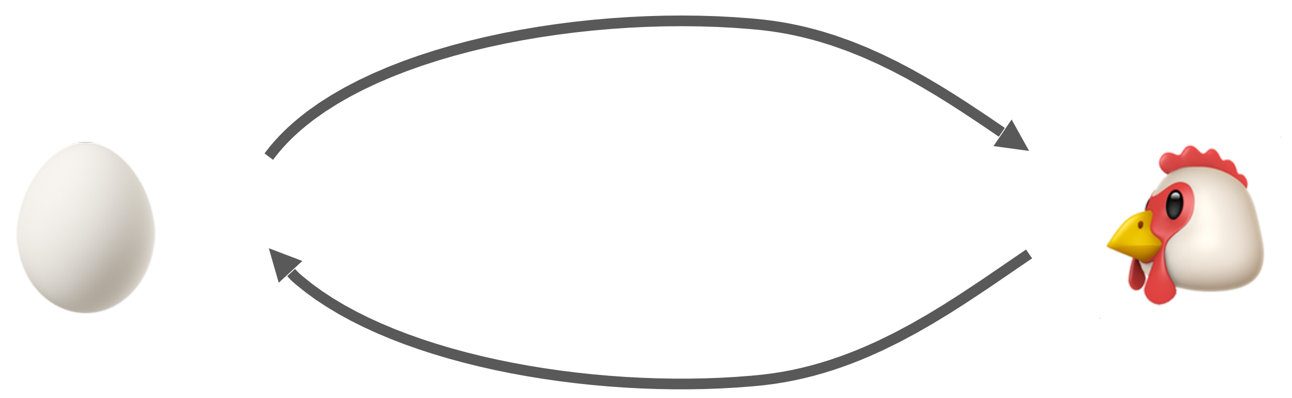
meta-training

Goal: Quickly learn tasks in a new kitchen.

meta-testing

Why is End-to-End Training Hard?

End-to-end approach: optimize exploration and execution episode behaviors end-to-end to maximize reward of execution



Learning to cook (execution)

Learning to find ingredients (exploration)

Ingredient not found (bad exploration) → Cannot learn to cook (bad execution)

Cannot cook (bad execution) → Low reward for **any** exploration

Coupling problem: learning exploration and execution depend on each other

—> can lead to poor local optima, poor sample efficiency

Liu, Raghunathan, Liang, Finn. *Explore then Execute: Adapting without Rewards via Factorized Meta-RL*. 2020

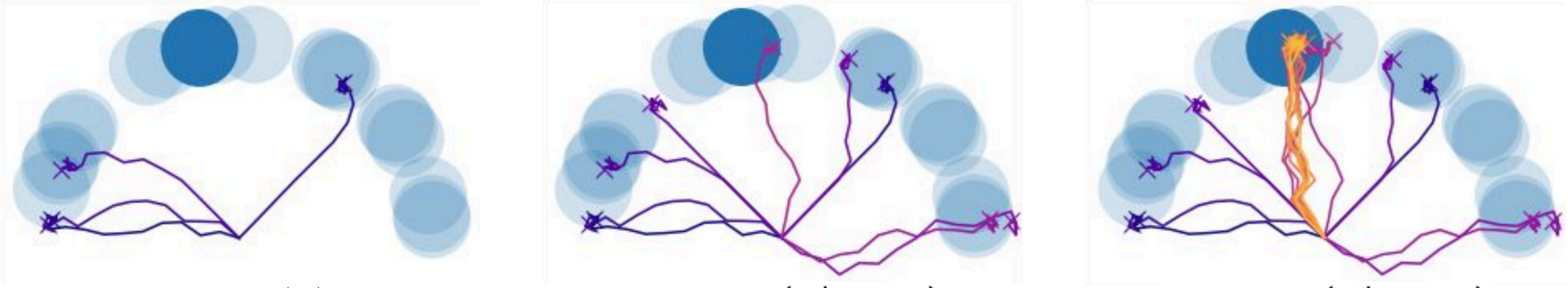
Solution #2: Leverage Alternative Exploration Strategies

1a. Use posterior sampling
(also called Thompson sampling)

PEARL (Rakelly, Zhou, Quillen, Finn, Levine. ICML '19)

i. Learn distribution over latent task variable $p(\mathbf{z}), q(\mathbf{z} \mid \mathscr{D}_{\text{tr}})$ and corresponding task policies $\pi(\mathbf{a} \mid \mathbf{s}, \mathbf{z})$

ii. Sample $\mathbf{z}$ from current *posterior* and sample from policy $\pi(\mathbf{a} \mid \mathbf{s}, \mathbf{z})$



$\mathbf{z} \sim p(\mathbf{z})$ $\mathbf{z} \sim q_\phi(\mathbf{z}|c_{1:10})$ $\mathbf{z} \sim q_\phi(\mathbf{z}|c_{1:30})$

When might posterior sampling be bad? Eg. Goals far away & sign on wall that tells you the correct goal.

