BLENDING IMITATION AND REINFORCEMENT LEARNING FOR ROBUST POLICY IMPROVEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

While reinforcement learning (RL) has shown promising performance, its sample complexity continues to be a substantial hurdle, restricting its broader application across a variety of domains. Imitation learning (IL) utilizes oracles to improve sample efficiency, yet it is often constrained by the quality of the oracles deployed. To address the demand for robust policy improvement in real-world scenarios, we introduce a novel algorithm, Robust Policy Improvement (RPI), which actively interleaves between IL and RL based on an online estimate of their performance. RPI draws on the strengths of IL, using oracle queries to facilitate exploration—an aspect that is notably challenging in sparse-reward RL—particularly during the early stages of learning. As learning unfolds, RPI gradually transitions to RL, effectively treating the learned policy as an improved oracle. This algorithm is capable of learning from and improving upon a diverse set of black-box oracles. Integral to RPI are Robust Active Policy Selection (RAPS) and Robust Policy Gradient (RPG), both of which reason over whether to perform state-wise imitation from the oracles or learn from its own value function when the learner's performance surpasses that of the oracles in a specific state. Empirical evaluations and theoretical analysis validate that RPI excels in comparison to existing state-of-the-art methodologies, demonstrating superior performance across various benchmark domains.

1 INTRODUCTION

Reinforcement learning (RL) has shown significant advancements, surpassing human capabilities in diverse domains such as Go (Silver et al., 2017), video games (Berner et al., 2019; Mnih et al., 2013), and Poker (Zhao et al., 2022). Despite such achievements, the application of RL is largely constrained by its substantial computational and data requirements and high sample complexity, particularly in fields like robotics (Singh et al., 2022) and healthcare (Han et al., 2023), where the extensive online interaction for trial and error is often impractical.

Imitation learning (IL) (Osa et al., 2018) improves sample efficiency by allowing the agent to replace some or all environment interactions with demonstrations provided by an oracle policy. However, the efficacy of IL heavily relies on access to near-optimal oracles for approaches like behavior cloning (Pomerleau, 1988; Zhang et al., 2018) or inverse reinforcement learning (Abbeel and Ng, 2004; Finn et al., 2016; Ho and Ermon, 2016; Ziebart et al., 2008). Interactive IL techniques, such as DAgger (Ross et al., 2011) and AggreVate(D) (Ross and Bagnell, 2014; Sun et al., 2017), similarly assume that the policy we train (i.e., *learner* policy) can obtain demonstrations from a near-optimal oracle. When we have access to rewards, the learner has the potential to improve and outperform the oracle. THOR (Sun et al., 2018) exemplifies this capability by utilizing a near-optimal oracle for cost shaping, optimizing the k-step advantage relative to the oracle's value function (referred to as "cost-to-go oracle").

However, in realistic settings, obtaining optimal or near-optimal oracles is often infeasible. Typically, learners have access to *suboptimal* and *black-box* oracles that may not offer optimal trajectories or quantitative performance measures in varying states, requiring substantial environment interactions to identify state-wise optimality. Recent approaches, including LOKI (Cheng et al., 2018) and TGRL (Shenfeld et al., 2023) aim to tackle this issue by combining IL and RL. They focus on a single-oracle setting, whereas MAMBA (Cheng et al., 2020) and MAPS (Liu et al., 2023) learn from multiple oracles. These approaches demonstrate some success, but often operate under the

assumption that at least one oracle provides optimal actions in any given state, which does not always hold in practice. In situations where no oracle offers beneficial advice for a specific state, it is more effective to learn based on direct reward feedback. Our work intend to bridge this gap by adaptively blending IL and RL in a unified framework.

Our contributions. In this paper, we present max$^+$, a learning framework devised to enable robust learning in unknown Markov decision processes (MDP) by interleaving RL and IL, leveraging multiple suboptimal, black-box oracles. Within this framework, we introduce *Robust Policy Improvement* (RPI), a novel policy gradient algorithm designed to facilitate learning from a set of black-box oracles. RPI comprises two innovative components:

1. *Robust Active Policy Selection* (RAPS), improving value function estimators of black-box oracles efficiently, and
2. *Robust Policy Gradient* (RPG), executing policy gradient updates within an actor-critic framework based on a newly devised advantage function.

Our algorithm strikes a balance between learning from these suboptimal oracles and self improvement through active exploration in states where the learner has surpassed the oracle's performance. We provide a theoretical analysis of our proposed method, proving that it ensures a performance lower bound no worse than that of the competing baseline (Cheng et al., 2020). Through extensive empirical evaluations on eight different tasks from DeepMind Control Suite (Tassa et al., 2018) and Meta-World (Yu et al., 2020), we empirically demonstrate that RPI outperforms contemporary methods and then ablate its core components.

2 Related Work

Online selection of suboptimal experts. CAMS (Liu et al., 2022b;a) learns from multiple suboptimal black-box experts to perform model selection based on a given context, but is only applicable in stateless online learning settings. Meanwhile, SAC-X (Riedmiller et al., 2018) learns the intention policies (oracles), each of which optimizes their own auxiliary reward function, and then reasons over which of these oracles to execute as a form of curriculum learning for the task policy. LfGP (Ablett et al., 2023) combines adversarial IL with SAC-X to improve exploration. Defining auxiliary rewards requires the task be decomposed into smaller subtasks, which may not be trivial. Further, they query the intention policies several times within a single episode. Unlike CAMS and SAC-X, which rely on selecting expert policies to perform sub-tasks, our approach trains an independent learner policy. It acquires expertise from sub-optimal experts using only a single oracle query per episode, thus having the potential to surpass these oracles through global exploration.

Policy improvement with multiple experts. Recent works attempt to learn from suboptimal black-box oracles while also utilizing rewards observed under the learner's policy. SFQL (Barreto et al., 2017) proposes *generalized policy improvement* with successor features. MAMBA (Cheng et al., 2020) utilizes an advantage function with geometric weighted generalization and achieves a larger policy improvement over SFQL. MAPS (Liu et al., 2023) improves on the sample efficiency and performance of MAMBA by proposing active policy selection and state exploration. However, when the oracle set is poor or the learner has outperformed every oracle, these algorithms will still resort to imitation learning with the inferior oracles. In contrast, our algorithm performs self-improvement, employing imitation learning only on states for which an oracle outperforms the learner.

3 Preliminaries

We consider a finite-horizon Markov decision process (MDP) $\mathcal{M}_0 = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, H \rangle$ with state space $\mathcal{S}$, action space $\mathcal{A}$, unknown stochastic transition dynamics $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, unknown reward function $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$, and episode horizon H. We define total number of episodes (training step) as N and assume access to a (possibly empty) set of K oracles, defined as $\Pi = \{\pi^k\}_{k=1}^K$, where $\pi_k : \mathcal{S} \to \Delta(\mathcal{A})$. The *generalized Q-function* with respect to a general function $f : \mathcal{S} \to \mathbb{R}$ is defined as:

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

When $f(s)$ is the value function of a particular policy π, the generalized Q-function can be used to recover the policy's Q-function $Q^\pi(s,a)$. We denote the *generalized advantage function* with respect to f as

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

Given an initial state distribution $d_0 \in \Delta(\mathcal{S})$, let d_t^π denote the distribution over states at time t under policy π. The state visitation distribution under π can be expressed as $d^\pi := \frac{1}{H}\sum_{t=0}^{H-1} d_t^\pi$. The value function of the policy π under d_0 is denoted as:

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

where $\rho^\pi(\tau_t \mid s_t)$ 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 $\pi = \arg\max_\pi J(\pi)$ maximizing the expected return

$$J(\pi) = \mathbb{E}_{s\sim d_0}\left[V^\pi(s)\right]. \tag{1}$$

4 Policy Improvement with Perfect Knowledge of Oracle Set

We now present a reinforcement learning framework in the presence of an imitation learning oracle set. In this section, we assume that we have perfect knowledge of the underlying MDP and each oracle's value function. We will relax these assupmtions in the next section.

Max-following. Given a collection of k imitation learning *oracles* $\Pi^{\mathrm{o}} = \{\pi^k\}_{k\in[K]}$, the *max-following* policy (Cheng et al., 2020; Liu et al., 2023) is a greedy policy that selects the oracle with the highest expertise in any given state. The max-following policy is sensitive to the quality of the oracles. Specifically, if all oracles perform worse than the learner policy at a given state, the max-following policy will still naively imitate the best (but poor) oracle. Instead, it would be more prudent to follow the learner's guidance in these cases.

Definition 4.1. (**Extended Oracle Set**). Let $\Pi^{\mathrm{o}} = \{\pi^k\}_{k\in[K]}$ be the given black-box oracle set, $\Pi^{\mathcal{L}} = \{\pi_n\}_{n\in[N]}$ be the learner's policy class, where π_n denotes that the policy has been updated for n episodes. We define the *extended oracle set* at the n-th episode as

$$\Pi^{\mathcal{E}} = \Pi^{\mathrm{o}} \cup \{\pi_n\} = \left\{\pi^1, \ldots, \pi^K, \pi_n\right\}. \tag{2}$$

Remark 4.2. The learner policy in the extended oracle set is updated at each episode.

