Exploration Benefits of Multitask Reinforcement Learning With Diverse Tasks

Ziping Xu
Postdoctoral Fellow at Harvard University

This work was done during my PhD at University of Michigan

Role of Exploration in RL

Exploration is a significant topic in online Reinforcement Learning (RL)

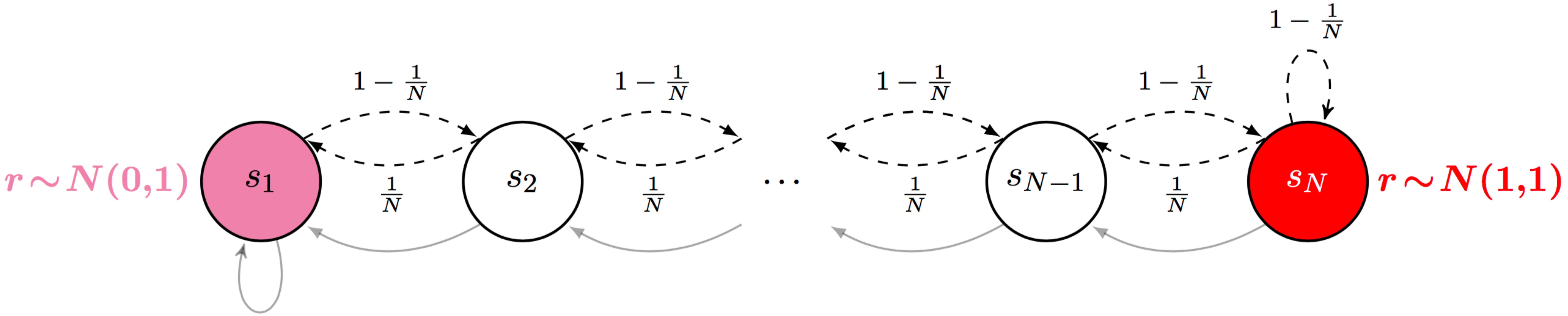
- RL: an agent takes a sequence of actions in an environment in order to maximize cumulative rewards
- Online RL: an agent actively explores an unknown environment to learn a (near)-optimal policy
- 

Figure 1: a typical hard-to-explore environment called River Swim

Random exploration has a probability of $\mathcal{O}(1/2^N)$ to visit state s_N

Leading to a poor coverage of the online dataset

- Naive exploration can easily fail
  - In a River Swim environment (Figure 1) with two actions and N states
    - Agent starts from s_1
    - Dashed lines: the transitions resulted from action 1
    - Grey lines: the transitions resulted from action 2
    - Higher expected reward at state s_N

Strategic Exploration Design

Previous provable sample-efficient online algorithm requires strategic exploration design

- Strategic exploration design accounts for the uncertainty of the environment
  - UCB (Upper Confidence Bound):
    - Construct a high confidence set of the uncertain parameters
    - Explore with the policy that is the most optimistic in the confidence set
- **Issues:**
  - They either heavily rely on tabular and linear MDP assumptions
  - Or they require intractable computational oracle (e.g., non-convex optimization)
    - For example, GOLF (Jin et al. 2021) approximate value function (function that predicts potential cumulative rewards) with general function class $\mathscr{F}$ (deep RL)
    - Step 1: find the set of functions with low error: $\mathscr{F}^t = \left\{ f \in \mathscr{F} : f \text{ has low empirical Bellman error} \right\}$
    - Step 2: maximize the function in this set: $f^{(t)} = \arg\max_{f \in \mathscr{F}^t} f(s_1, \pi(s_1 \mid f))$

[1] Jin, Chi, Qinghua Liu, and Sobhan Miryoosefi. "Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms." *Advances in neural information processing systems* 34 (2021): 13406-13418.

Exploration in Practice

- **UCB is hard to implement in Deep RL**
  - Non-convex optimization are intractable in general
- **Naive exploration:**
  - ϵ-greedy: explore randomly with a probability ϵ
  - Boltzmann exploration: draws actions from a Boltzmann distribution over the “advantage function” of action
- **Heuristic exploration:**
  - Uncertainty-oriented: measure the uncertainty of the value function [1]
  - Intrinsic motivation-oriented: set intrinsic reward the inverse proportion to the visit counts [2]