Solution #2: Leverage Alternative Exploration Strategies

1a. Use posterior sampling (also called Thompson sampling) — PEARL (Rakelly, Zhou, Quillen, Finn, Levine. ICML '19)

i. Learn distribution over latent task variable $p(\mathbf{z}), q(\mathbf{z} \mid \mathcal{D}_{\text{tr}})$ and corresponding task policies $\pi(\mathbf{a} \mid \mathbf{s}, \mathbf{z})$

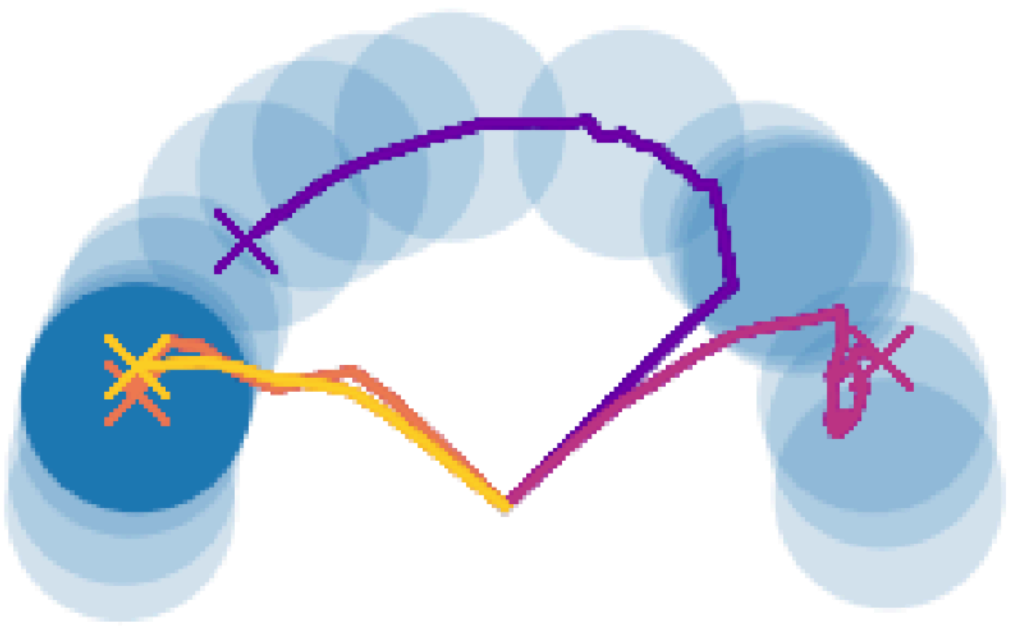
ii. Sample $\mathbf{z}$ from current *posterior* and sample from policy $\pi(\mathbf{a} \mid \mathbf{s}, \mathbf{z})$

1b. Use intrinsic rewards — MAME (Gurumurthy, Kumar, Sycara. CoRL '19)

1c. Task dynamics & reward prediction — MetaCURE (Zhang, Wang, Hu, Chen, Fan, Zhang. '20)

i. Train model $f(\mathbf{s}', r \mid \mathbf{s}, \mathbf{a}, \mathcal{D}_{\text{train}})$

ii. Collect $\mathcal{D}_{\text{train}}$ so that model is accurate.



When might this be bad?

Lots of distractors,
or complex, high-dim state dynamics

Solution #2: Leverage Alternative Exploration Strategies

1a. Use posterior sampling (also called Thompson sampling) — PEARL (Rakelly, Zhou, Quillen, Finn, Levine. ICML '19)

i. Learn distribution over latent task variable $p(\mathbf{z}), q(\mathbf{z} \mid \mathscr{D}_{\text{tr}})$ and corresponding task policies $\pi(\mathbf{a} \mid \mathbf{s}, \mathbf{z})$

ii. Sample $\mathbf{z}$ from current *posterior* and sample from policy $\pi(\mathbf{a} \mid \mathbf{s}, \mathbf{z})$

1b. Use intrinsic rewards — MAME (Gurumurthy, Kumar, Sycara. CoRL '19)

1c. Task dynamics & reward prediction — MetaCURE (Zhang, Wang, Hu, Chen, Fan, Zhang. '20)

i. Train model $f(\mathbf{s}', r \mid \mathbf{s}, \mathbf{a}, \mathscr{D}_{\text{train}})$

ii. Collect $\mathscr{D}_{\text{train}}$ so that model is accurate.

