Absolute Policy Optimization: Enhancing Lower Probability Bound of Performance with High Confidence

Weiye Zhao [*1] **Feihan Li** [*1] **Yifan Sun** [1] **Rui Chen** [1] **Tianhao Wei** [1] **Changliu Liu** [1]

Abstract

In recent years, trust region on-policy reinforcement learning has achieved impressive results in addressing complex control tasks and gaming scenarios. However, contemporary state-of-the-art algorithms within this category primarily emphasize improvement in expected performance, lacking the ability to control over the worst-case performance outcomes. To address this limitation, we introduce a novel objective function, optimizing which leads to guaranteed monotonic improvement in the lower probability bound of performance with high confidence. Building upon this groundbreaking theoretical advancement, we further introduce a practical solution called Absolute Policy Optimization (APO). Our experiments demonstrate the effectiveness of our approach across challenging continuous control benchmark tasks and extend its applicability to mastering Atari games. Our findings reveal that APO as well as its efficient variation Proximal Absolute Policy Optimization (PAPO) significantly outperforms state-of-the-art policy gradient algorithms, resulting in substantial improvements in worst-case performance, as well as expected performance.

1. Introduction

Existing reinforcement learning algorithms focus on improving expected cumulative rewards (referred to as **performance**). Within this framework, trust region-based on-policy reinforcement learning algorithms have achieved the most promising results. However, the representative trust-region policy optimization (TRPO) (Schulman et al., 2015) only ensures the monotonic improvement of the expected performance, it fails to exert control over the worst-case performance. Simultaneously, the deployment of reinforcement learning policies in real-world scenarios demands a high level of consistency, where the performance distribution must be carefully controlled to ensure that even under worst-case conditions, undesirable outcomes are avoided. This is particularly critical in applications such as autonomous driving and intelligent robot manipulation, where robust performance is essential to guarantee safety, reliability, and adherence to desired behaviors.

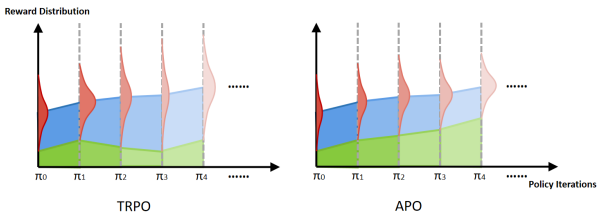


Figure 1: Explanation of APO principles. Here we simplify the distribution of rewards to a Gaussian distribution for displaying. Green and blue represent the worst case and expectation respectively. Y-axis represents the distribution of performance samples introduced in Section 2.2. In comparison to TRPO, APO is designed to consistently elevate the lower bound of overall performance.

Distribution control represents a prominent research area across various domains. Existing works in this field can be categorized into three main groups: (i) modeling the entire performance distribution, (ii) controlling worst-case costs, and (iii) mitigating model discrepancies during generalization. Many **distributional RL** methods (Bellemare et al., 2017; Dabney et al., 2017) fall into the first category, aiming to model the entire distribution of returns (Dabney et al., 2018). By incorporating more information about the distribution of rewards into policy gradients, these methods contribute to more stable and efficient learning (Yang et al., 2019; Zhou et al., 2020; Kuang et al., 2023; Fujimoto et al., 2018). While distributional RL is commonly applied in the context of off-policy methods, the convergence characteristics of these methods are not fully understood, typically explored under strict assumptions such as infinite sampling and Q-updates (Fujimoto et al., 2018). It's important to

[*]Equal contribution [1]Robotics Institute, Carnegie Mellon University, USA. Correspondence to: Weiye Zhao <weiyezha@andrew.cmu.edu>, Changliu Liu <cliu6@andrew.cmu.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

note that distributional RL does not inherently provide direct guarantees for worst-case performance. **Risk-sensitive RL** (Alexander & Baptista, 2004), which falls into the second category, stands as a potent class of safe RL methods designed to ensure high-confidence satisfaction of worst-case costs (Chow et al., 2015; Berkenkamp et al., 2017; Chow et al., 2018; Tang et al., 2019; Jain et al., 2021; Chen et al., 2023; Yu & Ying, 2023). However, these methods rely on a Lagrangian approach and necessitate intricate assumptions for convergence (Chow et al., 2018). In the third category, **robust RL** (Peng et al., 2017) addresses the challenge of minimizing performance fluctuations caused by applying a policy to different environments with dynamic discrepancies (Rajeswaran et al., 2016; Jiang et al., 2021; You et al., 2022). Typically, these methods involve training policies on a diverse set of environments to enhance performance robustness (Peng et al., 2017; Panaganti et al., 2022). Nevertheless, Robust RL does not address the enhancement of performance robustness within a single environment.

Distinct from all the aforementioned categories, **ensuring the improvement of worst-case policy performance for a given environment** remains an uncharted area. In this paper, we address this challenge by introducing a novel trust-region policy optimization method. Unlike distributional RL and risk-sensitive RL, our approach is an on-policy trust-region method that explicitly ensures the monotonic improvement of the worst-case performance without relying on complex assumptions. In contrast to robust RL, our method directly enhances the worst-case performance for a given environment. We delve into existing works and establish connections to our approach in detail in Appendix A.

Specifically, we introduce a novel theoretical breakthrough that ensures the monotonic improvement of the lower probability bound of performance with high confidence. Subsequently, we implement a series of approximations to transform this theoretically-grounded algorithm into a practical solution, which we refer to as **A**bsolute **P**olicy **O**ptimization (APO). The main idea of APO is illustrated in Figure 1. Remarkably, APO exhibits scalability and can efficiently optimize nonlinear policies characterized by tens of thousands of parameters. Our experimental results underscore the effectiveness of APO and its efficient variation PAPO, demonstrating substantial performance improvements in terms of both worst-case performance and expected performance compared to state-of-the-art policy gradient algorithms. These improvements are evident across challenging continuous control benchmark tasks and extend to the realm of playing Atari games. Our code is available on Github.[1]

Our contribution is summarized below:

- To the best of the authors' knowledge, the proposed approach is the first to guarantee the monotonic improvement of lower probability bound of performance with high confidence.

2. Problem Formulation

2.1. Notations

Consider an infinite-horizon discounted Markov decision process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, \gamma, \mathcal{R}, P, \mu)$, where $\mathcal{S}$ is the state space, and $\mathcal{A}$ is the control space, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is a bounded reward function, $0 \leq \gamma < 1$ is the discount factor, $\mu : \mathcal{S} \mapsto \mathbb{R}$ is the bounded initial state distribution, and $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ is the transition probability. $P(s'|s, a)$ is the probability of transitioning to state s' when the agent takes action a at state s. A stationary policy $\pi : \mathcal{S} \mapsto \mathcal{P}(\mathcal{A})$ is a mapping from states to a probability distribution over actions, with $\pi(a|s)$ denoting the probability of selecting action a in state s. We denote the set of all stationary policies by Π. Subsequently, we denote π_θ as the policy that is parameterized by the parameter θ.

The standard goal for MDP is to learn a policy π that maximizes a performance measure $\mathcal{J}(\pi)$ which is computed via the discounted sum of reward $\mathcal{J}(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})\right]$, where $\tau = [s_0, a_0, s_1, \cdots]$, and $\tau \sim \pi$ is shorthand for that the distribution over trajectories depends on $\pi : s_0 \sim \mu, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)$. Mathematically,

$$\max_{\pi\in\Pi} \mathcal{J}(\pi) \,. \tag{1}$$

Additionally, a performance sample is defined here as $R_\pi(s_0) \doteq \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})$, where the state action sequence $\hat{\tau} = [a_0, s_1, \ldots] \sim \pi$ starts with an initial state s_0, which follows initial state distribution μ.

The value function is denoted as $V_\pi(s) \doteq \mathbb{E}_{\tau\sim\pi}[R_\pi(s)|s_0 = s]$, the action-value function as $Q_\pi(s, a) = \mathbb{E}_{s'\sim P}[Q_\pi(s, a, s')] \doteq \mathbb{E}_{\tau\sim\pi}[R_\pi(s)|s_0 = s, a_0 = a]$, and the advantage function as $A_\pi(s, a) = \mathbb{E}_{s'\sim P}[A_\pi(s, a, s')] \doteq Q_\pi(s, a) - V_\pi(s)$. We also define $\bar{A}_{\pi',\pi}(s)$ as the expected advantage of π' over π at state s: $\bar{A}_{\pi',\pi}(s) \doteq \mathbb{E}_{a\sim\pi'}[A_\pi(s, a)]$.

2.2. Lower Probability Bound of Performance

Notice that maximizing the expected reward performance ($\mathcal{J}$), unfortunately, does not provide control over each individual performance sample ($R_\pi(s_0)$) derived from the policy π. In a practical reinforcement learning setting, unexpected poor performance samples can lead to training instability, compromising the reliability of solutions in real-world applications. To tackle this issue, our insight is that policy optimization should not be solely fixated on enhancing ex-

[1] https://github.com/intelligent-control-lab/absolute-policy-optimization

pected performance, but also on improving the lower bound originating from the distribution of the variable $R_\pi(s_0)$. For the continuous random variable $R_\pi(s_0)$, the best we can do is to improve the lower probability bound (Pishro-Nik, 2014) of performance with high confidence, defined as:

Definition 1 (Lower Probability Bound of Performance)**.** Given a tuple $(\mathcal{B} \in \mathbb{R}, p \in \mathbb{R}^+)$, $\mathcal{B}$ is defined as the lower probability bound of performance with confidence p. Mathematically:

$$Pr\big(R_\pi(s_0) \geq \mathcal{B}\big) \geq p, \tag{2}$$

For an unknown performance distribution of policy π, we first define $\mathcal{V}(\pi)$ as the variance of the performance distribution. Then, we can leverage the Selberg's inequality theory (Saw et al., 1984) to obtain an lower probability bound of performance as $\mathcal{B}_k(\pi) \doteq \mathcal{J}(\pi) - k\mathcal{V}(\pi)$, which is guaranteed to satisfy Definition 1 (proved in Proposition 1) with confidence $p_k^\psi \doteq 1 - \frac{1}{k^2\psi+1} \in (0,1)$. Here k is the probability factor ($k \geq 0$, $k \in \mathbb{R}$) and $\psi = \mathcal{V}_{min} \in \mathbb{R}^+$, where $\mathcal{V}_{min}$ is minima of $\mathcal{V}(\pi)$.

Remark 1. Definition 1 shows that more than p_k^ψ of the samples from the distribution of $R_\pi(s_0)$ will be larger than the bound $\mathcal{B}_k(\pi)$. Given a positive constant ψ, we can make $p_k^\psi \to 1$ by setting a large enough k, so that $\mathcal{B}_k(\pi)$ represents the lower probability bound of performance with with a confidence level close to 1.

2.3. Problem

In this paper, we focus on improving the lower probability bound of performance in Markov Decision Processes (MDP). In accordance with Proposition 1, the overarching objective is to identify a policy π that effectively improves $\mathcal{B}_k(\pi)$. Mathematically,

$$\max_{\pi \in \Pi} \ \mathcal{J}(\pi) - k\mathcal{V}(\pi). \tag{3}$$

3. Absolute Policy Optimization

To optimize equation 3, we need to evaluate the objective with respect to an unknown π. Our main intuition is to find a surrogate function for the objective, such that (i) it represents a tight lower bound of the objective; and (ii) it can be easily estimated from the samples on the most recent policy. To solve large and continuous MDPs, policy search algorithms look for the optimal policy within a set $\Pi_\theta \subset \Pi$ of parametrized policies. Mathematically, APO updates solve the following optimization:

$$\pi_{j+1} = \mathop{\mathbf{argmax}}_{\pi \in \Pi_\theta} \mathcal{J}^l_{\pi,\pi_j} - k\left(MV_{\pi,\pi_j} + VM_{\pi,\pi_j}\right) \tag{4}$$

where $\mathcal{J}^l_{\pi,\pi_j}$ represents the lower bound surrogate function for $\mathcal{J}(\pi)$ and $\big(MV_{\pi,\pi_j} + VM_{\pi,\pi_j}\big)$ represents the upper bound surrogate function for $\mathcal{V}(\pi)$ in the (j+1)-th iteration.