4.1 Max$^+$ Aggregation

Based on the extended oracle set, we first introduce the advantage function $\boldsymbol{A}^+$ and the baseline value function f^+ as follows:

Definition 4.3. (**$\boldsymbol{A}^+$ Advantage Function**). Given k oracles $\pi^1, \ldots, \pi^k$ and the learner policy π_n, we define $\boldsymbol{A}^+$ advantage function as :

$$\boldsymbol{A}^+(s,a) := r(s,a) + \mathbb{E}_{s'\sim\mathcal{P}|\pi,s}[f^+(s')] - f^+(s), \tag{3}$$

where $f^+(s)$ is the baseline value function, defined as:

$$f^+(s) = \max_{k\in[|\Pi^{\mathcal{E}}|]} V^k(s), \text{ where } \left[V^k\right]_{k\in[|\Pi^{\mathcal{E}}|]} := \left[V^{\pi^1}, \ldots, V^{\pi^K}, V^{\pi_n}\right]. \tag{4}$$

$f^+(s)$ focuses exclusively on optimizing oracle selection for a single state, assuming that the selected policy will be followed for the remainder of the trajectory. To optimize the oracle selection for every encountered state, we introduce the max$^+$-following policy, which acts as a greedy policy, adhering to the optimal policy within the *extended* oracle set for any given state.

Definition 4.4. (**Max$^+$-Following Policy**). Given an extended oracle set $\Pi^{\mathcal{E}}$, we define the *max$^+$-following* policy as

$$\pi^\circ(a \mid s) := \pi^{k^\star}(a \mid s), \text{ where } k^\star := \arg\max_{k\in[|\Pi^{\mathcal{E}}|]} V^k(s), |\Pi^{\mathcal{E}}| = K+1, V^{K+1} = V^{\pi_n}. \tag{5}$$

Proposition 4.5. *Following π° is as good or better than imitating the single-best policy in $\Pi^{\mathcal{E}}$.*

With a slight abuse of notation, we use $\boldsymbol{A}^+(s, \pi^\circ)$ to denote the generalized advantage function of the policy π° at s. As proved in the Appendix, the function $\boldsymbol{A}^+(s, \pi^\circ) \geq 0$ (Appendix C.1) and that the value function for the max$^+$-following policy satisfies $V^{\pi^\circ}(s) \geq f^+(s) = \max_{k \in [|\Pi^{\mathcal{E}}|]} V^k(s)$ (Appendix C.1). This indicates that following π° is at least as good as or better than imitating a single best policy in $\Pi^{\mathcal{E}}$. Thus, π° is a valid approach to robust policy learning in the multiple oracle setting. The max$^+$-following policy π° is better than the max-following policy when the learner's policy is better than any oracle for a given state. On the other hand, when the value of a specific oracle V^k is always better than all other policies for all states, π° simply reduces to the corresponding oracle π^k. This is not ideal because the value $V^k(s)$ assumes to keep rolling out the same oracle π^k from state s until termination, without making improvement by looking one step ahead and searching for a better action. To address this, we propose the *max$^+$-aggregation* policy as follows.

Definition 4.6. (**Max$^+$-Aggregation Policy**). For state s, the max$^+$-aggregation policy $\pi^{\circledcirc}$ performs one-step improvement and takes the action with largest advantage over f^+,

$$\pi^{\circledcirc}(a \mid s) = \delta_{a=a^\star}, \text{where } a^\star = \arg\max_{a \in \mathcal{A}} \boldsymbol{A}^+(s, a) \text{ and } \delta \text{ is the Dirac delta distribution.} \quad (6)$$

Although the max$^+$-following policy π° improves upon the max-following policy, it does not perform self-improvement. In contrast, the max$^+$-aggregation policy $\pi^{\circledcirc}$ looks one step ahead and makes the largest one-step advantage improvement with respect to f^+. Thus, in the degenerate case where max$^+$-following π° is equivalent to the single best policy, $\pi^{\circledcirc}$ outperforms the best single policy in $\Pi^{\mathcal{E}}$ for all states. Since $\boldsymbol{A}^+(s, \pi^{\circledcirc}) \geq \boldsymbol{A}^+(s, \pi^\circ) \geq 0$ for any state s by Corollary C.2 and Proposition 4.5, we conclude that the max$^+$-aggregation policy $\pi^{\circledcirc}$ is a suitable policy benchmark for the robust policy learning setting as well. We note that the baseline $f^+(s)$ corresponds to the value of choosing the single-best policy in $\Pi^{\mathcal{E}}$ at state s and rolling it out throughout the rest of the episode. In contrast, π° and $\pi^{\circledcirc}$ optimize the oracle selection at every remaining step in the trajectory. In this work, we build our algorithm on top of $\pi^{\circledcirc}$.

Remark 4.7. **(Empty Oracle Set)** Up to this point, we have primarily assumed a non-empty oracle set Π^{o} and an extended oracle set of size $|\Pi^{\mathcal{E}}| \geq 2$. Given an empty oracle set Π^{o}, $\Pi^{\mathcal{E}}$ will only contain the learner policy. In this case, $f^+ \equiv V^{\pi^\circ}$ and π° will not improve, while $\pi^{\circledcirc}$ reduces to pure reinforcement learning, performing self-improvement by using the advantage function $\boldsymbol{A}^+$.

5 Robust Policy Improvement with Black-Box Oracle Set

Improving a policy from the max$^+$ baseline f^+ (Eqn. 4) is the key to learning robustly via IL and RL. Nominally, this requires knowledge of the MDP and the oracles' value functions, however, the oracles are presented to the learner as black-box policies with unknown value functions.

A critical challenge to use f^+ as a baseline is that $f^+(s) = \max_{k \in [|\Pi^{\mathcal{E}}|]} V^k(s)$ is dynamic in the training process (i.e., wrt episode number) and is improving as the learner improves. To motivate our learning algorithm, in the following analysis we resort to a slightly weaker baseline, $f_m^+ := \max_{k \in [|\Pi^{\text{o}} \cup \{\pi_m\}|]} V^k(s)$, where $m \ll N$ is an intermediate step in the learning process, and N is the total number of episodes. Similarly, we define $\boldsymbol{A}_m^+(s, a) := r(s, a) + \mathbb{E}_{s' \sim \mathcal{P}|\pi, s}[f_m^+(s')] - f_m^+(s)$, as the corresponding advantage function, and $\pi_m^{\circledcirc}$ as the corresponding max$^+$-aggregation policy by plugging $\boldsymbol{A}^+ = \boldsymbol{A}_m^+$ in Definition 4.6.

In the following, we use the baseline value f_m^+, and reformulate the problem in an online learning setting (Ross et al., 2011; Ross and Bagnell, 2014; Sun et al., 2017; Cheng et al., 2020; Liu et al., 2023) for black-box oracles. Following the analysis framework from MAMBA (Cheng et al., 2020), we first assume that the oracle value functions are known, but the MDP is unknown, and then consider the case that the value functions are unknown.

Unknown MDP with known value functions. If the MDP is unknown, we can regard d^{π_n} as an adversary in online learning and establish the online loss for episode n as

$$\ell_n(\pi) := -H \mathbb{E}_{s \sim d^{\pi_n}} \mathbb{E}_{a \sim \pi|s} [\boldsymbol{A}^+(s, a)]. \quad (7)$$

Lemma C.1 and Proposition 4.5 suggest that making $\ell_n(\pi)$ small ensures that $V^{\pi_n}(d_0)$ achieves better performance than $f_m^+(d_0)$ for $m < n$. Averaging over N episodes of online learning, we obtain

$$\frac{1}{N}\sum_{n\in[N]} V^{\pi_n}(d_0) = f_m^+(d_0) + \Delta_N - \epsilon_N\left(\Pi^{\mathcal{L}}\right) - \text{Regret}_N^{\mathcal{L}}, \tag{8}$$

where $\text{Regret}_N^{\mathcal{L}} := \frac{1}{N}(\sum_{n=1}^N \ell_n(\pi_n) - \min_{\pi\in\Pi^{\mathcal{L}}} \sum_{n=1}^N \ell_n(\pi))$ depends the learning speed of an online algorithm, $\Delta_N := -\frac{1}{N}\sum_{n=1}^N \ell_n(\pi_m^{\circledcirc})$ is the loss of the baseline max$^+$-aggregation policy $\pi_m^{\circledcirc}$, and $\epsilon_N(\Pi^{\mathcal{L}}) := \min_{\pi\in\Pi^{\mathcal{L}}} \frac{1}{N}(\sum_{n=1}^N \ell_n(\pi) - \sum_{n=1}^N \ell_n(\pi_m^{\circledcirc}))$ expresses the quality of oracle class, where $\Pi^{\mathcal{L}}$ is specified in Definition 4.1. If $\pi_m^{\circledcirc} \in \Pi^{\mathcal{L}}$, we have $\epsilon_N(\Pi^{\mathcal{L}}) = 0$. Otherwise, $\epsilon_N(\Pi^{\mathcal{L}}) > 0$. By Proposition 4.5, $\boldsymbol{A}^+(s, \pi_m^{\circledcirc}) \geq 0$ and, in turn, $\Delta_N \geq 0$. If $\pi^{\circledcirc} \in \Pi^{\mathcal{L}}$, using a no-regret algorithm to address this online learning problem will produce a policy that achieves performance of at least $\mathbb{E}_{s\sim d_0}[f_m^+(s)] + \Delta_N + O(1)$ after N iterations.