+ easy to optimize
+ many based on principled strategies

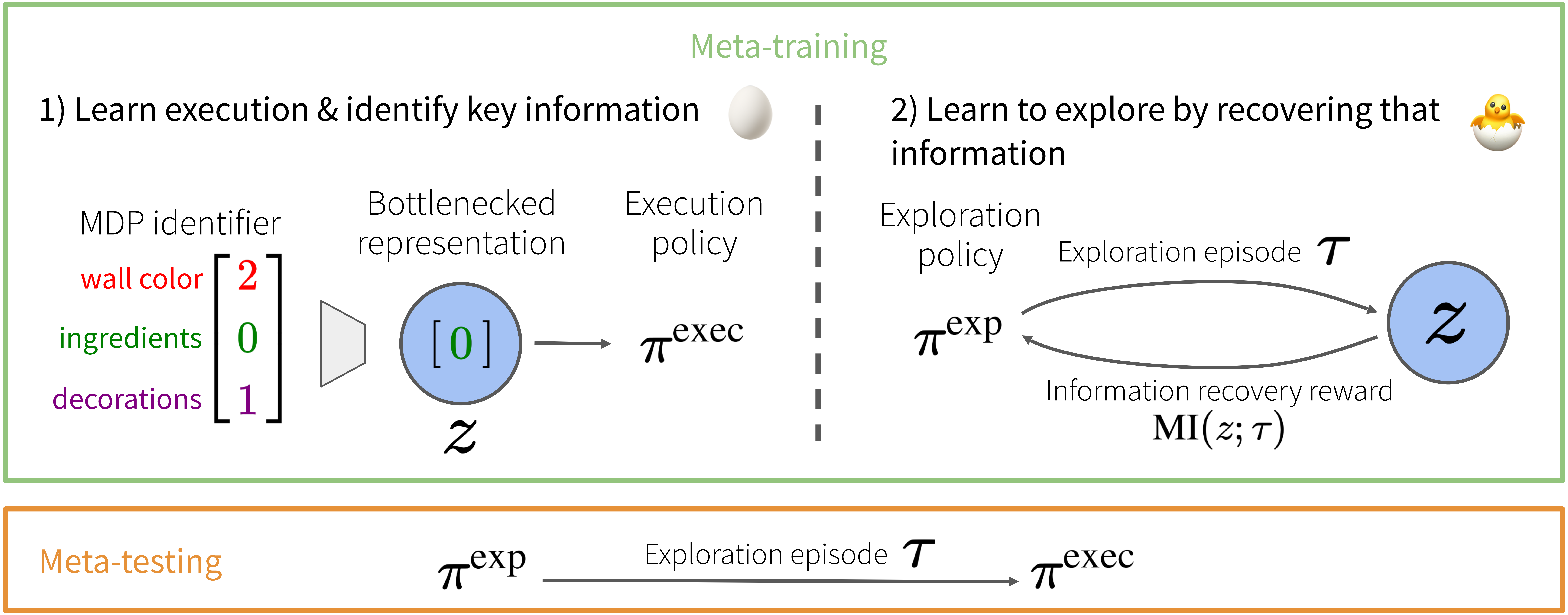
-- suboptimal by arbitrarily large amount in some environments.

Can we avoid the chicken-and-egg problem without sacrificing optimality?

Yes!

Evan Z. Liu

Solution #3: Decouple by acquiring representation of task relevant information



Decoupled **R**eward-free **E**xplor**A**tion and Execution in **M**eta-Reinforcement Learning (DREAM)

Liu, Raghunathan, Liang, Finn. *Explore then Execute: Adapting without Rewards via Factorized Meta-RL*. 2020

Solution #3: Decouple by acquiring representation of task relevant information

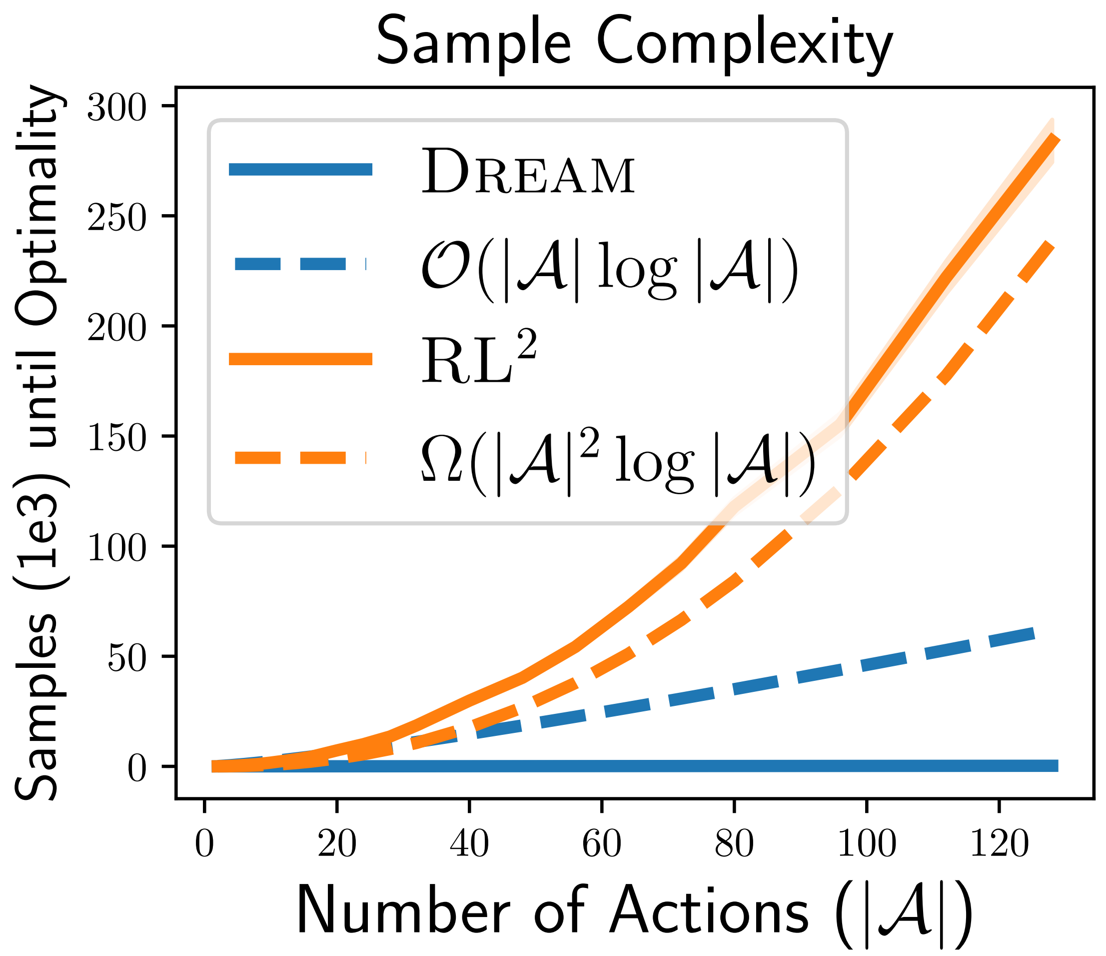
(Informal) Theoretical Results

(1) DREAM objective is ***consistent*** with end-to-end optimization. [under mild assumptions]

-> can in principle recover the optimal exploration strategy

(2) Consider a bandit-like setting with $|\mathcal{A}|$ arms.