Remark 2. MV_{π,π_j} reflects the upper bound of expected variance of the return over different start states. VM_{π,π_j} reflects the upper bound of variance of the expected return of different start states. The detailed interpretations are shown in Figure 2 and discussed in Equation (14), Proposition 2, and Proposition 4.

Remark 3 (**Balance of Exploration and Exploitation**). Intuitively, equation 4 improves performance expectation and minimizes performance variance, where k controls the importance of the two folder objectives. Higher performance variance (decreasing k) allows for more exploration, introducing the possibility of bad outcomes, thereby hindering exploitation. Conversely, reducing performance variance (increase k) limits exploration, impeding exploration and resulting in convergence to local optima. Hence, a moderate k is desirable, which is further discussed in Section 5.6. **The objective of TRPO can be viewed as a special case of equation 4, where $k = 0$.**

Here $\mathcal{J}^l_{\pi,\pi_j}, MV_{\pi,\pi_j}, VM_{\pi,\pi_j}$ are defined as:

$$\begin{aligned}
\mathcal{J}^l_{\pi,\pi_j} &\doteq \mathcal{J}(\pi_j) \qquad (5)\\
&+ \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_j}\\ a\sim\pi}} \left[A_{\pi_j}(s,a) - \frac{2\gamma\epsilon^\pi}{1-\gamma}\sqrt{\frac{1}{2}\mathcal{D}_{KL}(\pi\|\pi_j)[s]}\right]\\
MV_{\pi,\pi_j} &\doteq \frac{\|\mu^\top\|_\infty}{1-\gamma^2} \mathbf{max}_s \Bigg| \mathop{\mathbb{E}}_{\substack{a\sim\pi\\ s'\sim P}} \left[A_{\pi_j}(s,a,s')^2\right] \qquad (6)\\
&- \mathop{\mathbb{E}}_{\substack{a\sim\pi_j\\ s'\sim P}} \left[A_{\pi_j}(s,a,s')^2\right] + |H(s,a,s')|^2_{max}\\
&+ 2 \mathop{\mathbb{E}}_{\substack{a\sim\pi\\ s'\sim P}} \left[A_{\pi_j}(s,a,s')\right] \cdot |H(s,a,s')|_{max} \Bigg|\\
&+ MV_{\pi_j} + \frac{2\gamma^2\|\mu^\top\|_\infty}{(1-\gamma^2)^2}\sqrt{\frac{1}{2}\mathcal{D}^{max}_{KL}(\pi\|\pi_j)} \cdot \|\Omega_{\pi_j}\|_\infty\\
VM_{\pi,\pi_j} &\doteq \|\mu^\top\|_\infty \mathbf{max}_s \Big| |\eta(s)|^2_{max} + 2|V_{\pi_j}(s)| \cdot |\eta(s)|_{max} \Big| \qquad (7)\\
&- \mathbf{min}\ \left(\mathcal{J}(\pi)\right)^2 + \mathop{\mathbb{E}}_{s_0\sim\mu}[V^2_{\pi_j}(s_0)]
\end{aligned}$$

where $\mathcal{D}_{KL}(\pi\|\pi_j)[s]$ is the KL divergence between (π,π_j) at state s and $\mathcal{D}^{max}_{KL}(\pi\|\pi_j) = \max_s \mathcal{D}_{KL}(\pi\|\pi_j)[s]$; $\epsilon^\pi \doteq \mathbf{max}_s |\mathbb{E}_{a\sim\pi}[A_{\pi_j}(s,a)]|$ is the maximum expected advantage; $d^{\pi_j} \doteq (1-\gamma)\sum_{t=0}^\infty \gamma^t P(s_t = s|\pi_j)$ is the discounted future state distribution; $\omega_{\pi_j}(s) \doteq \mathop{\mathbb{E}}_{\substack{a\sim\pi_j\\ s'\sim P}}\left[Q_{\pi_j}(s,a,s')^2\right] - V_{\pi_j}(s)^2$ is the variance of action value; $\Omega_{\pi_j} \doteq \begin{bmatrix}\omega_{\pi_j}(s^1) & \omega_{\pi_j}(s^2) & \dots\end{bmatrix}^\top$ is the vector of variance of action value; $MV_{\pi_j} \doteq \mathop{\mathbb{E}}_{s_0\sim\mu}[\mathop{\mathbb{V}ar}_{\tau\sim\pi_j}[R_{\pi_j}(s_0)]$ is the expectation of performance variance over initial state

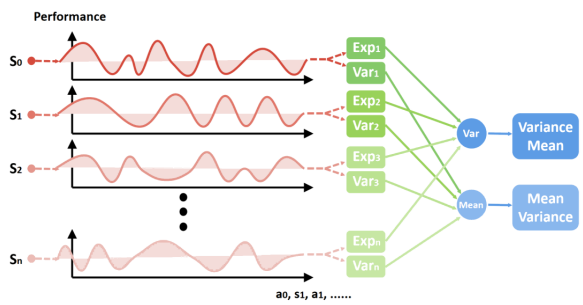


Figure 2: Explanation of MV and VM. Since performance from different start states belong to a mixture of one-dimensional distributions, the variance of performance can be deconstructed into two components: MeanVariance and VarianceMean.

distribution; **min** $(\mathcal{J}(\pi))^2 \doteq \min_{\mathcal{J}(\pi)\in[\mathcal{J}^l_{\pi,\pi_j},\mathcal{J}^u_{\pi,\pi_j}]} (\mathcal{J}(\pi))^2$ is the minimal squared performance, where the upper bound of $\mathcal{J}(\pi)$ is defined as $\mathcal{J}^u_{\pi,\pi_j} \doteq \mathcal{J}(\pi_j) + \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_j}\\ a\sim\pi}}\Big[A_{\pi_j}(s,a) + \frac{2\gamma\epsilon^\pi}{1-\gamma}\sqrt{\frac{1}{2}\mathcal{D}_{KL}(\pi\|\pi_j)[s]}\Big]$. Additionally,

$$
\begin{aligned}
|H(s,a,s')|_{max} &\doteq \frac{2\gamma(1+\gamma)\epsilon}{(1-\gamma)^2}\mathcal{D}^{max}_{KL}(\pi||\pi_j) \qquad (8)\\
&+ \left|\gamma \mathop{\mathbb{E}}_{\substack{s_0=s'\\ \hat{\tau}\sim\pi_j}}\left[\sum_{t=0}^{\infty}\gamma^t \bar{A}_{\pi,\pi_j}(s_t)\right] - \mathop{\mathbb{E}}_{\substack{s_0=s\\ \hat{\tau}\sim\pi_j}}\left[\sum_{t=0}^{\infty}\gamma^t \bar{A}_{\pi,\pi_j}(s_t)\right]\right|\\
|\eta(s)|_{max} &\doteq \left|\mathop{\mathbb{E}}_{\substack{s_0=s\\ \hat{\tau}\sim\pi_j}}\left[\sum_{t=0}^{\infty}\gamma^t \bar{A}_{\pi,\pi_j}(s_t)\right]\right|\\
&+ \frac{2\gamma\epsilon}{(1-\gamma)^2}\mathcal{D}^{max}_{KL}(\pi\|\pi_j),
\end{aligned}
$$

where $\epsilon \doteq \max_{s,a}|A_{\pi_j}(s,a)|$.

3.1. Theoretical Guarantees for APO

Theorem 1 (Monotonic Improvement of Absolute Performance)**.** *Suppose π, π' are related by equation 4, then absolute performance bound $\mathcal{B}_k(\pi) = \mathcal{J}(\pi) - k\mathcal{V}(\pi)$ satisfies $\mathcal{B}_k(\pi') \geq \mathcal{B}_k(\pi)$.*

The proof for Theorem 1 is summarized in Appendix C. In essence, we introduce the function $\mathcal{M}^j_k(\pi) = \mathcal{J}^l_{\pi,\pi_j} - k\left(MV_{\pi,\pi_j} + VM_{\pi,\pi_j}\right)$ (the right-hand side of equation 4). Our demonstration establishes that $\mathcal{M}^j_k(\pi)$ serves as the lower bound for the absolute bound $\mathcal{B}_k(\pi)$ through the application of Proposition 2, Proposition 4, and Proposition 5.

Subsequently, leveraging the facts that $\mathcal{B}_k(\pi_j) = \mathcal{M}^j_k(\pi_j)$ (proved in [Lemma 5, Appendix C.6]) and $\mathcal{B}_k(\pi_{j+1}) \geq \mathcal{M}^j_k(\pi_{j+1})$, the following inequality holds:

$$
\mathcal{B}_k(\pi_{j+1}) - \mathcal{B}_k(\pi_j) \geq \mathcal{M}^j_k(\pi_{j+1}) - \mathcal{M}^j_k(\pi_j) \qquad (9)
$$

Therefore, through the maximization of $\mathcal{M}^j_k$ at each iteration (equation 4), we ensure the true absolute performance $\mathcal{B}_k$ is non-decreasing. Furthermore, it is noteworthy that π_j is always a feasible solution for equation 4.

4. Practical Implementation

In this section, we show how to (i) encourage larger update steps with trust region constraint, (ii) simplify complex computations. The full APO pseudocode is provided as Algorithm 1 in Appendix D.

Trust Region Constraint In practice, strictly following the theoretical recommendations for the coefficients of KL divergence terms in equation 4 often leads to very small step sizes. Instead, a practical approach is to enforce a constraint on the KL divergence between the new and old policies (Schulman et al., 2015), commonly known as a trust region constraint. This strategy allows for taking larger steps in a robust way:

$$
\begin{aligned}
\pi_{j+1} = \mathop{\mathbf{argmax}}_{\pi\in\Pi_\theta} &\frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_j}\\ a\sim\pi}}\left[A_{\pi_j}(s,a)\right] \qquad (10)\\
&- k\left(\overline{MV}_{\pi,\pi_j} + \overline{VM}_{\pi,\pi_j}\right)\\
\textbf{s.t. } &\bar{\mathcal{D}}_{KL}(\pi||\pi_j) \leq \delta
\end{aligned}
$$

where δ is the step size, $\overline{MV}_{\pi,\pi_j} \doteq MV_{\pi,\pi_j} - \frac{2\gamma^2\|\mu^\top\|_\infty}{(1-\gamma^2)^2}\sqrt{\frac{1}{2}\mathcal{D}^{max}_{KL}(\pi\|\pi_j)} \cdot \|\Omega_{\pi_j}\|_\infty$ and $\overline{VM}_{\pi.\pi_j} = VM_{\pi,\pi_j} - \mathop{\mathbb{E}}_{s_0\sim\mu}[V^2_{\pi_j}(s_0)]$. The set $\{\pi \in \Pi_\theta : \bar{\mathcal{D}}_{KL}(\pi||\pi_j) = \mathop{\mathbb{E}}_{s\sim\pi_j}[\mathcal{D}_{KL}(\pi\|\pi_j)[s]] \leq \delta\}$ is called *trust region*. Notice that MV_{π_j} and $\mathop{\mathbb{E}}_{s_0\sim\mu}[V^2_{\pi_j}(s_0)]$ are computable constant.

Special Parameters Handling When implementing Equation (10), we first treat two items as hyperparameters. (i) $\|\boldsymbol{\mu}^\top\|_\infty$: Although the infinity norm of $\mu^\top$ is theoretically equal to 1, we found that treating it as a hyperparameter in $\mathbb{R}^+$ enhances performance in practical implementation. (ii) $|\boldsymbol{H(s,a,s')}|_{\boldsymbol{max}}$: We can either compute $|H(s,a,s')|_{max}$ from the most recent policy with equation 8 or treat it as a hyperparameter since $|H(s,a,s')|_{max}$ is bounded for any system with a bounded reward function. In practice, we found that the hyperparameter option helps increase the performance. This setting will be discussed more detailedly in Section 5.5. (iii) $|\boldsymbol{\eta(s)}|_{\boldsymbol{max}}$ and $\max_{\boldsymbol{s}}|\cdots|$: Furthermore, we find that taking the average of the state s instead of the maximum will be more superior

and stable in terms of convergence results. Note that a similar trick is also applied in (Schulman et al., 2015) to handle maximum KL divergence.