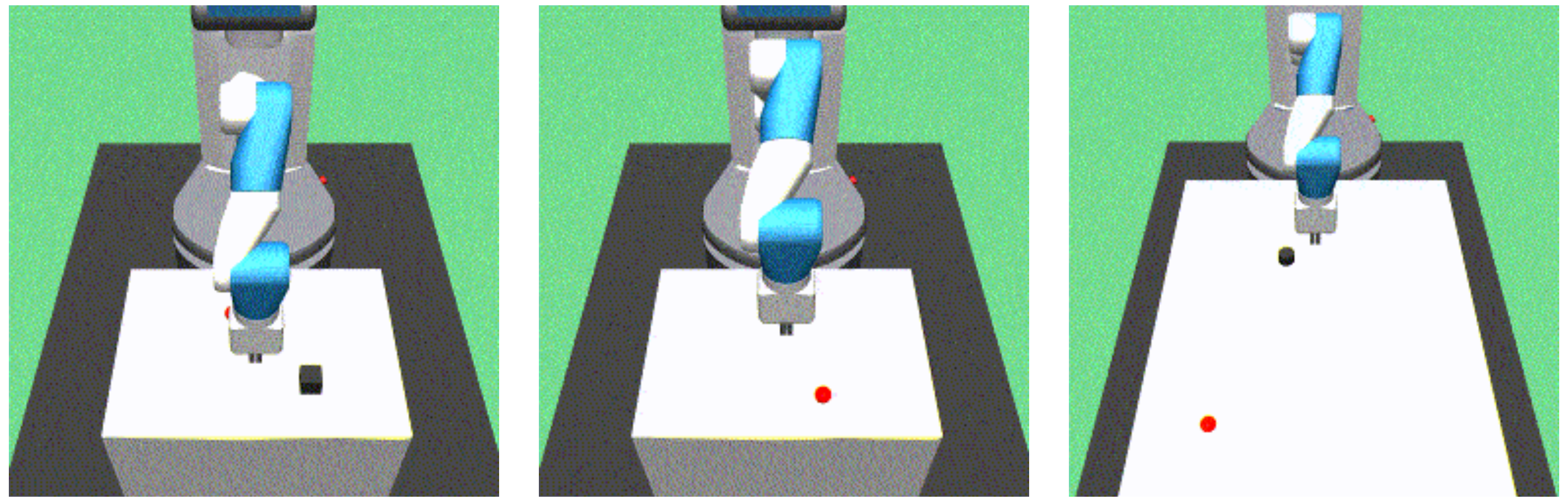
[1] Dearden, Richard, Nir Friedman, and Stuart Russell. "Bayesian Q-learning." *Aaai/iaai* 1998 (1998): 761-768.
[2] Tang, Haoran, et al. "# exploration: A study of count-based exploration for deep reinforcement learning." *Advances in neural information processing systems* 30 (2017).

A Gap Between Theory and Practice

ϵ-greedy, despite being sample-inefficient in the worst-case, performs well in a wide range of applications:

- Atari games control (reaches human level) [1]
- Robotic control [2]

Many real-world RL problems are multi-task RL (MTRL) problems:



Fetch and Push [1] Fetch and Place Fetch and Slide

In many multitask RL algorithms, exploration policies are (implicitly) shared across tasks

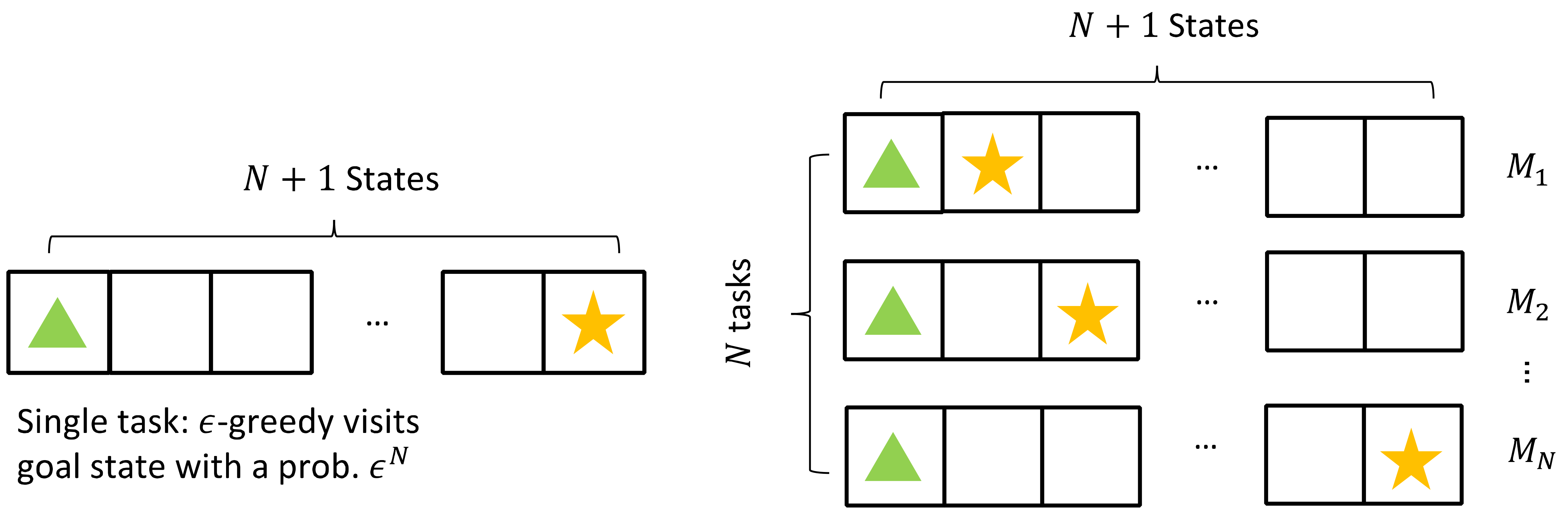
- For example, Andrychowicz et al. (2017) [3] shares the explored trajectories across tasks by relabeling the rewards

Does policy sharing in MTRL benefit exploration by allowing ϵ-greedy to be sample efficient in the worst case?

[1] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." *nature* 518.7540 (2015): 529-533.
[2] Kalashnikov, Dmitry, et al. "Scalable deep reinforcement learning for vision-based robotic manipulation." *Conference on Robot Learning*. PMLR, 2018.
[3] Andrychowicz, Marcin, et al. "Hindsight experience replay." *Advances in neural information processing systems* 30 (2017).