In MDP i, arm i yields reward. In all MDPs, arm 0 reveals the rewarding arm.



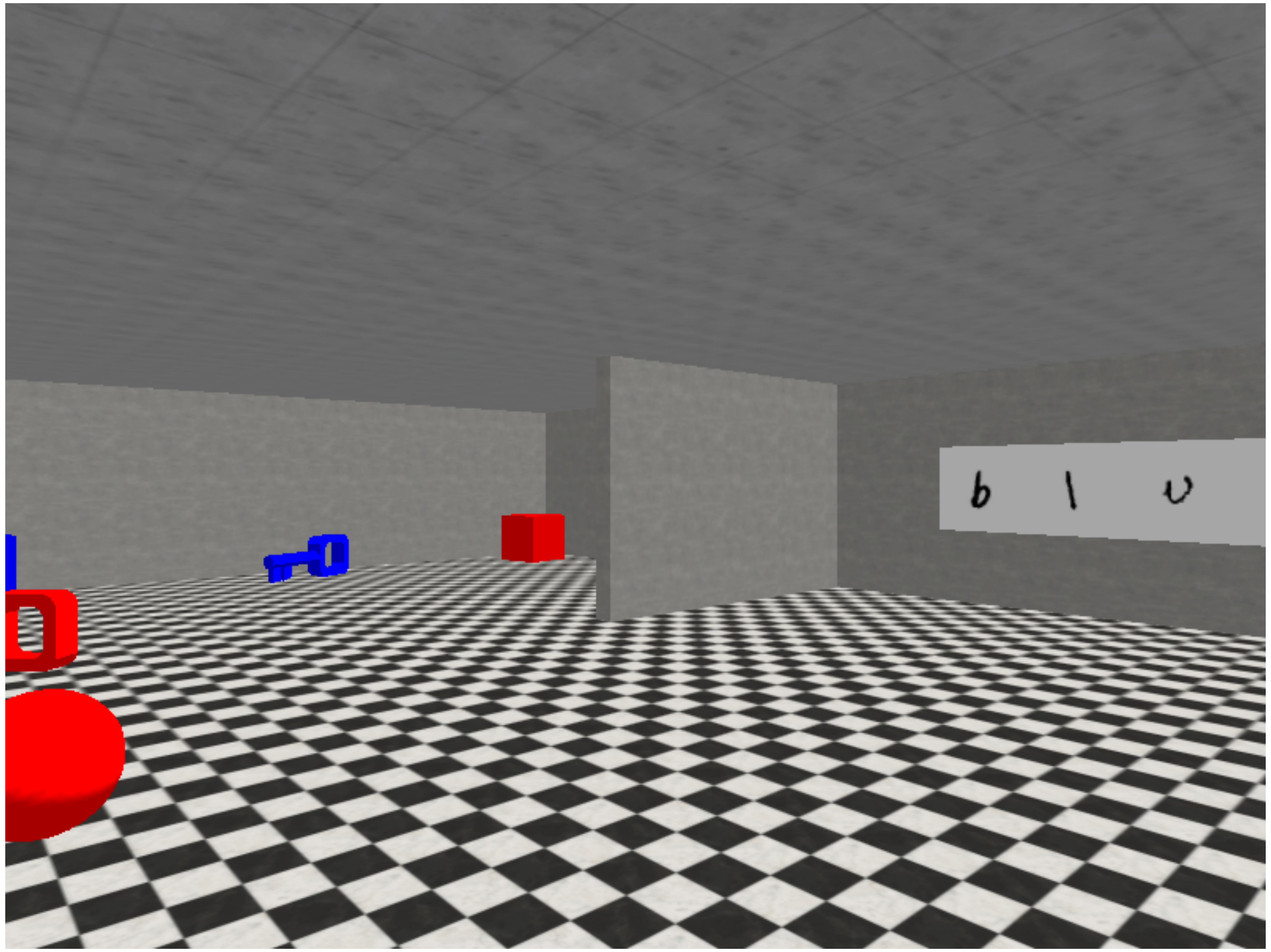
RL^2 requires $\Omega(|\mathcal{A}|^2 \log |\mathcal{A}|)$ samples for meta-optimization.

DREAM requires $\mathcal{O}(|\mathcal{A}| \log |\mathcal{A}|)$ samples for meta-optimization.