Unknown MDP with unknown value function. In reality, the value functions of the oracle set are unavailable. f^+ and $\boldsymbol{A}^+$ need to be approximated by $\hat{f}^+$ and $\hat{\boldsymbol{A}}^+$. We compute the sample estimate of the gradient as follows:

$$\nabla\hat{\ell}_n(\pi_n) = -H\mathbb{E}_{s\sim d^{\pi_n}}\mathbb{E}_{a\sim\pi_n|s}\left[\nabla\log\pi_n(a\mid s)\,\hat{\boldsymbol{A}}^+(s,a)\right] \tag{9}$$

The approximation of the value function and gradient introduces bias and variance terms in the online learning regret bound $\text{Regret}_N^{\mathcal{L}}$. We propose a general theorem to lower bound the performance:

Proposition 5.1 (Adapted from (Cheng et al., 2020))**.** *Define Δ_N, $\epsilon_N\left(\Pi^{\mathcal{L}}\right)$, f_m^+, and $\text{Regret}_N^{\mathcal{L}}$ as above, where $f_m^+ := \max_{k\in[|\Pi^o\cup\{\pi_m\}|]} V^k(s)$ for $m \leq N$, and $\text{Regret}_N^{\mathcal{L}}$ corresponds to the regret of a first-order online learning algorithm based on Eqn. 9. It holds that*

$$\mathbb{E}\left[\max_{n\in[N]} V^{\pi_n}(d_0)\right] \geq \mathbb{E}_{s\sim d_0}[f_m^+(s)] + \mathbb{E}\left[\Delta_N - \epsilon_N\left(\Pi^{\mathcal{L}}\right) - \text{Regret}_N^{\mathcal{L}}\right],$$

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

Remark 5.2. Without loss of generality, assume that $f_0^+(s) := \arg\max_{k\in[K]} V^k(s)$. Note that $f_m^+(s)$ admits a weaker baseline value than $f_n^+(s)$ for $m < n$, but *no weaker than* the max value of any oracle, $f_0^+(s)$. Therefore, as the learner improves $f_m^+(s)$, max$^+$-aggregation will have an improved lower bound. Let's consider a scenario where $m = o(N)$. In episode $n = m$, we instantiate f_m^+, and perform one-step advantage improvement over f_m^+. Since we have $f_m^+(s) > f_0^+(s)$ when $V^{\pi_m(s)} > f_0^+(s), s \sim d^{\pi_n}$, we can view max$^+$-aggregation as adding improved learner policies into Π^{o} at the end of each episode and perform one-step improvement over f^+ on the *expending* oracle set. As $\mathbb{E}_{s\sim d_0}[f_m^+(s)]$ improves, it will lead to the improvement over the original bound in Proposition 5.1.

6 Robust Policy Improvement via Actively Blending RL and IL

In this section, we present RPI, an algorithm for robust policy improvement that builds upon the max$^+$-aggregation policy. RPI consists of two main components: Robust Active Policy Selection (RAPS) and Robust Policy Gradient (RPG) that enable the algorithm to combine the advantages of reinforcement and imitation learning.

6.1 Robust Active Policy Selection

To improve the sample efficiency in learning from multiple oracles and lower the bias in $\text{Regret}_N^{\mathcal{L}}$ in Proposition 5.1 caused by the approximator of the max$^+$ baseline function $\hat{f}^+$, we propose a *robust active policy selection* strategy. We employ an ensemble of prediction models to estimate the value function for a policy, where we estimate both the mean $\hat{V}_\mu^k(s)$ and the uncertainty $\sigma_k(s)$ for a particular state s. We generate a few independent value prediction networks that are initialized randomly, and then train them using random samples from the trajectory buffer of the corresponding oracle π^k.

Algorithm 1 Robust Policy Improvement (RPI)

Input: Learner policy π_1, oracle set $\Pi = \{\pi^k\}_{k\in[K]}$, function approximators $\{\hat{V}^k\}_{k\in[K]}$, $\hat{V}_n$.
Output: The best policy among $\{\pi_1, ..., \pi_N\}$.
1: **for** $n = 1, \ldots, N-1$ **do**
2: Construct an extended oracle set $\Pi^{\mathcal{E}} = \left[\pi^1, \ldots, \pi^k, \pi_n\right]_{k\in[|\Pi|]}$.
3: Sample $t_e \in [H-1]$ uniformly random.
4: Roll-in π_n up to t_e, select $k_\star$ (Eqn. 10), and roll out $\pi^{k_\star}$ to collect data $\mathcal{D}^k$.
5: Update $\hat{V}^{k_\star}$ using $\mathcal{D}^k$.
6: Roll-in π_n for full H-horizon to collect data $\mathcal{D}'_n$.
7: Update $\hat{V}_n$ using $\mathcal{D}'_n$.
8: Compute advantage $\hat{\boldsymbol{A}}^{\text{GAE+}}$ (Eqn. 11) and gradient estimate $\hat{g}_n$ (Eqn. 14) using $\mathcal{D}'_n$.
9: Update π_n to π_{n+1} by giving $\hat{g}_n$ to a first-order online learning algorithm.

In the single oracle case, the motivation of rolling in a learner policy and rolling out an oracle policy (referred to RIRO) in prior work (e.g., DAgger, AggrevateD) is to address the distribution shift. In our work, in addition to addressing distribution shift, we aim to improve the value function estimator $\hat{V}$ of the most promising oracle on the switch state s to reduce the bias term of $\hat{f}^+$. Moreover, we seek to reduce the roll-out cost associated with querying oracles, particularly when the learner exhibits a higher expected value for the switching state. In such cases, we roll-out the learner to collect additional data to enhance its policy. We achieve this goal by comparing the UCB of oracle policies' value function and LCB of learner policy to improve the estimation of $\hat{f}^+$. We design the strategy as follows:

Let $\overline{\hat{V}^k}(s) = \hat{V}^k_\mu(s) + \sigma_k(s), \underline{\hat{V}^k}(s) = \hat{V}^k_\mu(s) - \sigma_k(s)$ be the UCB and LCB of policy k's value function for state s, respectively. We obtain the best oracle $\pi^{k_\star}$ for state s as follows:

$$k_\star = \underset{k\in[|\Pi^{\mathcal{E}}|]}{\arg\max} \left\{ \overline{\hat{V}^1}(s), \overline{\hat{V}^2}(s), ..., \overline{\hat{V}^K}(s), \underline{\hat{V}^{K+1}}(s) \right\}, \tag{10}$$

where $\underline{\hat{V}^{K+1}}$ is the confidence-aware value function approximator for the learner's policy, while $[\overline{\hat{V}^k}]_{k\in[K]}$ represents the value function approximators associated with oracle policies.

Remark 6.1. The insight behind using a confidence-aware policy selection strategy in RAPS is to find the most promising oracle for a given state and improve its value function estimate. This necessitates accounting for estimation uncertainties, which leads to the adoption of a UCB-based approach to identify the optimal oracle. Using LCB for the learner encourages exploration unless we are very sure that the learner surpasses all oracles for the given state. We empirically evaluate this in Section 7.2.

Remark 6.2. MAPS (Liu et al., 2023) introduced active policy selection strategy by selecting the best oracle to roll out and improve the value function approximation on switching state s_{t_e} according to f_0^+. In this work, we empirically improve its active policy selection strategy by utilizing the learner policy in $\Pi^{\mathcal{E}}$.

6.2 Robust Policy Gradient

We now propose robust policy gradient based on a novel advantage function, denoted by $\boldsymbol{A}^{\text{GAE+}}$ and a novel max$^+$ actor-critic framework.

$\boldsymbol{A}^{\text{GAE+}}$ advantage function. The policy gradient methods maximize the expected total reward by repeatedly estimating the gradient $g := \nabla_\theta \mathbb{E}[\sum_{t=0}^{H-1} r_t]$. The policy gradient has the form $g = \mathbb{E}_t[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{\boldsymbol{A}}_t]$ (Schulman et al., 2015; 2017), where π_θ is a stochastic learner policy and $\hat{\boldsymbol{A}}_t$ is an estimator of the advantage function at timestep t and $\mathbb{E}[\cdot]$ indicates the empirical average over a finite batch of samples, for an algorithm that alternates between sampling and optimization. $\boldsymbol{A}_t$ measures whether the action is better or worse than the current policy. Hence, the gradient term $\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{\boldsymbol{A}}_t$ points in the direction of increased $\pi_\theta(a_t|s_t)$ if and only if

Figure 1: We consider 8 tasks from DeepMind Control Suite and Meta-World. Extended results on different variants of these tasks are provided in Appendices E.2 & E.3.

$\hat{\boldsymbol{A}}_t = \hat{\boldsymbol{A}}\left(s_t, a_t\right) > 0$. For $\hat{\boldsymbol{A}}$, we propose a novel advantage function $\boldsymbol{A}^{\text{GAE+}}$ based on general advantage estimation (Schulman et al., 2015), the max$^+$ baseline f^+ and the $\boldsymbol{A}^+$ advantage function (3).

$$\hat{\boldsymbol{A}}_t^{\text{GAE}(\gamma,\lambda)+} = \hat{\delta}_t + (\gamma\lambda)\,\hat{\delta}_{t+1} + \ldots + (\lambda\gamma)^{T-t+1}\,\hat{\delta}_{T-1}, \text{ where } \hat{\delta}_t = r_t + \gamma\hat{f}^+\left(s_{t+1}\right) - \hat{f}^+\left(s_t\right), \quad (11)$$

where $T \ll H$, and γ and λ are the predefined parameters that control the bias-variance tradeoff. In this work, we use $\lambda = 0.9$ and $\gamma = 1$, and thus simplify $\hat{\boldsymbol{A}}_t^{\text{GAE}(\gamma,\lambda)+}$ as $\hat{\boldsymbol{A}}_t^{\text{GAE+}}$.

We propose a variant of the max$^+$ baseline f^+ that includes a confidence threshold Γ_s for an oracle's value estimate:

$$\hat{f}^+\left(s\right) = \max_{k \in [|\Pi^{\mathcal{E}}|]} \hat{V}_\mu^k\left(s\right), \text{ where } \hat{V}_\mu^k\left(s\right) = \hat{V}_\mu^{|\Pi^{\mathcal{E}}|}\left(s\right), \text{ when } \sigma_k\left(s\right) > \Gamma_s. \quad (12)$$

Remark 6.3. We use a threshold to control the reliability of taking the advice of an oracle, where a lower value indicates greater confidence. In our experiments, we use $\Gamma_s = 0.5$, which we have found to exhibit robust behavior (Appendix E.5).