A Motivating Example

Recall the River Swim environment



- Single-task: ϵ-greedy requires $(2/\epsilon)^N$ many episodes to visit the target state
- Diverse multi-task (right panel) is sample-efficient. To see this
  - If task $i \in [N]$ knows its optimal policy $\pi_i^\star$, then state i can be reached with a high probability
  - The target state of task $i+1$, can be reached with probability ϵ with $\pi_i^\star$ + ϵ-greedy
  - We have sufficient exploration for task $i+1$
  - ***Diversity (or richness) of the task set plays an important role here***

Problem Formulation

MDP (Markov Decision Process) formulation

- Episodic MDP $M = (\mathcal{S}, \mathcal{A}, H, P_M, R_M)$
  - $\mathcal{S}$: state space; $\mathcal{A}$ action space; $H \in \mathbb{N}$ horizon length
  - $P_M = (P_{h,M})_{h\in[H]}$ and $R_M = (R_{h,M})_{h\in[H]}$ are transition and reward functions
- Agent interacts with the environment:
  - At each step $h \in [H]$, the agent chooses an action $A_h \in \mathcal{A}$
  - The environment samples the next state $S_{h+1} \sim P_{h,M}(\cdot \mid S_h, A_h)$ and $R_h = R_{h,M}(S_h, A_h)$
- An episode is a sequence $(S_1, A_1, R_1, \ldots, S_H, A_h, R_H, S_{H+1})$
- Goal: maximize the cumulative reward $\sum_{h=1}^{H} R_h$ by optimizing action sequence

Problem Formulation

- **Policy:** the agent chooses actions based on Markovian policies
  - $\pi = (\pi_h)_{h \in [H]}$ and each π_h is a mapping from state to a distribution over $\mathscr{A}$
  - Let Π denote the space of all such policies
- **Value function:**

$$Q^{\pi}_{h,M}(s,a) = \mathbb{E}^{M}_{\pi}\left[r_h + V^{\pi}_{h+1,M}\left(s_{h+1}\right) \mid s_h = s, a_h = a\right]$$

$$V^{\pi}_{h,M}(s) = \mathbb{E}^{M}_{\pi}\left[Q^{\pi}_{h,M}\left(s_h, a_h\right) \mid s_h = s\right]$$

- **General value function approximation:**
  - The algorithm has access to a function class $\mathscr{F} = (\mathscr{F}_h)_{h \in [H+1]}$. Each $\mathscr{F}_h : \mathscr{S} \times \mathscr{A} \mapsto \mathbb{R}$
  - Each $\mathscr{F}_h$ is used to approximate the optimal value function $Q^{\star}_{h,M}$
  - We assume Bellman completeness [1] that ensures the richness of $\mathscr{F}$

[1] Jin, Chi, Qinghua Liu, and Sobhan Miryoosefi. "Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms." *Advances in neural information processing systems* 34 (2021): 13406-13418.

Proposed Generic Algorithm

Multitask learning scenario

- Algorithm interacts with a set of tasks $\mathscr{M}$ sequentially for T rounds
- At the each round t, the algorithm chooses an $M \in \mathscr{M}$ and an exploratory policy, which is used to collect one episode on M
- At the end of T rounds, the algorithm outputs a set of policies $\{\pi_M\}_{M \in \mathscr{M}}$.
- Goal: learn a near-optimal policy π_M for each task $M \in \mathscr{M}$

Algorithm 1 Generic Algorithm for MTRL with Policy-Sharing

Input: function class $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_{H+1}$, task set $\mathcal{M}$
Initialize $\mathcal{D}_{0,M} \leftarrow \emptyset$ for all $M \in \mathcal{M}$
for round $t = 1, 2, \ldots, \lfloor T/|\mathcal{M}| \rfloor$ **do**
    Offline learning oracle outputs $\hat{f}_{t,M} \leftarrow \mathcal{Q}(\mathcal{D}_{t-1,M})$ for each M
    Set myopic exploration policy $\hat{\pi}_{t,M} \leftarrow \mathrm{expl}(\pi^{\hat{f}_{t,M}})$ for each M
    Set $\hat{\pi}_t \leftarrow \mathrm{Mixture}(\{\hat{\pi}_{t,M}\}_{M \in \mathcal{M}})$
    for $M \in \mathcal{M}$ **do**
        Sample one episode $\tau_{t,M}$ on MDP M with policy $\hat{\pi}_t$
        Add $\tau_{t,M}$ to the dataset: $\mathcal{D}_{t,M} \leftarrow \mathcal{D}_{t-1,M} \cup \{\tau_{t,M}\}$
    end for
end for
Return $\hat{\pi}_M = \mathrm{Mixture}(\{\hat{\pi}_{t,M}\}_{t \in \lfloor T/|\mathcal{M}| \rfloor})$ for each M

Additional notations

- $\mathcal{Q}$: an offline learning oracle that outputs a value function given a dataset
- expl(π): ϵ-greedy exploration with greedy policy π
- Policy mixture:
  - Given a set of policies $\{\pi_i\}_{i=1}^N$, Mixture($\{\pi_i\}_{i=1}^N$) randomly draw an index i, then run policy π_i for the whole episode

When Does ϵ-greedy in Single Task Work?

