Active Policy Improvement from Multiple Black-box Oracles

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

Abstract

Reinforcement learning (RL) has made significant strides in various complex domains. However, identifying an effective policy via RL often necessitates extensive exploration. Imitation learning aims to mitigate this issue by using expert demonstrations to guide exploration. In real-world scenarios, one often has access to multiple *suboptimal* black-box experts, rather than a single optimal oracle. These experts do not universally outperform each other across all states, presenting a challenge in actively deciding which oracle to use and in which state. We introduce MAPS and MAPS-SE, a class of policy improvement algorithms that perform imitation learning from multiple suboptimal oracles. In particular, MAPS actively selects which of the oracles to imitate and improve their value function estimates, and MAPS-SE additionally leverages an active state exploration criterion to determine which states one should explore. We provide a comprehensive theoretical analysis and demonstrate that MAPS and MAPS-SE enjoy sample efficiency advantage over the state-of-the-art policy improvement algorithms. Empirical results show that MAPS-SE significantly accelerates policy optimization via state-wise imitation learning from multiple oracles across a broad spectrum of control tasks in the DeepMind Control Suite.

1. Introduction

Reinforcement learning (RL) has achieved exceptional performance in many domains, such as robotics (Kober et al., 2011; 2013), video games (Mnih et al., 2013), and Go (Silver et al., 2017). However, RL tends to be highly sample inefficient in high-dimensional environments due to the need for exploration (Sutton & Barto, 2018). The sample complexity of RL limits its application to many real-world domains for which interactions with the environment can be costly. To address this challenge, imitation learning (IL) has emerged as a promising alternative. IL improves the sample efficiency of RL by training a policy to imitate the actions of an expert policy, which is typically assumed to be optimal or near-optimal. This allows the agent to learn from expert demonstrations, reducing the need for costly trial-and-error exploration.

Previous works such as behavioral cloning (Pomerleau, 1988), DAgger (Ross et al., 2011), and AggreVaTe(D) (Ross & Bagnell, 2014; Sun et al., 2017) demonstrate the effectiveness of IL with a single, near-optimal oracle. In real-world settings, however, access to an optimal oracle[1] may be prohibitively expensive or they may simply be unavailable. Instead, it is often the case that the learner has access to *multiple* suboptimal oracles (Cheng et al., 2020) that are presented to the learner as black boxes, such as in the case of autonomous driving (Lee et al., 2020), robotics (Yang et al., 2020), and medical diagnosis (Le et al., 2023). A naive approach to IL in these settings—choosing one of the oracles to imitate—can result in suboptimal performance, particularly when the relative utility of the oracles differs according to the state.

These considerations emphasize the importance of attaining sample efficiency when learning from black-box oracles by harnessing their state-wise expertise. Although practically relevant, this problem remains largely unexplored. Inspired by prior works (Cheng et al., 2020; Liu et al., 2022a;b), we propose MAPS-SE (**M**ax-aggregation **A**ctive **P**olicy **S**election with Active **S**tate **E**xploration), an algorithm that actively learns from multiple suboptimal oracles by exploiting their complementary utilities to train a policy that aims to surpass each oracle in a sample-efficient manner. MAPS-SE actively selects the oracle to roll out and uses the resulting trajectory to improve its value function estimate. Furthermore, it is integrated with an active state exploration criterion that actively selects a state to explore based on the uncertainty of its state value.

To offer a better understanding of the proposed algorithm, we first investigate a special setting that does not include

[*] Equal contribution [1] Department of Computer Science, University of Chicago, Chicago, IL, USA [2] Toyota Technological Institute at Chicago, Chicago, IL, USA. Correspondence to: Xuefeng Liu <xuefeng@uchicago.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[1] We use the terms "expert" and "oracle" interchangeably.

the active state exploration component. We refer to this algorithm as MAPS. MAPS generalizes MAMBA (Cheng et al., 2020)—the current state-of-the-art (SOTA) approach to learning from multiple oracles—by a novel utilization of the active policy selection component. We then prove improvements of MAPS over MAMBA in terms of the sample complexity. Additionally, we pinpoint issues with MAMBA, such as the uncontrolled switching of roll-in (learner policy) and roll-out (expert policy) (RIRO) and approximation errors of the gradient estimates. We then provide a theoretical analysis that demonstrates the advantages of the active state exploration component in MAPS-SE.

Lastly, we conduct extensive experiments on the DeepMind Control Suite benchmark that compare MAPS with MAMBA (Cheng et al., 2020), PPO (Schulman et al., 2017) with GAE (Schulman et al., 2015), and the best oracle from the oracle set. We empirically show that MAPS outperforms the current state-of-the-art (MAMBA). We present an analysis of these performance gains as well as the sample efficiency of our algorithm, which aligns with our theory. We then evaluate our proposed MAPS-SE algorithm and demonstrate that it provides performance gains over MAPS through various experiments.

2. Related Work

In this section, we provide a review of existing literature on model selection and imitation learning from one or multiple oracles. We refer the reader to Table 2 of Appendix B for a detailed comparison with prior art.

2.1. Policy / Model Selection

Policy selection, commonly referred to as *model selection* (Yang et al., 2022), considers the problem of selecting from or ranking a given set of policies and arises in various aspects of reinforcement learning. These include the selection of a state abstraction (Jiang, 2017; Jiang et al., 2015), the choice of a learning algorithm, and the process of feature selection (Foster et al., 2019; Pacchiano et al., 2020; Liu et al., 2022b). The significance of policy selection has garnered increased attention due to its practical implications (Paine et al., 2020; Fu et al., 2021).

Recent advancements in this field include active offline policy selection (A-OPS) (Konyushova et al., 2021), which leverages policy similarities to improve value predictions and emphasizes an active setting in which a small number of environment evaluations can enhance the prediction of the optimal policy. Existing policy selection methods have several limitations, such as assuming knowledge of an optimal policy that can be queried for each state, the inability to incorporate a learner policy as part of the selection set, being restricted to a state-less online learning setting, or poor sample efficiency. In this work, we aim to overcome these limitations by actively approximating the value function of multiple blackbox oracles and learning a state-dependent learner policy in a manner that achieves a better sample efficiency than the state-of-the-art method.

2.2. Active / Interactive Imitation Learning

Offline imitation learning (IL) methods, such as behavioral cloning, require an offline dataset of trajectories collected from one or more experts, which can lead to cascading errors in the learner policy. Interactive IL methods, such as DAgger (Ross et al., 2011) and AggreVaTe (Ross & Bagnell, 2014), assume that the learner can actively request a demonstration starting from the current state. With DAgger, the learner aims to imitate the oracle, regardless of its quality. AggreVaTe and its policy gradient variant AggreVaTeD (Sun et al., 2017) improve upon DAgger by incorporating the cost-to-go into policy training to prevent the learner from blindly following potentially unreasonable actions suggested by the oracle. However, these interactive IL approaches do not consider the cost of querying the oracle. Active imitation learning methods, on the other hand, reason over the utility of requesting a demonstration (Shon et al., 2007; Judah et al., 2012). Among them, LEAQI (Brantley et al., 2020) uses a heuristic algorithm to actively decide when not to query experts to reduce interaction costs, but it focuses on a single-oracle setting. Our work lies at the intersection of active/interactive IL and RL, allowing our method to handle both single- and multiple-oracle settings, with a focus on efficiently learning from multiple oracles.

2.3. Learning from Multiple Oracles

Several methods, including EXP3 and EXP4 (Auer et al., 2002), EXP4.P (Beygelzimer et al., 2011), and Hedge (Freund & Schapire, 1997), frame the problem of learning from multiple oracles as a contextual bandit or online learning problem. Similarly, CAMS (Liu et al., 2022a;b) considers active learning from multiple experts, but only in an online learning setting (with full observations of the oracles' losses and no state transitions). However, these methods cannot handle sequential decision-making problems such as Markov decision processes (MDPs) due to their inability to incorporate state information. In RL and IL, ILEED (Beliaev et al., 2022) differentiates between oracles based on their expertise at each state, but is limited to pure offline IL settings. MAMBA (Cheng et al., 2020) considers imitation learning from multiple experts, but it selects an oracle to query at random, compromising its sample efficiency.In contrast, our work improves sample efficiency in a multi-oracle setting while reasoning over their state-dependent relative performance by implementing active policy selection and active state exploration.

