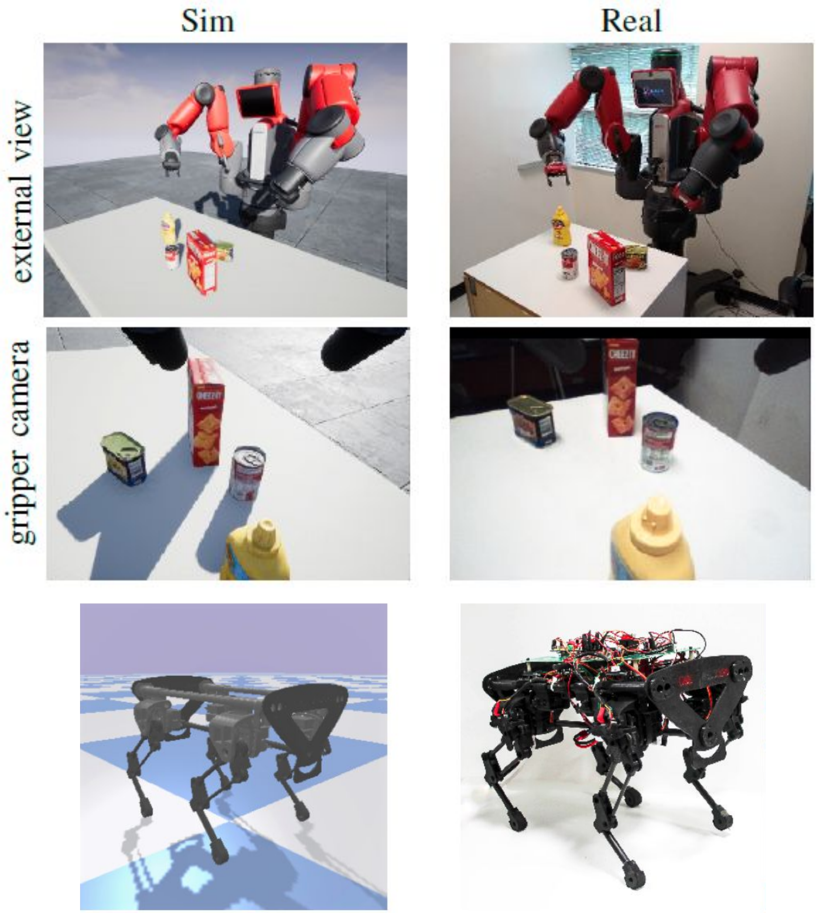
Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning

Presenter: Jake Grigsby

11/1/2022

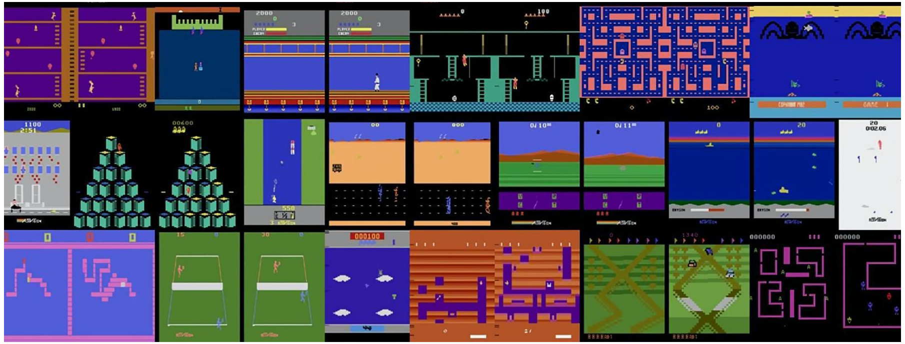
Motivation: RL Generalization

- Training Deep RL algorithms takes millions of timesteps **per task**
- We want to use one policy to solve **multiple tasks**
- We also want to be able to adapt to slight changes in the environment
    - Key special-case in robotics: **sim2real transfer** [1]
    - Different degrees of the same core problem