Dann et al. (2022) [1] proposed myopic exploration gap (MEG)

For the ease of presentation, we keep our discussion within tabular MDPs (finite state and action space)

- For the tabular case, myopic exploration gap $\alpha(\pi)$ of policy π is the value of

$$\sup_{\pi' \in \Pi', c \geq 1} \frac{1}{\sqrt{c}} \left(V_1^{\pi'}(s_1) - V_1^{\pi}(s_1) \right) \text{ s.t.}$$

$$\frac{\mu_h^{\pi'}(s,a)}{\mu_h^{\text{expl}(\pi)}(s,a)} \leq c, \text{ for all } s, a, h$$

Single-policy Concentrability / Density Ratio for covariate shift

- Here $\mu_h^{\pi}(s,a) = \Pr^{\pi}(S_h = s, A_h = a)$ is called the occupancy measure
- Intuition: out of all policies that are sufficiently covered by π, there exists one policy π' that makes significant value improvement
- Dann et al. (2022) showed that
  - if $\alpha(\pi)$ is lower bounded for all β-suboptimal policies,
  - then Algorithm 1 (single-task case) has sample-complexity bound that is polynomial in all parameters

[1] Dann, Chris, et al. "Guarantees for epsilon-greedy reinforcement learning with function approximation." International Conference on Machine Learning. PMLR, 2022.

Extending to MTRL

We need large MEG to hold for at least one task

Definition 1 (Multitask MEG). Let $\pi \in \Pi^{\mathscr{M}}$ be a joint policy and π_M is the component for task M. We say that π has $\alpha(\pi, \mathscr{M})$ multitask myopic exploration gap, where $\alpha(\pi, \mathscr{M})$ is the value to:

Hold for at least one task

$$\max_{M \in \mathscr{M}} \sup_{\pi' \in \Pi', c \geq 1} \frac{1}{\sqrt{c}} \left(V_{1,M}^{\pi'}(s_1) - V_{1,M}^{\pi_M}(s_1) \right) \text{ s.t.}$$

$$\frac{\mu_{h,M}^{\pi'}(s,a)}{\mu_{h,M}^{\mathrm{expl}(\pi)}(s,a)} \leq c, \text{ for all } s, a, h$$

For a joint policy, let $\mathrm{expl}(\pi) = \mathrm{Mixture}(\{\mathrm{expl}(\pi_M)\}_{M \in \mathscr{M}})$

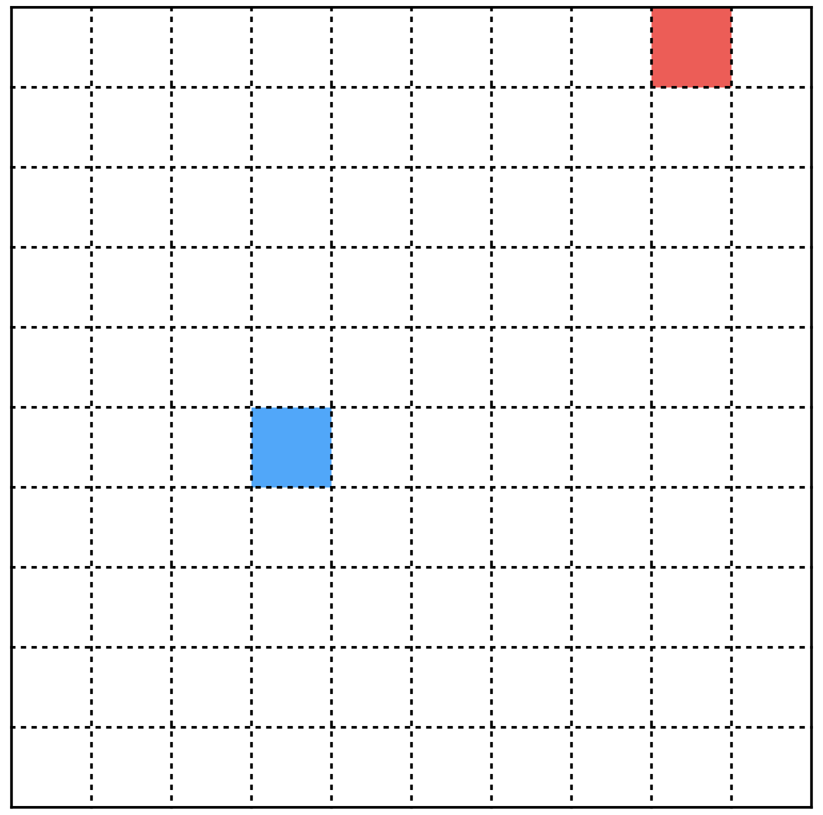
The current exploration policy $\mathrm{expl}(\pi)$ can significantly improve π_M for at least one task M.

We want this to happen whenever some π_M is β suboptimal.