5. Experiment

In our experiments, we want to answer the following questions:
Q1: How does APO compare with state-of-the-art on-policy RL algorithms?
Q2: What benefits are demonstrated by directly optimizing the absolute performance?
Q3: Is treating H_{max} as a hyperparameter necessary?
Q4: What are the impacts of different probability factor k choices?
Q5: What potential does APO hold?
Q6: How does APO compare with state-of-the-art algorithms in terms of computational cost?
Q7: What are the scenarios where APO is most beneficial or less effective?

5.1. Experiment Setup

To answer the above, we run experiments on both continuous domain and the discrete domain.

Continuous Tasks Our continuous experiments are conducted on GUARD (Zhao et al., 2023), a challenging robot locomotion benchmark build upon Mujoco (Todorov et al., 2012) and Gym. Seven different robots are included: (i) **Point:** (Figure 15a) A point-mass robot ($\mathcal{A} \subseteq \mathbb{R}^2$) that can move on the ground. (ii) **Swimmer:** (Figure 15b) A three-link robot ($\mathcal{A} \subseteq \mathbb{R}^2$) that can move on the ground. (iii) **Arm3:** (Figure 15c) A fixed three-joint robot arm($\mathcal{A} \subseteq \mathbb{R}^3$) that can move its end effector around with high flexibility. (iv) **Drone:** (Figure 15d) A quadrotor robot ($\mathcal{A} \subseteq \mathbb{R}^4$) that can move in the air. (v) **Hopper:** (Figure 15e) A one-legged robot ($\mathcal{A} \subseteq \mathbb{R}^5$) that can move on the ground. (vi) **Ant:** (Figure 15f) A quadrupedal robot ($\mathcal{A} \subseteq \mathbb{R}^8$) that can move on the ground. (vii) **Walker:** (Figure 15g) A bipedal robot ($\mathcal{A} \subseteq \mathbb{R}^{10}$) that can move on the ground. Furthermore, three different types of tasks are considered, including (i) **Goal:** (Figure 14a) robot navigates towards a series of 2D or 3D goal positions. (ii) **Push:** (Figure 14b) robot pushes a ball toward different goal positions. (iii) **Chase:** (Figure 14c) robot tracks multiple dynamic targets. Considering these different robots and tasks, we design 8 low-dim test suites and 4 high-dim test suits with 7 types of robots and 3 types of tasks, which are summarized in Table 2 in Appendix. We name these test suites as `{Task Type}_{Robot}`. Further details are listed in Appendix E.1.

Additionally, we conduct continuous control experiments on Mujoco Openai Gym (Brockman et al., 2016b) and Gymnasium Robotics (Plappert et al., 2018). Five high dimensional tasks are considered: (i) **Humanoid:** (Figure 14d) The 3D bipedal robot ($\mathcal{A} \subseteq \mathbb{R}^{17}$) is designed to simulate a human. And the goal of the environment is to walk forward as fast as possible without falling over. (ii) **Humanoid Standup:** (Figure 14e) The robot ($\mathcal{A} \subseteq \mathbb{R}^{17}$) is same with task **Humanoid**, but the goal is to make the humanoid standup and then keep it standing. These two tasks are also summarized in Table 2. (iii) **HandReach:** (Figure 14f) The goal of the task is for the fingertips of the hand ($\mathcal{A} \subseteq \mathbb{R}^{24}$) to reach a predefined target Cartesian position. (iv) **HandManipulateEgg:** (Figure 14g) The task is to manipulate the egg such that a target pose is achieved (v) **HandManipulateBlock:** (Figure 14h) The task is to manipulate the block such that a target pose is achieved. Notice that tasks (iii)-(v) employ dense rather than discrete reward setup for learning, see (Plappert et al., 2018) for details.

Discrete Tasks We also test APO in all 62 Atari environments of (Brockman et al., 2016b) which are simulated on the Arcade Learning Environment benchmark (Bellemare et al., 2018). All experiments are based on 'v5' environments and 'ram' observation space.

Comparison Group We compare APO to the state-of-the-art on-policy RL algorithms: (i) TRPO (Schulman et al., 2015) (ii) Advantage Actor Critic (A2C) (Mnih et al., 2016) (iii) PPO by clipping (Schulman et al., 2017) in both continuous and discrete tasks and additionally compare (iv) AlphaPPO (Xu et al., 2023) (v) ESPO (Sun et al., 2022) (vi) V-MPO (Song et al., 2020) on continuous tasks.For all experiments, we take the best specific parameters mentioned in the original (Xu et al., 2023) paper and keep the common parameters as the same. In particular, we selected the optimal parameters based on the tuning method of the paper for each task individually. The policy π, the value V^π are all encoded in feedforward neural networks using two hidden layers of size (64,64) with tanh activations. The full list of parameters of all methods and tasks compared can be found in Appendix E.2.

5.2. Comparison to Other Algorithms on the Atari Domain

Figure 11 shows 12 representative test suites in Atari Domain. All 62 learning curves can be found in Appendix F. The hyperparameters for Atari Domain are also provided in Appendix E.2. For the other three algorithms, we used hyperparameters that were tuned to maximize performance on this benchmark. Then we follow the metrics of (Schulman et al., 2017) to quantitatively evaluate the strengths of APO: (i) average expected reward per episode over **all epochs of training** (which favors fast learning), and (ii) average expected reward per episode over **last 100 (20 of GUARD) epochs of training** (which favors final perfor-

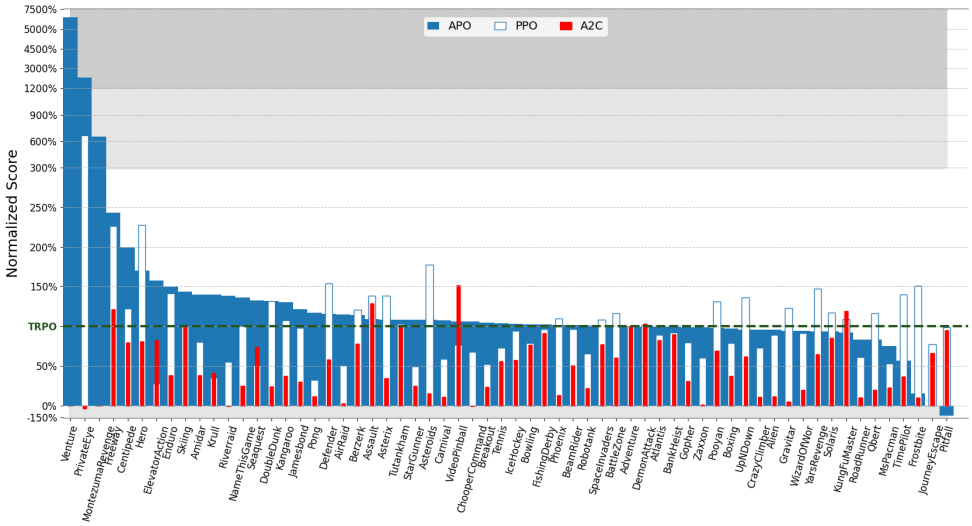


Figure 3: Stacked bar chart for all 62 atari games

Performance	Metrics	APO	PPO	TRPO	A2C	Tie
Expected Performance	(1) average expected reward over all epochs(Atari)	**26**	22	10	3	1
	(2) average expected reward over last 100 epochs(Atari)	**29**	17	12	3	1
	(1) average expected reward over all epochs(GUARD)	**9**	3	0	0	0
	(2) average expected reward over last 20 epochs(GUARD)	**9**	2	1	0	0
Worst Performance	(1) average worst reward over all epochs(Atari)	**26**	24	7	3	2
	(2) average worst reward over last 100 epochs(Atari)	**27**	20	10	3	2
	(3) average worst reward over all epochs(GUARD)	**8**	2	1	1	0
	(4) average worst reward over last 20 epochs(GUARD)	**9**	1	2	0	0

Table 1: The number of highest evaluation scores obtained by each algorithm across all test suites

mance). Table 1 records the number of highest evaluation scores obtained by each algorithm across all games. To compare the performance of all testing algorithm to the TRPO baseline across games, we slightly change the normalization algorithm proposed by (van Hasselt et al., 2015) to obtain more reasonable score (See Appendix E.3 for further explanation) in percent. The score we used is average reward per episode over last 100 (20 of GUARD) epochs of training:

$$\Delta_1 \doteq score_{agent} - score_{random} \tag{11}$$
$$\Delta_2 \doteq score_{TRPO} - score_{random}$$
$$score_{normalized} = \frac{\Delta_2}{\Delta_1} \; if \; \Delta_1 < 0 \; and \; \Delta_2 < 0 \; else \; \frac{\Delta_1}{\Delta_2}$$

Then we use a stacked bar chart in Figure 3 to visualize APO's capabilities. Figure 3 show that APO has a superior combination of capabilities compared to other algorithms. The statistics of the algorithm in terms of expected performance improvement are presented in Table 1. So far the above experimental comparison answers **Q1**.

5.3. Comparison to Other Algorithms in GUARD Continuous Domain

Low dimension Figure 4 shows representative comparison results on a low dimensional system (See Appendix F for all results). APO is successful at getting a more steady and higher final reward. We notice that PPO only gains faster convergence in part of the simplest task owing to its exploration abilities, the advantage decreases rapidly with more complex tasks such as PUSH. In difficult tasks, APO can perform best at the combined level of convergence speed and final performance.

High dimension Figure 5 reports the comparison results on challenging high-dimensional {PUSH, CHASE}_Ant and {PUSH, CHASE}_Walker tasks, where APO outperforms other baselines in getting higher reward and convergence speed. It is worth noting that we can clearly observe that PPO-related methods perform more erratically in complex tasks (PUSH and CHASE) and are not as good as TRPO in benchmark performance.

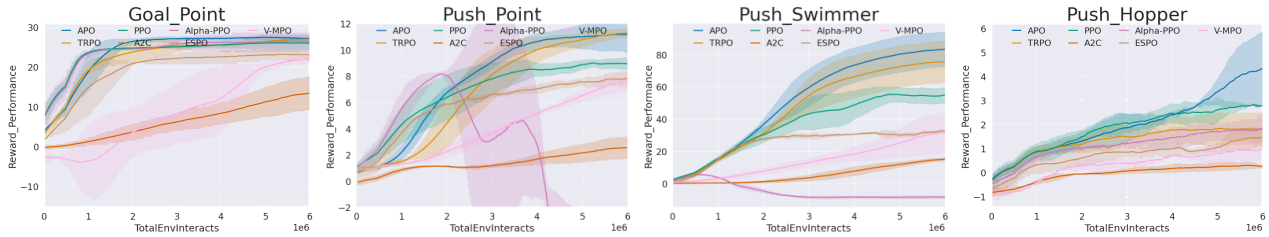


Figure 4: Comparison of results from four representative test suites in low dimensional continuous systems

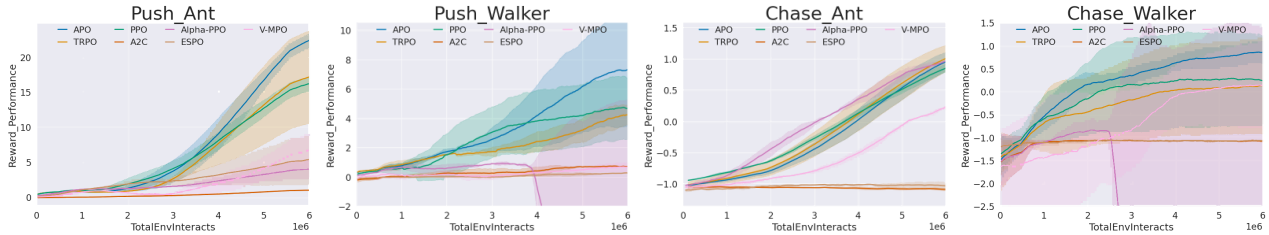


Figure 5: Comparison of results from four representative test suites in high dimensional continuous systems

We also performed quantitative statistics on the 12 sets of continuous tasks tested. The results are presented in Table 1 and metrics we used are presented in the next section.