Motivation: Multi-task RL and Generalization

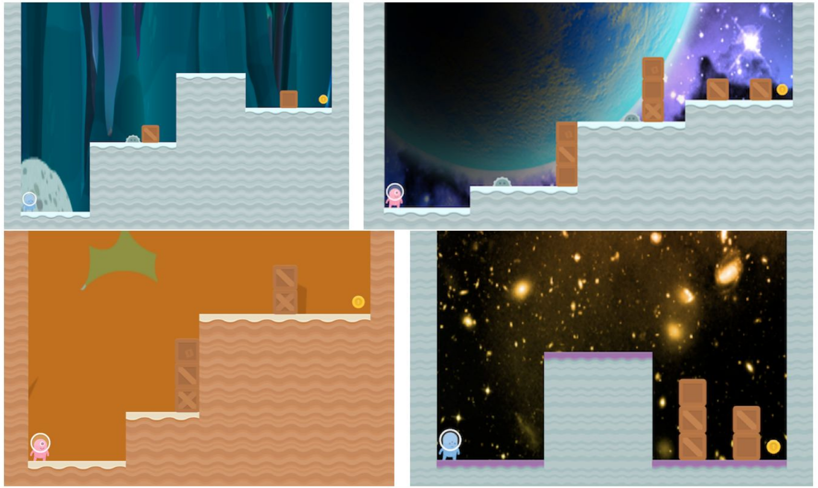
- "Generalization" initially focused on applying **one algorithm to multiple tasks <u>independently</u>**
  - E.g, 1 set of DQN hyperparameters, 57 Atari games [2] [3]



- Atari games are too distinct for positive transfer→ instead try different levels of the same game [4]

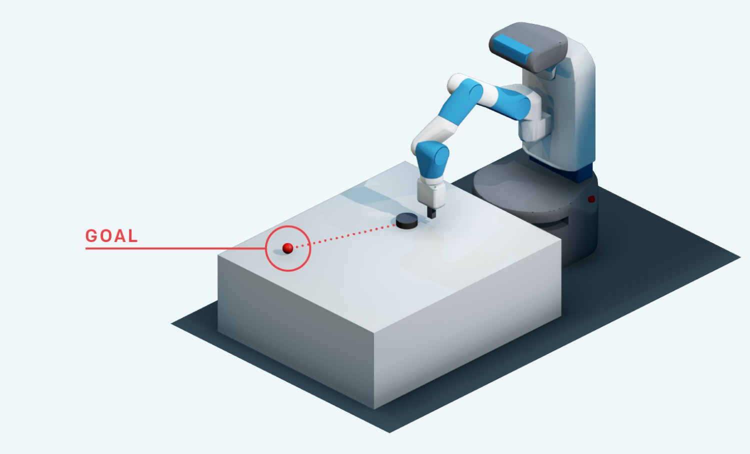
Motivation: Multi-task RL and Generalization

- ➢ Supervised learning sometimes needs millions of images or text fragments
- ➢ How many different "levels"/tasks does RL need to generalize?
    - ○ We can find out by generating near-infinite variations of the same environment [5] [6]



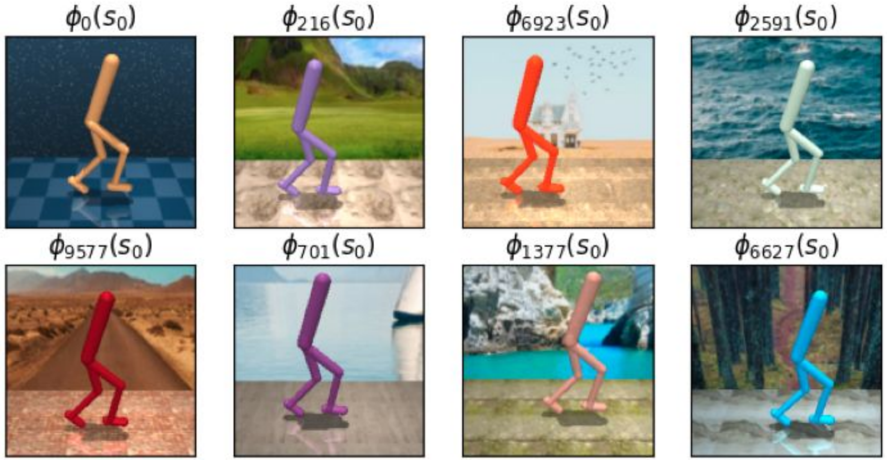
Motivation: Multi-task RL and Generalization