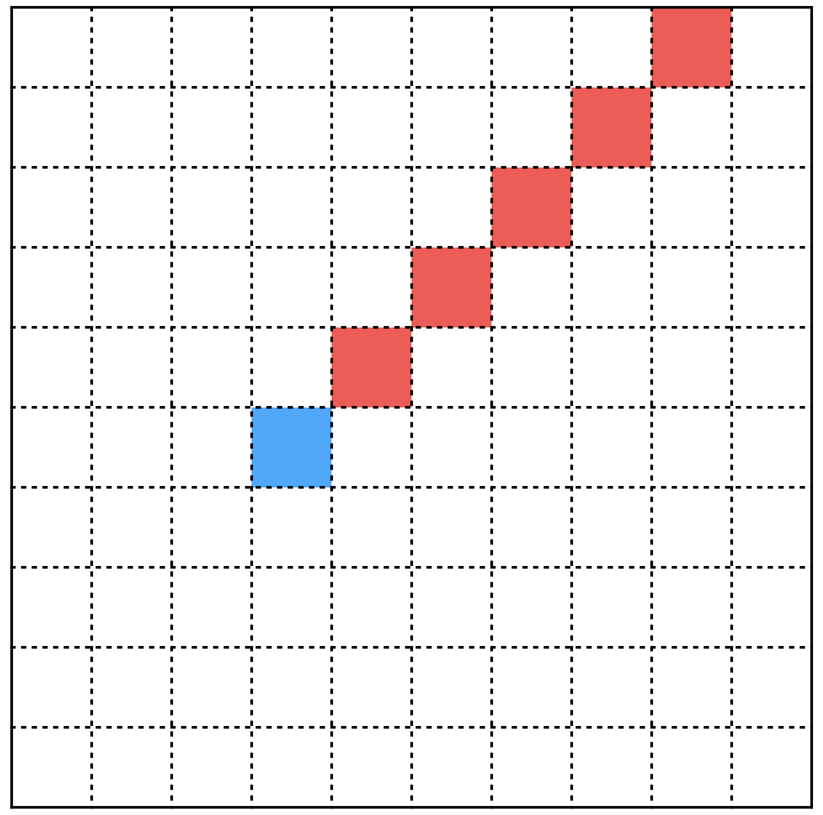
Comparing MEG and Multitask MEG

There can be an exponential gap between Single-task MEG and Multitask MEG

- For all $\pi \in \Pi^{\mathscr{M}}$, we have $\alpha(\pi, \mathscr{M}) \geq \alpha(\pi_M, \{M\})/\sqrt{M}$ for all $M \in \mathscr{M}$
- For a goal-orientated environment (sparse reward on a goal (s_g, h_g)): $\alpha(\pi, \{M\}) \leq \sqrt{\Pr^{\text{expl}(\pi)}(S_{h_g} = s_g)}$
- When the initial state and the goal state are far, $\Pr^{\text{expl}(\pi)}(S_{h_g} = s_g)$ can be exponentially small



Blue: initial state; Red: goal state
Grid-world with small MEG



Each red cell is a goal state for a task
Grid-world with large Multitask MEG

Sample Complexity Guarantee

A definition of diverse task set

Definition 2 (Multitask Suboptimality). $\Pi_\beta \subset (\Pi)^{\mathscr{M}}$ is the β-suboptimal policy class, such that for any $\pi \in \Pi_\beta$, there exists π_M that is β-suboptimal for MDP M, i.e.
$V_{1,M}^{\pi_M} \leq \max_{\pi \in \Pi} V_{1,M}^{\pi} - \beta$

Definition 3 (Diverse Tasks). For some function $\tilde{\alpha} : [0,1] \mapsto \mathbb{R}$, a tasks set is $\tilde{\alpha}$-diverse if any $\pi \in \Pi_\beta$ has multitask myopic exploration gap $\alpha(\pi, \mathscr{M}) \geq \tilde{\alpha}(\beta)$ for any constant $\beta > 0$

Theorem 1 (Upper Bound for Sample Complexity). If $\mathscr{M}$ is $\tilde{\alpha}$-diverse, Algorithm 1 with ϵ-greedy exploration function has a sample-complexity

$$\mathscr{C}(\beta, \delta) = \tilde{\mathcal{O}}\left(\frac{\mathscr{M}^{\;2} H^2}{\tilde{\alpha}^2(\beta)} \ln(1/\delta) \right),$$

Sample complexity: with $\mathscr{C}(\beta, \delta)$ total number of episodes, Algorithm 1 outputs a β-optimal policy for each M with a probability at least $1 - \delta$.

Examples of Diverse Tasks

Tabular case

- Diverse task: for each $(s, h) \in \mathcal{S} \times [H]$, there exists $M_{s,h} \in \mathcal{M}$, such that $R_{h',M_{s,h}}(s') = \mathbb{I}[s' = s, h' = h]$ ($M_{s,h}$ has sparse reward function on goal state (s, h))
- Note that this construction is also used for the reward-free exploration under the tabular MDP setting
- Multitask MEG lower bound: $\alpha(\pi, \mathcal{M}) = \Omega(\beta^2/(\ \mathcal{A}\ \ \mathcal{M}\ \ H))$