Finally, we have the n-th episode online loss as

$$\hat{\ell}_n\left(\pi_n\right) := -H\mathbb{E}_{s\sim d^{\pi_n}}\mathbb{E}_{a\sim\pi|s}\left[\hat{\boldsymbol{A}}^{\text{GAE+}}\left(s,a\right)\right]|_{\pi=\pi_n}, \quad (13)$$

and gradient estimator as

$$\hat{g}_n = \nabla\hat{\ell}_n\left(\pi_n\right) = -H\mathbb{E}_{s\sim d^{\pi_n}}\mathbb{E}_{a\sim\pi|s}\left[\nabla\log\pi\left(a \mid s\right)\hat{\boldsymbol{A}}_t^{\text{GAE+}}\left(s,a\right)\right]|_{\pi=\pi_n}. \quad (14)$$

Max$^+$ actor-critic. We note that the RPG component (Algorithm 1, lines 8–9) can be viewed as a variant of the *actor-critic* framework, with the actor sampling trajectories that are then evaluated by the max$^+$ critic based on the $\boldsymbol{A}^{\text{GAE+}}$ advantage function (11). The policy gradient in Eqn. 14 enables the learner policy π_n to learn from high-performing oracles and to improve its own value function $\hat{V}^k$ for the states in which the oracles perform poorly.

Remark 6.4. When $\gamma = 1$, Eqn. 11 disregards the accuracy of $\hat{f}^+$, but it has high variance due to the sum of the reward terms. When $\gamma = 0$, it introduces bias, but has much lower variance. Moreover, when $\lambda = 0$ and $\gamma = 1$, the loss (13) of RPI reduced to the loss (7) under max$^+$-aggregation (6), and the performance bound for the max$^+$-aggregation policy and RPI will be equal. Thus, performing no-regret online learning with regards to Eqn. 13 has the guarantee in Proposition 5.1 and Remark 5.2. However, when $\lambda > 0$, RPI will optimize the multi-steps advantage over f^+ in Eqn. 13, while the max$^+$-aggregation policy $\pi^{\circledcirc}$ only optimizes the one-step advantage over f^+. Thus, RPI will have a smaller $\epsilon_N\left(\Pi^{\mathcal{L}}\right)$ term than max$^+$-aggregation, which improves the performance lower bound in Proposition 5.1.

Imitation, Blending and Reinforcement. Instances of $\hat{f}^+$ in Eqn. 11 may involve a combination of oracles' and learner's value functions. In a case that this does not involve the learner's value function—this is likely in the early stage of training since the learner's performance is poor—RPI performs imitation learning on the oracle policies. Once the learner policy improves and $\hat{f}^+$ becomes identical to the learner's value function, RPI becomes equivalent to the vanilla actor-critic that performs self-improvement. When it is a combination of the two, RPI learns from a blending of the learner and oracles.

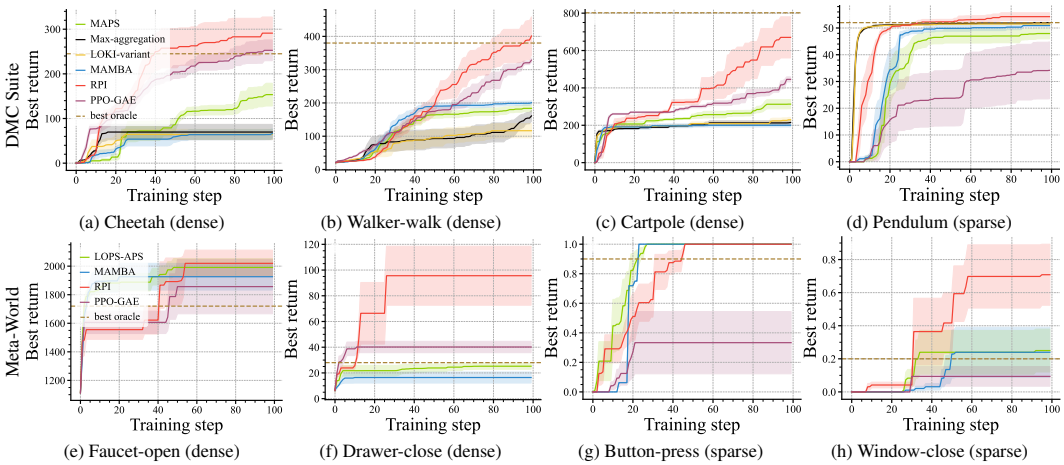


Figure 2: **Main results.** A comparison between RPI with five baselines and the best oracle (horizontal line) on Cheetah, Cartpole, Pendulum, and Walker-walk from DMC; and Window-close, Button-press, Faucet-open, and Drawer-close from Meta-World in terms of best-return-so-far with three diversified oracles. The shaded area represents the standard error based on 10 random seeds. RPI outperforms all baselines in all benchmarks.

7 Experiments

Environments. We evaluate our method on eight continuous state and action space domains: Cheetah-run, CartPole-swingup, Pendulum-swingup, and Walker-walk from the DeepMind Control Suite (Tassa et al., 2018); and Window-close, Faucet-open, Drawer-close and Button-press from Meta-World (Yu et al., 2020). In addition, we conduct experiments on a modified sparse reward Meta-World environment, which is considered to be a more challenging task. Appendix D provides further details.

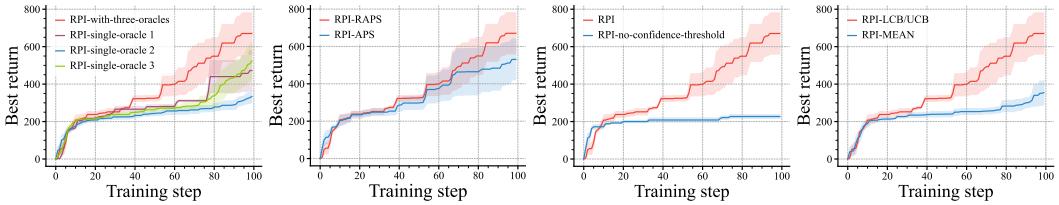
Oracles. We implement our oracles as policies trained using PPO (Schulman et al., 2017) with generalized advantage estimate (GAE) (Schulman et al., 2015) and SAC (Haarnoja et al., 2018). We save the policy weights at different points during training to achieve oracles that perform differently in different states.

Baselines. We compare RPI with five baselines: (1) PPO with GAE as a pure RL baseline; (2) Max-Aggregation (Cheng et al., 2020) as a pure IL baseline (a multiple-oracle variant of AggreVaTe(D)); (3) a variant of LOKI adapted to the multiple-oracle setting that initially performs pure IL and then pure RL; (4) MAMBA; (5) MAPS (the current state-of-the-art method)[1]; and also the best oracle in the oracle set as a reference. Appendix D provides further details.

7.1 Main Results

Figure 2 visualizes the performance of RPI and the baselines. The results show that RPI surpasses the baselines on all domains, despite variations in the black-box oracle set. Notably, the RL-based PPO-GAE baseline outperforms the IL methods in the later stages of training in most of the dense reward environments, while IL-based approaches perform better in the sparse reward domains . Pendulum-swingup (Fig. 2(d)) and window-close (Fig. 2(h)) are particularly difficult domains that involve non-trivial dynamics and sparse reward (i.e., the agent receives a reward of 1 only when the pole is near vertical, the window is closed exactly). Due to the sparse reward, the IL-based approaches are significantly more sample efficient than the RL-based approach, but their performance plateaus quickly. RPI initially bootstraps from the oracles, and due to their suboptimality, it switches to self-improvement (i.e., learning from its own value function), resulting in better performance than both IL and RL methods. These results demonstrate the robustness of RPI as it actively combines the advantages of IL and RL to adapt to various environment.

[1]Our experimental setup including the oracle set differs from that of MAPS. In this work, the learner for all baselines has access to approximately the same number of transitions and the learner does not have access to the oracle's trajectory. We reproduce the baseline performance for the MAPS' setting in Appendix E.1.



(a) Learning state-wise expertise. (b) RAPS vs APS. (c) Confidence-aware RPG (d) Confidence-aware RAPS

Figure 3: Results of ablation studies on the Cartpole environment.

7.2 Ablation Studies

Learning state-wise oracle expertise. In Fig. 3(a), we examine the ability of RPI to aggregate the expertise of multiple oracles on Cartpole. We created three diversified oracles and find that RPI achieves a return of 645 when it is able to query all three oracles, while the best return falls below 600 when given access to only a single oracle. This result demonstrates RPI's utilization of the oracles' state-wise expertise, as it achieves better performance when given access to more oracles.

Ablation on robust active policy selection. In order to understand the effectiveness of RPI's robust active policy selection strategy (RAPS), we compare it to active policy selection (APS) (Liu et al., 2023) (without the learner in RIRO (Algorithm 1, line 4)) on Cartpole. Fig. 3(b) shows that RAPS has the advantage of selecting the learner policy to roll out in states for which it outperforms the oracles, resulting in self-improvement. This leads to RAPS outperforming the APS-based approach.