5.4. Worst-case Performance Comparison

We use a large probability factor k in practical implementation, which means we are close to optimizing the lower bound for all samples. Thus we use another two similar metrics to evaluate the effectiveness of algorithms for lower bound lifting: (iii) average worst reward per episode over **all epochs of training**, and (iv) average worst reward per episode over **last 100 or 20 epochs of training**. We summarize the worst-case performance of APO in Atari games and GUARD in Table 1, which answers **Q2**.

5.5. Ablation on H_{max} Hyperparameter Trick

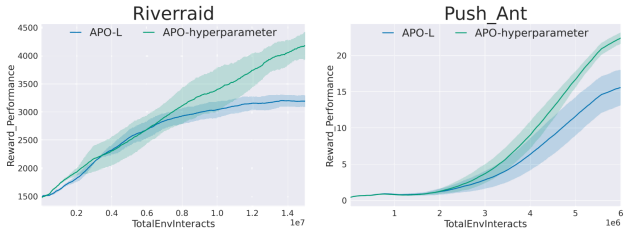


Figure 6: Ablation on H_{max} hyperparameter trick

We chose Riverraid of discrete tasks and PUSH_Ant of continuous tasks to perform ablation experiments against the $|H(s,a,s')|_{max}$ implementation. Figure 6 shows that although both boosts are similar in the early stages of tasks, hyperparameter methods can more consistently converge to a higher reward value. Thus, the figures and description answer **Q3**.

5.6. Ablation on Probability Factor k

For ablation, we selected Riverraid to investigate the impact of different choices for the probability factor k. As illustrated in Figure 9, when k takes on a very small value, indicating optimization of only a limited portion of performance samples, the effectiveness diminishes. This is attributed to the loss of control over the lower probability bound of performance because of the small confidence level. Conversely, when k becomes excessively large, the optimization shifts its focus towards the most extreme worst-case performance scenarios. This ultra-conservative approach tends to render the overall optimization less effective. Therefore, a moderate choice of k will be favorable to the overall improvement of the effect, which answers **Q4**.

In the following subsection, we will show a simple strategy can greatly improve the APO computation efficiency and performance.

5.7. Proximal Absolute Policy Optimization (PAPO)

With APO's proven success in addressing demanding tasks in both continuous control and Atari game playing, a natural question arises: What potential does APO hold? To shed light on this inquiry, we undertake a natural extension of APO by incorporating a successful variation from TRPO to PPO, i.e. introduction of a **Clipped Surrogate Objective** (Schulman et al., 2017).

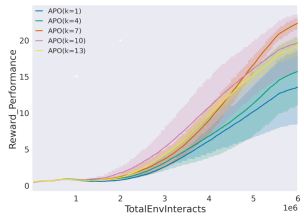


Figure 9: Ablation on probability factor k

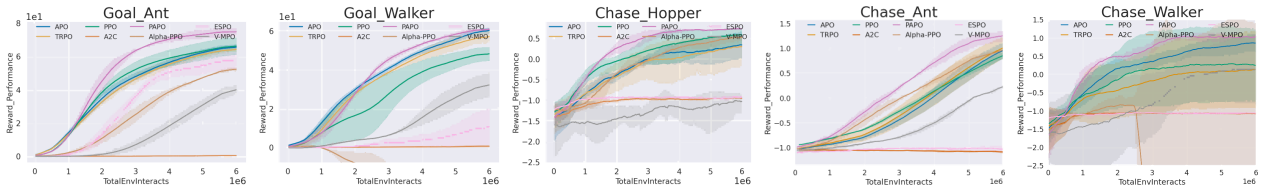


Figure 7: Comparison results of PAPO from five representative test suites in GUARD

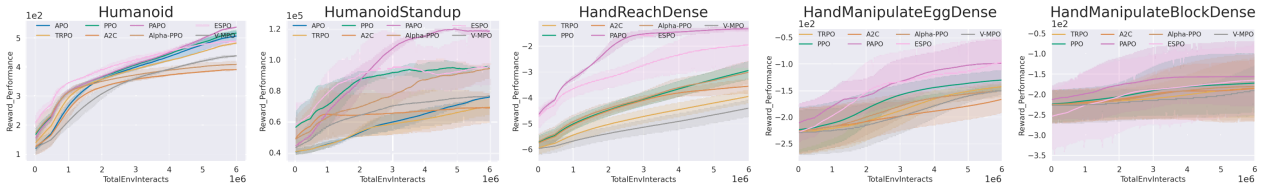


Figure 8: Comparison results of PAPO from five representative test suites in Mujoco and Gymnasium benchmark

In our experiment, this clipping is implemented with a simple method: **early stopping**. Specifically, multiple steps of stochastic gradient descent are taken to maximize the objective of equation 10. If the mean KL-divergence of the new policy from the old grows beyond a threshold (δ), we stop taking gradient steps. We call this enhanced version of APO as *Proximal Absolute Policy Optimization (PAPO)*.

5.7.1. Compare PAPO with Baseline Algorithms

Next we will focus on challenging and high-dimensional tasks in GUARD where trust-region based approaches usually struggle to get good results. The detailed test suites can be found in Table 3. The summarized comparison results are presented in Figure 7, highlighting PAPO's superior performance over previous baseline methods, including APO, across all these GUARD challenging control environments. Notably, PAPO significantly improves APO's performance in challenging continuous tasks and outperforms PPO by a learning speed and converged reward, as evidenced by the learning curves.

5.7.2. Showcase in the Continuous Domain: Mujoco and Gymnasium Benchmark

To demonstrate the efficacy of PAPO in **super-high dimensional** continuous benchmarks, we conducted additional experiments in scenarios that were highly integrated with the reality frontier such as dexterous hand manipulation and humanoid robot movement. The results, compared with all baseline methods, including APO, are presented in Table 5 with detailed hyperparameters and learning curves in Figure 8. We can see that the more exploratory proximal methods can achieve better results in this type of super-high dimensional control task compared to the trust region methods. PAPO consistently outperforms all algorithms including all other proximal methods, particularly showcasing enhanced learning efficiency and converged performance in the *HumanoidStandup* task and *HandReach*. Thus the figures and description answer **Q5**.

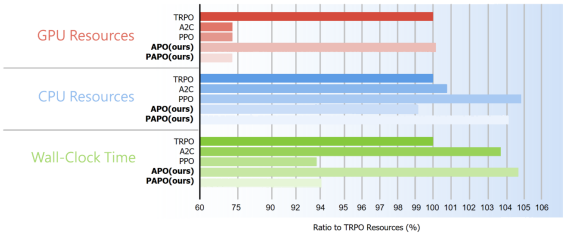


Figure 10: Computational cost(GPU occupancy, CPU occupancy and wall-clock time) comparison

5.8. APO, PAPO Computational Cost Comparison

We test the comparison of APO, PAPO, PPO, TRPO and A2C in terms of GPU and CPU resource usage, wall-clock time using TRPO as a benchmark and present in Figure 10, where the horizontal axis is the percentage value of the rest of the algorithms compared to TRPO. The experiments are built on Goal_Ant task and were run and averaged over five different seeds. We can see that APO as a trust region method takes more significant and stable results with essentially the same hardware resource footprint as TRPO and only less than 5% additional wall-clock time spent, demonstrating that our surrogate absolute bound may seem complex but does not introduce additional unacceptable computational costs. At the same time, there is basically no difference between PAPO and PPO in terms of performance of the indicators, but PAPO achieves better results on both high and low dimensional, complex and simple tasks which demonstrates the superiority of our method. Additional wall-clock time experiment results can be found in Figures 12 and 13. Thus, all of these answers **Q6**.

5.9. Effective and Less Effective Scenarios for APO and PAPO

Through extensive experimentation, we have found that APO/PAPO represents a more robust iteration of TRPO/PPO, offering significantly improved and more consistent performance while maintaining the same computational efficiency (sampling efficiency). Consequently, any application scenarios currently utilizing PPO or other on-policy RL methods can readily benefit from the application of APO. Real-world scenarios demanding high performance consistency, such as video games and robotics, are particularly advantageous domains for deploying APO.

For less effective scenarios, APO shares typical limitations inherent to on-policy algorithms. Notably, its sample efficiency may be poor due to the requirement of collecting data exclusively from the current policy's interactions with the environment, making APO impractical for environments where data acquisition is costly or time-consuming. This discussion addresses **Q7**.

6. Conclusion and Future Work

This paper proposed APO, the first general-purpose policy search algorithm that ensures monotonic improvement of the lower probability bound of performance with high confidence. We demonstrate APO's effectiveness on challenging continuous control benchmark tasks and playing Atari games, showing its significant performance improvement compared to existing methods and ability to enhance both expected performance and worst-case performance.

Furthermore, we integrate proven techniques that enhance TRPO into Proximal Absolute Policy Optimization (PAPO), resulting in substantial performance improvements on continuous tasks. This effort underscores the promising capability of APO to significantly enhance existing methodologies.

In anticipation of future endeavors, we aspire to leverage APO as the inaugural phase in our exploration of probability bound control employing the trust region method. Subsequently, our objective is to extend this investigation into a more expansive domain within scientific research.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

This work is partially supported by the National Science Foundation, Grant No. 2144489.

References

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. PMLR, 2017.

Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Alexander, G. J. and Baptista, A. M. A comparison of var and cvar constraints on portfolio selection with the mean-variance model. *Management science*, 50(9):1261–1273, 2004.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 449–458. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/bellemare17a.html.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: an evaluation platform for general agents. *Journal of Artificial Intelligence Research*, pp. 253–279, Jul 2018. doi: 10.1613/jair.3912. URL http://dx.doi.org/10.1613/jair.3912.

Berkenkamp, F., Turchetta, M., Schoellig, A. P., and Krause, A. Safe model-based reinforcement learning with stability guarantees, 2017.

Brillinger, D. R. Information and Information Stability of Random Variables and Processes. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 13(2):134–135, 12 2018. ISSN 0035-9254. doi: 10.2307/2985711. URL https://doi.org/10.2307/2985711.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016a.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv: Learning,arXiv: Learning*, Jun 2016b.

Chen, W., Subramanian, D., and Paternain, S. Probabilistic constraint for safety-critical reinforcement learning. *arXiv preprint arXiv:2306.17279*, 2023.

Chow, Y., Ghavamzadeh, M., Janson, L., and Pavone, M. Risk-constrained reinforcement learning with percentile risk criteria. *CoRR*, abs/1512.01629, 2015. URL http://arxiv.org/abs/1512.01629.

Chow, Y., Ghavamzadeh, M., Janson, L., and Pavone, M. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18 (167):1–51, 2018.

Dabney, W., Rowland, M., Bellemare, M. G., and Munos, R. Distributional reinforcement learning with quantile regression. *CoRR*, abs/1710.10044, 2017. URL http://arxiv.org/abs/1710.10044.

Dabney, W., Ostrovski, G., Silver, D., and Munos, R. Implicit quantile networks for distributional reinforcement learning. *CoRR*, abs/1806.06923, 2018. URL http://arxiv.org/abs/1806.06923.

Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, pp. 1329–1338. PMLR, 2016.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Gu, S., Lillicrap, T., Sutskever, I., and Levine, S. Continuous deep q-learning with model-based acceleration. In *International conference on machine learning*, pp. 2829–2838. PMLR, 2016.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Hausknecht, M. and Stone, P. Deep recurrent q-learning for partially observable mdps. In *2015 aaai fall symposium series*, 2015.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Jain, A., Patil, G., Jain, A., Khetarpal, K., and Precup, D. Variance penalized on-policy and off-policy actor-critic, 2021.