Linear MDP

- Definition: there exists a feature mapping $\phi_h : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}^d$, such that $P_h(s' \mid s, a) = \langle \phi_h(s, a), \nu_h(s') \rangle$, for some measure $\nu_{h,i}$ over $\mathcal{S}$ and $R_h(s, a) = \langle \phi_h(s, a), \theta_h \rangle$ for $\theta_h \in \mathbb{R}^d$
- Diverse task: for any $h \in [H]$, there exists a subset $\mathcal{M}^{(h)} \subset \mathcal{M}$, such that for all $M \in \mathcal{M}^{(h)}$, $\theta_{h',M} = 0$ with $h' \neq h$ and

  $$\sigma_{\min}\left(\sum_{M \in \mathcal{M}^{(h)}} \mathbb{E}_M^{\pi_M^*} \phi_h(s_h, a_h) \phi_h^\top(s_h, a_h) \right) \geq c$$
- Multitask MEG lower bound: $\alpha(\pi, \mathcal{M}) = \Omega(\beta^2 c/(\ \mathcal{A}\ \ \mathcal{M}\ \ H))$

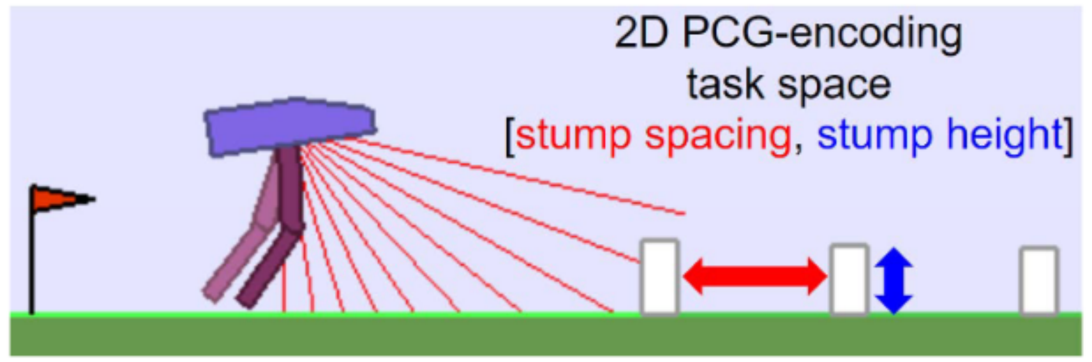
Implications of Diversity in Deep RL

Motivations

- Deep RL: a pre-trained feature extractor generates embeddings for Q-value function followed by a linear mapping
- This manner is similar to the setup in linear MDPs
- Diversity for Linear MDPs requires a full rank covariance matrix of $\phi_h(s_h, a_h)$ at each h if the optimal policy is executed.
- We conduct simple simulation studies on a robotic control environment, to verify that whether a more spread spectrum of the covariance matrix of the embeddings would lead to better sample efficiency

Experiment setup

- Environment: BipedalWalker environment [1]
- The walker has controllable motors with torque
- The objective of the agent is to move forward, while crossing stumps with varying heights at regular intervals
- Task $M_{p,q}$: p and q denote the heights of the stumps and the spacings between the stumps



(a) BipedalWalker environment

[1] Portelas, Rémy, et al. "Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments." *Conference on Robot Learning*. PMLR, 2020.