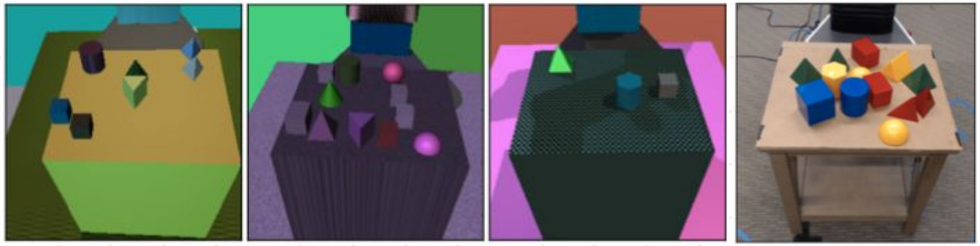
- Supervised learning sometimes needs millions of images or text fragments
- How many different “levels”/tasks does RL need to generalize?
  - We can find out by generating near-infinite variations of the same environment
  - Robotics examples: **manipulation environments with random object locations** [7]



Motivation: Multi-task RL and Generalization

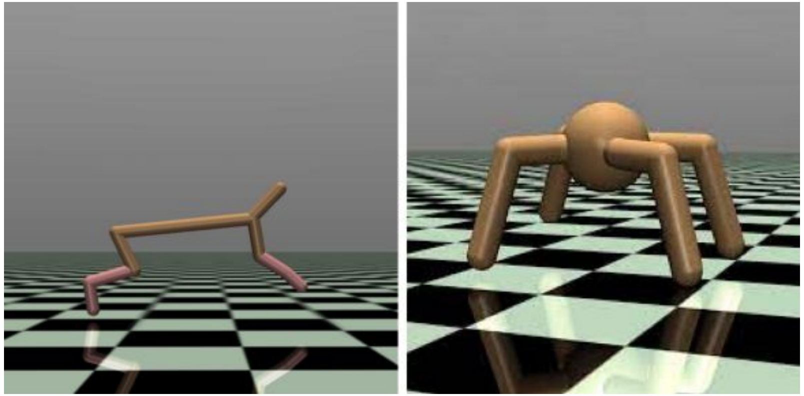
- Procedural generation for diverse task collections is a common theme [8]
  - We've seen one example already with dexterous hand sim2real [9]
  - Especially for visual generalization, where **graphics are easily randomized** [10] [11] [12]





Motivation: Multi-task RL and Generalization

- ➢ Examples so far have leaned towards easily visualized differences
- ➢ But variations in reward functions, goals, and dynamics are also studied [13]
  - ○ Especially reward function changes in classic gym envs [14]



The Spectrum of RL Generalization

Completely Distinct Tasks

Shared Structure With Varied Elements

Minor Variations in Dynamics

The Spectrum of RL Generalization

Completely Distinct Tasks

Shared Structure With Varied Elements

Minor Variations in Dynamics

E.g., 57 Atari Games

(Different robots, levels of mobility)

E.g., Randomly Generated Levels, Reward Functions

(Different goal locations, table-top manipulation tasks, same robot)

E.g., anti-overfitting benchmarks

(Different lighting, object colors, friction)

The Spectrum of RL Generalization

Completely Distinct Tasks

E.g., 57 Atari Games

(Different robots, levels of mobility)

Shared Structure With Varied Elements

E.g., Randomly Generated Levels, Reward Functions

(Different goal locations, table-top manipulation tasks, same robot)

Minor Variations in Dynamics

E.g., anti-overfitting benchmarks

(Different lighting, object colors, friction)

Meta-World

Meta-World

Meta-World provides a suite of table-top manipulation tasks with the same robot arm