Confidence-awareness in RPI. *(1) RPG*: We first perform an ablation on Cartpole to investigate the benefits of using a confidence threshold on an oracle's value estimate for RPG (Eqn. 12). We see in Fig. 3(c) that the confidence threshold enables RPG to benefit from both state-wise imitation learning from oracles with high confidence and the execution of reinforcement learning when oracles exhibit high uncertainty. Without the threshold, RPG is more vulnerable to the quality of oracle set. *(2) RAPS*: We then consider the benefits of reasoning over uncertainty to the policy selection strategy, comparing uncertainty-aware RPI-LCB/UCB (Eqn. 10) to RPI-MEAN, which does not consider uncertainty. Fig. 3(d) demonstrates the benefits of using LCB/UCB for policy selection. Addition results in Appendix E.6 reveal that RPI-LCB/UCB outperforms RPI-MEAN across all benchmarks by an *overall* margin of 40%, supporting the advantage of incorporating confidence to policy selection.

Visualizing active IL and RL. Figure 4 visualizes the active state-wise imitation and reinforcement process employed by RPI in the gradient estimator on Pendulum. The figure includes three oracle policies (in blue, orange, and green) and the learner's policy (in red). Each oracle exhibits different expertise at different stages. In the beginning, RPI only imitates the oracles, which initially have greater state-wise expertise than the learner. As the learner improves, the frequency with which RPI samples the leaner policy increases, corresponding to self-improvement. As training continues, the expertise of the learner increasingly exceeds that of the oracles, resulting in RPI choosing self-improvement more often than imitating the oracles.

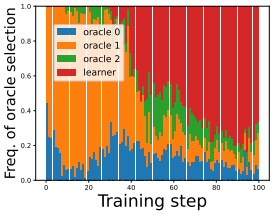


Figure 4: IL and RL.

8 Conclusion

We present max$^+$, a robust framework for IL and RL in the presence of a set of black-box oracles. Within this framework, we introduce RPI, a policy gradient algorithm comprised of two innovative components: a robust active policy selection strategy (RAPS) that enhances sample efficiency and a robust policy gradient (RPG) for policy improvement. We provide a rigorous theoretical analysis of RPI, demonstrating its superior performance compared to the current state-of-the-art. Moreover, empirical evaluations on a diverse set of tasks demonstrate that RPI consistently outperforms all IL and RL baselines, even in scenarios with limited oracle information (favoring RL) or sparse rewards (favoring IL). RPI effectively adapts to the nature of the domain and the quality of the oracles by actively interleaving IL and RL. Our work introduces new avenues for robust imitation and reinforcement learning and encourages future research on addressing more challenging tasks in robust settings, such as handling missing state or oracle information.

REFERENCES

Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2004. 1

Trevor Ablett, Bryan Chan, and Jonathan Kelly. Learning from guided play: Improving exploration for adversarial imitation learning with simple auxiliary tasks. *IEEE Robotics and Automation Letters*, 8(3):1263–1270, 2023. 2, 13

André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2

Mark Beliaev, Andy Shih, Stefano Ermon, Dorsa Sadigh, and Ramtin Pedarsani. Imitation learning by estimating expertise of demonstrators. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1732–1748, July 2022. 13

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d. O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019. 1

Kianté Brantley, Amr Sharaf, and Hal Daumé III. Active imitation learning with noisy guidance. *arXiv preprint arXiv:2005.12801*, 2020. 13

Ching-An Cheng, Xinyan Yan, Nolan Wagener, and Byron Boots. Fast policy learning through imitation and reinforcement. *arXiv preprint arXiv:1805.10413*, 2018. 1, 13, 16

Ching-An Cheng, Andrey Kolobov, and Alekh Agarwal. Policy improvement via imitation of multiple oracles. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5587–5598, 2020. 1, 2, 3, 4, 5, 8, 13, 14, 15, 16

Hal Daumé, John Langford, and Daniel Marcu. Search-based structured prediction. *Machine Learning*, 75(3):297–325, 2009.

Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 49–58, 2016. 1

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1861–1870, 2018. 8

Dong Han, Beni Mulyana, Vladimir Stankovic, and Samuel Cheng. A survey on deep reinforcement learning algorithms for robotic manipulation. *Sensors*, 23(7):3762, 2023. 1

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4565–4573, 2016. 1

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2002. 14

Ksenia Konyushova, Yutian Chen, Thomas Paine, Caglar Gulcehre, Cosmin Paduraru, Daniel J Mankowitz, Misha Denil, and Nando de Freitas. Active offline policy selection. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 24631–24644, 2021. 13

Xuefeng Liu, Fangfang Xia, Rick L Stevens, and Yuxin Chen. Cost-effective online contextual model selection. *arXiv preprint arXiv:2207.06030*, 2022a. 2, 13

Xuefeng Liu, Fangfang Xia, Rick L Stevens, and Yuxin Chen. 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. 2

Xuefeng Liu, Takuma Yoneda, Chaoqi Wang, Matthew R Walter, and Yuxin Chen. Active policy improvement from multiple black-box oracles. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 22320–22337, 2023. 1, 2, 3, 4, 6, 9, 13, 14, 15, 16, 18

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 1

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 278–287, 1999. 14

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1–2): 1–179, 2018. 1

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988. 1, 13

Martin Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom Wiele, Vlad Mnih, Nicolas Heess, and Jost Tobias Springenberg. Learning by playing solving sparse reward tasks from scratch. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4344–4353, 2018. 2, 13

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 661–668, 2010. 13

Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014. 1, 4

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 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)*, pages 627–635, 2011. 1, 4, 13

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. 6, 7, 8, 13, 16

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 6, 8, 13, 16

Idan Shenfeld, Zhang-Wei Hong, Aviv Tamar, and Pulkit Agrawal. TGRL: An algorithm for teacher guided reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023. 1, 13

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017. 1

Bharat Singh, Rajesh Kumar, and Vinay Pratap Singh. Reinforcement learning in robotic applications: a comprehensive survey. *Artificial Intelligence Review*, pages 1–46, 2022. 1

Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. Deeply AggreVaTeD: Differentiable imitation learning for sequential prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3309–3318, 2017. 1, 4, 13, 15

Wen Sun, J Andrew Bagnell, and Byron Boots. Truncated horizon policy search: Combining reinforcement learning & imitation learning. *arXiv preprint arXiv:1805.11240*, 2018. 1, 13

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1999. 13

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind control suite. *arXiv preprint arXiv:1801.00690*, 2018. 2, 8, 16

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992. 13

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning (CoRL)*, pages 1094–1100, 2020. 2, 8, 16

Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635, 2018. 1

Enmin Zhao, Renye Yan, Jinqiu Li, Kai Li, and Junliang Xing. AlphaHoldem: High-performance artificial intelligence for heads-up no-limit poker via end-to-end reinforcement learning. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 4689–4697, 2022. 1

Brian D. Ziebart, A. Maas, J. Andrew Bagnell, and Anind. K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2008. 1

A SELECTIVE COMPARISON AGAINST RELATED WORKS

Table 2: A qualitative comparison of related algorithms. The publication years are included in parentheses for reference. Algorithms designed to fit a particular criterion are marked by "✓"; criteria that are not explicitly considered in the algorithm design are marked by "×".

Algorithm	Criterion	Online	Stateful	Active	Interactive	Multiple oracles	Sample efficiency (in multiple oracles)	Robust
Behavioral Cloning (Pomerleau, 1988)	IL	×	✓	×	×	×	–	×
REINFORCE (Williams, 1992) (Sutton et al., 1999)	RL	✓	✓	×	×	×	×	–
SMILe (Ross and Bagnell, 2010)	IL	×	✓	×	×	×	–	×
DAgger (Ross et al., 2011)	IL	✓	✓	×	✓	×	–	×
PPO with GAE (Schulman et al., 2017) (Schulman et al., 2015)	RL	✓	✓	×	×	×	×	–
AggreVateD (Sun et al., 2017)	IL	✓	✓	×	✓	×	–	×
THOR (Sun et al., 2018)	IL+RL	✓	✓	×	✓	×	–	×
LOKI (Cheng et al., 2018)	IL+RL	✓	✓	×	✓	×	–	×
SAC-X (Riedmiller et al., 2018)	RL	✓	✓	×	✓	✓	×	×
LEAQI (Brantley et al., 2020)	IL	✓	✓	✓	✓	×	–	×
MAMBA (Cheng et al., 2020)	IL+RL	✓	✓	×	✓	✓	×	×
A-OPS (Konyushova et al., 2021)	Policy Sel.	×	×	✓	×	✓	✓	–
ILEED (Beliaev et al., 2022)	IL	×	✓	×	×	✓	×	×
CAMS (Liu et al., 2022a)	Model Sel.	✓	×	✓	×	✓	✓	✓
TGRL (Shenfeld et al., 2023)	IL+RL	✓	✓	×	✓	×	–	×
LfGP (Ablett et al., 2023)	IL	✓	✓	×	✓	✓	×	×
MAPS (Liu et al., 2023)	IL+RL	✓	✓	✓	✓	✓	✓	×
RPI (Ours)	IL+RL	✓	✓	✓	✓	✓	✓	✓

B ADDITIONAL BACKGROUND

B.1 ADDITIONAL ALGORITHMS FOR LEARNING FROM MULTIPLE ORACLES

In this section, we introduce a few baselines that learn from a set of black-box oracles $\Pi = \{\pi^k\}_{k\in[K]}$.

Single-best expert $\pi^\star$: The first baseline that we consider imitates a single oracle that achieves the best performance in hindsight among the oracle set, i.e., $\pi^\star := \arg\max_{\pi\in\Pi} \mathbb{E}_{s_0\sim d_0}\left[V^\pi(s_0)\right]$. After figuring out the single-best expert, this strategy simply keeps rolling out the expert. In practice, this is often inadequate as it neglects the potential benefits of suboptimal oracles at the state level.