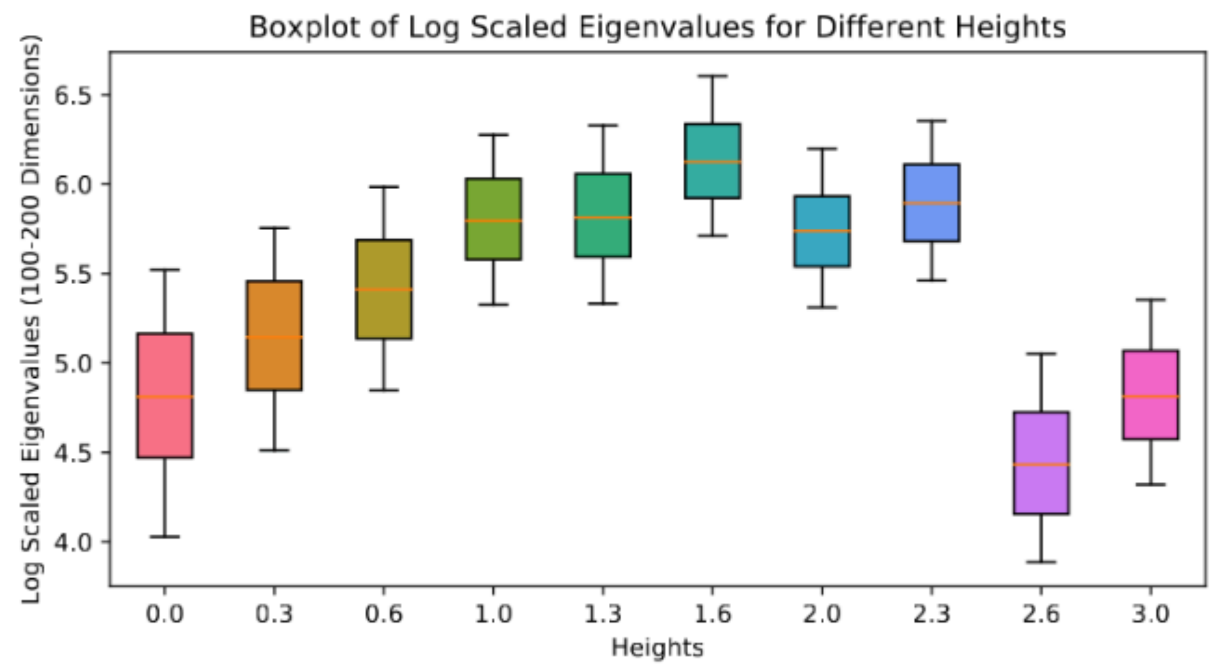
Experiments

Investigating feature covariance matrix

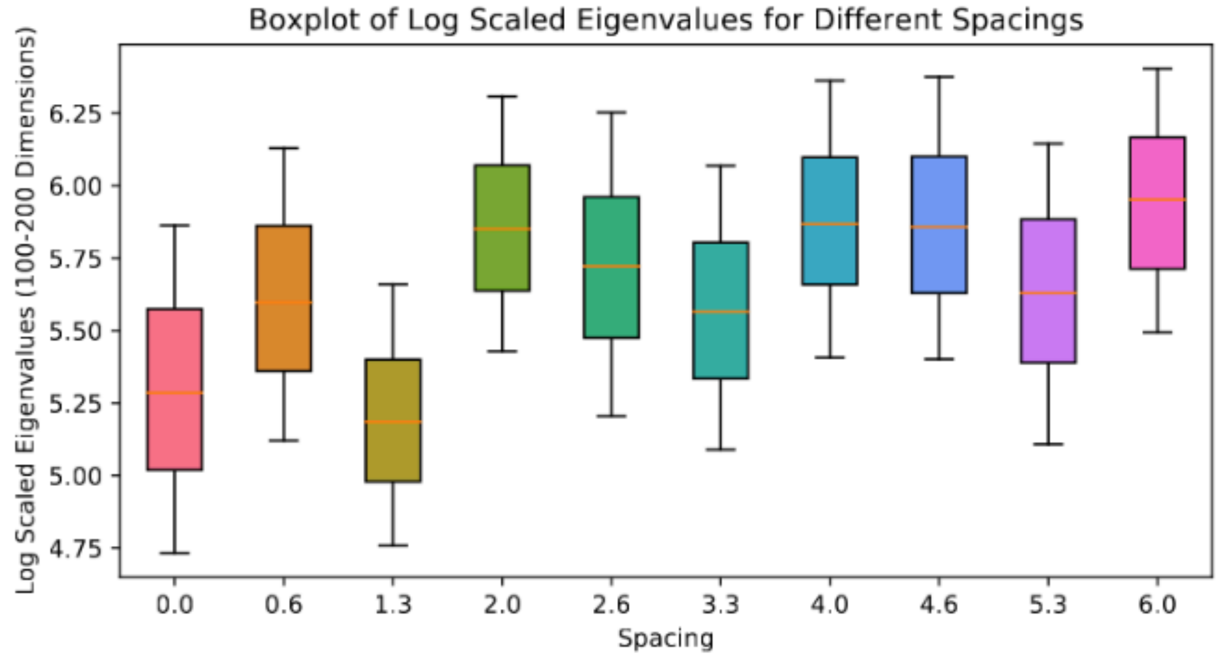
- Train PPO (Proximal Policy Optimization Algorithms) [1] agent on 100 tasks with different parameter vectors (p, q)
- Evaluate $\phi(s, a) \in \mathbb{R}^{300}$ at the end of the training generated by near-optimal policies π
- Compute the covariance matrix $V_{p,q} = \mathbb{E}_{\pi}^{M_{p,q}} \sum_{h=1}^{H} \phi(s_h, a_h)\phi(s_h, a_h)^T$
- We observe that the stump heights p has a more significant impact on the spectrum compared to spacing q ((b) and (c))
- Tasks with $p \in [1.0, 2.3]$ leads to better diversity