3. Preliminaries

In this work, we focus on a finite-horizon Markov decision process (MDP) denoted as $\mathcal{M}_0 = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, H \rangle$, where $\mathcal{S}$ represents the state space, $\mathcal{A}$ represents the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ denotes the unknown state transition function with $\Delta(\mathcal{S})$ being the set of distributions over set $\mathcal{S}$, $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ denotes the unknown reward function, and H represents the horizon of the episode. The policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ maps the current state to a distribution over actions. We define the set of K oracles as $\Pi = \{\pi^k\}_{k=1}^K$ and total number of episodes as N. We denote $\doteq$ as definition. For a given function $f : \mathcal{S} \to \mathbb{R}$, we define the generalized Q-function with respect to f as:

$$Q^f(s, a) \doteq r(s, a) + \mathbb{E}_{s' \sim \mathcal{P}|s,a}[f(s')] .$$

When $f(s)$ is the value function of some policy π, then the above generalized form recovers the Q-function of the policy π, i.e., $Q^\pi(s, a)$. Let $d_\pi^t \in \Delta(\mathcal{S})$ represent the distribution over states at time t under policy π given an initial state distribution $d_0 \in \Delta(\mathcal{S})$. The state visitation distribution under π can then be written as $d^\pi \doteq \frac{1}{H} \sum_{t=0}^{H-1} d_t^\pi$. The value function of the policy π under d_0 is then

$$\begin{aligned} V^\pi(d_0) &\doteq \mathbb{E}_{s_0 \sim d_0}[V^\pi(s)] \\ &\doteq \mathbb{E}_{s_0 \sim d_0}\left[\mathbb{E}_{\tau_0 \sim \rho^\pi|s_0}\left[\sum_{t=0}^{H-1} r(s_t, a_t)\right]\right], \end{aligned}$$

where $\rho^\pi(\tau_t|s_t)$ is the distribution over trajectories $\tau_t = \{s_t, a_t, \ldots, s_{H-1}, a_{H-1}\}$ under policy π. The goal is to find a policy π that maximizes the H-step return with respect to the initial state distribution d_0. The associated advantage function is expressed as

$$\begin{aligned} A^f(s, a) &\doteq Q^f(s, a) - f(s) \\ &\doteq r(s, a) + \mathbb{E}_{s' \sim \mathcal{P}|s,a}[f(s')] - f(s) . \end{aligned}$$

3.1. Algorithms for Learning from Multiple Oracles

We consider a setting in which an agent has access to a set of black-box oracles $\Pi = \{\pi^k\}_{k \in [K]}$ and propose several different approaches to learning from this set.

Single-best oracle $\pi^\star$: The most fundamental baseline is the single-best oracle $\pi^\star$, characterized by its hindsight-optimal performance, i.e., $\pi^\star := \arg\max_{\pi \in \Pi} V^\pi(d_0)$. This baseline is clearly not sufficient to demonstrate the superiority of the algorithm, as it does not take into account state-wise optimality of different oracles.

Max-following $\pi^\bullet$: The choice of the optimal oracle varies according to the state. We express this *state-wise expertise* by each oracle's value at the state $V^k(s),\ k \in 1, \ldots, K$ (we use $V^k(s)$ to denote $V^{\pi^k}(s)$ for simplicity). The *max-following* baseline then selects the best-performing oracle independently at each state according to its value function, resulting in the *max-following* policy $\pi^\bullet$

$$\pi^\bullet(a \mid s) \doteq \pi^{k^*}(a \mid s), \quad k^* \doteq \arg\max_{k \in [K]} V^k(s) . \tag{1}$$

We can interpret *max-following* policy $\pi^\bullet$ as a greedy policy that follows the best oracle in any state.

Max-aggregation $\pi^{\max}$: In this work, we use *max-aggregation* (Cheng et al., 2020) as our benchmark that looks one step ahead based on the *max-following* policy $\pi^\bullet$. We denote a natural value baseline $f^{\max}(s_t)$ for studying imitation learning with multiple oracles as

$$f^{\max}(s) \doteq \max_{k \in [K]} V^k(s) . \tag{2}$$

We then define the *max-aggregation* policy as

$$\pi^{\max}(a \mid s) \doteq \delta_{a=a^*},\ a^* = \arg\max_{a \in \mathcal{A}} A^{f^{\max}}(s, a) , \tag{3}$$

where δ is Dirac delta distribution.

When the oracle set Π contains only a single oracle, denoted as π^e, one solution is to perform one-step policy improvement from π^e. We define the resulting policy, π_+^e, as follows:

$$\pi_+^e = \arg\max_{a \in \mathcal{A}} r(s, a) + \mathbb{E}_{s' \sim \mathcal{P}|s,a}\left[V^{\pi^e}(s')\right] . \tag{4}$$

Since $V^{\pi_+^e}(s) \geq V^{\pi^e}(s)$ is guaranteed, π_+^e is uniformly better than π^e in all states. For a single oracle, $\pi^\bullet$ reduces to π^e, and $\pi^{\max}$ reduces to π_+^e. Consequently, $\pi^{\max}$ performs better than $\pi^\bullet$. In the multi-oracle case, $\pi^{\max}$ and $\pi^\bullet$ are generally not directly comparable, except when one of the oracles is uniformly better than all others. In this scenario, $\pi^\bullet$ reduces to π^e and $\pi^{\max}$ reduces to π_+^e. Therefore, $\pi^{\max}$ still outperforms $\pi^\bullet$.

To perform online imitation learning from the max-aggregation policy, $f^{\max}$ is crucial for learning the state-wise expertise from multiple oracles, as shown in Equation 3. However, $f^{\max}$ requires knowledge of each oracle's value function. In the episodic interactive IL setting, the oracles are provided in a black-box manner, i.e., without access to their value functions. To address this issue, we follow previous work (Ross & Bagnell, 2010; 2014; Sun et al., 2017) and reduce IL to an online learning problem.

Since the MDP transition and reward models are unknown, we regard d^{π_n} as the adversary (i.e., d^{π_n} could be an arbitrary distribution) in online learning, where π_n is the policy used for round n. Consequently, we define the n-th round online imitation learning loss as follows:

$$\ell_n^{\mathrm{IL}}(\pi) \doteq -H \mathbb{E}_{s \sim d^{\pi_n}}\left[A^{f^{\max}}(s, \pi)\right] . \tag{5}$$

In this work, we adapt the online loss $\ell_n(\pi;\lambda)$ of Cheng et al. (2020) to balance the effect of reinforcement leanring (i.e., explore the environment using the learner's policy only) and imitation learning (i.e., imitating the $\pi^{\max}$ policy), resulting the following n-th round loss,:

$$\begin{aligned}\ell_n(\pi;\lambda) \doteq & \underbrace{-(1-\lambda)H\mathbb{E}_{s\sim d^{\pi_n}}\left[A_\lambda^{f^{\max},\pi}(s,\pi)\right]}_{\text{Imitation Learning Loss (Eqn. 5)}} \\ & \underbrace{-\lambda\mathbb{E}_{s\sim d_0}\left[A_\lambda^{f^{\max},\pi}(s,\pi)\right]}_{\text{Reinforcement Learning Loss}},\end{aligned} \tag{6}$$

where $A_\lambda^{f^{\max},\pi}(s,a)$ is a λ-weighted advantage:

$$A_\lambda^{f^{\max},\pi}(s,a) \doteq (1-\lambda)\sum_{i=0}^{\infty}\lambda^i A_{(i)}^{f^{\max},\pi}(s,a), \tag{7}$$

which combines various i-step advantages:

$$\begin{aligned}A_{(i)}^{f^{\max},\pi}(s_t,a_t) \doteq \mathbb{E}_{\tau_t\sim\rho^\pi(\cdot|s_t)}[r(s_t,a_t)+\ldots \\ + r(s_{t+i},a_{t+i}) + f^{\max}(s_{t+i+1}))] \\ - f^{\max}(s_t).\end{aligned}$$

Consider the scenario in which we repeatedly roll-out the k-th oracle starting at state s_t. With $N_k(s_t)$ trajectories $\tau_{1,k},\tau_{2,k},\ldots,\tau_{N_k,k}$, we compute the return estimate for state s_t as the average return achieved for the trajectories,

$$\hat{V}^k(s_t) \doteq \hat{V}^{\pi_k}(s_t) \doteq \frac{1}{N_k(s_t)}\sum_{i=1}^{N_k(s_t)}\sum_{j}^{H}\lambda^j r(s_j,a_j), \tag{8}$$

where $N_k(s_t)$ is the number of trajectories starting from the initial state s_t collected by the k-th oracle.

3.2. Estimator for the Policy Gradient

We define the empirical estimate of the $\ell_n(\pi,\lambda)$ gradient as

$$\begin{aligned}\nabla\hat{\ell}_n(\pi_n;\lambda) = \\ -H\mathbb{E}_{s\sim d^{\pi_n},a\sim\pi_n(\cdot|s)}\left[\nabla\log\pi_n(a|s)A_\lambda^{\hat{f}^{\max},\pi_n}(s,a)\right],\end{aligned} \tag{9}$$

where the partial derivative is taken with respect to the parameters of the policy, denoted as π_n. Since the true value function of each oracle is unknown, we use a separate function approximator $\hat{V}^k(\cdot)$ to represent the value function of each oracle k. The approximation error then affects the estimation of $f^{\max}(\cdot)$, which is essential for computing the policy gradient in Equation 9. Therefore, the learning speed, which refers to how quickly the error in estimating $f^{\max}(\cdot)$ decreases, plays a crucial role in determining the sample efficiency. In the following sections, we discuss the limitations of the existing state-of-the-art.