Max-following $\pi^\bullet$: Given a collection of k imitation learning *oracles* $\Pi^{\mathrm{o}} = \{\pi^k\}_{k\in[K]}$, the *max-following* policy (Cheng et al., 2020; Liu et al., 2023) is a greedy policy that selects the oracle with the highest expertise in any given state:

$$\pi^\bullet(a \mid s) := \pi^{k^\star}(a \mid s), \quad k^\star := \arg\max_{k\in[K]} V^k(s)$$

where $V^k(s) = V^{\pi^k}(s)$ is the value function for oracle $k \in [K]$.

Max-aggregation $\pi^{\max}$: Max-aggregation (Cheng et al., 2020) performs one-step improvement based on the *max-following* policy $\pi^\bullet$. Denote a natural value baseline $f^{\max}(s_t)$ for IL with multiple oracles as

$$f^{\max}(s_t) := \max_{k\in[K]} V^k(s). \tag{15}$$

We then denote the *max-aggregation* policy as

$$\pi^{\max}(a \mid s) := \delta_{a=a^\star}, \text{where } a^\star = \arg\max_{a \in \mathcal{A}} \boldsymbol{A}^{f^{\max}}(s,a),$$
$$\boldsymbol{A}^{f^{\max}}(s,a) = r(s,a) + \mathbb{E}_{s' \sim \mathcal{P}|s,a}[f^{\max}(s')] - f^{\max}(s), \text{and } \delta \text{ is the Dirac delta distribution.} \tag{16}$$

The max-aggregation policy is a function of $f^{\max}$ and, in turn, requires knowledge of the MDP and each oracle's value function (Eqn. 15). However, in the episodic interactive IL setting, oracles are provided as black boxes and their value functions are unknown. MAMBA (Cheng et al., 2020) and MAPS (Liu et al., 2023) deal with this by reducing IL to an online learning problem and adapt the online loss defined at episode n as:

$$\ell_n(\pi;\lambda) := -(1-\lambda)H\mathbb{E}_{s\sim d^{\pi_n}}\left[\boldsymbol{A}_\lambda^{f^{\max},\pi}(s,\pi)\right] - \lambda\mathbb{E}_{s\sim d_0}\left[\boldsymbol{A}_\lambda^{f^{\max},\pi}(s,\pi)\right]. \tag{17}$$

Here, $\boldsymbol{A}_\lambda^{f^{\max},\pi}(s,a)$ is a λ-weighted advantage defined as:

$$\boldsymbol{A}_\lambda^{f^{\max},\pi}(s,a) := (1-\lambda)\sum_{i=0}^{\infty}\lambda^i \boldsymbol{A}_{(i)}^{f^{\max},\pi}(s,a), \tag{18}$$

which integrates various i-step advantages:

$$\boldsymbol{A}_{(i)}^{f^{\max},\pi}(s_t,a_t) := \mathbb{E}_{\tau_t\sim\rho^\pi(\cdot|s_t)}[r(s_t,a_t) + \cdots + r(s_{t+i},a_{t+i}) + f^{\max}(s_{t+i+1}))] - f^{\max}(s_t).$$

Limitations of the prior art. MAPS (Liu et al., 2023) and MAMBA (Cheng et al., 2020) suffer from two limitations related to their non-robustness to the choice of the oracle set. First, their online loss function (17) relies on a predetermined λ value that combines imitation learning and reinforcement learning, making their performance sensitive to the quality of the oracle set. Second, their gradient estimator utilizes the max-aggregation policy $\pi^{\max}$ and the value baseline function $f^{\max}$, both of which are dependent on the given black-box oracle set. If the oracle set includes only adversarial oracles, these methods will still try to perform imitation learning, thereby impeding policy enhancement.

B.2 Value function approximator for discrete environment

In the interactive episodic MDP, we roll out a selected oracle k, resulting in $N_k(s_t)$ trajectories $\tau_{1,k}, \tau_{2,k}, \ldots, \tau_{N_k,k}$ starting from state s_t for episode N. We determine an estimate for the return in state s_t by averaging the returns obtained across the trajectories:

$$\hat{V}^{\pi_k}(s_t) = \frac{1}{N_k(s_t)}\sum_{i=1}^{N_k(s_t)}\sum_{j}^{H}\lambda^j r(s_j,a_j). \tag{19}$$

B.3 Active policy selection

To address the sample efficiency challenge in learning from multiple experts, we reference active policy selection technique in MAPS work to select the best oracle $k^\star$ for state s_t as follows:

$$k_\star = \arg\max_{k\in[K]} \begin{cases} \hat{V}^k(s_t) + \sqrt{\frac{2H^2\log\frac{2}{\delta}}{N_k(s_t)}} & \mathcal{S} \text{ discrete} \\ \hat{V}_\mu^k(s_t) + \sigma_k(s_t) & \mathcal{S} \text{ continuous} \end{cases} \tag{20}$$

C Proofs

In the following, we provide proofs for the theoretical claims in the main paper.

Lemma C.1. *(Kakade and Langford, 2002; Ng et al., 1999) Let $f : \mathcal{S} \to \mathbb{R}$ such that $f(s_H) = 0$. For any MDP and policy π,*

$$V^\pi(d_0) - f(d_0) = H\mathbb{E}_{s\sim d^\pi}\left[\mathbf{A}^f(s,\pi)\right] \tag{21}$$

From Lemma C.1, we get the following corollary:

Corollary C.2. *(Cheng et al., 2020) If f is improvable with respect to π, then $V^{\pi}(s) \geq f(s)$, $\forall s \in \mathcal{S}$.*

Corollary C.2 indicates that a policy π outperforms all policies in $\Pi^{\mathcal{E}}$, if, for every state, there is a baseline value function f superior to that of all policies ($f(s) \geq V^{k}(s), \forall k \in [|\Pi^{\mathcal{E}}|], s \in \mathcal{S}$), while f can be improved by π (i.e., $\boldsymbol{A}^{f}(s,\pi) \geq 0$).

C.1 Proof of Proposition 4.5

Proof. Without loss of generality, let us assume the optimal oracle is oracle 1 (the first oracle) in oracle set Π,

$$\boldsymbol{A}^{+}(s,\pi^{\circ}) = r(s,\pi^{\circ}) + \mathbb{E}_{a\sim\pi^{\circ}|s}\mathbb{E}_{s'\sim\mathcal{P}|s,a}\left[f^{+}(s')\right] - f^{+}(s) \tag{22a}$$

$$\geq r(s,\pi^{\circ}) + \mathbb{E}_{a\sim\pi^{\circ}|s}\mathbb{E}_{s'\sim\mathcal{P}|s,a}\left[V^{1}(s')\right] - V^{1}(s) \tag{22b}$$

$$\geq r(s,\pi^{\bullet}) + \mathbb{E}_{a\sim\pi^{\circ}|s}\mathbb{E}_{s'\sim\mathcal{P}|s,a}\left[V^{1}(s')\right] - V^{1}(s) \tag{22c}$$

$$= \boldsymbol{A}^{V^{1}}\left(s,\pi^{1}\right) \geq 0, \tag{22d}$$

where the last step follows since $\pi^{\circ}(a|s) \geq \pi^{\bullet}(a|s) = \pi^{1}(a|s)$. Since we have $\boldsymbol{A}^{+}(s,\pi^{\circ}) \geq 0$, together with Lemma C.1, we have

$$V^{\pi^{\circ}}(s) \geq f^{+}(s) = \max_{k\in[|\Pi^{\mathcal{E}}|]} V^{k}(s). \tag{23}$$

$V^{\pi^{\circ}}(s) \geq f^{+}(s)$ indicates that following π° is equally good or superior to imitating a single best policy in $\Pi^{\mathcal{E}}$. □

C.2 Proof of Proposition 5.1

We denote $f_0^{+}(s) := \arg\max_{k\in[K]} V^{k}(s)$. According to Theorem 1 of Cheng et al. (2020), we obtain

$$\mathbb{E}\left[\max_{n\in[N]} V^{\pi_n}(d_0)\right] \geq \mathbb{E}_{s\sim d_0}[f_0^{+}(s)] + \mathbb{E}\left[\Delta_N - \epsilon_N\left(\Pi^{\mathcal{L}}\right) - \text{Regret}_N^{\mathcal{L}}\right]. \tag{24}$$

Now let $\Pi_m^{\mathcal{E}} = \Pi^{\text{o}} \cup \pi_m$. Following the same reasoning strategy as above, we will have lower bound for RPI as $\mathbb{E}_{s\sim d_0}[f_m^{+}(s)] + \mathbb{E}\left[\Delta_N - \epsilon_N\left(\Pi^{\mathcal{L}}\right) - \text{Regret}_N^{\mathcal{L}}\right]$. Since $\mathbb{E}_{s\sim d_0}[f_m^{+}(s)] \geq \mathbb{E}_{s\sim d_0}[f_0^{+}(s)]$, we have performance lower bound of RPI no worse than MAMBA.

Remark. MAPS (Liu et al., 2023) retains MAMBA's lower bound but enhances sample efficiency and reduces the bias in $\text{Regret}_N^{\mathcal{L}}$. The inherent uncertainty of the optimal policy $\pi^{\star} \in \Pi_m^{\mathcal{E}}$ makes an unbiased f^{+} estimate challenging. The regret term $\text{Regret}_N^{\mathcal{L}}$ is bounded by:

$$\mathbb{E}\left[\text{Regret}_N^{\mathcal{L}}\right] \leq O\left(\left(\beta^{+} + \beta^{\epsilon}\right)N + \sqrt{vN}\right),$$

where β^{+} is the estimation bias that results from selecting the non-optimal policy $\hat{\pi}^{\star}$ in $\Pi_m^{\mathcal{E}}$ for a given state, and β^{ϵ} is the value estimation error w.r.t. the true value for given state of selected policy $\hat{\pi}^{\star}$ and v represents the variance term.