[assuming Q-learning with uniform outer-loop exploration]

Liu, Raghunathan, Liang, Finn. ***Explore then Execute: Adapting without Rewards via Factorized Meta-RL***. 2020

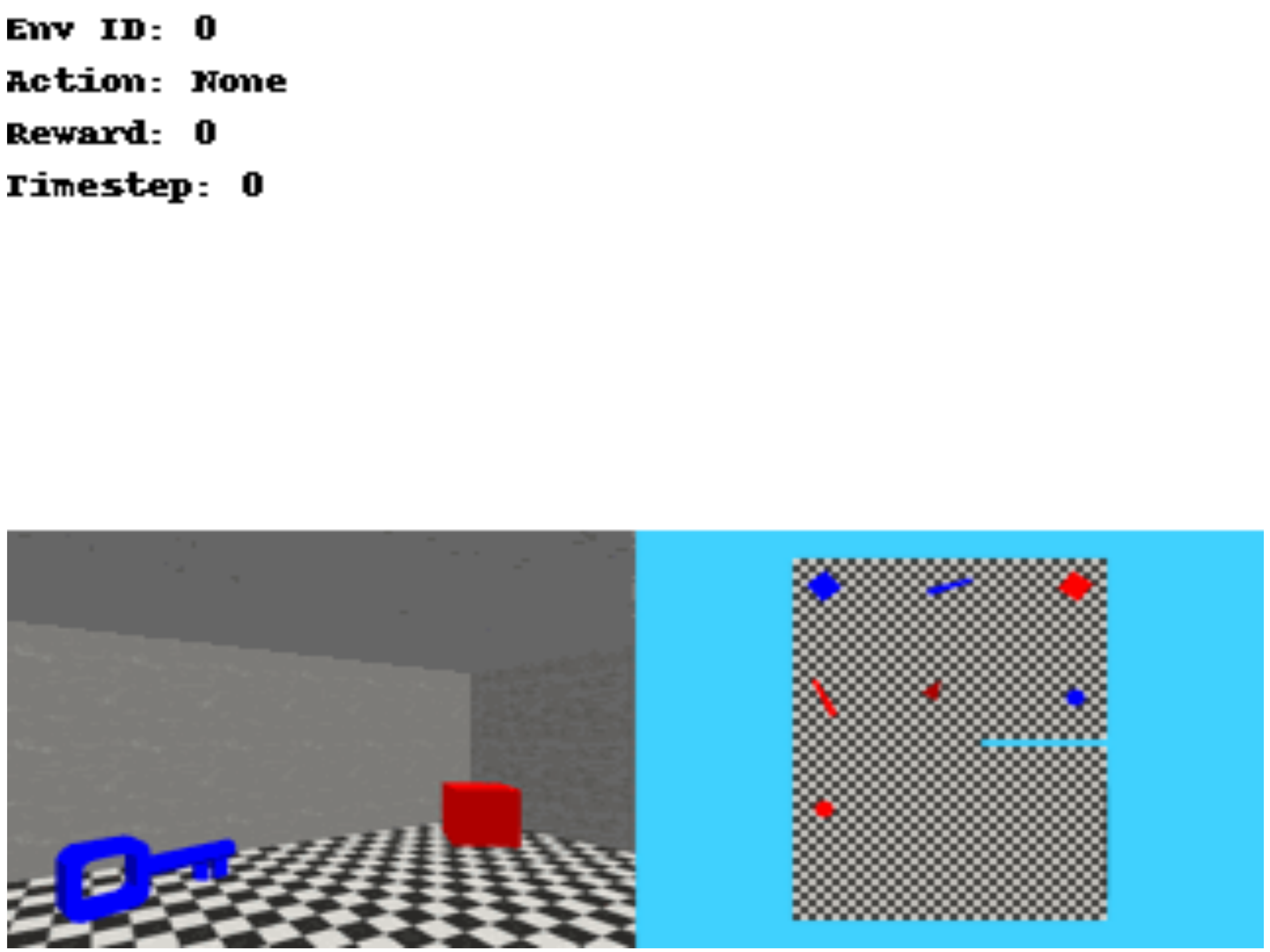
Empirical Results: Sparse Reward 3D Visual Navigation Problem



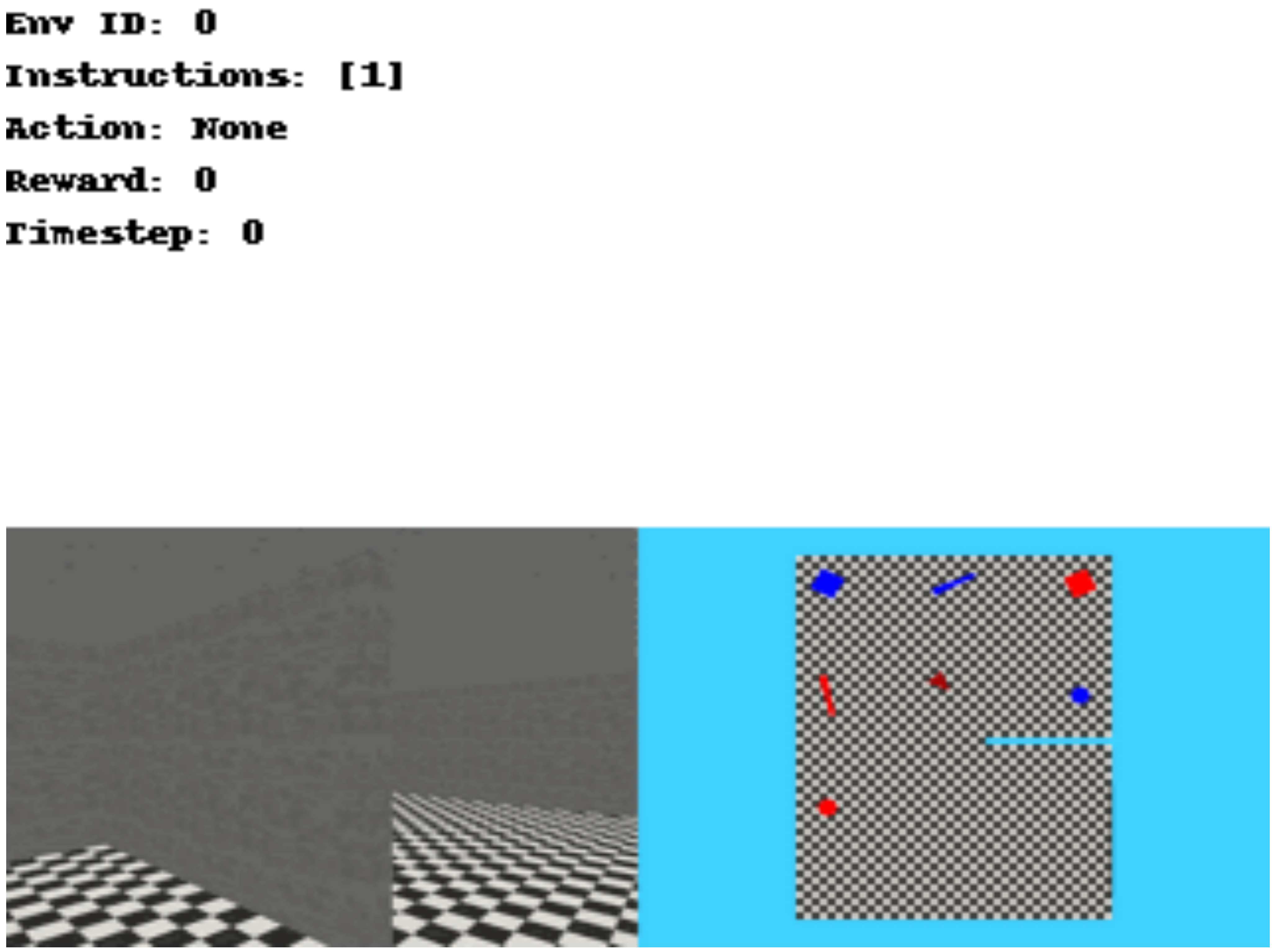
More challenging variant of task from Kamienny et al., 2020