Limitations of the prior state-of-the-art: A limitation of MAMBA (Cheng et al., 2020) is its high sample complexity. MAMBA estimates the policy gradient based on $\hat{f}^{\max}(s)$ and requires prolonged episodes to identify the optimal oracle for a given state due to its strategy of sampling an oracle uniformly at random. As a result, MAMBA suffers from a large accumulation of the error (and hence the regret) when identification fails (See Theorem 1). Additionally, MAMBA has no control over the approximation error of the gradient estimates when selecting which state to roll-out. Our work aims to reduce the approximation error of the estimator by actively selecting an oracle and controlling the state-wise uncertainty through active state exploration.

4. Algorithm

In this section, we focus on addressing the inefficiency of MAMBA with regards to two aspects. Firstly, MAMBA's uniform policy selection fails to effectively balance exploration and exploitation. Secondly, the selection of the state for rolling out the oracle policy plays a critical role in minimizing the sample complexity required to identify the state-wise optimal oracle. To address these issues, we introduce MAPS-SE, a novel policy improvement algorithm that actively selects among multiple oracles to imitate, as well as actively determines which state to explore. We separately introduce the active policy selection and active state exploration components in Sections 4.1 and 4.2, respectively. The full algorithm is outlined in Algorithm 1, with further implementation details provided in Appendix E.3.

4.1. Active Policy Selection

MAPS-SE reinterprets online imitation learning as an online optimization problem, thereby permitting the utilization of any readily available online learning algorithm. For our problem, the performance (or regret) crucially relies on the accuracy of the estimated $\hat{f}^{\max}(\cdot)$, which is computed as the maximum over the predicted value functions ($\hat{V}^k(\cdot)$) of every oracle $k\in[K]$. Consequently, this challenge necessitates the formulation of a method that identifies the optimal oracle in a sample-efficient fashion.

For clarity and simplicity of discussion, we refer to special case of MAPS-SE with only the *active policy selection* component as MAPS. This is equivalent to setting active state exploration SE $=$ FALSE in Line 2 of Algorithm 1. MAPS incorporates the concept of the upper confidence bound (UCB) to determine which oracle should be selected during the online learning process. In every state, MAPS decides to roll out the oracle with the highest upper confidence bound value, as this oracle has a higher probability of yielding the best performance. The data collected are subsequently used to further improve the approximation of the corresponding value function. Unlike the uniform selection

