Active Policy Improvement from Multiple Blackbox Oracles

Xuefeng Liu[1*] Takuma Yoneda[2*] Chaoqi Wang[1*] Matthew R. Walter[2] Yuxin Chen[1]

[1] University of Chicago [2] Toyota Technological Institute at Chicago (TTIC)

Motivation

- Reinforcement learning (RL) tends to be highly sample inefficient
- Imitation learning improves the sample efficiency of RL
- In real-world scenarios, accessing an optimal oracle can be costly or even not possible
- However, one often has access to multiple *suboptimal* oracles
- **Goal:** Learning from black-box oracles by combining their state-wise expertise

How can an agent actively learn from ***multiple black-box oracles*** by taking advantage of their complementary expertise to learn a better policy in a ***sample-efficient*** manner?

Learning from Multiple Oracles

- **Single-best oracle:** $\pi^\star \doteq \arg\max_{\pi\in\Pi} V^\pi(d_0)$
  - weak baseline that does not consider the state-wise optimality of different oracles
- **Max-following policy:** $\pi^\bullet(a \mid s) \doteq \pi^{k^*}(a \mid s)$, $k^* \doteq \arg\max_{k\in[K]} V^k(s)$
  - a greedy policy that follows the best oracle in any state

✓ **Max-aggregation policy:** $\pi^{\max}(a|s) \doteq \delta_{a=a^\star}$,

$$a^\star = \arg\max_{a\in\mathcal{A}} A^{f^{\max}}(s,a),\ f^{\max}(s_t) \doteq \max_{k\in[K]} V^k(s),$$
$$A^{f^{\max}}(s,a) = r(s,a) + \mathbb{E}_{s'\sim\mathcal{P}|s,a}\left[f^{\max}(s')\right] - f^{\max}(s)$$

Max-aggregation in Online Learning Setting

- Black-box oracle
  - × true value function of each oracle is unknown to the learner
  - ✓ reduce IL algorithm to online learning
- We adapt the online loss

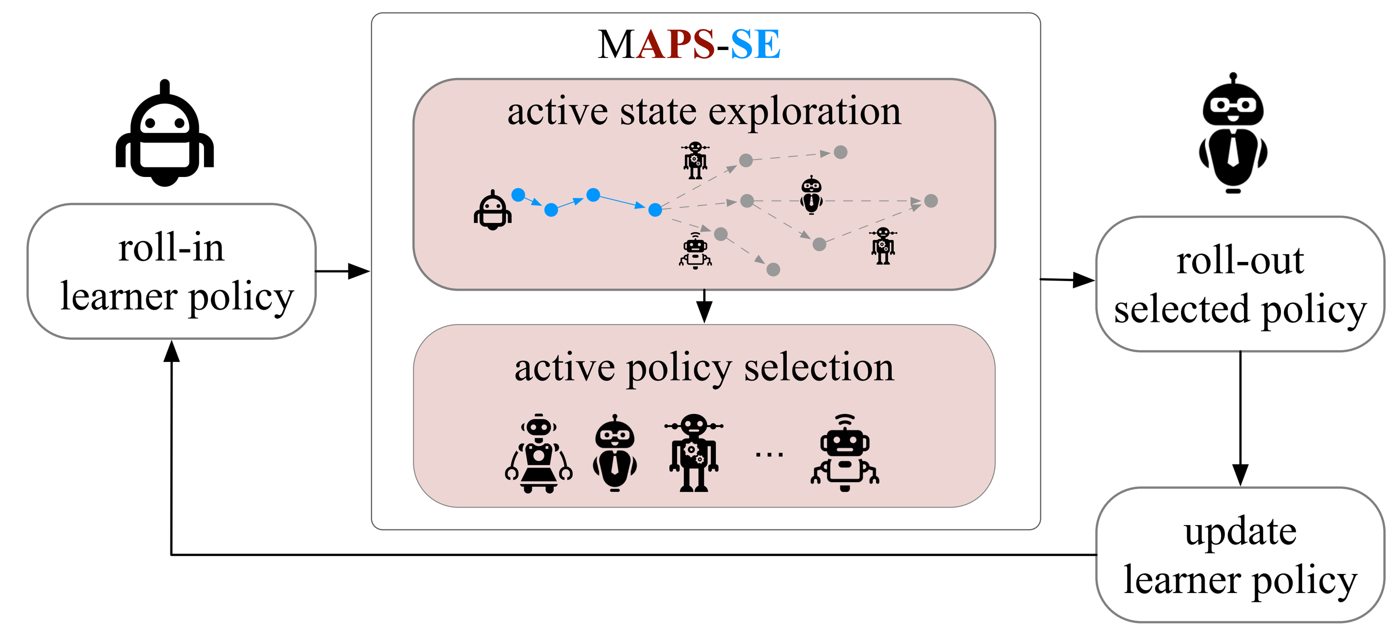
$$\ell_n(\pi;\lambda) \doteq \underbrace{-(1-\lambda)H\mathbb{E}_{s\sim d^{\pi_n}}\left[A_\lambda^{f^{\max},\pi}(s,\pi)\right]}_{\text{Imitation Learning Loss}}\ \underbrace{-\lambda\mathbb{E}_{s\sim d_0}\left[A_\lambda^{f^{\max},\pi}(s,\pi)\right]}_{\text{Reinforcement Learning Loss}}$$