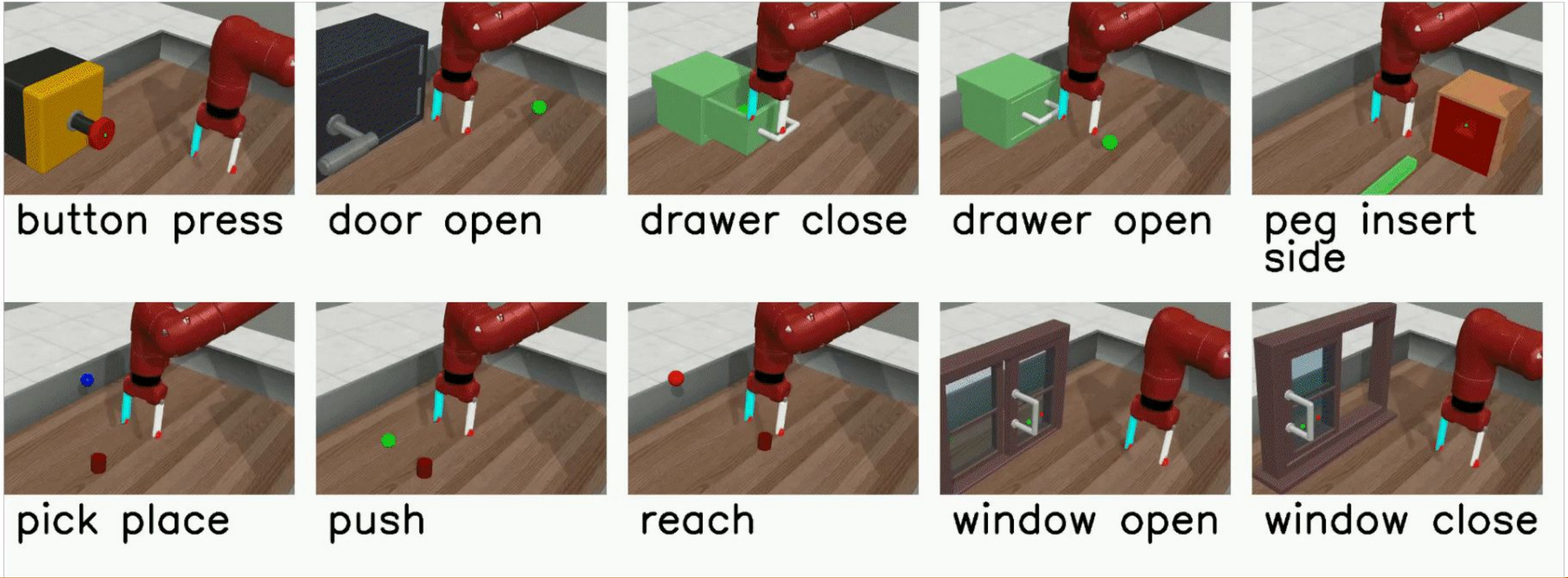
→ (same state and action space)

The range of tasks is formalized by the task distribution $p(\mathcal{T})$, where each task $\mathcal{T}$ is defined by its:

- reward function $R_{\mathcal{T}}(s, a)$
- Initial state s_0
- Goal g

Meta-World

The set of 50 distinct manipulation tasks creates **non-parametric** variation



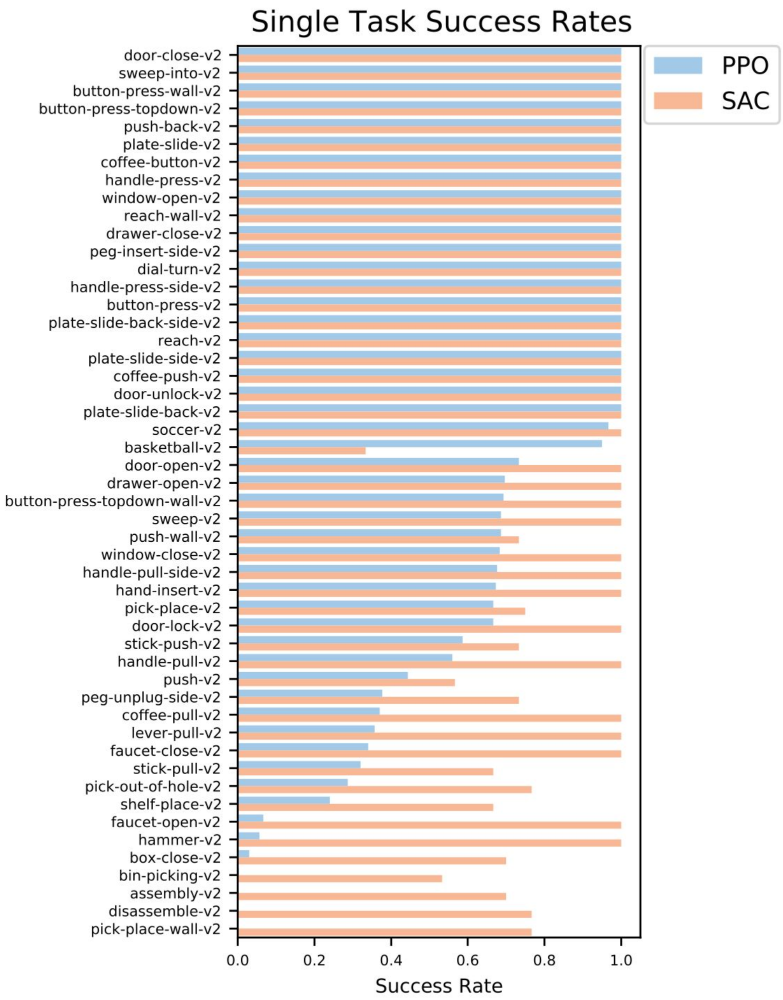
Meta-World

Meta-World creates **parametric** variation by sampling from a distribution over initial states ($p_{\mathcal{T}}(s_0)$) and goals ($p_{\mathcal{T}}(g)$)

...

Meta-World

- ➢ First step is to show every task is solvable individually
  - ○ Requires dense (hand-engineered) reward function
  - ○ Reward scale varies by task so we compare based on binary "success" metric
- ➢ Single-Task SAC and PPO
  - ○ Train on parametric variation with goal provided
  - ○ Can succeed on **at least 50% of <u>goals per task</u>**
    - ■ Slightly inconsistent vocab here



Multi-Task vs. Meta-Learning

Multi-Task Learning: tell the policy which task we are solving

- ➢ one-hot encoding of non-parametric task ID
- ➢ array of parametric goal information
- ➢ connections to goal-conditioned RL [15]

Multi-Task vs. Meta-Learning

Meta-Learning: the policy needs to *discover* which task we are solving

Two main categories of approaches:

1. *Optimization-based* methods quickly finetune on the current task with gradient updates
    a. MAML [16] and its many variants

2. *Context-based* methods infer the current task by remembering all the past attempts

Context-Based Meta-Learning

Informally: *figure out the task by looking at everything we've tried and all the rewards we've received*

→ *See what worked and what didn't and avoid past mistakes*

Full task trajectory (ignoring episode resets) up until time t:

$$\tau_{:t} \coloneqq (s_0, a_0, r_0, d_0, s_1, a_1, r_1, d_1, \dots, s_{t-1}, a_{t-1}, r_{t-1}, d_{t-1}, s_t)$$

Learn a trajectory-conditioned policy to maximize multi-episode return

$$\pi(a \mid s) \to \pi(a \mid \tau_{:t})$$

Context-Based Meta-Learning

- Simplest and earliest implementations are **RL^2** [17] and **L2RL** [18]
  - On-policy policy gradient RNNs that roll through episode boundaries

- More complex variants include **PEARL** [19] and **variBAD** [20]
  - Better ways to drive exploration and model how uncertain we are of the current task
  - For more formal reading: check out connections between Meta-Learning and CMDPs / BAMDPs [21][22]