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 $n = 1, 2, \dots, N-1$ 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 12 and 13 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 12.
    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.
```

strategy in MAMBA, MAPS establishes a more effective balance between exploration and exploitation when rolling out the oracle, thereby achieving better sample efficiency. It should be highlighted that in a single-oracle setting, MAPS reduces to MAMBA. In the case of multiple oracles, our experimental results indicate that by applying reasoning to determine which oracle should be rolled out, MAPS consistently surpasses MAMBA in performance. Next, we present the details of MAPS for both discrete and continuous state spaces.

Discrete state space: When the state space is discrete, we define the best oracle $k_\star$ to select for a given state s_t as

$$k_\star = \arg\max_{k\in[K]} \hat{V}^k(s_t) + \sqrt{\frac{2H^2 \log \frac{2}{\delta}}{N_k(s_t)}}, \tag{10}$$

where $\hat{V}^k(s_t)$, $N_k(s_t)$ is defined immediately following Equation 8, and δ is a small-valued hyperparameter commonly seen in high-probability bounds. The exploration bonus term $\sqrt{\frac{2H^2 \log \frac{2}{\delta}}{N_k(s_t)}}$ is derived from Lemma D.1, which captures the standard deviation of the estimated value as well as our confidence over the estimation.

Continuous state space: In the case of a continuous state space, we employ an ensemble of prediction models to approximate the mean value $\hat{V}^k(s_t)$ and the bonus representing uncertainty, denoted as $\sigma_k(s_t)$. For each oracle policy $\pi^k \in \Pi$, we initiate a set of n independent value prediction networks with random values, and proceed to train them using random samples obtained from the oracle's trajectory buffer. We formulate the UCB term and estimate the optimal oracle policy $\pi^{k_\star}$ using the following expression:

$$k_\star = \arg\max_{k\in[K]} \hat{V}^k(s_t) + \sigma_k(s_t). \tag{11}$$

To summarize, we determine the best oracle as

$$k_\star = \arg\max_{k\in[K]} \begin{cases} \hat{V}^k(s_t) + \sqrt{\frac{2H^2 \log \frac{2}{\delta}}{N_k(s_t)}} & \text{discrete} \\ \hat{V}^k(s_t) + \sigma_k(s_t) & \text{continuous} \end{cases} \tag{12}$$

4.2. Active State Exploration

The second limitation of MAMBA is that it doesn't reason over which state the exploration should occur. As a result, MAMBA may choose to roll out an oracle policy in states for which it already has good confidence on. Therefore, building upon MAPS, we propose an *active state exploration* variant of MAPS (MAPS-SE) that decides whether to continue rolling in the current learner policy or switch to the most promising oracle, similar to MAPS, based on an uncertainty measure for the current state. In this way, MAPS-SE aims to actively select the state in which to minimize uncertainty.

The bias and variance of the gradient estimates decrease when $f^{\max}(s_t)$ returns the best-performing oracle and the associated uncertainty of the value estimation on state s_t is minimized. For a specific state s_t, MAPS-SE determines whether to proceed with the roll-out using the selected oracle policy (Eqn. 12) or continue using the learner's policy, based on the optimal oracle's uncertainty. The means by which we estimate the oracle's uncertainty varies depending on whether the state space is discrete or continuous.

When the state space is discrete, MAPS-SE identifies the best oracle $k_\star$ for state s_t according to Equation 10. In continuous state space domains, $N_{k_\star}(s_t)$ becomes intractable. In this case, as in Section 4.1, we use an ensemble of value networks to measure the uncertainty $\Gamma_{k_\star}(s_t)$. MAPS-SE then measures the exploration bonus associated with this oracle for state s_t as $\Gamma_{k_\star}(s_t)$

$$\Gamma_{k_\star}(s_t) = \begin{cases} \sqrt{\frac{2H^2 \log \frac{2}{\delta}}{N_{k_\star}(s_t)}} & \text{discrete} \\ \sigma_{k_\star}(s_t) & \text{continuous} \end{cases} \tag{13}$$

MAPS-SE decides whether to roll out the best oracle in state s_t according to how confident we are in the selection of the best oracle. We define the uncertainty threshold according to Theorem 4 as

$$\Gamma_s = \alpha \cdot \left(\sqrt{\frac{2H^2 \log \frac{2}{\delta}}{K + \left(\sum_i \frac{1}{\Delta_i^2}\right) \log\left(\frac{K}{\delta}\right)}} \right), \tag{14}$$

where α is a tunable hyperparameter. If $\Gamma_{k_\star}(s_t) \geq \Gamma_s$, MAPS-SE rolls out the identified oracle $k_\star$ at state s_t. Otherwise, MAPS-SE transitions to the next state using the learner's policy. Thus, by applying MAPS-SE, we aim to reduce the $f^{\max}$ uncertainty of state s_t under learner's trajectory below the threshold Γ_s.

5. Theoretical Analysis

5.1. Performance Guarantee of MAPS-SE

In this section, we present a theoretical analysis of the benefits offered by MAPS in both the APS and ASE configurations. For the sake of simplicity, our analysis concentrates on employing the online imitation learning loss $\ell_n^{\mathrm{IL}}(\cdot)$, or its equivalent $\ell_n(\cdot)$ with $\lambda = 0$. This setting is the same as that in Cheng et al. (2020). The proofs of the theorems in this section are deferred to the Appendix D.

Let π_n be the learner's policy generated in the n-th round of online learning in Algorithm 1. Define $\zeta_N = -\frac{1}{N}\sum_{n=1}^{N} \ell_n^{\mathrm{IL}}(\pi^{\max})$ and let

$$\epsilon_N(\Pi) = \frac{1}{N}\left(\min_{\pi\in\Pi}\sum_{n=1}^{N}\ell_n^{\mathrm{IL}}(\pi) - \sum_{n=1}^{N}\ell_n^{\mathrm{IL}}(\pi^{\max})\right),$$

$$\mathrm{Regret}_N = \sum_{n=1}^{N}\ell_n^{\mathrm{IL}}(\pi_n) - \min_{\pi\in\Pi}\sum_{n=1}^{N}\ell_n^{\mathrm{IL}}(\pi).$$

Here, $\epsilon_N(\Pi)$ captures the quality of Π, and Regret_N characterizes the convergence rate of the online algorithm.

Cheng et al. (2020) establish a meta theorem for a class of max-aggregation algorithms based on the gradient estimator provided in Equation 9. As MAPS falls into this category, the following general result also applies to Algorithm 1:

Theorem 1 (Cheng et al. (2020))**.** *Define ζ_N, $\epsilon_N(\Pi)$, and Regret$_N$ as above, where Regret$_N$ corresponds to the regret of a first-order online learning algorithm based on Equation 9. It holds that*

$$\mathbb{E}\left[\max_{n\in[N]} V^{\pi_n}(d_0)\right] \geq \mathbb{E}_{s\sim d_0}\left[\max_{k\in[K]} V^k(s)\right] + \mathbb{E}\left[\zeta_N - \epsilon_N(\Pi) - \frac{\mathit{Regret}_N}{N}\right],$$

where the expectation is over the randomness in feedback and the online algorithm.

Theorem 1 suggests that the performance of the learning algorithm (i.e., measured by $\mathbb{E}\left[\max_{n\in[N]} V^{\pi_n}(d_0)\right]$) is directly impacted by the regret $\mathbb{E}[\mathrm{Regret}_N]$. Therefore, to improve the above lower bound, it suffices to design an online learning algorithm with a better regret bound.

Consider a first-order online algorithm that satisfies

$$\mathbb{E}[\mathrm{Regret}_N] \leq O\left(\beta \cdot N + \sqrt{vN}\right), \tag{15}$$

where β and v are the bias and the variance of the gradient estimates, respectively. In the following, we show that MAPS and MAPS-SE can effectively reduce the bias term and variance term with a smaller number of oracle calls per state compared to its passive counterpart.

5.2. Advantage of MAPS Over Uniform Policy Sampling

We consider the discrete-state setting and provide an upper bound on the number of state visitations required by MAPS to identify the best oracle at any given state s.

Theorem 2. *For any state s and K oracles, the best oracle an be identified with probability at least $1-\delta$ using Algorithm 1 with active policy selection[2] if the number of visitation on this state is*

$$T = \mathcal{O}\left(K + \left(\sum_i \frac{H^2}{\Delta_i(s)^2}\right)\log\left(\frac{K}{\delta}\right)\right), \tag{16}$$

where $\Delta_i(s) := \max_{j\neq i^\star} V_{i^\star}(s) - V_j(s)$, $i^\star = \max_i V_i(s)$ is the suboptimality gap of oracle i at state s, and H is the task horizon.

Note that the bias term β in Equation (15) is caused by two factors: (a) the bias in estimating $V^{k_\star}(s)$ when $f^{\max}$ actually selects the optimal oracle at state s, and (b) the bias due to $f^{\max}$ selecting the suboptimal oracle at state s. The bias from (a) will be eliminated as N grows. According to Theorem 2, the bias introduced by (b) over N episodes is upper bounded by $T\beta^{\max}/N$, where $\beta^{\max}$ denotes the maximum difference in loss between π^n and the optimal oracle at s. Suppose that the learner visited $\mathcal{S}' \subseteq \mathcal{S}$ states in total, with $|\mathcal{S}'| \leq \mathcal{S}$. Then by the union bound, we get with probability $1 - |\mathcal{S}'|\delta$, the biased caused by (b) is