Coincidence with automatic curriculum learning (ACL) task selection

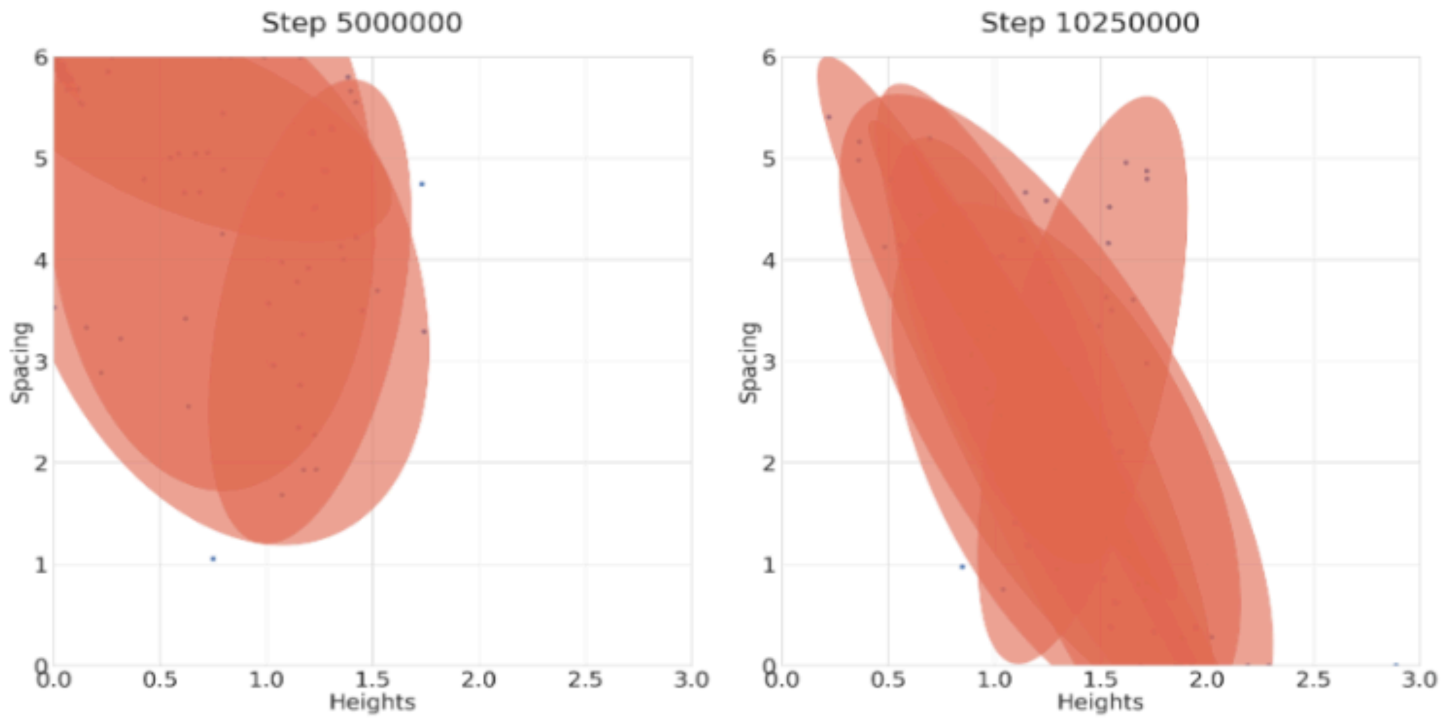
- ALP-GMM [2], a well-established ACL algorithm, for BipedalWalker environment
- Figure (d) gives the density plots of the ACL task sampler during the training process
- It shows a significant preference over heights p in the middle range, with little preference over spacing q



(b) Controlling heights



(c) Controlling spacings



(d) Preference of automatic CL

[1] Schulman, John, et al. "Proximal policy optimization algorithms." *arXiv preprint arXiv:1707.06347* (2017).
[2] Portelas, Rémy, et al. "Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments." *Conference on Robot Learning*. PMLR, 2020.

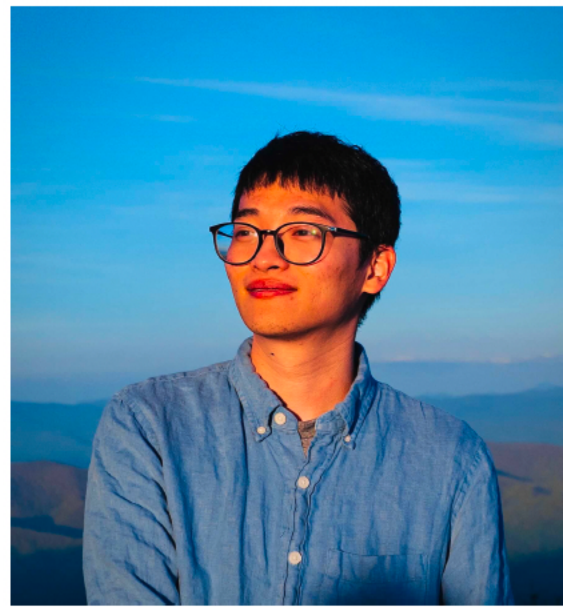
Generalization Performance

Training on different parameters

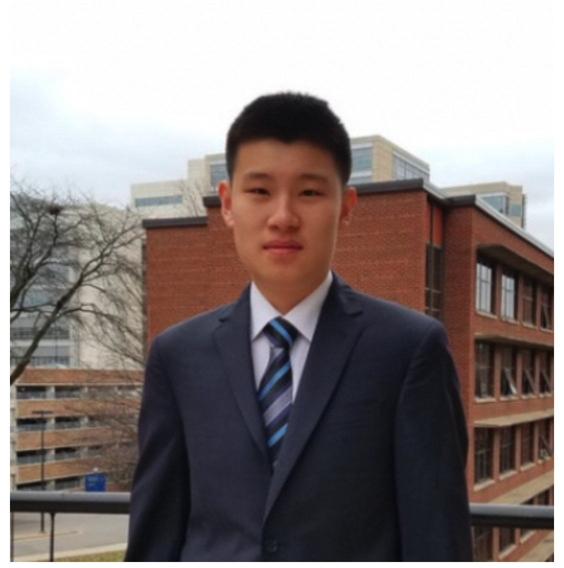
- We train the same agent with different means of the stump heights p, then fine tune them on all tasks
- Evaluation: the number of tasks the agent masters in the end of training
- Algorithm trained on tasks with stump heights in the middles masters significantly more tasks

Obstacle spacing	Stump height	Mastered task
[2, 4]	[0.0, 0.3]	28.1 ± 6.1
[2, 4]	[1.3, 1.6]	41.6 ± 9.8
[2, 4]	[2.6, 3.0]	11.5 ± 10.9

Collaborators



Zifan Xu, PhD student at UT Austin, CS department



Runxuan Jiang: undergrad at University of Michigan

Ambuj Tewari, Professor at University of Michigan, Department of Statistics

Peter Stone, Professor at UT Austin, CS Department