THE UNIVERSITY OF CHICAGO

TOYOTA TECHNOLOGICAL INSTITUTE AT CHICAGO

Active Policy Improvement from Multiple Black-box Oracles

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

[1]UChicago [2]TTIC

Motivation

- *Reinforcement Learning* (RL) tends to be highly sample inefficient
  - especially in high-dimensional or sparse reward environments

- *Imitation learning* improves the sample efficiency of RL
  - train a policy to imitate an expert policy
  - methods typically assume expert is optimal or near-optimal
  - in real-world scenarios, accessing an optimal oracle can be costly or may not even be possible
  - it is often the case that one has access to multiple suboptimal oracles

- **Goal:** Sample efficient 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 skills to learn a better policy in a sample-efficient manner?

Algorithms for Learning from Multiple Oracles

- **Max-following $\pi^\bullet$**

$$\pi^\bullet(a \mid s) \doteq \pi^{k^*}(a \mid s), \quad k^* \doteq \arg\max_{k \in [K]} V^k(s).$$

  - a greedy policy that follows the best oracle in any state
  - better than single-best oracle $\pi^\star \doteq \arg\max_{\pi \in \Pi} V^\pi(d_0)$

Is there a better baseline?

Algorithms for Learning from Multiple Oracles

A natural value function for multiple oracles:

$$f^{\mathsf{max}}(s_t) \doteq \max_{k \in [K]} V^k(s).$$

✓ *Max-aggregation* π^{max}

$$\pi^{\mathsf{max}}(a|s) \doteq \delta_{a=a^\star}, \text{where } a^\star = \arg\max_{a \in \mathcal{A}} \; A^{f^{\mathsf{max}}}(s,a),$$
$$A^{f^{\mathsf{max}}}(s,a) = r(s,a) + \mathbb{E}_{s' \sim \mathcal{P}|s,a}[f^{\mathsf{max}}(s')] - f^{\mathsf{max}}(s)$$

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 adapted the online loss from Cheng et al. (2020)

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

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

$$\nabla\hat{\ell}_n(\pi_n;\lambda) = -H\mathbb{E}_{s\sim d^{\pi_n},a\sim\pi_n(\cdot|s)}\left[\nabla\log\pi_n(a|s)\,A_\lambda^{\hat{f}^{\text{max}},\pi_n}(s,a)\right] \tag{2}$$

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

Limitations of the prior state-of-the-art

- MAMBA (Cheng et al., 2020)
  - Sample inefficiency
    - caused by uniform policy sampling
    - a large accumulation of error (regret) when identification fails
  - no control over the state uncertainty

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

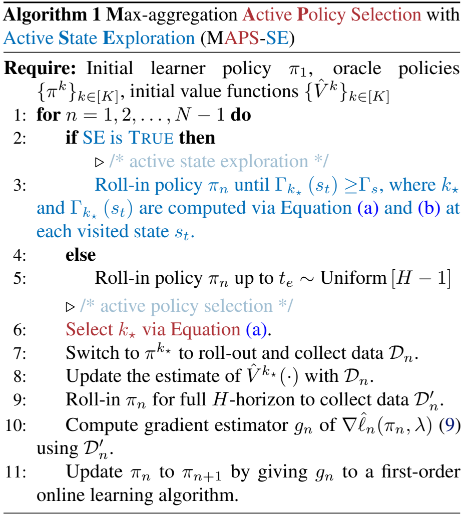
The MAPS-SE Algorithm

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

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

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

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



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

```
Require: Initial learner policy $\pi_1$, 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 $\pi_n$ until $\Gamma_{k_\star}(s_t) \geq \Gamma_s$, where $k_\star$ and $\Gamma_{k_\star}(s_t)$ are computed via Equation (a) and (b) at each visited state $s_t$.
    else
        Roll-in policy $\pi_n$ up to $t_e \sim$ Uniform $[H-1]$
    ▷ /* active policy selection */
    Select $k_\star$ via Equation (a).
    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 $\pi_n$ for full $H$-horizon to collect data $\mathcal{D}'_n$.
    Compute gradient estimator $g_n$ of $\nabla\hat{\ell}_n(\pi_n, \lambda)$ (9) using $\mathcal{D}'_n$.
    Update $\pi_n$ to $\pi_{n+1}$ by giving $g_n$ to a first-order online learning algorithm.
```