$$\mathcal{O}\left(\left(K + \left(\frac{KH^2}{\min_{i,s}\Delta_i(s)^2}\right)\log\left(\frac{K}{\delta}\right)\right)\frac{|\mathcal{S}'|\beta^{\max}}{N}\right).$$

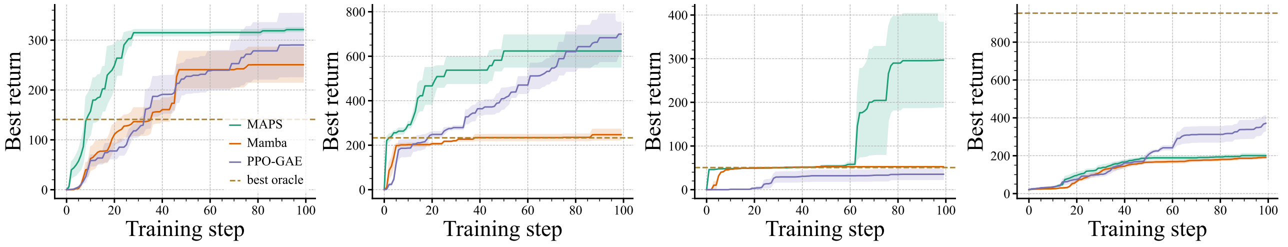
In contrast, if we replace Line 6 of Algorithm 1 by uniform sampling, MAPS reduces to MAMBA, and we have

Theorem 3. *Under the same conditions as in Theorem 2, if we adopt the uniform selection strategy, then with probability at least $1-\delta$, the best oracle can be identified if*

$$T = \mathcal{O}\left(\left(\sum_i \frac{KH^2}{\Delta_i(s)^2}\right)\log\left(\frac{K}{\delta}\right)\right). \tag{17}$$

The uniform policy selection strategy incurs a cost of $\widetilde{\mathcal{O}}(KH^2\log(K))$, while MAPS necessitates merely $\widetilde{\mathcal{O}}(K+$

[2] For analysis, we assume that the upper confidence bound of each oracle i is computed by $\widehat{V}_i(s) + \sqrt{a/N_i(t)}$ for each iteration t, and $0 \leq a \leq (25(T-K))/(36(\sum_i \Delta_i^{-2}))$ for simplicity.



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

Figure 1. 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.

$H^2 \log(K))$ oracle calls. Consequently, when dealing with a large number of oracles (that is, when K is sizeable), the superiority of MAPS over the uniform policy selection strategy becomes increasingly pronounced.

MAPS-SE, with its active state exploration, bypasses unnecessary exploration in states for which we are sufficiently confident regarding the best oracle based on its state value. Theorem 4 provides a stopping criterion for MAPS-SE:

Theorem 4. *For any state s and K experts, with probability at least $1-\delta$,* MAPS-*SE identifies the best oracle when the exploration bonus reaches the uncertainty threshold*

$$\Gamma_s = o\left(\sqrt{\frac{2H^2 \log \frac{4}{\delta}}{K + \left(\sum_i \frac{H^2}{\Delta_i^2}\right) \log\left(\frac{2K}{\delta}\right)}}\right),$$

where $\Delta_i(s) := \max_{j \neq i^\star} V_{i^\star}(s) - V_j(s)$, $i^\star = \max_i V_i(s)$ is the suboptimality gap of oracle i at state s.

When Γ_s is set too large, the learner runs the risk of not calling any oracle and consequently having too much uncertainty on the estimate $\hat{V}^{k_\star}(s)$, leading to a large bias in the gradient estimates (and therefore the regret). On the other hand, when Γ_s is too small, the learner may stop rolling in π_n early (at Line 3 of Algorithm 1), and therefore waste collecting samples on states that are sufficient confident. In the next section, we will show that with proper choices of Γ_s as suggested by Theorem 4, MAPS-SE strikes a good empirical balance between exploring uncertainty states and collecting useful training examples for improving π_n.

6. Experiments

In this section, we perform an empirical study of MAPS and MAPS-SE, comparing them to the best-oracle, PPO-GAE, and MAMBA baselines. We find that both MAPS and MAPS-SE outperform these baselines in most scenarios.

6.1. Experiment Setup

Environments. We evaluate our method on four continuous state and action environments: Cheetah-run, CartPole-swingup, Pendulum-swingup, and Walker-walk, which are part of the DeepMind Control Suite (Tassa et al., 2018).

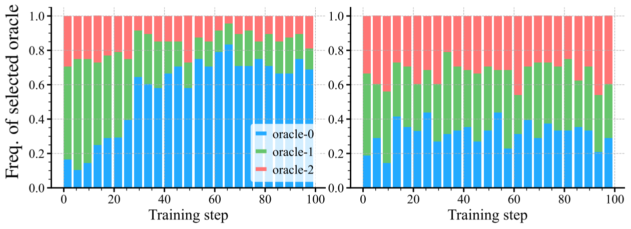
Oracle Policies. We train the oracle policies for each environment using proximal policy optimization (PPO) (Schulman et al., 2017) integrated with a generalized advantage estimate (GAE) (Schulman et al., 2015) and soft actor-critic (SAC) (Haarnoja et al., 2018). The weights of the learner policy are periodically saved as checkpoints. Generally, the average performance of the oracles increases monotonically as training progresses, which implies that each checkpoint represents an oracle of progressively improving quality.

Baseline Methods. We evaluate MAPS and MAPS-SE against three representative baselines: (1) the best oracle; (2) proximal policy optimization with a generalized advantage estimate (PPO-GAE) serving as a pure RL baseline; and (3) MAMBA, the state-of-the-art method for online imitation learning from multiple black-box oracles. For further details, we refer the reader to Appendix E.1.

Setup. To guarantee a fair evaluation, MAPS, MAPS-SE, MAMBA, and PPO-GAE are assessed based on an equal number of environment interactions. Specifically, for MAPS, MAPS-SE, and MAMBA, the oracle roll-out is used to update the learner policy, with each training iteration involving RIRO through both the learner's policy and the selected oracle's policy. Hence, we ensure that the total number of environment interactions for PPO-GAE is the same as that of MAPS, MAPS-SE and MAMBA.

6.2. Active Policy Selection

Performance Figure 1 compares MAPS against the best oracle, PPO-GAE, and MAMBA on Cheetah-run, Cartpole-swingup, Pendulum-swingup, and Walker-walk with a multi-oracle set. We observe that MAPS outperforms MAMBA and the other baselines including the best oracle in all do-



(a) MAPS (b) MAMBA

Figure 2. 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.

mains except for Walker-walk. In the case of Walker-walk, we suspect that relatively high quality of the oracles with a limited transition buffer size result in less accurate value function estimates in states that the learner encounters early in training. Further, we see that the performance MAPS improves sooner during training, demonstrating the sample efficiency advantages of MAPS.

Effect of active policy selection The superior performance of MAPS can be explained by how MAPS actively selects a better oracle, in contrast with the random selection process employed by MAMBA. Figure 2 illustrates the frequency with which each oracle is chosen by an algorithm at each training iteration. MAPS indeed has a clear preference in its oracle selection, quickly identifying the qualities of the oracles and efficiently learning from the best one to improve performance. This observation is aligned with the results from Theorem 2 and is directly evident from the improved performance and decreased bias term of the gradient estimates. In contrast, MAMBA spares a lot of resources to improve the value function estimate of bad or mediocre oracles that do not necessarily help the learner policy improve, negatively affecting sample efficiency.

6.3. Active State Exploration

Performance Active state exploration requires a threshold, Γ_s that indirectly modulates the uncertainty of a state's predicted value. The theoretical analysis in Theorem 4 reveals that this threshold is directly connected to the bias term of the gradient estimates. In essence, the threshold Γ_s trades off between bias term and sample efficiency. When the threshold is set sufficiently high, the learner policy never transitions to an oracle, resulting in substantial bias error due to the inaccurate estimate of the oracles' value functions. On the other hand, an extremely low threshold prompts the learner to immediately switch to an oracle, unless the uncertainty of the predicted value diminishes significantly. Consequently, the model spends large number of environment steps to improve the estimation of the value function.

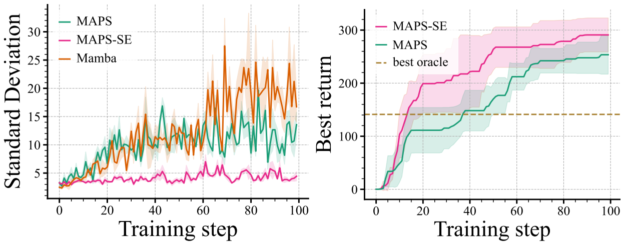


Figure 3. **Left**: A demonstration of the benefits of active state exploration in terms of a comparison between the standard deviation for switch-state for MAPS-SE (in blue), MAPS (in orange) and MAMBA (in green) in the Cheetah-run environment with the same set of oracles used in Figure 1. under a threshold $\Gamma_s = 2.5$. 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.

As illustrated in Figure 3, the MAPS-SE algorithm outperforms the MAPS algorithm by preventing unnecessary exploration. However, it is worth noting that MAPS-SE has limitations regarding its dependence on the environment and the need for non-trivial hyper-parameter tuning.

Effect of active state exploration Figure 3 illustrates that active state exploration significantly decreases the standard deviation of the predicted value at switching states. This is in contrast to the increasing trend of MAMBA and MAPS. MAMBA randomly chooses a switching timestep as well as the oracle to switch to. This risks exposing the value function (of the selected oracle) to an unfamiliar state, resulting in an increase in the standard deviation. A small standard deviation in MAPS-SE may result from a small variance v of the gradient estimates, resulting in improved performance.