Jiang, Y., Li, C., Dai, W., Zou, J., and Xiong, H. Monotonic robust policy optimization with model discrepancy. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4951–4960. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/jiang21c.html.

Kaufmann, E., Bauersfeld, L., Loquercio, A., Müller, M., Koltun, V., and Scaramuzza, D. Champion-level drone racing using deep reinforcement learning. *Nature*, 620 (7976):982–987, 2023.

Kuang, Q., Zhu, Z., Zhang, L., and Zhou, F. Variance control for distributional reinforcement learning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17874–17895. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kuang23a.html.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Lin, Z., Wu, X., Sun, M., Ye, D., Fu, Q., Yang, W., and Liu, W. Sample dropout: A simple yet effective variance reduction technique in deep policy optimization. *arXiv preprint arXiv:2302.02299*, 2023.

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*, 2013a.

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*, 2013b.

Mnih, V., Badia, A., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. *arXiv: Learning,arXiv: Learning*, Feb 2016.

Pajarinen, J., Thai, H. L., Akrour, R., Peters, J., and Neumann, G. Compatible natural gradient policy search. *Machine Learning*, 108:1443–1466, 2019.

Panaganti, K., Xu, Z., Kalathil, D., and Ghavamzadeh, M. Robust reinforcement learning using offline data, 2022.

Papini, M., Binaghi, D., Canonaco, G., Pirotta, M., and Restelli, M. Stochastic variance-reduced policy gradient. In *International conference on machine learning*, pp. 4026–4035. PMLR, 2018.

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. Sim-to-real transfer of robotic control with dynamics randomization. *CoRR*, abs/1710.06537, 2017. URL http://arxiv.org/abs/1710.06537.

Pishro-Nik, H. *Introduction to probability, statistics, and random processes*. Kappa Research, LLC Blue Bell, PA, USA, 2014.

Plappert, M., Andrychowicz, M., Ray, A., McGrew, B., Baker, B., Powell, G., Schneider, J., Tobin, J., Chociej, M., Welinder, P., Kumar, V., and Zaremba, W. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *CoRR*, abs/1802.09464, 2018. URL http://arxiv.org/abs/1802.09464.

Queeney, J., Paschalidis, Y., and Cassandras, C. G. Generalized proximal policy optimization with sample reuse. *Advances in Neural Information Processing Systems*, 34: 11909–11919, 2021.

Rajeswaran, A., Ghotra, S., Levine, S., and Ravindran, B. Epopt: Learning robust neural network policies using model ensembles. *CoRR*, abs/1610.01283, 2016. URL http://arxiv.org/abs/1610.01283.

Saw, J. G., Yang, M. C., and Mo, T. C. Chebyshev inequality with estimated mean and variance. *The American Statistician*, 38(2):130–132, 1984.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

Schulman, J., Zoph, B., Kim, C., Hilton, J., Menick, J., Weng, J., Uribe, J. F. C., Fedus, L., Metz, L., Pokorny, M., et al. Chatgpt: Optimizing language models for dialogue. *OpenAI blog*, 2022.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *International conference on machine learning*, pp. 387–395. Pmlr, 2014.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Sobel, M. J. The variance of discounted markov decision processes. *Journal of Applied Probability*, 19(4):794–802, 1982.

Song, H., Abdolmaleki, A., Springenberg, J., Clark, A., Soyer, H., Rae, J., Noury, S., Ahuja, A., Liu, S., Tirumala, D., Heess, N., Belov, D., Riedmiller, M., and Botvinick, M. V-mpo: On-policy maximum a posteriori policy optimization for discrete and continuous control. *International Conference on Learning Representations,International Conference on Learning Representations*, Apr 2020.

Sun, M., Kurin, V., Liu, G., Devlin, S., Qin, T., Hofmann, K., and Whiteson, S. You may not need ratio clipping in ppo. 2022.

Sun, M., Ellis, B., Mahajan, A., Devlin, S., Hofmann, K., and Whiteson, S. Trust-region-free policy optimization for stochastic policies. *arXiv preprint arXiv:2302.07985*, 2023.

Tang, Y. C., Zhang, J., and Salakhutdinov, R. Worst cases policy gradients, 2019.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep 2012. doi: 10.1109/iros.2012.6386109. URL http://dx.doi.org/10.1109/iros.2012.6386109.

Tomczak, M. B., Kim, D., Vrancx, P., and Kim, K.-E. Policy optimization through approximate importance sampling. *arXiv preprint arXiv:1910.03857*, 2019.

van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning, 2015.

Wang, Y., He, H., and Tan, X. Truly proximal policy optimization. In *Uncertainty in Artificial Intelligence*, pp. 113–122. PMLR, 2020.

Xu, H., Yan, Z., Xuan, J., Zhang, G., and Lu, J. Improving proximal policy optimization with alpha divergence. *Neurocomputing*, 534:94–105, 2023.

Xu, T., Liu, Q., and Peng, J. Stochastic variance reduction for policy gradient estimation. *arXiv preprint arXiv:1710.06034*, 2017.

Yang, D., Zhao, L., Lin, Z., Qin, T., Bian, J., and Liu, T. Fully parameterized quantile function for distributional reinforcement learning. *CoRR*, abs/1911.02140, 2019. URL http://arxiv.org/abs/1911.02140.

You, H., Yu, B., Jin, H., Yang, Z., and Sun, J. User-oriented robust reinforcement learning, 2022.

Yu, X. and Ying, L. On the global convergence of risk-averse policy gradient methods with expected conditional risk measures, 2023.

Yuan, H., Li, C. J., Tang, Y., and Zhou, Y. Policy optimization via stochastic recursive gradient algorithm, 2019. In *URL https://openreview. net/forum*, 2019.

Zhao, W., Chen, R., Sun, Y., Liu, R., Wei, T., and Liu, C. Guard: A safe reinforcement learning benchmark. *arXiv preprint arXiv:2305.13681*, 2023.

Zhou, F., Wang, J., and Feng, X. Non-crossing quantile regression for deep reinforcement learning. 2020. URL https://api.semanticscholar.org/CorpusID:231187649.

Zhu, W. and Rosendo, A. Proximal policy optimization smoothed algorithm. *CoRR*, abs/2012.02439, 2020. URL https://arxiv.org/abs/2012.02439.

A. Related Works

Model-Free Deep Reinforcement Learning Model-free deep reinforcement learning (RL) algorithms have found applications from the realm of games (Mnih et al., 2013b; Silver et al., 2016) to the intricate domain of robotic control (Schulman et al., 2015).

The leading contenders of the model free reinforcement learning algorithms include (i) deep Q-learning (Mnih et al., 2013a; Hausknecht & Stone, 2015; van Hasselt et al., 2015; Hessel et al., 2018), (ii) off-policy policy gradient methods (Silver et al., 2014; Lillicrap et al., 2015; Gu et al., 2016; Fujimoto et al., 2018; Haarnoja et al., 2018), and (iii) trust region on-policy policy gradient methods (Schulman et al., 2015; 2017).

Among those categories, Q-learning-based techniques (Mnih et al., 2013a) , augmented with function approximation, have exhibited remarkable prowess over tasks with discrete action spaces, e.g. Atari game playing (Bellemare et al., 2013). However, these methods perform poorly in the realm of continuous control benchmarks, notably exemplified in OpenAI Gym (Brockman et al., 2016a; Duan et al., 2016).

In contrast, off-policy policy gradient methods extend Q-learning-based strategies via introducing an independent actor network to handle continuous control tasks, as exemplified by the Deep Deterministic Policy Gradient (DDPG)(Lillicrap et al., 2015). However, off-policy methods suffer from stability issues and susceptibility to hyperparameter tuning nuances(Haarnoja et al., 2018). Recently, enhancements have been made to incorporate entropy to foster exploration (Haarnoja et al., 2018) and mitigate the overestimation bias through target networks (Fujimoto et al., 2018). Additionally, Distributional RL emerges as powerful off-policy methods to model the entire distribution of returns instead of focusing solely on expected values, providing more information about the distribution of rewards, leading to potentially more stable and efficient learning. Early efforts try to model the distribution of value function with various parameterizations, such as quartiles, in algorithms like C51 (Bellemare et al., 2017), QR-DQN(Dabney et al., 2017), IQN(Dabney et al., 2018) and FQF (Yang et al., 2019). Recent advancements address issues like the non-decreasing property of learned quartiles (Zhou et al., 2020) and use distributional RL to reduce bias and variance in Q function estimation (Kuang et al., 2023). Despite these advancements, the convergence characteristics of off-policy policy gradient methods remain incompletely understood, primarily explored under stringent assumptions such as infinite sampling and Q-updates (Fujimoto et al., 2018). Moreover, off-policy policy gradient methods are primarily tailored for continuous action spaces.

Conversely, trust region on-policy policy gradient methods harmoniously accommodate both continuous and discrete action spaces while showcasing superior stability and dependable convergence properties. Notably, the representative Trust Region Policy Optimization (TRPO) (Schulman et al., 2015), complemented by its pragmatic counterpart, Proximal Policy Optimization (PPO) (Schulman et al., 2017), have consistently delivered impressive performance across an array of demanding benchmark tasks. Furthermore, PPO has largely helped training of groundbreaking artificial intelligence applications, including ChatGPT (Schulman et al., 2022), the automated Rubik's Cube solver with a robotic hand (Akkaya et al., 2019), and the championship-level drone racing (Kaufmann et al., 2023), thereby reaffirming their profound impact on advancing the frontiers of AI technology.

Attempts to Improve Trust Region Methods Recently, many efforts are made to improve trust region on-policy methods, including (i) *improve computation efficiency.* TREFree (Sun et al., 2023) introduced a novel surrogate objective that eliminates trust region constraints. (ii) *encourage exploration.* COPOS (Pajarinen et al., 2019) applied compatible value function approximation to effectively control entropy during policy updates. (iii) *improve training stability and data-efficiency.* Truly PPO (TR-PPO) (Wang et al., 2020) introduced a new clipping function and trust region-based triggering condition. PPO Smooth (PPOS) (Zhu & Rosendo, 2020) use a functional clipping method instead of a flat clipping method. Generalized PPO (GePPO) (Queeney et al., 2021) extended PPO to an off-policy variant, thereby enhancing sampling efficiency through data reuse. Early Stopping Policy Optimization (ESPO) (Sun et al., 2022) argued that the clip method in PPO is not reasonable and proposed an early stopping method to replace it. AlphaPPO (Xu et al., 2023) introduced alpha divergence, a metric that offers a more effective description of policy differences, resulting in more stable training performance.

More relevantly, there are improvements considering variance control, including (i) *variance reduction of policy gradient.* Xu et al. and Papini et al. applied the stochastic variance reduced gradient descent (SVRG) technique for getting stochastic variance-reduced version of policy gradient (SVRPO) to improve the sample efficiency. Yuan et al. incorporates the StochAstic Recursive grAdient algoritHm (SARAH) into the TRPO framework to get more stable variance. Song et al. uses on-policy adaptation of Maximum a Posteriori Policy Optimization to replace policy gradients which may have

large variance. (ii) *variance reduction of performance update.* Tomczak et al. introduced a surrogate objective with approximate importance sampling to strike a balance between performance update bias and variance. (iii) *variance reduction of importance sampling.* (Lin et al., 2023) introduced sample dropout to bound the variance of importance sampling estimate by dropping out samples when their ratio deviation is too high.

Although trust-region-based methods have achieved notable success, there hasn't been any approach to exert control over the worst case performance. Unforeseen instances of poor performance can result in training instability, thereby jeopardizing the reliability of solutions in real-world applications. In our research, we bridge this gap by introducing novel theoretical results that ensure a monotonic improvement of the lower bound of near-total performance samples.

Simultaneously, there exist additional domains warranting our consideration, particularly those aligned with the conceptual framework of optimizing worst-case performance. However, it is crucial to acknowledge the fundamental distinctions between these areas and our chosen approach.