- Task: go to the (key / block / ball), color specified by the sign
- Agent starts on other side of barrier, must walk around to read the sign
- Pixels observations (80 x 60 RGB)
- Sparse binary reward

Qualitative Results for DREAM



Exploration episode

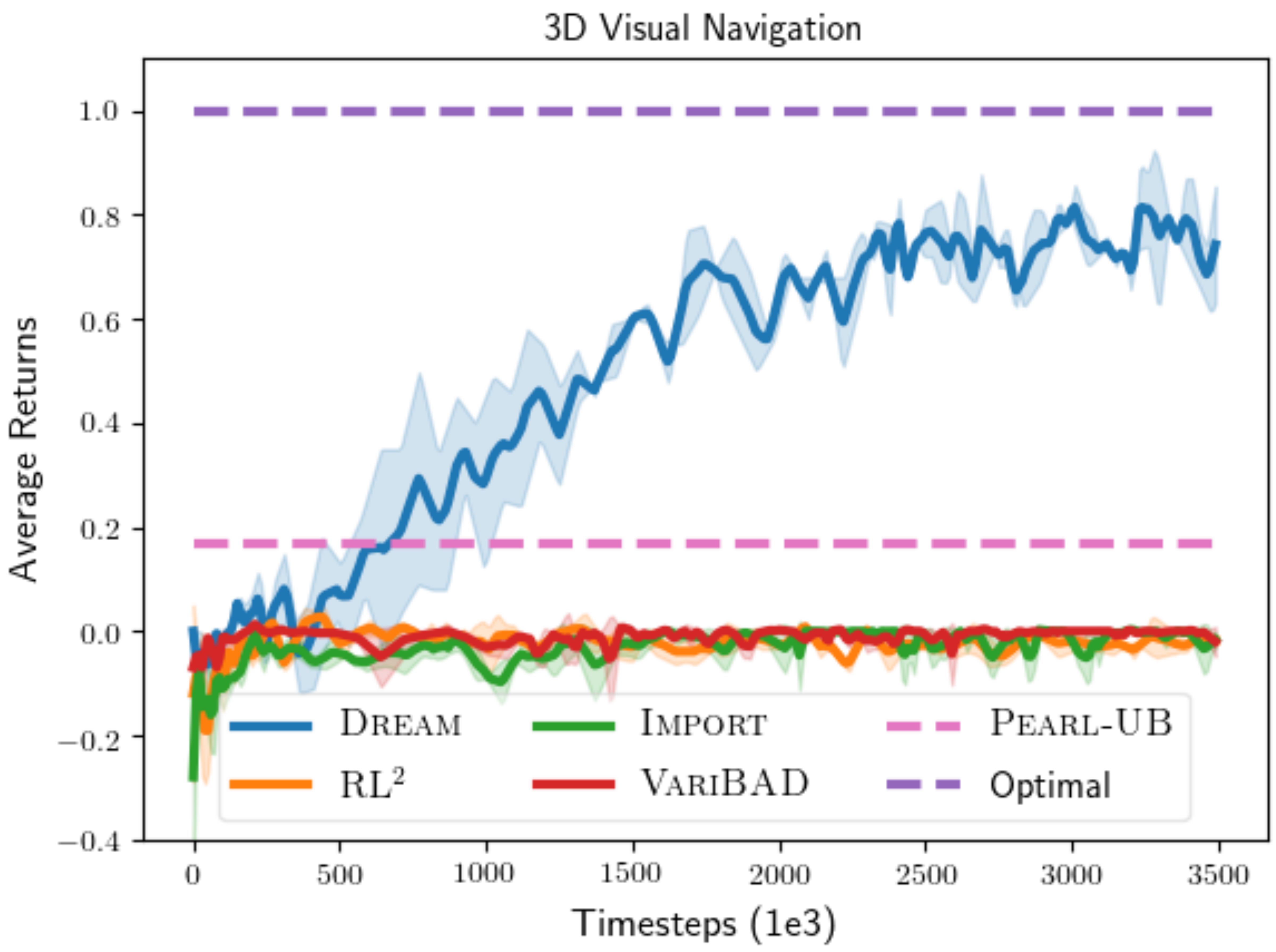


Execution episode

Task: Go to key

Liu, Raghunathan, Liang, Finn. *Explore then Execute: Adapting without Rewards via Factorized Meta-RL*. 2020

Quantitative Results



- DREAM achieves near-optimal reward
- Existing state-of-the-art algorithms perform poorly due to **coupling**
- Alternate exploration strategies, e.g., Thompson Sampling do not learn the optimal exploration strategy
- PEARL-UB: Upper-bound on PEARL, reward achieved with optimal policy and Thompson-Sampling exploration

RL² (Duan et al., 2016), IMPORT (Kamienny et al., 2020), VARIBAD (Zintgraf et al., 2019), PEARL (Rakelly, et. al., 2019), Thompson, 1933

Liu, Raghunathan, Liang, Finn. *Explore then Execute: Adapting without Rewards via Factorized Meta-RL*. 2020

How Do We Learn to Explore?

End-to-End

+ leads to optimal strategy in principle

-- challenging optimization when exploration is hard

Alternative Strategies

+ easy to optimize
+ many based on principled strategies

-- suboptimal by arbitrarily large amount in some environments.