7. Conclusion

We've introduced MAPS, a novel active policy improvement algorithm that learns from multiple black-box oracles. It is motivated by real-world scenarios where one wants to learn a state-dependent policy efficiently from multiple suboptimal oracles. MAPS incorporates active policy selection and active state exploration that provably improve sample complexity over the current state-of-the-art approach, MAMBA. Empirically, our results demonstrate that MAPS excels at identifying the state-wise quality of black-box oracles, learns from the best oracle more efficiently than the other baselines on various benchmarks.

Acknowledgements

We thank Ching-An Cheng for constructive suggestions. We also thank Yicheng Luo, Ziyu Ye and Zixin Ding for initial discussion. This work is supported in part by the RadBio-AI project (DE-AC02-06CH11357), U.S. Department of Energy Office of Science, Office of Biological and Environment Research, the Improve project under contract (75N91019F00134, 75N91019D00024, 89233218CNA000001, DE-AC02-06-CH11357, DE-AC52-07NA27344, DE-AC05-00OR22725), the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration (DOE DE-EE0009505), and NSF HDR TRIPODS (2216899).

References

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002. 2

Beliaev, M., Shih, A., Ermon, S., Sadigh, D., and Pedarsani, R. Imitation learning by estimating expertise of demonstrators. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1732–1748, 2022. 2, 12

Beygelzimer, A., Langford, J., Li, L., Reyzin, L., and Schapire, R. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 19–26, 2011. 2

Brantley, K., Sharaf, A., and Daumé III, H. Active imitation learning with noisy guidance. *arXiv preprint arXiv:2005.12801*, 2020. 2, 12

Cheng, C.-A., Kolobov, A., and Agarwal, A. Policy improvement via imitation of multiple oracles. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5587–5598, 2020. 1, 2, 3, 4, 6, 12

Daumé, H., Langford, J., and Marcu, D. Search-based structured prediction. *Machine Learning*, 75(3):297–325, 2009. 12

Foster, D. J., Krishnamurthy, A., and Luo, H. Model selection for contextual bandits. *arXiv preprint arXiv:1906.00531*, 2019. 2

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. 2

Fu, J., Norouzi, M., Nachum, O., Tucker, G., Wang, Z., Novikov, A., Yang, M., Zhang, M. R., Chen, Y., Kumar, A., Paduraru, C., Levine, S., and Paine, T. L. Benchmarks for deep off-policy evaluation. *arXiv preprint arXiv:2103.16596*, 2021. 2

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1861–1870, 2018. 7

Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, pp. 13–30, March 1963. 16

Jiang, N. *A theory of model selection in reinforcement learning*. PhD thesis, University of Michigan, 2017. 2

Jiang, N., Kulesza, A., and Singh, S. Abstraction selection in model-based reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 179–188, 2015. 2

Judah, K., Fern, A., and Dietterich, T. G. Active imitation learning via reduction to iid active learning. *arXiv preprint arXiv:1210.4876*, 2012. 2

Kober, J., Oztop, E., and Peters, J. Reinforcement learning to adjust robot movements to new situations. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2011. 1

Kober, J., Bagnell, J. A., and Peters, J. Reinforcement learning in robotics: A survey. *Journal of Machine Learning Research*, 32(11):1238–1274, 2013. 1

Konyushova, K., Chen, Y., Paine, T., Gulcehre, C., Paduraru, C., Mankowitz, D. J., Denil, M., and de Freitas, N. Active offline policy selection. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 24631–24644, 2021. 2, 12

Le, K. H., Tran, T. V., Pham, H. H., Nguyen, H. T., Le, T. T., and Nguyen, H. Q. Learning from multiple expert annotators for enhancing anomaly detection in medical image analysis. *IEEE Access*, 11:14105–14114, 2023. 1

Lee, G., Kim, D., Oh, W., Lee, K., and Oh, S. MixGAIL: Autonomous driving using demonstrations with mixed qualities. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5425–5430, 2020. 1

Liu, X., Xia, F., Stevens, R. L., and Chen, Y. Cost-effective online contextual model selection. *arXiv preprint arXiv:2207.06030*, 2022a. 1, 2, 12

Liu, X., Xia, F., Stevens, R. L., and Chen, Y. Contextual active online model selection with expert advice. In

Proceedings of the ICML Workshop on Adaptive Experimental Design and Active Learning in the Real World, 2022b. 1, 2

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 1

Pacchiano, A., Phan, M., Abbasi-Yadkori, Y., Rao, A., Zimmert, J., Lattimore, T., and Szepesvari, C. Model selection in contextual stochastic bandit problems. *arXiv preprint arXiv:2003.01704*, 2020. 2

Paine, T. L., Paduraru, C., Michi, A., Gulcehre, C., Zolna, K., Novikov, A., Wang, Z., and de Freitas, N. Hyperparameter selection for offline reinforcement learning. *arXiv preprint arXiv:2007.09055*, 2020. 2

Pomerleau, D. A. ALVINN: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1988. 1, 12

Ross, S. and Bagnell, D. Efficient reductions for imitation learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 661–668, 2010. 3, 12

Ross, S. and Bagnell, J. A. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014. 1, 2, 3, 12

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 627–635, 2011. 1, 2

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. 2, 7, 12

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2, 7, 12

Shon, A. P., Verma, D., and Rao, R. P. Active imitation learning. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 756–762, 2007. 2

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017. 1

Sun, W., Venkatraman, A., Gordon, G. J., Boots, B., and Bagnell, J. A. Deeply AggreVaTeD: Differentiable imitation learning for sequential prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 3309–3318, 2017. 1, 2, 3, 12

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018. 1

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D. d. L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., Lillicrap, T., and Riedmiller, M. DeepMind Control Suite. *arXiv preprint arXiv:1801.00690*, 2018. 7, 17

Yang, C., Yuan, K., Zhu, Q., Yu, W., and Li, Z. Multi-expert learning of adaptive legged locomotion. *Science Robotics*, 5(49):eabb2174, 2020. 1

Yang, M., Dai, B., Nachum, O., Tucker, G., and Schuurmans, D. Offline policy selection under uncertainty. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 4376–4396, 2022. 2

A. Notations

Table 1 summarizes the notation used in the main paper.

Notation	Description
	Problem Statement
$V, \widehat{V}$	value function , value function estimator
$\mathcal{M}_0$	$(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, finite-horizon MDP
$\mathcal{S}$	state space
$\mathcal{A}$	action space
$\mathcal{P}, \mathcal{P}\left(s' \mid s, a\right)$	transition dynamics, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta\mathcal{S}$
N	total number of episodes
$r, r\left(s, a\right)$	reward function, $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$
Π	$\Pi = \left\{\pi^k\right\}, k \in [K]$
π	policy, $\pi : \mathcal{S} \to \Delta\mathcal{A}$ maps the current state to a distribution over actions
π^k, π_n	k_{th} oracle policy, n_{th} learner policy
f	$f : \mathcal{S} \to \mathcal{R}$
$Q_t^\pi\left(s, a\right)$	$r\left(s, a\right) + \mathbb{E}_{s' \sim \mathcal{P} \mid s,a}[f\left(s'\right)]$
d_π^t	the distribution over states at time t
$\mathcal{D}_\pi$	dataset
γ	discount factor
λ	blending value of RL and IL objective
τ_t	$s_t, a_t, ..., s_{H-1}, a_{H-1}$
$\rho^\pi\left(\tau_t \mid s_t\right)$	trajectory distribution
$A^f\left(s, a\right)$	advantage function
$\pi^\bullet\left(a \mid s\right)$	$\pi^{k_s}\left(a \mid s\right)$, where $k_s := \arg\max_{k \in [K]} V^k\left(s\right)$
$f^{max}\left(s_t\right)$	$\max_{k \in [K]} V^k\left(s\right)$
$\pi^{max}\left(a \mid s\right)$	$\delta_{a=a_s}$, where $a_s := \arg\max_{a \in \mathcal{A}} A^{max}\left(s, a\right)$
π^e	single oracle
π_+^e	one-step policy improvement from π^e
$A_\lambda^{max,\pi}\left(s, a\right)$	λ-weighted advantage
$\ell_n\left(\pi; \lambda\right)$	$-(1-\lambda)H\mathbb{E}_{s \sim d}[A_\lambda^{max,\pi}\left(s, \pi\right)] - \lambda\mathbb{E}_{s \sim d_0}[A_\lambda^{max,\pi}\left(s, \pi\right)]$
$\nabla\widehat{\ell}_n\left(\pi_n; \lambda\right)$	$-H\mathbb{E}_{s \sim d^{\pi_n}}\mathbb{E}_{a \sim \pi \mid s}\left[\nabla \log \pi\left(a \mid s\right) \widehat{A}_\lambda^\pi\left(s, a\right)\right] \mid_{\pi=\pi_n}$
$\widehat{V}^{\pi_k}\left(s_t\right)$	$\frac{1}{N_k(s_t)} \sum_{i=1}^{N_k(s_t)} \sum_j^H \lambda^j r\left(s_j, a_j\right)$