Risk-sensitive RL / Probabilistic-constrained RL This method primarily operates in the realm of safe RL, focusing on minimizing risk-sensitive criteria like variance-related measures or percentile performance. Common metrics include Value-at-Risk (VaR) and conditional Value-at-Risk (CVaR) (Alexander & Baptista, 2004), aiming to quantify costs in the tail of a distribution. (Chow et al., 2015; Berkenkamp et al., 2017; Chow et al., 2018; Tang et al., 2019; Jain et al., 2021; Chen et al., 2023; Yu & Ying, 2023) addressed gradient computation under the Lagrangian function for percentile risk-constrained Markov Decision Processes (MDPs). While the objective of risk-sensitive RL aligns with ours, it is noteworthy that (i) Our commitment lies in the enhancement of rewards, excluding consideration for safe constraints. (ii) Diverging from the paradigm of risk-sensitive Reinforcement Learning, we employ surrogate functions and trust region methods to optimize confidence lower bounds. This strategic choice inherits the advantages inherent in trust region methods while preserving the extensibility of our broader research framework. (iii) Our approach, devoid of intricate assumptions, guarantees the monotonic ascent of the lower probability bound, surpassing mere adherence to constraints.

Robust Reinforcement Learning Robust RL research on the topic of mitigating the model discrepancy in the process of model generalization. (Peng et al., 2017) propose domain randomization (DR) method to solve the problem, which randomizes the simulator to generate a variety of environments for training a same policy in the source domain to get better generalization performance. (Rajeswaran et al., 2016) takes another way to learn policies that are robust to environment perturbations. They train the policy solely on the worst performance subset to get better worst case performance at the cost of sacrificing average performance. (Jiang et al., 2021) (MRPO) concurrently improve the average and worst-case performance by derive a lower bound for worst performance which is related to the expected performance. (Panaganti et al., 2022) proposes Robust Fitted Q-Ieration (RFQI) algorithm which uses only offline data and solves problems of data collection, optimization over models and unbiased estimation. (You et al., 2022) integrate user preference into policy learning in robust RL, and propose a novel User-Oriented Robust RL (UOR-RL) framework which gets rid of traditional max-min robustness. It is noteworthy that their emphasis lies on variable environments as opposed to a fixed and stable setting. This nuanced choice implies that their interpretation of "worst performance" pertains to the most adverse outcome arising from performance fluctuations induced by model generalization, diverging from a conventional reliance on the lower bound within a distribution.

B. Additional Experiment Results

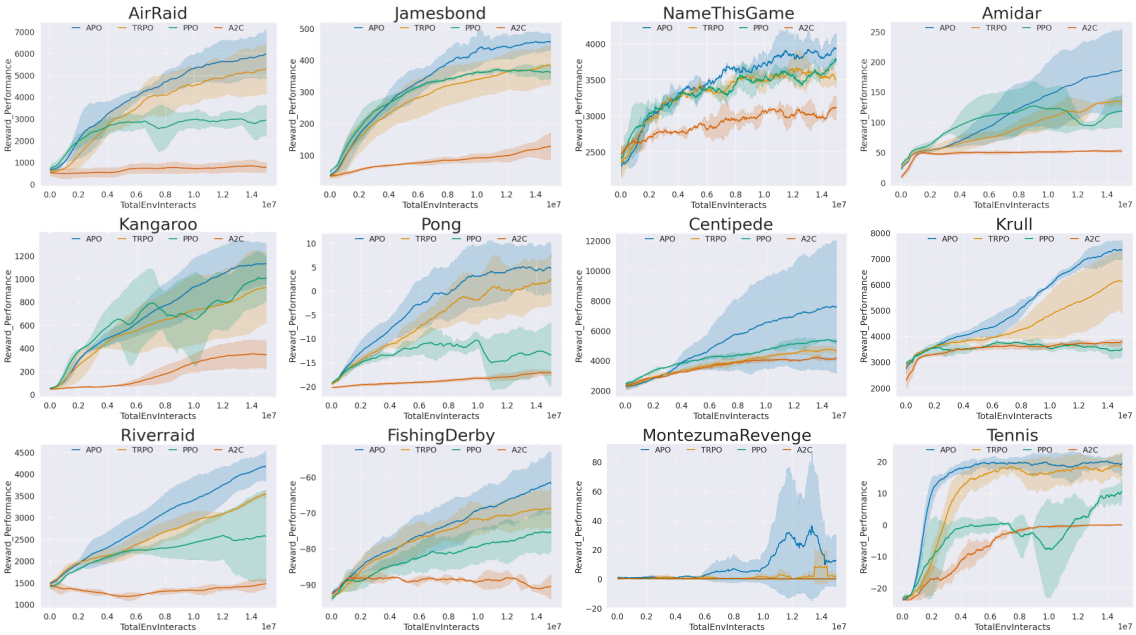


Figure 11: Comparison results from part of the total test suites in Atari Domain

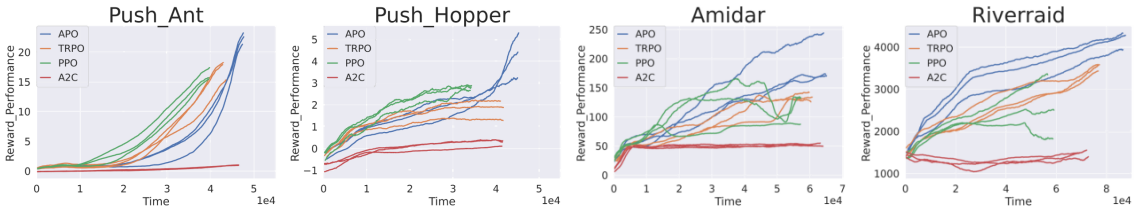


Figure 12: Comparison of wall-clock time from four representative test suites in continuous and discrete domain

Figure 13: Comparison of wall-clock time of PAPO

C. Lower Probability Bound

C.1. Preliminaries

First we define $R_\pi(s) = \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})$ as infinite-horizon discounted return starts at state s and define expected return $V_\pi(s) = \mathop{\mathbb{E}}_{\hat{\tau}\sim\pi}\big(R_\pi(s)\big)$ as the value of state s. Then for all trajectories $\hat{\tau} \sim \pi$ start from state $s_0 \sim \mu$, the expectation and variance of $R_\pi(s_0)$ can be respectively defined as $\mathcal{J}(\pi)$ and $\mathcal{V}(\pi)$. Formally:

$$\mathcal{J}(\pi) = \mathop{\mathbb{E}}_{\substack{s_0\sim\mu \\ \hat{\tau}\sim\pi}}\Big[R_\pi(s_0)\Big] = \mathop{\mathbb{E}}_{s_0\sim\mu}\Big[V_\pi(s_0)\Big] \tag{12}$$

$$\begin{aligned}
\mathcal{V}(\pi) &= \mathop{\mathbb{E}}_{\substack{s_0\sim\mu \\ \hat{\tau}\sim\pi}}\Big[\big(R_\pi(s_0) - \mathcal{J}(\pi)\big)^2\Big] \qquad (13)\\
&= \mathop{\mathbb{E}}_{s_0\sim\mu}\Big[\mathop{\mathbb{V}ar}_{\hat{\tau}\sim\pi}\big[R_\pi(s_0)\big] + \Big[\mathop{\mathbb{E}}_{\hat{\tau}\sim\pi}\big[R_\pi(s_0)\big]\Big]^2\Big] - \mathcal{J}(\pi)^2 \\
&= \mathop{\mathbb{E}}_{s_0\sim\mu}\Big[\mathop{\mathbb{V}ar}_{\hat{\tau}\sim\pi}\big[R_\pi(s_0)\big] + V_\pi(s_0)^2\Big] - \mathcal{J}(\pi)^2 \\
&= \mathop{\mathbb{E}}_{s_0\sim\mu}\Big[\mathop{\mathbb{V}ar}_{\hat{\tau}\sim\pi}\big[R_\pi(s_0)\big]\Big] + \mathop{\mathbb{E}}_{s_0\sim\mu}\Big[V_\pi(s_0)^2\Big] - \mathcal{J}(\pi)^2 \\
&= \underbrace{\mathop{\mathbb{E}}_{s_0\sim\mu}\Big[\mathop{\mathbb{V}ar}_{\hat{\tau}\sim\pi}\big[R_\pi(s_0)\big]\Big]}_{MeanVariance} + \underbrace{\mathop{\mathbb{V}ar}_{s_0\sim\mu}\big[V_\pi(s_0)\big]}_{VarianceMean}
\end{aligned}$$

Note that for the derivation of $\mathcal{V}(\pi)$ we treat the return of all trajectories as a mixture of one-dimensional distributions. Each distribution consists of the returns of trajectories from the same start state. The variance can then be divided into two parts:
1. **MeanVariance** reflects the expected variance of the return over different start states.
2. **VarianceMean** reflects the variance of the average return of different start states.

Proposition 1. *$\mathcal{B}_k(\pi) = \mathcal{J}(\pi) - k\mathcal{V}(\pi)$ is guaranteed to be an lower probability bound of performance defined by Definition 1.*

Proof. According to Selberg's inequality theory (Saw et al., 1984), if random variable $R_\pi(s_0)$ has finite non-zero variance $\mathcal{V}(\pi)$ and finite expected value $\mathcal{J}(\pi)$. Then for any real number $k \geq 0$, following inequality holds:

$$Pr(R_\pi(s_0) < \mathcal{J}(\pi) - k\mathcal{V}(\pi))) \leq \frac{1}{k^2\mathcal{V}(\pi) + 1} \quad , \tag{14}$$

which equals to:

$$Pr\big(R_\pi(s_0) \geq \mathcal{B}_k(\pi)\big) \geq 1 - \frac{1}{k^2\mathcal{V}(\pi) + 1} \quad . \tag{15}$$

Considering that $\pi \in \Pi$, then $\mathcal{V}(\pi)$ belongs to a corresponding variance space which has a non-zero minima of $\mathcal{V}(\pi)$ denoted as $\mathcal{V}_{min} \in \mathbb{R}^+$. Therefore, by treating $\psi = \mathcal{V}_{min}$, the following condition holds:

$$\begin{aligned}
\text{There exists } \psi > 0, \text{ s.t. } Pr\big(R_\pi(s_0) \geq \mathcal{B}_k(\pi)\big) &\geq 1 - \frac{1}{k^2\mathcal{V}(\pi) + 1} \qquad (16)\\
&\geq 1 - \frac{1}{k^2\mathcal{V}_{min} + 1} \\
&= \underbrace{1 - \frac{1}{k^2\psi + 1}}_{p_k^\psi} \quad .
\end{aligned}$$

□

Now to prove Theorem 1, we need to prove the optimization object in equation 4 serves as the lower bound of $\mathcal{B}_k(\pi')$. To do this, we first try to obtain the following terms:
1. Upper bound of **MeanVariance** with π'. (Appendix C.2)
2. Upper bound of **VarianceMean** with π'. (Appendix C.3)
3. Lower bound of $\mathcal{J}(\pi')$.(Appendix C.4)
And then we can proof our theorem by leveraging this lower bound in Appendix C.5

C.2. MeanVariance Bound

Proposition 2 (Bound of MeanVariance)**.** *Denote **MeanVariance** of policy π as $MV_\pi = \mathop{\mathbb{E}}_{s_0\sim\mu}[\mathbb{V}ar[R_\pi(s_0)]$. Given two policies π', π, the following bound holds:*

$$
\begin{aligned}
|MV_{\pi'} - MV_\pi| \leq \|\mu^\top\|_\infty \Bigg(\frac{1}{1-\gamma^2} \max_s \Bigg| & \mathop{\mathbb{E}}_{\substack{a\sim\pi \\ s'\sim P}} \left[\left(\frac{\pi'(a|s)}{\pi(a|s)} - 1 \right) A_\pi(s,a,s')^2 \right] \\
& + 2 \mathop{\mathbb{E}}_{\substack{a\sim\pi \\ s'\sim P}} \left[\left(\frac{\pi'(a|s)}{\pi(a|s)} \right) A_\pi(s,a,s') \right] |H(s,a,s')|_{max} + |H(s,a,s')|^2_{max} \Bigg| \\
& + \frac{2\gamma^2}{(1-\gamma^2)^2} \sqrt{\frac{1}{2} \mathcal{D}^{max}_{KL}(\pi' \| \pi)} \cdot \|\mathbf{\Omega}_\pi\|_\infty \Bigg)
\end{aligned}
\tag{17}
$$

where $\mathbf{\Omega}_\pi = \begin{bmatrix} \omega_\pi(s^1) \\ \omega_\pi(s^2) \\ \vdots \end{bmatrix}$ and $\omega_\pi(s) = \mathop{\mathbb{E}}_{\substack{a\sim\pi \\ s'\sim P}} \left[Q_\pi(s,a,s')^2 \right] - V_\pi(s)^2$ is defined as the variance of the state-action value function Q_π at state s. Additionally,

$$
\begin{aligned}
|H(s,a,s')|_{max} &= |L(s,a,s')| + \frac{2(1+\gamma)\gamma\epsilon}{(1-\gamma)^2} \mathcal{D}^{max}_{KL}(\pi' \| \pi) \\
L(s,a,s') &= \gamma \mathop{\mathbb{E}}_{\substack{s_0=s' \\ \hat{\tau}\sim\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \bar{A}_{\pi',\pi}(s_t) \right] - \mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \bar{A}_{\pi',\pi}(s_t) \right] \\
\epsilon &= \max_{s,a} |A_\pi(s,a)|
\end{aligned}
\tag{18}
$$

where $\mathcal{D}^{max}_{KL}(\pi' \| \pi) = \max_s \mathcal{D}_{KL}(\pi' \| \pi)[s]$.