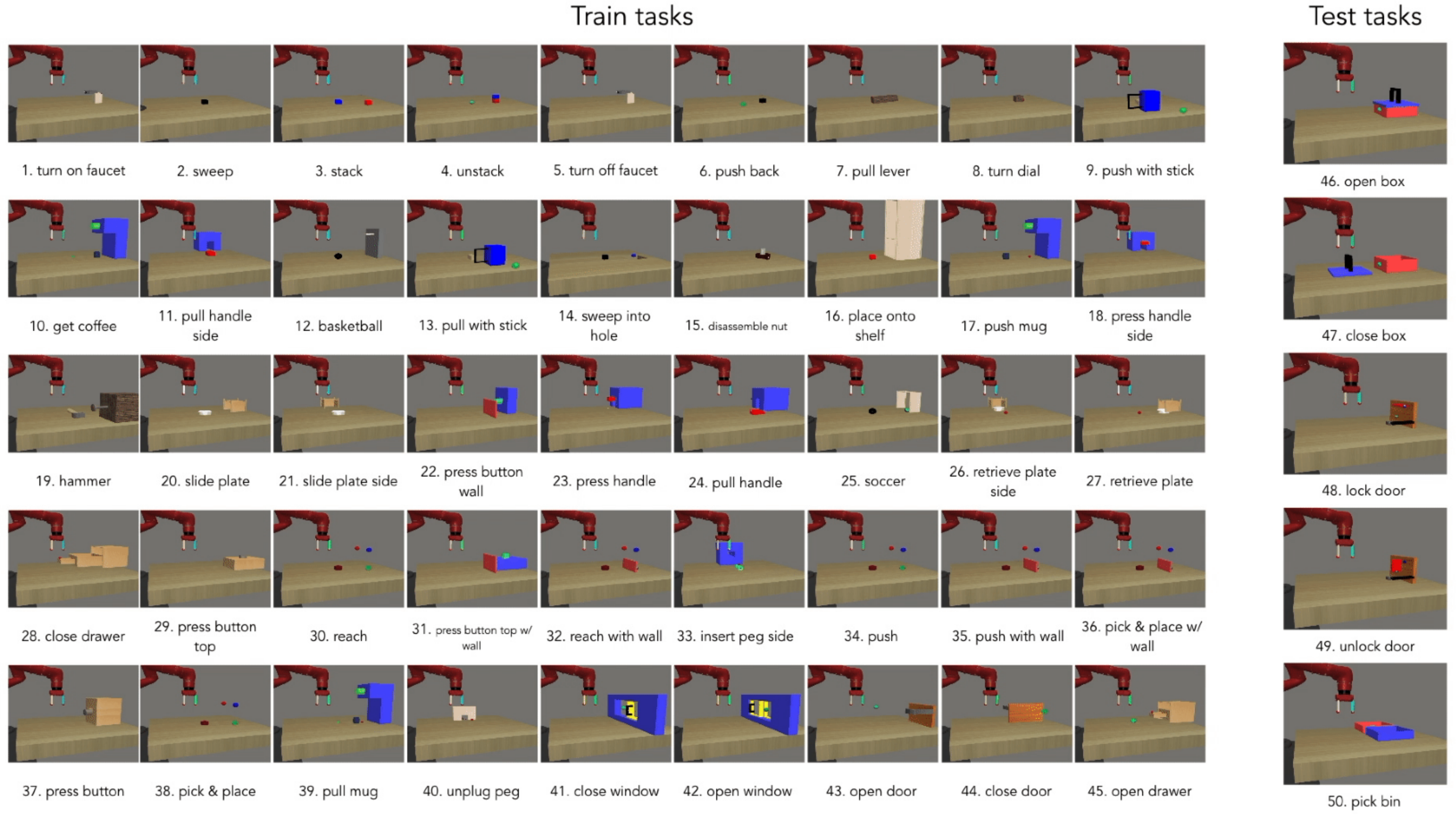
Decoupled Exploration & Execution

+ leads to optimal strategy in principle
+ easy to optimize in practice

-- requires task identifier

Other Challenges in Meta-Reinforcement Learning

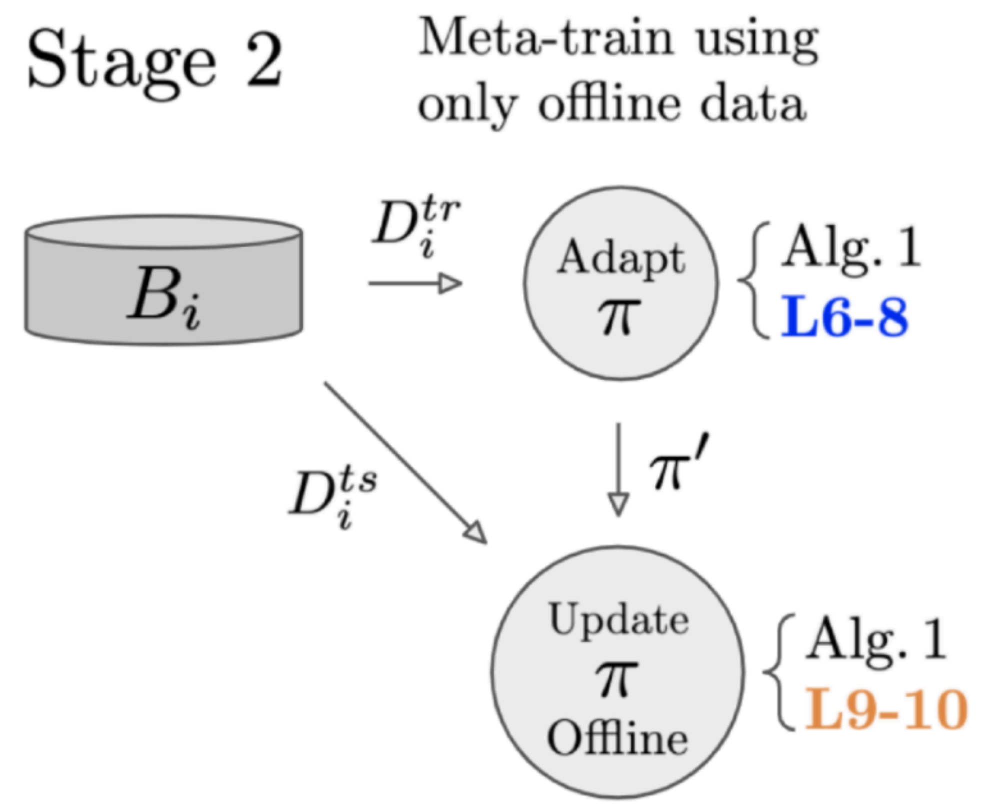
Handling Broad Task Distributions



T Yu, D Quillen, Z He, R Julian, K Hausman, C Finn, S Levine. ***Meta-World***. CoRL '19

Meta-RL from Offline Multi-Task Data

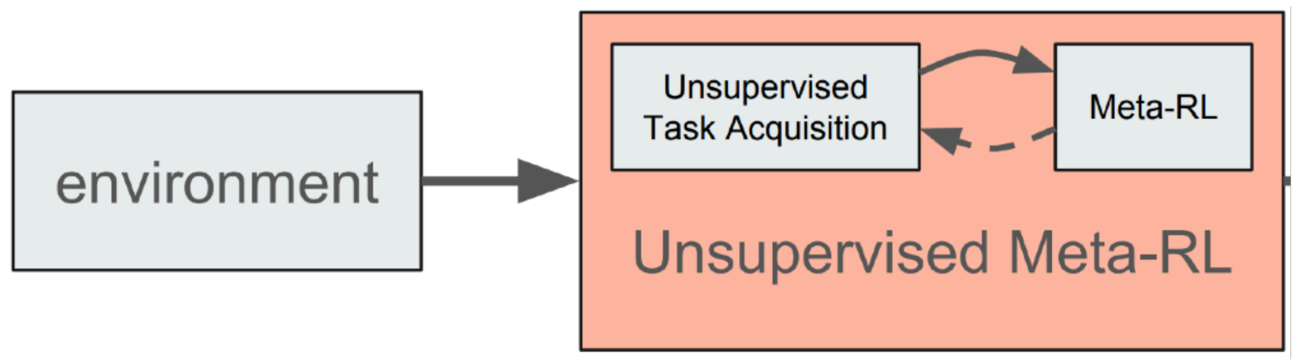
Initial work:



Mitchell, Rafailov, Peng, Levine, Finn. ***Offline Meta-RL with Advantage Weighting***. arXiv '20

Unsupervised Meta-RL

Meta-RL over discovered skills



Gupta, Eysenbach, Finn, Levine. ***Unsupervised Meta-Learning for Reinforcement Learning***. '18

Jabri, Hsu, Eysenbach, Gupta, Levine, Finn. ***Unsupervised Curricula for Visual Meta-Reinforcement Learning***. *NeurIPS '19*

Want to learn more?

Stanford CS330: Deep Multi-Task and Meta Learning
cs330.stanford.edu
All lecture videos online!

Questions?