MAPS improves upon MAMBA's sample complexity, reducing bias in its regret bound via an active policy selection mechanism. Our work builds on MAPS, emphasizing empirical enhancements in active policy selection with the integration of the learner policy in $\Pi_m^{\mathcal{E}}$.

D Experimental Details

D.1 Baselines

AggreVaTeD AggreVaTeD (Sun et al., 2017) 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 models the policy as a deep neural network and trains the policy using differentiable imitation learning. By applying differentiable imitation learning, it minimize the difference between the expert's demonstration and the learner policy behavior. AggreVaTeD learns from the expert's demonstration while interact with the environment to outperform the expert.

Max-Aggregation We have developed a variant of the Max-aggregation policy as outlined in Equation (16) that is specifically designed for pure imitation learning using multiple oracle sets. When utilizing a single oracle, it effectively reduces to AggreVateD. Our approach builds on the existing MAMBA framework by setting the lambda value in the loss function to zero. While max-aggregation may not always yield the optimal policy, it offers the advantage of being able to achieve results with fewer samples, making it a more sample-efficient option.

LOKI-variant LOKI (Cheng et al., 2018) is strategy for policy learning that combines the imitation learning and reinforcement learning objectives in a two-stage manner for the single oracle setting. In the first stage, LOKI performs imitation learning for a small but random number of iterations and then switches to policy gradient reinforcement learning method for the second stage. LOKI is able to outperform a sub-optimal expert and converge faster than running policy gradient from scratch. In this work, we propose a variation of LOKI that adapts to multiple experts. During the first-half of training (i.e., the first stage) we perform Max-aggregation style imitation learning, and then perform pure reinforcement learning as the second stage.

PPO-GAE Schulman et al. (2015) proposed the generalized advantage estimator (GAE) as a means of solving high-dimensional continuous control problems using reinforcement learning. 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 (Schulman et al., 2017) achieved both sample efficiency and stability in high-dimensional continuous control problems.

MAMBA MAMBA (Cheng et al., 2020) 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 imitates 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 selects the oracle to roll out. It effectively combines the strengths of multiple experts and able to handle the case of conflicting expert demonstrations.

MAPS MAPS (Liu et al., 2023) is a policy improvement algorithm that performs imitation learning from multiple suboptimal oracles. It actively chooses the oracle to imitate based on their value function estimates and identifies the states that require exploration. By introducing two variations, Active Policy Selection (APS) and Active State Exploration (ASE), MAPS improves the sample efficiency of MAMBA. The MAPS variant selects the most promising oracle, denoted as $k_\star$, for rollout, utilizing the resulting trajectory to refine the value function estimate $\hat{V}^{k_\star}(s_t)$. This approach aims to minimize the chances of selecting an inferior oracle for a given state s_t, thereby reducing both the sample complexity and gradient estimation bias. On the other hand, the ASE variant of MAPS deliberates whether to continue with the current policy or switch to what is believed to be the most promising oracle, similar to APS, by leveraging an uncertainty measure over the current state. In this study, we adopt MAPS variant as our baseline method.

D.2 GYM ENVIRONMENTS

We evaluate RPI and compare its performance to the aforementioned baselines on the Cheetah-run, CartPole-swingup, Pendulum-swingup, and Walker-walk tasks from the DeepMind Control Suite (Tassa et al., 2018) and Window-close, Faucet-open, Drawer-close and Button-press from Meta-World (Yu et al., 2020). In addition, we conduct experiments on a modified sparse reward Meta-World environment, which is considered to be a more challenge task.

D.3 SETUP

Setup. In order to ensure a fair evaluation, all baselines are assessed using an equal number of environment interaction steps. Each training iteration involved a policy rollout for the same number of steps. We note that there is a discrepancy in the amount of data available to the learners of RPI and PPO-GAE. RPI (MAPS, MAMBA, Max-Aggregation) uses some of the interactions to learn the value function for each Oracle, which results in relatively less data for its learner, whereas PPO-GAE utilizes all the environment interactions to update all benefits for its learner policy. Thus, in this work, we balance the transition buffer size for each algorithm to make them have approximately same number of stored transitions for learner policy improvement. We average the result based on 5-10 trails.

D.4 IMPLEMENTATION DETAILS OF RPI

Algorithm 2 Robust Policy Improvement (RPI)

Input: Learner policy π_1, oracle set $\Pi = \{\pi^k\}_{k\in[K]}$, function approximators $\{\hat{V}^k\}_{k\in[K]}$, $\hat{V}_n$

Output: The best policy in $\{\pi_1, ..., \pi_N\}$.

1: **for** $n = 1, 2, \ldots, N-1$ **do**
2: Construct an extended oracle set $\Pi^{\mathcal{E}} = \left[\pi^1, \pi^2, \ldots, \pi^k, \pi_n\right]_{k\in[|\Pi|]}$.
3: Sample $t_e \in [H-1]$ uniformly random. 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.
4: Switch to $\pi^{k_\star}$
5: Roll-in π_n up to t_e, select $k_\star$ (10), and roll out $\pi^{k_\star}$ to collect data $\mathcal{D}^k$.
6: Update $\hat{V}^{k_\star}$ using $\mathcal{D}^k$.
7: Roll-out the learner policy π_n 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.
8: Update $\hat{V}_n$ using $\mathcal{D}'_n$.
9: Compute advantage $\hat{\boldsymbol{A}}^{\text{GAE+}}$ (11) and gradient estimate $\hat{g}_n$ (14) using $\mathcal{D}'_n$.
10: Perform PPO style policy update on policy π_n to π_{n+1}.

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

- In line 5, we use a buffer with a fixed size ($\|\mathcal{D}_n\| = 19,200$) for each oracle, and discard the oldest data when it fills up.
- In line 7, we roll-out the learner policy until the buffer reaches a fixed size ($\|\mathcal{D}'_n\|$ = 2,048), and then empty it once we use the roll outs to update the learner policy. This stabilizes the training compared to storing a fixed number of trajectories in the buffer, as MAMBA does.
- In line 10, we use PPO with a max$^+$ actor-critic style policy update.
- We pretrain the value function $\hat{V}^k$ of oracle k before the main training loop, with trajectories generated by rolling out π^k from the initial states. In the main training loop, we train $\hat{V}^k$ using the corresponding rolled-out trajectories, bootstrapped only by itself. This is the same strategy as in MAMBA and MAPS. Similarly, we train the learner value function $\hat{V}_n$ using only the trajectories rolled-in with π_n, bootstrapped only by itself.

D.5 COMPUTING INFRASTRUCTURE

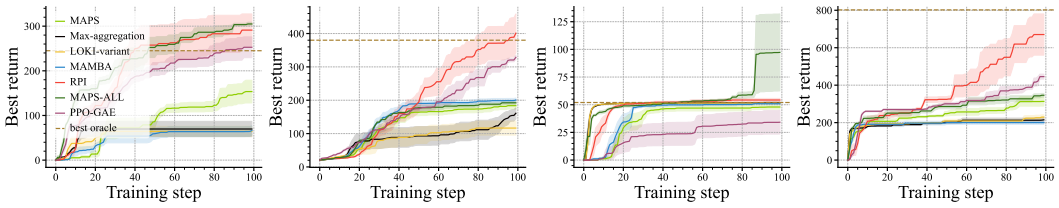
We performed our experiments on a cluster that includes CPU nodes (approximately 280 cores) and GPU nodes (approximately 110 Nvidia GPUs, ranging from Titan X to A6000, set up mostly in 4- and 8-GPU configurations).

E SUPPLEMENTAL EXPERIMENTS

E.1 COMPARING RPI AGAINST BASELINES WITH A DATA ADVANTAGE

In the main paper, we followed an experimental setup where we assumed that the learner had access to approximately the same quantity of transitions, while lacking access to the oracle's offline trajectory. However, some of the baseline algorithms, such as MAPS (Liu et al., 2023) were originally evaluated under a different setting in the literature: They assume that the learner can access additional data from the oracles' pre-trained offline dataset. In this section, we run MAPS under such a setting, in order to provide more comprehensive evaluation that is consistent with the literature. Note that under this experimental setup, MAPS has approximately twice the amount of data compared to RPI. We refer to this variant as MAPS-ALL.

In contrast to MAPS in our original configuration, the performance of MAPS-ALL doubles in the Cheetah environment (as shown in Figure 5(a)) and the Pendulum environment (as depicted in Figure 5(c)). Moreover, the performance of MAPS-ALL surges between middle and end of episodes in the Pendulumn environment. This behavior mirrors what was reported in the original MAPS paper as well. In the Walker-walk environment (as illustrated in Figure 5(b)), MAPS-ALL demonstrates an approximate 10% improvement. For the Cartpole environment (Figure 5(d)), MAPS-ALL's performance increases by around 20%. MAPS-ALL exhibits overall performance similar to that of its original paper, with any differences caused by the difference in the oracle set. Notably, as a result, MAPS-ALL distinctly outperforms RPI only in the Pendulum environment. RPI's performance remains comparable to MAPS-ALL in the Cheetah environment and significantly surpasses the MAPS-ALL baseline in the Walker-Walker and Cartpole environments, despite utilizing much less data.