Proof. According to (Sobel, 1982), the following Proposition holds

Proposition 3 (Theorem 1, (Sobel, 1982))**.** *Define $\boldsymbol{X}_\pi = \begin{bmatrix} \mathbb{V}ar[R_\pi(s^1)] \\ \mathbb{V}ar[R_\pi(s^2)] \\ \vdots \end{bmatrix}$ and $\hat{P}_\pi = P_\pi^\top$, where $\hat{P}_\pi(i,j)$ denotes the probability of the transfer from i-th state to j-th state, the following equation holds*

$$
\boldsymbol{X}_\pi = (I - \gamma^2 \hat{P}_\pi)^{-1} \mathbf{\Omega}_\pi \ . \tag{19}
$$

With $\boldsymbol{X}_\pi$, **MeanVariance** can be computed via

$$
\begin{aligned}
MV_\pi &= \mathop{\mathbb{E}}_{s_0\sim\mu}[\mathbb{V}ar[R_\pi(s_0)] \\
&= \mu^\top \boldsymbol{X}_\pi \\
&= \mu^\top \left((I - \gamma^2 \hat{P}_\pi)^{-1} \mathbf{\Omega}_\pi \right)
\end{aligned}
\tag{20}
$$

The divergence of **MeanVariance** we want to bound can be written as:

$$\begin{aligned}|MV_{\pi'} - MV_{\pi}| &= \Big|\mathop{\mathbb{E}}_{s_0\sim\mu}[\mathbb{V}ar[R_{\pi'}(s_0)] - \mathop{\mathbb{E}}_{s_0\sim\mu}[\mathbb{V}ar[R_{\pi}(s_0)]\Big| \\ &= \|\mu^\top\big(\boldsymbol{X}_{\pi'} - \boldsymbol{X}_{\pi}\big)\|_\infty \\ &\leq \|\mu^\top\|_\infty \|\boldsymbol{X}_{\pi'} - \boldsymbol{X}_{\pi}\|_\infty\end{aligned} \tag{21}$$

To bound $\|\boldsymbol{X}_{\pi'} - \boldsymbol{X}_{\pi}\|_\infty$, consider the following conditions:

$$\begin{aligned}\boldsymbol{X}_{\pi} - \gamma^2 \hat{P}_{\pi}\boldsymbol{X}_{\pi} &= \boldsymbol{\Omega}_{\pi} \\ \boldsymbol{X}_{\pi'} - \gamma^2 \hat{P}_{\pi'}\boldsymbol{X}_{\pi'} &= \boldsymbol{\Omega}_{\pi'}\end{aligned} \tag{22}$$

Let $G_\pi = (I - \gamma^2 \hat{P}_\pi)^{-1}$, then $\boldsymbol{X}_\pi$ can be written as:

$$\boldsymbol{X}_{\pi} = G_{\pi}\boldsymbol{\Omega}_{\pi} \tag{23}$$

Then we have:

$$\begin{aligned}\boldsymbol{X}_{\pi'} - \boldsymbol{X}_{\pi} &= G_{\pi'}\boldsymbol{\Omega}_{\pi'} - G_{\pi}\boldsymbol{\Omega}_{\pi} \\ &= G_{\pi'}\boldsymbol{\Omega}_{\pi'} - G_{\pi'}\boldsymbol{\Omega}_{\pi} + G_{\pi'}\boldsymbol{\Omega}_{\pi} - G_{\pi}\boldsymbol{\Omega}_{\pi} \\ &= G_{\pi'}\big(\boldsymbol{\Omega}_{\pi'} - \boldsymbol{\Omega}_{\pi}\big) + \big(G_{\pi'} - G_{\pi}\big)\boldsymbol{\Omega}_{\pi}\end{aligned} \tag{24}$$

Now $\|\boldsymbol{X}_{\pi'} - \boldsymbol{X}_{\pi}\|_\infty$ can be bounded by:

$$\|\boldsymbol{X}_{\pi'} - \boldsymbol{X}_{\pi}\|_\infty \leq \|G_{\pi'}\|_\infty \|\boldsymbol{\Omega}_{\pi'} - \boldsymbol{\Omega}_{\pi}\|_\infty + \|G_{\pi'} - G_{\pi}\|_\infty \|\boldsymbol{\Omega}_{\pi}\|_\infty \ . \tag{25}$$

Notice that $\|G_\pi\|_\infty$ is bounded by:

$$\|G_\pi\|_\infty = \|(I - \gamma^2 \hat{P}_\pi)^{-1}\|_\infty \leq \sum_{t=0}^{\infty} (\gamma^2)^t \|\hat{P}_\pi\|_\infty^t = (1-\gamma^2)^{-1} \ . \tag{26}$$

Therefore, $||G_\pi||_\infty = ||G_{\pi'}||_\infty = (1-\gamma^2)^{-1}$ and $\|\boldsymbol{\Omega}_\pi\|_\infty$ can be obtained with $\boldsymbol{\Omega}_\pi$. We only need to tackle with $\|G_{\pi'} - G_\pi\|_\infty$ and $\|\boldsymbol{\Omega}_{\pi'} - \boldsymbol{\Omega}_\pi\|_\infty$.
To get $\|G_{\pi'} - G_\pi\|_\infty$, we have

$$\begin{aligned}G_\pi^{-1} - G_{\pi'}^{-1} &= \big(I - \gamma^2 \hat{P}_\pi\big) - \big(I - \gamma^2 \hat{P}_{\pi'}\big) \\ &= \gamma^2\big(\hat{P}_{\pi'} - \hat{P}_\pi\big) \\ &= \gamma^2 \Delta\end{aligned} \tag{27}$$

where $\Delta = \hat{P}_{\pi'} - \hat{P}_\pi$. Then we have

$$G_{\pi'} - G_\pi = \gamma^2 G_{\pi'} \Delta G_\pi \tag{28}$$

With $\|\Delta\|_\infty = 2\mathcal{D}_{TV}^{max}(\pi'\|\pi)$, where $\mathcal{D}_{TV}^{max}(\pi'\|\pi) = \max_s \mathcal{D}_{TV}(\pi'\|\pi)[s]$, we have

$$\begin{aligned}\|\gamma^2 G_{\pi'} \Delta G_\pi\|_\infty &\leq \gamma^2 \|G_{\pi'}\|_\infty \|\Delta\|_\infty \|G_{\pi'}\|_\infty \\ &= \gamma^2 \cdot \frac{1}{1-\gamma^2} \cdot 2\mathcal{D}_{TV}^{max}(\pi'\|\pi) \cdot \frac{1}{1-\gamma^2} \\ &= \frac{2\gamma^2}{(1-\gamma^2)^2} \mathcal{D}_{TV}^{max}(\pi'\|\pi)\end{aligned} \tag{29}$$

According to (Brillinger, 2018), $\mathcal{D}_{TV}^{max}(\pi'\|\pi) \leq \sqrt{\frac{1}{2}\mathcal{D}_{KL}^{max}(\pi'\|\pi)}$, and given equation 21, we have

$$\begin{aligned}
|MV_{\pi'} - MV_{\pi}| &= \big|\mathop{\mathbb{E}}_{s_0\sim\mu}[\mathbb{V}ar[R_{\pi'}(s_0)] - \mathop{\mathbb{E}}_{s_0\sim\mu}[\mathbb{V}ar[R_{\pi}(s_0)]\big| \\
&\leq \|\mu^\top\|_\infty \|\boldsymbol{X}_{\pi'} - \boldsymbol{X}_{\pi}\|_\infty \\
&\leq \|\mu^\top\|_\infty \big(\frac{1}{1-\gamma^2}\|\boldsymbol{\Omega}_{\pi'} - \boldsymbol{\Omega}_{\pi}\|_\infty + \frac{2\gamma^2}{(1-\gamma^2)^2}\mathcal{D}_{TV}^{max}(\pi'\|\pi)\|\boldsymbol{\Omega}_{\pi}\|_\infty\big) \\
&\leq \|\mu^\top\|_\infty \big(\frac{1}{1-\gamma^2}\|\boldsymbol{\Omega}_{\pi'} - \boldsymbol{\Omega}_{\pi}\|_\infty + \frac{2\gamma^2}{(1-\gamma^2)^2}\sqrt{\frac{1}{2}\mathcal{D}_{KL}^{max}(\pi'\|\pi)} \cdot \|\boldsymbol{\Omega}_{\pi}\|_\infty\big)
\end{aligned} \tag{30}$$

To address $\|\boldsymbol{\Omega}_{\pi'} - \boldsymbol{\Omega}_{\pi}\|$, we notice that $\omega_\pi(s) = \mathop{\mathbb{V}ar}_{\substack{a\sim\pi \\ s'\sim P}}[Q_\pi(s,a,s')] = \mathop{\mathbb{V}ar}_{\substack{a\sim\pi \\ s'\sim P}}[A_\pi(s,a,s')]$, which means:

$$\|\boldsymbol{\Omega}_{\pi'} - \boldsymbol{\Omega}_{\pi}\|_\infty = \max_s \left|\mathop{\mathbb{V}ar}_{\substack{a\sim\pi' \\ s'\sim P}}[A_{\pi'}(s,a,s')] - \mathop{\mathbb{V}ar}_{\substack{a\sim\pi \\ s'\sim P}}[A_\pi(s,a,s')]\right| \tag{31}$$

Where

$$\begin{aligned}
A_\pi(s,a,s') &= R(s,a,s') + \gamma V_\pi(s') - V_\pi(s) \\
A_{\pi'}(s,a,s') &= R(s,a,s') + \gamma V_{\pi'}(s') - V_{\pi'}(s)
\end{aligned} \tag{32}$$

Define $H(s,a,s') = A_{\pi'}(s,a,s') - A_\pi(s,a,s')$, we have:

$$H(s,a,s') = \gamma(V_{\pi'}(s') - V_\pi(s')) - (V_{\pi'}(s) - V_\pi(s)) \tag{33}$$

Similar to TRPO (Schulman et al., 2015):

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi'}}\left[\sum_{t=0}^{\infty}\gamma^t A_\pi(s_t,a_t,s_{t+1})\right] \\
&= \mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi'}}\left[\sum_{t=0}^{\infty}\gamma^t (R_\pi(s_t,a_t,s_{t+1}) + \gamma V_\pi(s_{t+1}) - V_\pi(s_t))\right] \\
&= \mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi'}}\left[-V_\pi(s_0) + \sum_{t=0}^{\infty}\gamma^t R_\pi(s_t,a_t,s_{t+1})\right] \\
&= \mathop{\mathbb{E}}_{s_0=s}\left[-V_\pi(s_0)\right] + \mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi'}}\left[\sum_{t=0}^{\infty}\gamma^t R_\pi(s_t,a_t,s_{t+1})\right] \\
&= -V_\pi(s) + V_{\pi'}(s)
\end{aligned} \tag{34}$$