$N_k\left(s_t\right)$	the number of trajectories that start from initial state s_t collected by the k_{th} oracle
$N\left(s_t\right)$	the number of trajectories that start from initial state s_t
	Algorithm
APS	Active Policy Selection
ASE	Active State Exploration
$\Gamma_{k_\star}\left(s_t\right)$, $\sigma_k\left(s_t\right)$	exploration bonus, uncertainty
Γ_s	threshold
t_e	the round switch to oracle from learner
s_{t_e}	switching state
	Analysis
ζ_N	$-\frac{1}{N}\sum_{n=1}^N \ell_n\left(\pi^{\text{max}}\right)$
$\epsilon_N\left(\Pi\right)$	$\min_{\pi \in \Pi} \frac{1}{N}\left(\sum_{n=1}^N \ell_n\left(\pi\right) - \sum_{n=1}^N \ell_n\left(\pi^{\text{max}}\right)\right)$
Regret_N	$\text{Regret}_N = \sum_{n=1}^N \ell_n\left(\pi_n\right) - \min_{\pi \in \Pi}\sum_{n=1}^N \ell_n\left(\pi\right)$
Δ_N	$\frac{-1}{N}\sum_{n=1}^N \ell_n\left(\pi^{\text{max}}\right)$
$\Delta_i(s)$	$V_{i^\star}(s) - V_i(s)$
β^{max}	the maximum difference in loss between π^n and the optimal oracle at s
v	variance

Table 1. Notations used in the main paper

B. Related Work: Problem Setup

For better positioning of this work, we highlighted the key differences and compared our setting against a few related works in this domain in the problem setup in Table 2.

Algorithm	**Criterion**	**Online**	**Stateful**	**Active**	**Interactive**	**Multiple oracles**	**Sample efficient under multi-oracles**
Behavioral cloning (Pomerleau, 1988)	IL	×	✓	×	×	×	–
SMILE (Daumé et al., 2009)	IL	×	✓	×	×	×	–
DAgger (Ross & Bagnell, 2010)	IL	✓	✓	×	✓	×	–
AggreVaTe (Ross & Bagnell, 2014)	IL	✓	✓	×	✓	×	–
PPO with GAE (Schulman et al., 2017; 2015)	RL	✓	✓	×	×	×	×
AggreVaTeD (Sun et al., 2017)	IL	✓	✓	×	✓	×	–
LEAQI (Brantley et al., 2020)	IL	✓	✓	✓	✓	×	–
MAMBA (Cheng et al., 2020)	IL + RL	✓	✓	×	✓	✓	×
A-OPS (Konyushova et al., 2021)	Policy Selection	×	×	✓	×	✓	✓
ILEED (Beliaev et al., 2022)	IL	×	✓	×	×	×	×
CAMS (Liu et al., 2022a)	Model Selection	✓	×	✓	×	✓	✓
MAPS (ours)	IL + RL	✓	✓	✓	✓	✓	✓

Table 2. Algorithms Characteristics

C. Related Work: Theoretical Guarantees

We summarize the sample complexity for identifying the best oracle per state of related algorithms in Table 3.

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

D. Proofs for Theorems

Theorem 2. *For any state s and K oracles, the best oracle an be identified with probability at least $1-\delta$ using Algorithm 1 with active policy selection[3] if the number of visitation on this state is*

$$T = \mathcal{O}\left(K + \left(\sum_i \frac{H^2}{\Delta_i(s)^2}\right)\log\left(\frac{K}{\delta}\right)\right), \tag{16}$$

where $\Delta_i(s) := \max_{j\neq i^\star} V_{i^\star}(s) - V_j(s)$, $i^\star = \max_i V_i(s)$ is the suboptimality gap of oracle i at state s, and H is the task horizon.

Proof. We first define the gap, for $i \neq i^\star$,

$$\Delta_i\left(s\right) = V_{i^\star}\left(s\right) - V_i\left(s\right). \tag{18}$$

For the case of $i = i^\star$,

$$\Delta_{i^\star} = \min_{i\neq i^\star} V_{i^\star}\left(s\right) - V_i\left(s\right). \tag{19}$$

Then, we define the following event for a state s

$$E\left(s\right) = \left\{\forall i \in [K], t\in[T], \left|\widehat{V}_{i,t}\left(s\right) - V_i\left(s\right)\right| \leq \frac{1}{5}\sqrt{\frac{a}{t}}\right\} \tag{20}$$

By Hoeffding's inequality, we will have that

$$\mathbb{P}\left[\left|\widehat{V}_{i,t}\left(s\right) - V_i\left(s\right)\right| \leq \frac{1}{5}\sqrt{\frac{a}{t}}\right] \geq 1 - 2\exp\left(\frac{-a}{50H^2}\right). \tag{21}$$

By applying the union bound, we have

$$\mathbb{P}\left[E\left(s\right)\right] \geq 1 - 2TK\exp\left(\frac{-a}{50H^2}\right). \tag{22}$$

In the next, we will show that, under the event $E\left(s\right)$, we identified the best policy if

$$\frac{1}{5}\sqrt{\frac{a}{N_i(T)}} \leq \frac{\Delta_i}{2},\ \forall i\in[K], \tag{23}$$

where $N_i(t)$ denotes the number of i.i.d samples used for estimating the value function of oracle i at t_{th} iteration. If the above holds, then we will have for $i \neq i^\star$,

$$\widehat{V}_{i,N_i(T)}(s) \leq V_i(s) + \frac{1}{5}\sqrt{\frac{a}{N_i(T)}} \tag{24a}$$

$$\leq \frac{V_{i^\star}(s) + V_i(s)}{2}. \tag{24b}$$

For $i = i^\star$, we simply have

$$\widehat{V}_{i^\star,N_{i^\star}(T)}(s) \geq V_{i^\star}(s) - \frac{1}{5}\sqrt{\frac{a}{N_{i^\star}(T)}} \tag{25a}$$

$$\geq V_{i^\star}(s) - \frac{\Delta_{i^\star}}{2} \tag{25b}$$

$$= V_{i^\star}(s) - \frac{\min_{i:i\neq i^\star} V_{i^\star}(s) - V_i(s)}{2} \tag{25c}$$

$$= \frac{\max_{i:i\neq i^\star} V_i(s) + V_{i^\star}(s)}{2}. \tag{25d}$$

[3] For analysis, we assume that the upper confidence bound of each oracle i is computed by $\widehat{V}_i(s) + \sqrt{a/N_i(t)}$ for each iteration t, and $0 \leq a \leq (25(T-K))/(36(\sum_i \Delta_i^{-2}))$ for simplicity.

Hence, by the results from equation 25d and equation 24b, we have

$$\widehat{V}_{i^\star,N_{i^\star}(T)}(s) \geq \widehat{V}_{i,N_i(T)}(s). \tag{26}$$

In the next, we will prove Equation 23 is true. This is equivalent to show the following,

$$N_i(T) \geq \frac{4}{25}\frac{a}{\Delta_i^2} \tag{27}$$

We first show that

$$N_i(t) \leq \frac{36}{25}\frac{a}{\Delta_i^2} + 1, \ \forall i \neq i^\star. \tag{28}$$

We prove the above result by induction. Firstly, when $t = 1$, this is obviously ture. Suppose that it also holds at time $t-1$. Therefore, the policy selected at time step t is not i, we will have $N_i(t) = N_i(t-1)$, and the above inequality stil holds. If the policy selected at time step t is i_{th} policy, then, this implies

$$\widehat{V}_{i,N_i(t-1)} + \sqrt{\frac{a}{N_i(t-1)}} \geq \widehat{V}_{i^\star,N_{i^\star}(t-1)} + \sqrt{\frac{a}{N_{i^\star}(t-1)}} \tag{29}$$

In addition, we have the event $E\left(s\right)$ is true. We further have

$$\widehat{V}_{i^\star,N_{i^\star}(t-1)} + \sqrt{\frac{a}{N_{i^\star}(t-1)}} \geq V_{i^\star}\left(s\right), \tag{30a}$$

$$\widehat{V}_{i,N_i(t-1)} + \sqrt{\frac{a}{N_i(t-1)}} \leq V_i\left(s\right) + \frac{6}{5}\sqrt{\frac{a}{N_i(t-1)}}, \tag{30b}$$

$$\Rightarrow \quad \frac{6}{5}\sqrt{\frac{a}{N_i(t-1)}} \geq \Delta_i. \tag{30c}$$

Since $N_i(t) = N_i(t-1)$, we get

$$N_i(t) \leq \frac{36}{25}\frac{a}{\Delta_i^2} + 1. \tag{31}$$

Then, we prove the following,

$$N_i(t) \geq \frac{4}{25}\min\left\{\frac{a}{\Delta_i^2}, \frac{25}{36}(N_{i^\star}(t)-1)\right\}, \ \ \forall i \neq i^\star. \tag{32}$$

Following the same inductive proof, we only need to show the above is true if the policy $i^\star$ is selected at iteration t. Since the event $E\left(s\right)$ is true, we further have

$$V_{i^\star}\left(s\right) + \frac{6}{5}\sqrt{\frac{a}{N_{i^\star}(t-1)}} \geq V_i\left(s\right) + \frac{4}{5}\sqrt{\frac{a}{N_i(t-1)}} \tag{33}$$

which gives us

$$N_i(t-1) \geq \frac{16}{25}\frac{a}{\left(\Delta_i + \frac{6}{5}\sqrt{\frac{a}{N_{i^\star}(t-1)}}\right)^2} \geq \frac{16}{25}\frac{a}{4\max\left\{\Delta_i, \frac{6}{5}\sqrt{\frac{a}{N_{i^\star}(t-1)}}\right\}^2}. \tag{34a}$$

$$\Rightarrow \quad N_i(t-1) \geq \frac{4}{25}\min\left\{\frac{a}{\Delta_i^2}, \frac{25}{36}(N_{i^\star}(t)-1)\right\}. \tag{34b}$$

Now, it remains to show for $i^\star$. We have

$$N_{i^\star}(T) - 1 = T - 1 - \sum_{i\neq i^\star} N_i(T) \geq T - K - \frac{36}{25}a\sum_{i\neq i^\star}\Delta_i^2 \tag{35}$$

By assumption, we have

$$a \leq \frac{25}{36} \frac{T-K}{\sum_i \Delta_i^{-2}} \tag{36}$$

we get

$$N_{i^\star}(T) - 1 \geq \frac{36}{25} a \Delta_{i^\star}^{-2}, \tag{37}$$

Now, let

$$2TK \exp\left(\frac{-2a}{50H^2}\right) = \delta. \tag{38}$$

We can conclude that, the optimal policy is identified with probability at least $1-\delta$, if

$$T = \mathcal{O}\left(K + \left(\sum_i \frac{H^2}{\Delta_i^2}\right) \log\left(\frac{K}{\delta}\right)\right) \tag{39}$$

□

Theorem 3. *Under the same conditions as in Theorem 2, if we adopt the uniform selection strategy, then with probability at least $1-\delta$, the best oracle can be identified if*

$$T = \mathcal{O}\left(\left(\sum_i \frac{KH^2}{\Delta_i(s)^2}\right) \log\left(\frac{K}{\delta}\right)\right). \tag{17}$$

Proof. We bound the following probability for iterations T,

$$\mathbb{P}\left[\widehat{V}_{i,T_i}(s) - \widehat{V}_{i^\star,T_{i^\star}}(s) \geq 0\right] = \mathbb{P}\left[\widehat{V}_{i,T_i}(s) - \widehat{V}_{i^\star,T_{i^\star}}(s) - (-\Delta_i) \geq \Delta_i\right] \tag{40}$$

By Hoeffding's inequality, we have

$$\mathbb{P}\left[\widehat{V}_{i,T_i}(s) - \widehat{V}_{i^\star,T_{i^\star}}(s) \geq 0\right] \leq \exp\left(\frac{-2\left(\lfloor T/K \rfloor \Delta_i\right)^2}{2\lfloor T/K \rfloor H^2}\right) \tag{41a}$$

$$= \exp\left(-\left\lfloor \frac{T}{K} \right\rfloor \Delta_i^2 / H^2\right). \tag{41b}$$