- Empirical estimate of the $\ell_n(\pi,\lambda)$ gradient

$$\nabla\hat{\ell}_n(\pi_n;\lambda) = -H\mathbb{E}_{s\sim d^{\pi_n}, a\sim\pi_n(\cdot|s)}\left[\nabla\log\pi_n(a|s)\, A_\lambda^{\widehat{f^{\max}},\pi_n}(s,a)\right] \quad (1)$$

  - may select suboptimal oracle policy due to bias in the value function approximator $\widehat{f^{\max}}$ for $\ell_n(\pi,\lambda)$

Method Overview



✓ Max-aggregation Active Policy Selection with Active State Exploration (MAPS-SE)
- Active policy selection (APS)
  - ∗ reduce approximation error
- Active state exploration (ASE)
  - ∗ control state-wise uncertainty

Algorithm Details

Algorithm 1 Max-aggregation Active Policy Selection with Active State Exploration (MAPS-SE)

Require: Initial learner policy π_1, oracle policies $\{\pi^k\}_{k\in[K]}$, initial value functions $\{\hat{V}^k\}_{k\in[K]}$

for $n = 1, 2, \ldots, N-1$ **do**
  if SE is TRUE **then**
    ▷ /* active state exploration */
    Roll-in policy π_n until $\Gamma_{k_\star}(s_t) \geq \Gamma_s$, where $k_\star$ and $\Gamma_{k_\star}(s_t)$ are computed via Equations (2) and (3) at each visited state s_t.
  else
    Roll-in policy π_n up to $t_e \sim \text{Uniform}[H-1]$
  ▷ /* active policy selection */
  Select $k_\star$ via Equation (2).
  Switch to $\pi^{k_\star}$ to roll-out and collect data $\mathcal{D}_n$.
  Update the estimate of $\hat{V}^{k_\star}(\cdot)$ with $\mathcal{D}_n$.
  Roll-in π_n for full H-horizon to collect data $\mathcal{D}'_n$.
  Compute gradient estimator g_n of $\nabla\hat{\ell}_n(\pi_n,\lambda)$ (1) using $\mathcal{D}'_n$.
  Update π_n to π_{n+1} by giving g_n to a first-order online learning algorithm.

- Active policy selection (MAPS)
  - define the best oracle $\pi^{k_\star}$ as

$$k_\star = \arg\max_{k\in[K]} \begin{cases} \hat{V}^k(s_t) + \sqrt{\frac{2H^2\log\frac{2}{\delta}}{N_k(s_t)}} & \text{discrete} \\ \hat{V}^k(s_t) + \sigma_k(s_t) & \text{continuous} \end{cases} \quad (2)$$

- Active state exploration (MAPS-SE)
  - define the state-wise uncertainty $\Gamma_{k_\star}(s_t)$ as

$$\Gamma_{k_\star}(s_t) = \begin{cases} \sqrt{\frac{2H^2\log\frac{2}{\delta}}{N_{k_\star}(s_t)}} & \text{discrete} \\ \sigma_{k_\star}(s_t) & \text{continuous} \end{cases} \quad (3)$$

Algorithmic Characteristics

Algorithm	Criterion	Online	Stateful	Active	Interactive	Multiple oracles	Sample efficient
Behavioral cloning	IL	×	✓	×	×	×	–
PPO with GAE	RL	✓	✓	×	×	×	×
AggreVaTeD	IL	✓	✓	×	✓	×	–
MAMBA	IL + RL	✓	✓	×	✓	✓	×
CAMS	Model Selection	✓	×	✓	×	✓	✓
MAPS (ours)	IL + RL	✓	✓	✓	✓	✓	✓