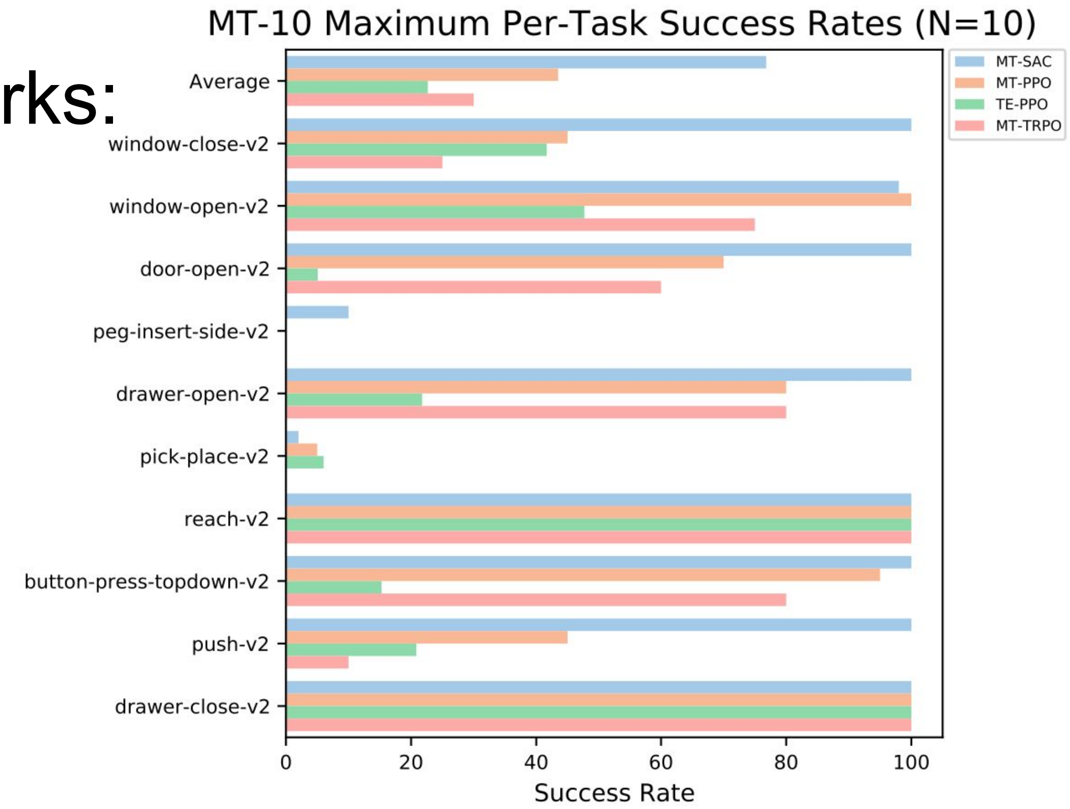
- In general, there is less activity here than gradient-based MAML variants

Meta-World Benchmarks: Multi-Task

Multi-Task (MT): MT1, MT10, MT50

Use standard RL algorithms to train policies that can see the one-hot task ID and goal array

Tasks are sampled from 1 manipulation task (MT**1**), or 10 (MT**10**), etc.



Methods	MT10	MT50
Multi-task PPO	30.5%	**35.4%**
Multi-task TRPO	31.3%	21.0%
Task embeddings	20.9%	11.8%
Multi-task SAC	**68.3%**	**38.5%**

Meta-World Benchmarks: Meta-Learning

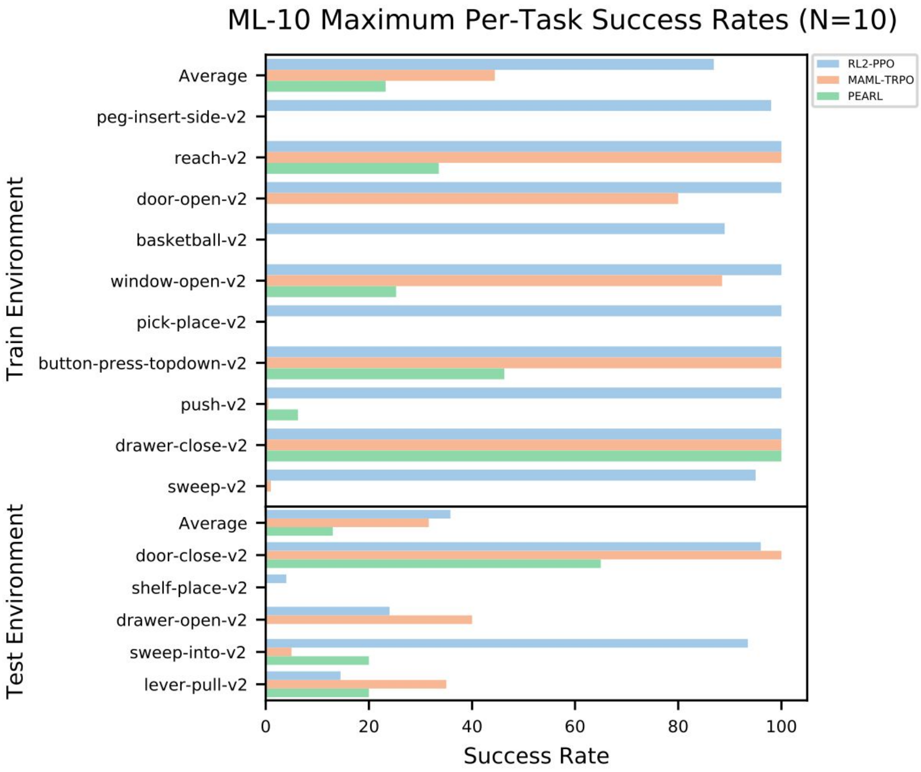
Meta-Learning (ML): ML1, ML10, ML45

Use meta-RL algorithms to train policies that cannot see the one-hot task ID or goal array