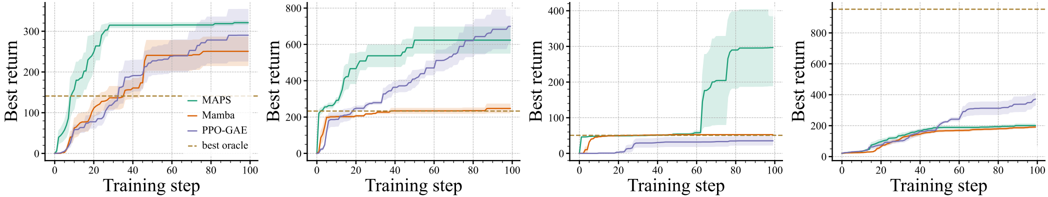
Theoretical Guarantees

The following table summarizes the sample complexity of related methods associated with 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)	$\mathcal{O}\left(K + \left(\sum_i \frac{H^2}{\Delta_i^2}\right)\log\left(\frac{K}{\delta}\right)\right)$	$o\left(\sqrt{\frac{2H^2\log(4/\delta)}{K+\left(\sum_i H^2/\Delta_i^2\right)\log(2K/\delta)}}\right)$

Table: APS (MAPS) achieves a significant reduction in sample complexity of scale K compared to uniform (MAMBA). ASE (MAPS-SE) exhibits the same sample complexity as APS, provided that a pre-set Γ_s condition is met.

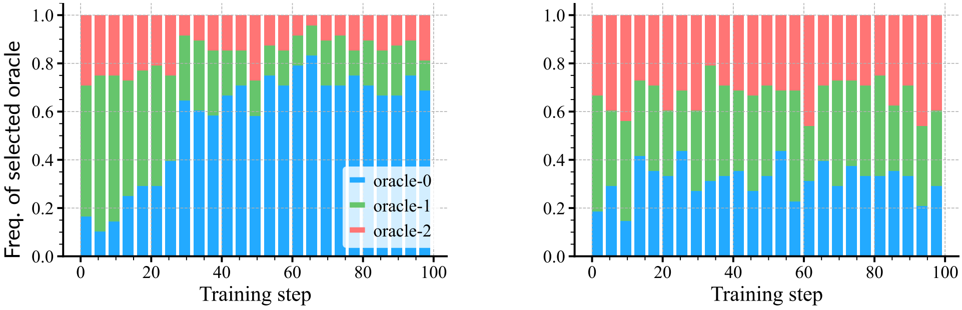
Experiments- MAPS (APS) -Performance



(a) Cheetah-run (b) Cartpole-swingup (c) Pendulum-swingup (d) Walker-walk

Figure: Comparing the performance of MAPS against the three baselines (MAMBA, PPO-GAE, and the best oracle) across four environments (Cheetah-run, Cartpole-swingup, Pendulum-swingup, and Walker-walk), using the best-return-so-far metric. Each domain includes three oracles, each representing a mixture of policies pretrained via PPO-GAE and SAC. The best oracle is depicted as dotted horizontal lines in the figure. The shaded areas denote the standard error calculated using five random seeds. Except for the Walker-walk environment, MAPS surpasses all baselines in every benchmark.

Experiments-Effect of Active Policy Selection



(a) MAPS

(b) MAMBA

Figure: A comparison of the frequency with which **(a)** MAPS and **(b)** MAMBA select among a bad (in red), mediocre (in green), and good (in blue) oracle in a three-oracle Cheetah-run experiment. MAPS efficiently identifies each oracle's quality as indicated by the frequency with which it queries the good oracle. In contrast, MAMBA maintains roughly the same selection frequency for the bad, mediocre, and good oracles throughout.

Experiments- MAPS-SE (ASE) -Performance

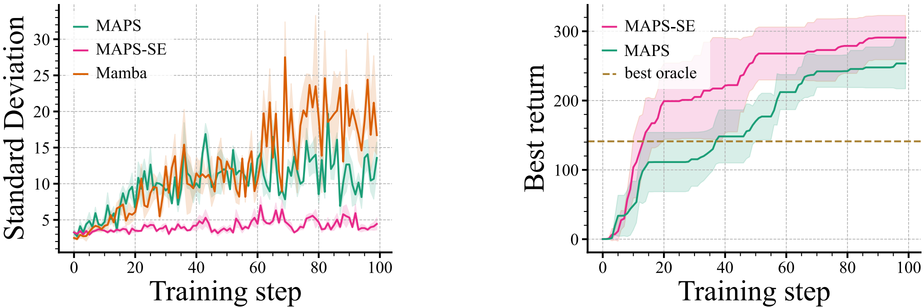


Figure: **Left**: A demonstration of the benefits of active state exploration in terms of a comparison between the standard deviation for switch-state for MAPS-SE (in blue), MAPS (in orange) and MAMBA (in green) in the Cheetah-run environment with the same set of oracles used in Figure 1. under a threshold $\Gamma_s = 2.5$. The predicted standard deviation is evaluated at the switching state from learner policy to oracle. **Right**: A comparison of the best-return-so-far in a multiple oracle set between MAPS-SE with MAPS and the best oracle baseline on the Cheetah-run environment.