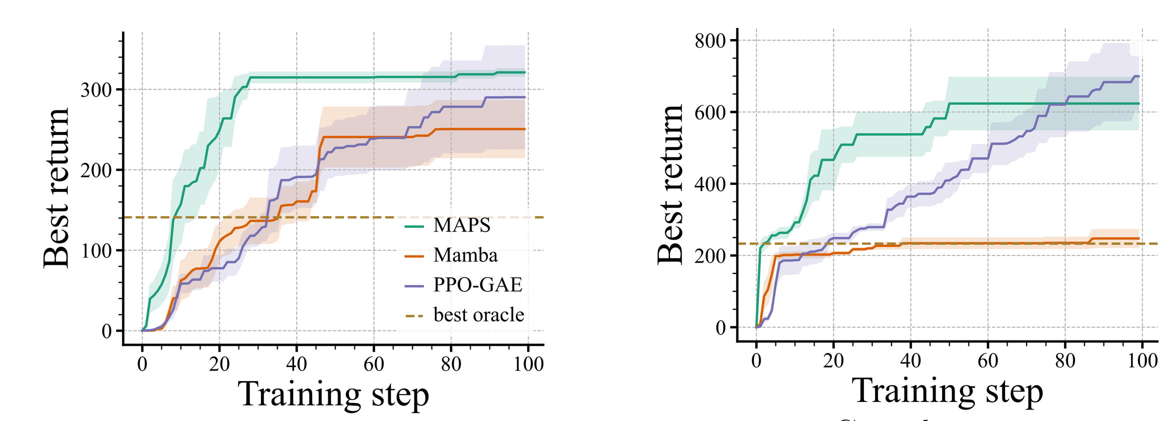
Theoretical Guarantees

- The sample complexity table for identifying the best oracle per state

Selection strategy	Sample complexity	Γ_s
Uniform (MAMBA)	$\mathcal{O}\left(\left(\sum_i \frac{KH^2}{\Delta_i^2}\right)\log\left(\frac{K}{\delta}\right)\right)$	—
APS (MAPS)	$\mathcal{O}\left(K + \left(\sum_i \frac{H^2}{\Delta_i^2}\right)\log\left(\frac{K}{\delta}\right)\right)$	—
ASE (MAPS-SE)	$\mathcal{O}\left(K + \left(\sum_i \frac{H^2}{\Delta_i^2}\right)\log\left(\frac{K}{\delta}\right)\right)$	$o\left(\sqrt{\frac{2H^2\log(4/\delta)}{K+(\sum_i H^2/\Delta_i^2)\log(2K/\delta)}}\right)$

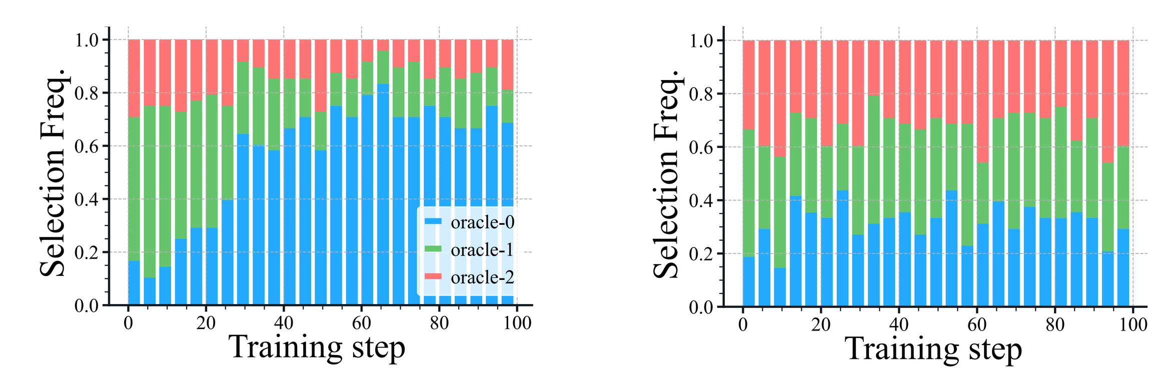
Experimental Results

- MAPS (APS) Performance



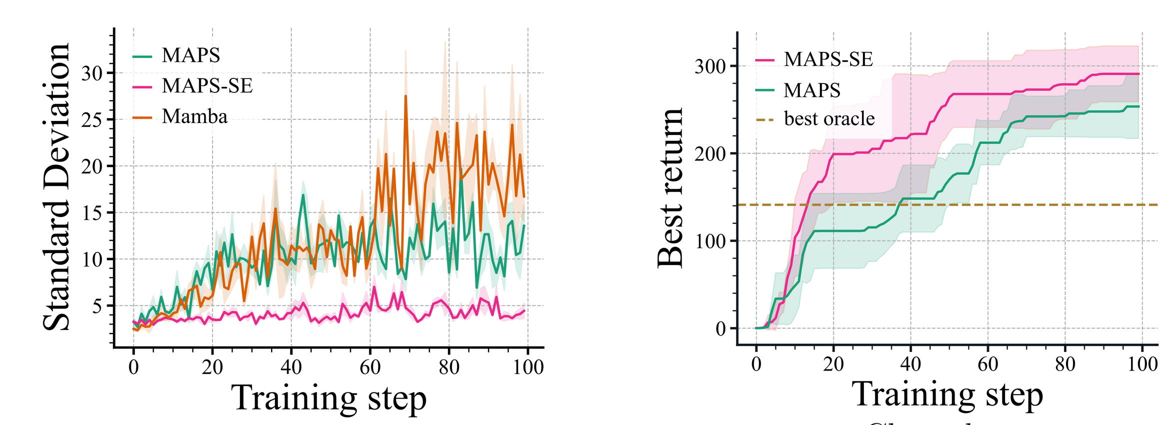
Cheetah-run Carpole-swingup

- Effect of Active Policy Selection



MAPS MAMBA

- MAPS-SE (ASE) Performance



Cheetah-run Cheetah-run