Tasks are sampled from 1 manipulation task (ML**1**), or 10 (ML**10**), or 45 (ML**45**)

5 manipulation tasks are held-out as “test” tasks

→ Measures non-parametric generalization



Methods	ML10 meta-train	ML10 meta-test	ML45 meta-train	ML45 meta-test
MAML	44.4%	31.6%	40.7%	**39.9%**
RL^2	**86.9%**	**35.8%**	**70%**	33.3%
PEARL	23.2%	13%	14.5%	22%

Results Discussion & Takeaways

- Single-Task RL is still brittle
  - Unconvincing PPO/SAC results
  - Finding one stable set of hyperparameters with reasonable compute remains hard

Methods	ML10 meta-train	ML10 meta-test	ML45 meta-train	ML45 meta-test
MAML	44.4%	31.6%	40.7%	**39.9%**
RL^2	**86.9%**	**35.8%**	**70%**	33.3%
PEARL	23.2%	13%	14.5%	22%

- Multi-Task RL is still difficult to get working
  - Algorithms are unstable enough that positive transfer is difficult empirically
  - Overlap with Goal-Conditioned RL gives us more tools for improvement

Methods	MT10	MT50
Multi-task PPO	30.5%	**35.4%**
Multi-task TRPO	31.3%	21.0%
Task embeddings	20.9%	11.8%
Multi-task SAC	**68.3%**	**38.5%**

- Meta-RL can extend beyond toy gym tasks
  - Revival of RL^2
  - Are 45 tasks enough to expect non-parametric generalization?

Future Work and Open Problems

- Scaling beyond 50 tasks will probably require sparse reward functions
- Realistic observation spaces (images vs. sensor states)
- Meta-Learning relies heavily on automatic resets

An example of image-based multi-task learning with image observations and a reset trick [23]:

References I

[1] Iqbal et al., 2020, **Toward Sim-to-Real Directional Semantic Grasping**

[2] Mnih et al., 2015, **Human-level control through deep reinforcement learning**

[3] Machado et al., 2017, **Revisiting the Arcade Learning Environment: Evaluation Protocols and Open Problems for General Agents**

[4] Nichol et al., 2018, **Gotta Learn Fast: A New Benchmark for Generalization in RL**

[5] Cobbe et al., 2018, **Quantifying Generalization in Reinforcement Learning**

[6] Cobbe et al., 2019, **Leveraging Procedural Generation to Benchmark Reinforcement Learning**

References II

[7] Andrychowicz et al., 2017, **Hindsight Experience Replay**
[8] Kirk et al., 2021, **A Survey of Generalisation in Deep Reinforcement Learning**
[9] OpenAI et al., 2018, **Learning Dexterous In-Hand Manipulation**
[10] Zhang et al., 2018, **Natural Environment Benchmarks for Reinforcement Learning**
[11] Grigsby and Qi, 2020, **Measuring Visual Generalization in Continuous Control from Pixels**
[12] Tobin et al., 2017, **Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World**

References III

[13] Zhao et al., 2019, **Investigating Generalisation in Continuous Deep Reinforcement Learning**

[14] Finn, 2018, **Learning to Learn with Gradients**

[15] Liu et al., 2022, **Goal-Conditioned Reinforcement Learning: Problems and Solutions**

[16] Finn et al., 2017, **Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks**

[17] Duan et al., 2016, **RL^2: Fast Reinforcement Learning via Slow Reinforcement Learning**

[18] Wang et al., 2016, **Learning to reinforcement learn**

References IV

[19] Rakelly et al., 2019, **Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables**

[20] Zintgraf et al., 2019, **VariBAD: A Very Good Method for Bayes-Adaptive Deep RL via Meta-Learning**

[21] Ghosh et al., 2021, **Why Generalization in RL is Difficult: Epistemic POMDPs and Implicit Partial Observability**

[22] Zintgraf, 2022, **Fast adaptation via meta reinforcement learning**

[23] Kalashnikov et al., 2021, **MT-Opt: Continuous Multi-Task Robotic Reinforcement Learning at Scale**