By union bound, we have

$$\mathbb{P}\left[\max_{i \neq i^\star} \widehat{V}_{i,T_i)}(s) - \widehat{V}_{i^\star,T_{i^\star}}(s) < 0\right] \geq 1 - \sum_i \exp\left(-\frac{T}{KH^2}\Delta_i^2\right). \tag{42}$$

By solving the following equation,

$$\sum_i \exp\left(-\frac{T}{KH^2}\Delta_i^2\right) = \delta. \tag{43}$$

We get

$$T = \mathcal{O}\left(\left(\sum_i \frac{KH^2}{\Delta_i^2}\right) \log\left(\frac{K}{\delta}\right)\right). \tag{44}$$

□

D.1. Proof of Theorem 4

Lemma D.1. *Given state s_t, policy π^k, $N_k(s_t)$ trajectories started from state s_t, with probability $1-\delta$, we have*

$$|V^{\pi^k}(s_t) - \widehat{V}^{\pi^k}(s_t)| \leq \sqrt{\frac{2H^2 \log \frac{2}{\delta}}{N_k(s_t)}}.$$

Proof of Lemma D.1. Hoeffding's inequality (Hoeffding, 1963) implies for $X_1, \ldots, X_n$ independent random variables, and W such that for all i, $|X_i| \leq W$ with probability 1, and $\mu = \frac{1}{n}\sum X_i$, we have for all $t > 0$ that

$$\mathbb{P}\left[|\bar{X} - \mu| > \frac{t}{\sqrt{n}}\right] \leq 2 \cdot \exp\left(-\frac{t^2}{2W^2}\right) \tag{45}$$

Let $\delta = 2\exp\left(-\frac{t^2}{2W^2}\right)$, we have

$$t = \sqrt{2W^2 \log\left(\frac{2}{\delta}\right)}. \tag{46}$$

By replacing t in Equation 45, we have

$$\mathbb{P}\left[|\bar{X} - \mu| > \sqrt{\frac{2W^2 \log\left(\frac{2}{\delta}\right)}{n}}\right] \leq \delta$$

$$\mathbb{P}\left[|\bar{X} - \mu| \leq \sqrt{\frac{2W^2 \log\left(\frac{2}{\delta}\right)}{n}}\right] \geq 1-\delta$$

$$\mathbb{P}\left[|V^{\pi^k}(s_t) - \widehat{V}^{\pi^k}(s_t)| \leq \sqrt{\frac{2H^2 \log \frac{2}{\delta}}{N_k(s_t)}}\right] \overset{(a)}{\geq} 1-\delta,$$

where (a) follows by replacing $\bar{X} = \widehat{V}^{\pi^k}(s_t), \mu = V^{\pi^k}(s_t), W = H, n = N_k(s_t)$.

So, with probability at least $1-\delta$, the following holds

$$|V^{\pi^k}(s_t) - \widehat{V}^{\pi^k}(s_t)| \leq \sqrt{\frac{2H^2 \log \frac{2}{\delta}}{N_k(s_t)}}.$$

□

Proof of Theorem 4. From Theorem 2, we have that for any state s, the optimal policy is identified with probability at least $1-\delta$ if

$$N(s) = \mathcal{O}\left(K + \left(\sum_i \frac{H^2}{\Delta_i^2}\right) \log\left(\frac{K}{\delta}\right)\right). \tag{47}$$

This means that, for state s, after $N(s) = \mathcal{O}\left(K + \left(\sum_i \frac{H^2}{\Delta_i^2}\right) \log\left(\frac{K}{\delta}\right)\right)$ rounds state visitation, $f^{\max}(s)$ will no long pick a suboptimal oracle with probability $1-\delta$. Thus, any increment on $N(s)$ will contribute to estimating the value of the best oracle $k_\star$. It will reduce the variance of the estimated value of oracle $k_\star$, which will lead to a decrease of the variance term v in Equation 15.

Now we would like to connect $N(s)$ with threshold Γ_s. By Lemma D.1 we know that we will require $N_k(s_t) = \frac{2H^2 \log \frac{2}{\delta}}{\Gamma^2}$ samples to achieve uncertainty Γ on state s. Combining this result with Equation (47), we know that by setting

$$\frac{2H^2 \log \frac{2}{\delta}}{\Gamma^2} = \mathcal{O}\left(K + \left(\sum_i \frac{H^2}{\Delta_i^2}\right) \log\left(\frac{K}{\delta}\right)\right)$$

and consequently,

$$\Gamma = o\left(\sqrt{\frac{2H^2 \log \frac{2}{\delta}}{K + \left(\sum_i \frac{H^2}{\Delta_i^2}\right) \log\left(\frac{K}{\delta}\right)}}\right),$$

we can ensure selecting $k_\star$ with probability $1 - 2\delta$ (by the union bound).

This is equivalent to stating that, with probability $1 - \delta$, MAPS-SE identifies $k_\star$ in state s with threshold

$$\Gamma = o\left(\sqrt{\frac{2H^2 \log \frac{4}{\delta}}{K + \left(\sum_i \frac{H^2}{\Delta_i^2}\right) \log\left(\frac{2K}{\delta}\right)}}\right), \tag{48}$$

hence completing the proof. □

E. Experiments

E.1. Baselines

PPO PPO uses a trust region optimization approach to update the policy. The trust region improves the stability of the learning process by ensuring the new policy is similar to the old policy. PPO also uses a clipped objective function for stability which limits the change in the policy at each iteration.

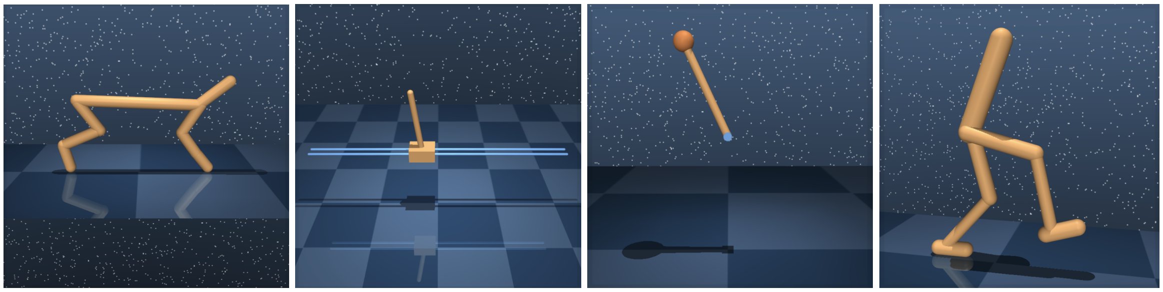
AggreVaTeD AggreVaTeD is a differentiable version of AggreVaTe which focuses on a single oracle scenario. AggreVaTeD allows us to train policies with efficient gradient update procedures. AggreVaTeD uses a deep neural network to model the policy and use differentiable imitation learning to train it. By applying differentiable imitation learning, it minimize the difference between the expert's demonstration and the learner policy behavior. AggreVaTeD learn from the expert's demonstration while interact with the environment to outperform the expert.

PPO-GAE GAE proposed a method to solving high-dimensional continuous control problems using reinforcement learning. GAE is based on a technique, *generalized advantage estimation (GAE)*. GAE is used to estimate the advantage function for updating the policy. The advantage function measures how much better a particular action is compared to the average action. Estimating the advantage function with accuracy in high-dimensional continuous control problems is challenging. In this work, we propose PPO-GAE, which combines PPO's policy gradient method with GAE's advantage function estimate, which is based on a linear combination of value function estimates. By combining the advantage of PPO and GAE, PPO-GAE achieved both sample efficiency and stability in high-dimensional continuous control problems.

MAMBA MAMBA is the SOTA work of learning from multiple oracles. It utilizes a mixture of imitation learning and reinforcement learning to learn a policy that is able to imitate the behavior of multiple experts. MAMBA is also considered as interactive imitation learning algorithm, it imitate the expert and interact with environment to improve the performance. MAMBA randomly select the state as switch point between learner policy and oracle. Then, it randomly select oracle to roll out. It effectively combines the strengths of multiple experts and able to handle the case of conflicting expert demonstrations.

E.2. DeepMind Control Suite Environments

We conduct an evaluation of MAPS, comparing its performance to the mentioned baselines, on the Cheetah-run, CartPole-swingup, Pendulum-swingup, and Walker-walk tasks sourced (see Fig. 4) from the DeepMind Control Suite (Tassa et al., 2018).



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

Figure 4. A visual illustration of the tasks we adopted for our experiments.

Algorithm 2 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 $\{\hat{V}^k\}_{k\in[K]}$
 1: for $n = 1, 2, \dots, N-1$ do
 2:     if SE is TRUE then
            ▷ /* active state exploration */
 3:         Roll-in learner 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 12 and 13 at
            each visited state $s_t$.
 4:     else
 5:         Roll-in learner policy $\pi_n$ up to $t_e \sim \text{Uniform}[H-1]$
        ▷ /* active policy selection */
 6:     Select $k_\star$ via Equation 12.
 7:     Switch to $\pi^{k_\star}$ to roll-out and collect data $\mathcal{D}_n$. We have a buffer with a fixed size ($\|\mathcal{D}_n\| = 19,200$) for each oracle,
        and we discard the oldest data when it fills up.
 8:     Update the estimate of $\widehat{V}^{k_\star}(\cdot)$ with $\mathcal{D}_n$.
 9:     Roll-out the learner policy until a buffer with a fixed size ($\|\mathcal{D}'_n\|$ = 2,048) fills up, and empty it once we use them to
        update the learner policy. This stabilizes the training compared to storing a fixed number of trajectories to the buffer
10:     Compute gradient estimator $g_n$ of $\nabla\widehat{\ell}_n(\pi_n, \lambda)$ in (9) using $\mathcal{D}'_n$.
11:     Perform PPO style policy update on policy $\pi_n$ to $\pi_{n+1}$.
```

E.3. Implementation Details of MAPS-SE

We provide the details of MAPS in Algorithm 1 as Algorithm 2. Algorithm 2 closely follows Algorithm 1 with a few modifications as follows:

- In line 7, we have a buffer with a fixed size ($\|\mathcal{D}_n\| = 19,200$) for each oracle, and we discard the oldest data when it fills up.
- In line 9, we roll-out the learner policy until a buffer with a fixed size ($\|\mathcal{D}'_n\|$ = 2,048) fills up, and empty it once we use them to update the learner policy. This stabilizes the training compared to storing a fixed number of trajectories to the buffer, as MAMBA does.
- In line 11, we adopted PPO style policy update.

E.4. Computing Infrastructure

We performed our experiments on a cluster that includes CPU nodes (about 280 cores) and GPU nodes, about 110 Nvidia GPUs, ranging from Titan X to A6000, set up mostly in 4- and 8-GPU nodes.