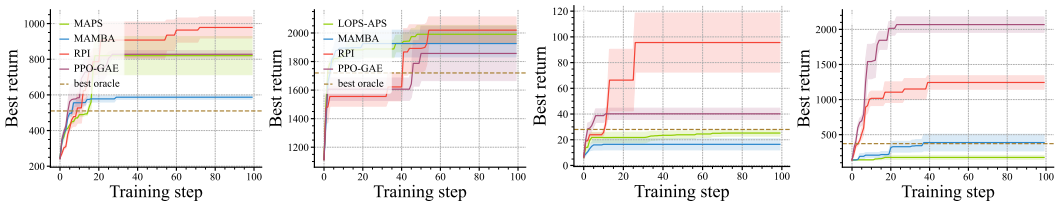


(a) Cheetah (dense) (b) Walker-walk (dense) (c) Pendulum (sparse) (d) Cartpole (dense)

Figure 5: Running MAPS in the original paper's setting.

E.2 META-WORLD EXPERIMENTS (DENSE REWARD)

In Fig. 6, we conducted additional experiments comparing RPI and state-of-the-art (SOTA) methods {MAMBA, MAPS}, as well as the best-performing oracle and PPO-GAE, across the Meta-World benchmarks. The tasks are including (1) `window-close`, (2) `faucet-open`, (3) `drawer-close`, and (4) `button-press`. RPI demonstrates superior performance compared to all baselines in the majority of environments, with the exception of the button-press task.

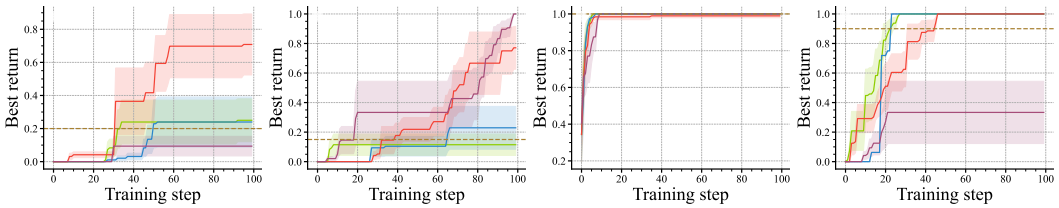


(a) Window-close (dense) (b) Faucet-open (dense) (c) Drawer-close (dense) (d) Button-press (dense)

Figure 6: Experimental results on the Meta-World benchmark with dense reward.

E.3 META-WORLD EXPERIMENTS (SPARSE REWARD)

In Fig. 7, to further demonstrate the advantages of imitation learning, we modified the Meta-World environment to create a more challenging sparse reward environment. In this environment, the agent only receives a reward of 1 upon success; otherwise, it receives a reward of 0. We then compared the performance of the RPI and state-of-the-art (SOTA) imitation learning-based methods MAMBA, MAPS, as well as the pure RL method PPO-GAE, and the best-performing oracle across the Meta-World benchmarks. The tasks include (1) `window-close`, (2) `faucet-open`, (3) `drawer-close`, and (4) `button-press`. In these sparse reward environments, when provided with a good oracle, the imitation learning-based approach demonstrates its advantage over the pure RL approach. RPI, MAPS, and Mamba outperform PPO-GAE by a factor of 3 in the `button-press` environment. When provided with a bad oracle, RPI can still outperform MAPS and MAMBA in the `faucet-open` environment. Moreover, even with a poor oracle, RPI outperforms both IL-based approaches (MAMBA, MAPS) and the RL-based approach (PPO-GAE) in the `window-close` environment, showcasing that RPI enjoys benefits from both RL and IL aspects.



(a) Window-close (sparse) (b) Faucet-open (sparse) (c) Drawer-close (sparse) (d) Button-press (sparse)

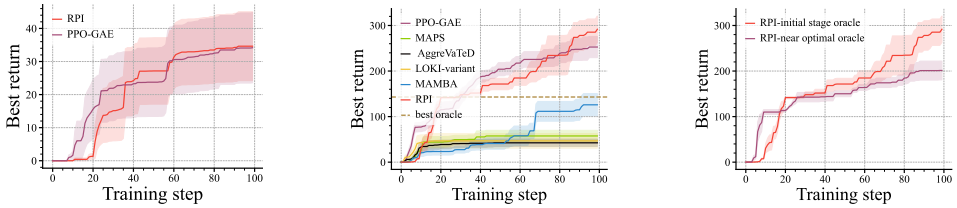
Figure 7: Experimental results on the Meta-World benchmark with sparse reward.

E.4 SINGLE/EMPTY ORACLE SET

In Fig. 2, we mainly discuss experiment on multiple oracle set. In Fig. 8, we demonstrate that RPI is also robust enough to handle single or empty oracle set.

Single oracle setting. In Fig. 8(b), we demonstrate that RPI outperforms all other baselines in a single oracle setting as well. This is consistent with the results observed in the multiple-experts setting. Fig. 8(c) demonstrates that providing an oracle with mediocre performance to RPI boosts the performance rather than providing a near-optimal oracle. Since we train oracles' value functions from the oracle rollouts, the value function of the near-optimal oracle may not have seen the "bad" states that the learner policy would encounter in the early stage. This leads to inaccuracy in the predicted values for such states. In comparison, the value function for a mediocre oracle would be able to produce accurate predictions on such states.

No oracle environment. When there are no experts available, the performance of imitation learning-based approaches will inevitably degrade. However, as shown in Fig. 8(a), RPI can adapt to such a scenario by regressing to pure reinforcement learning. Since we extend RPI based on PPO-GAE, it achieves a similar level of performance to PPO-GAE when the oracle set is empty.



(a) No oracles (b) Good single oracle (c) Helpful oracle ablation

Figure 8: **Ablation study.** (a) Comparing RPI and PPO-GAE under no oracle Pendulum environment. (b) Comparing RPI and six baseline under single expert Cheetah environment. (c) ablation study on oracle quality under Cheetah environment.

E.5 Ablation on confidence threshold Γ_s

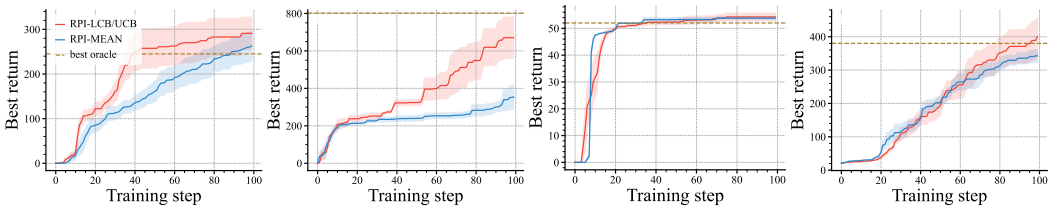
In Table 3, we perform an ablation study of threshold Γ_s in Eqn. 12 of robust policy gradient component. We treat Γ_s as a hyperparameter the choice of which depends on the user's risk aversion. Empirically, we find that a setting of $\Gamma_s = 0.5$ robustly works well in nearly every setting in our experiments across Deepmind Control suite, Meta-World (Dense reward), and Meta-world (Sparse reward) environment, the one exception being the Pendulum domain. To better understand how the choice of Γ_s effects overall performance, we conducted a set of experiments in which we ran RPI on the Deepmind Control Suite using different values for $\Gamma_s \in [0, 0.5, 1, 3, 5]$. As the following table shows, setting $\Gamma_s = 0.5$ yields the best performance for all but the Pendulum environment.

In practice, one can first use roll outs of each oracle to estimate the standard deviation and associated confidence intervals of their ensemble values. A conservative user could then start by setting Γ_s based on a probabilistic lower-bound of σ and subsequently tune the hyperparameter according to user's risk aversion preference.

Environment	Episode	$\Gamma_s = 0$	$\Gamma_s = 0.5$	$\Gamma_s = 1$	$\Gamma_s = 3$	$\Gamma_s = 5$
Cheetah	100	252.7 ± 23.2	**291.2** ± 36.3	251.4 ± 15.1	53.4 ± 20.0	81.3 ± 20.8
Walker-walk	100	328.7 ± 6.5	**402.2** ± 57.7	253.0 ± 43.5	31.8 ± 1.4	38.2 ± 1.9
Pendulum	100	34.2 ± 23.5	38.0 ± 10.4	45.6 ± 2.3	**54.3** ± 1.5	52.1 ± 0.1
Cartpole	100	445.7 ± 13.5	**670.4** ± 110.1	394.8 ± 50.6	301.7 ± 60.0	303.2 ± 4.0

Table 3: Tuning the confidence threshold Γ_s.

E.6 Ablation on UCB/LCB policy selection



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

Figure 9: Experimental results on ablation study on confidence aware UCB/LCB vs MEAN policy selection.

Environment	Episode	RPI-RAPS(LCB/UCB)	RPI-MEAN
Cheetah	100	**291.2** ± 36.3	263.0 ± 33.7
Walker-walk	100	**402.2** ± 57.7	342.7 ± 18.8
Pendulum	100	**54.3** ± 1.5	53.8 ± 0.5
Cartpole	100	**670.4** ± 110.1	354.3 ± 65.2
Overall	100	**1418.1**	1013.8

Table 4: Ablation study on confidence aware UCB/LCB vs MEAN policy selection

We conducted an ablation study that compares RPI-LCB/UCB, which takes uncertainty into account as follows:

$$K = \arg\max\ (\overline{\hat{V}^1(s)}, \overline{\hat{V}^2(s)}, \ldots, \underline{\hat{V}^{K+1}(s)})$$

against RPI-MEAN, which does not consider uncertainty as:

$$K = \arg\max\ (\hat{V}^1(s), \hat{V}^2(s), \ldots, \hat{V}^{K+1}(s)).$$

The experimental results presented in Fig. 9 and Table 4 demonstrate that the RPI-LCB/UCB strategy outperforms RPI-MEAN across all benchmarks by an *overall* margin of 40%. This highlights the significance of incorporating uncertainty in the policy selection strategy.