Then $H(s,a,s')$ can be written as:

$$H(s,a,s') = \gamma \mathop{\mathbb{E}}_{\substack{s_0=s' \\ \hat{\tau}\sim\pi'}}\left[\sum_{t=0}^{\infty}\gamma^t A_\pi(s_t,a_t,s_{t+1})\right] - \mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi'}}\left[\sum_{t=0}^{\infty}\gamma^t A_\pi(s_t,a_t,s_{t+1})\right] \tag{35}$$

Define $\bar{A}_{\pi',\pi}(s)$ to be the expected advantage of π' over π at state s:

$$\bar{A}_{\pi',\pi}(s) = \mathop{\mathbb{E}}_{a\sim\pi'}\left[A_\pi(s,a)\right] \tag{36}$$

Now $H(s,a,s')$ can be written as:

$$H(s,a,s') = \gamma \mathop{\mathbb{E}}_{\substack{s_0=s' \\ \hat{\tau}\sim\pi'}} \left[\sum_{t=0}^{\infty} \gamma^t \bar{A}_{\pi',\pi}(s_t) \right] - \mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi'}} \left[\sum_{t=0}^{\infty} \gamma^t \bar{A}_{\pi',\pi}(s_t) \right] \tag{37}$$

Define $L(s,a,s')$ as:

$$L(s,a,s') = \gamma \mathop{\mathbb{E}}_{\substack{s_0=s' \\ \hat{\tau}\sim\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \bar{A}_{\pi',\pi}(s_t) \right] - \mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \bar{A}_{\pi',\pi}(s_t) \right] \tag{38}$$

With $\epsilon = \mathbf{max}_{s,a}|A_\pi(s,a)|$, we have:

$$\begin{aligned}
&|H(s,a,s') - L(s,a,s')| \\
&= \Bigg| \gamma \Bigg(\mathop{\mathbb{E}}_{\substack{s_0=s' \\ \hat{\tau}\sim\pi'}} \left[\sum_{t=0}^{\infty} \gamma^t \bar{A}_{\pi',\pi}(s_t) \right] - \mathop{\mathbb{E}}_{\substack{s_0=s' \\ \hat{\tau}\sim\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \bar{A}_{\pi',\pi}(s_t) \right] \Bigg) \\
&- \Bigg(\mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi'}} \left[\sum_{t=0}^{\infty} \gamma^t \bar{A}_{\pi',\pi}(s_t) \right] - \mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \bar{A}_{\pi',\pi}(s_t) \right] \Bigg) \Bigg| \\
&\leq \gamma \Bigg| \mathop{\mathbb{E}}_{\substack{s_0=s' \\ \hat{\tau}\sim\pi'}} \left[\sum_{t=0}^{\infty} \gamma^t \bar{A}_{\pi',\pi}(s_t) \right] - \mathop{\mathbb{E}}_{\substack{s_0=s' \\ \hat{\tau}\sim\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \bar{A}_{\pi',\pi}(s_t) \right] \Bigg| \\
&+ \Bigg| \mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi'}} \left[\sum_{t=0}^{\infty} \gamma^t \bar{A}_{\pi',\pi}(s_t) \right] - \mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi}} \left[\sum_{t=0}^{\infty} \gamma^t \bar{A}_{\pi',\pi}(s_t) \right] \Bigg| \\
&\leq \frac{4\gamma(1+\gamma)\epsilon}{(1-\gamma)^2} (\mathcal{D}_{TV}^{max}(\pi' \| \pi))^2 \quad \leftarrow \texttt{([Lemma3, (Schulman et al., 2015)])}
\end{aligned} \tag{39}$$

Then according to (Brillinger, 2018) $\mathcal{D}_{TV}^{max}(\pi' \| \pi) \leq \sqrt{\frac{1}{2}\mathcal{D}_{KL}^{max}(\pi' \| \pi)}$, we can then bound $|H(s,a,s')|$ with:

$$\begin{aligned}
|H(s,a,s')| &\leq |L(s,a,s')| + \frac{4\gamma(1+\gamma)\epsilon}{(1-\gamma)^2} (\mathcal{D}_{TV}^{max}(\pi' \| \pi))^2 \\
&\leq |L(s,a,s')| + \frac{2\gamma(1+\gamma)\epsilon}{(1-\gamma)^2} \mathcal{D}_{KL}^{max}(\pi' \| \pi) \doteq |H(s,a,s')|_{max}
\end{aligned} \tag{40}$$

With $A_{\pi'}(s,a,s') = A_\pi(s,a,s') + H(s,a,s')$, we have:

$$
\begin{aligned}
&\underset{\substack{a\sim\pi'\\ s'\sim P}}{\mathbb{V}ar}[A_{\pi'}(s,a,s')] - \underset{\substack{a\sim\pi\\ s'\sim P}}{\mathbb{V}ar}[A_{\pi}(s,a,s')] \\
&= \underset{\substack{a\sim\pi'\\ s'\sim P}}{\mathbb{E}}[A_{\pi'}(s,a,s')^2] - \underset{\substack{a\sim\pi\\ s'\sim P}}{\mathbb{E}}[A_{\pi}(s,a,s')^2] \\
&= \underset{\substack{a\sim\pi'\\ s'\sim P}}{\mathbb{E}}[(A_{\pi}(s,a,s') + H(s,a,s'))^2] - \underset{\substack{a\sim\pi\\ s'\sim P}}{\mathbb{E}}[A_{\pi}(s,a,s')^2] \\
&= \underset{\substack{a\sim\pi'\\ s'\sim P}}{\mathbb{E}}[A_{\pi}(s,a,s')^2] - \underset{\substack{a\sim\pi\\ s'\sim P}}{\mathbb{E}}[A_{\pi}(s,a,s')^2] + 2\underset{\substack{a\sim\pi'\\ s'\sim P}}{\mathbb{E}}[A_{\pi}(s,a,s')H(s,a,s')] + \underset{\substack{a\sim\pi'\\ s'\sim P}}{\mathbb{E}}[H(s,a,s')^2] \\
&= \underset{\substack{a\sim\pi\\ s'\sim P}}{\mathbb{E}}\left[\left(\frac{\pi'(a|s)}{\pi(a|s)} - 1\right)A_{\pi}(s,a,s')^2\right] + 2\underset{\substack{a\sim\pi'\\ s'\sim P}}{\mathbb{E}}[A_{\pi}(s,a,s')H(s,a,s')] + \underset{\substack{a\sim\pi'\\ s'\sim P}}{\mathbb{E}}[H(s,a,s')^2] \\
&\leq \underset{\substack{a\sim\pi\\ s'\sim P}}{\mathbb{E}}\left[\left(\frac{\pi'(a|s)}{\pi(a|s)} - 1\right)A_{\pi}(s,a,s')^2\right] + 2\underset{\substack{a\sim\pi\\ s'\sim P}}{\mathbb{E}}\left[\left(\frac{\pi'(a|s)}{\pi(a|s)}\right)A_{\pi}(s,a,s')\right]|H(s,a,s')|_{max} + |H(s,a,s')|^2_{max}
\end{aligned}
\tag{41}
$$

Then we can bound $\|\mathbf{\Omega}_{\pi'} - \mathbf{\Omega}_{\pi}\|_{\infty}$ with:

$$
\begin{aligned}
&\|\mathbf{\Omega}_{\pi'} - \mathbf{\Omega}_{\pi}\|_{\infty} \\
&\leq \mathbf{max}_{s}\Bigg|\underset{\substack{a\sim\pi\\ s'\sim P}}{\mathbb{E}}\left[\left(\frac{\pi'(a|s)}{\pi(a|s)} - 1\right)A_{\pi}(s,a,s')^2\right] \\
&\quad + 2\underset{\substack{a\sim\pi\\ s'\sim P}}{\mathbb{E}}\left[\left(\frac{\pi'(a|s)}{\pi(a|s)}\right)A_{\pi}(s,a,s')\right]|H(s,a,s')|_{max} + |H(s,a,s')|^2_{max}\Bigg|
\end{aligned}
\tag{42}
$$

By substituting Equation (42) into Equation (30), Proposition 2 is proved. □

C.3. VarianceMean Bound

Proposition 4 (Bound of VarianceMean)**.** *Denote **VarianceMean** of policy π as $VM_{\pi} = \underset{s_0\sim\mu}{\mathbb{V}ar}[V_{\pi}(s_0)]$. Given two policies π', π, the **VarianceMean** of π' can be bounded by:*

$$
\begin{aligned}
VM_{\pi'} \leq &\underset{s_0\sim\mu}{\mathbb{E}}[V^2_{\pi}(s_0)] + \|\mu^{\top}\|_{\infty}\mathbf{max}_{s}\Big||\eta(s)|^2_{max} + 2|V_{\pi}(s)|\cdot|\eta(s)|_{max}\Big| \\
&- \left(\mathbf{min}\left\{\mathbf{max}\left\{0,\ \mathcal{J}^{l}_{\pi',\pi}\right\}, \mathcal{J}^{u}_{\pi',\pi}\right\}\right)^2
\end{aligned}
\tag{43}
$$

Proof.

$$
VM_{\pi'} = \underset{s_0\sim\mu}{\mathbb{E}}[V^2_{\pi'}(s_0)] - \mathcal{J}(\pi')^2 \tag{44}
$$

Since both terms on the right of Equation (44) are non-negative, we can bound $VM_{\pi'}$ with the upper bound of $\underset{s_0\sim\mu}{\mathbb{E}}[V^2_{\pi'}(s_0)]$ and the lower bound of $\mathcal{J}(\pi')^2$.

Define $\boldsymbol{Y}_{\pi} = \begin{bmatrix} V^2_{\pi}(s^1) \\ V^2_{\pi}(s^2) \\ \vdots \end{bmatrix}$, where $\underset{s_0\sim\mu}{\mathbb{E}}[V^2_{\pi}(s_0)] = \mu^{\top}\boldsymbol{Y}_{\pi}$. Then we have

$$
\begin{aligned}
&\left|\underset{s_0\sim\mu}{\mathbb{E}}[V^2_{\pi'}(s_0)] - \underset{s_0\sim\mu}{\mathbb{E}}[V^2_{\pi}(s_0)]\right| \\
&= \|\mu^{\top}\left(\boldsymbol{Y}_{\pi'} - \boldsymbol{Y}_{\pi}\right)\|_{\infty} \\
&\leq \|\mu^{\top}\|_{\infty}\|\boldsymbol{Y}_{\pi'} - \boldsymbol{Y}_{\pi}\|_{\infty}
\end{aligned}
\tag{45}
$$

To address $\|\boldsymbol{Y}_{\pi'} - \boldsymbol{Y}_{\pi}\|_{\infty}$, we have:

$$V_{\pi'}^2(s) - V_{\pi}^2(s) = \Big(V_{\pi'}(s) - V_{\pi}(s)\Big)\Big(V_{\pi'}(s) + V_{\pi}(s)\Big) \tag{46}$$

According to equation 34:

$$\begin{aligned} V_{\pi'}(s) - V_{\pi}(s) &= \mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi'}}\left[\sum_{t=0}^{\infty}\gamma^t A_{\pi}(s_t, a_t, s_{t+1})\right] \\ &= \mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi'}}\left[\sum_{t=0}^{\infty}\gamma^t \bar{A}_{\pi',\pi}(s_t)\right] \\ &\doteq \eta(s) \end{aligned} \tag{47}$$

Define $T(s) = \mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi}}\left[\sum_{t=0}^{\infty}\gamma^t \bar{A}_{\pi',\pi}(s_t)\right]$, then we have:

$$|\eta(s) - T(s)| = \left|\mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi'}}\left[\sum_{t=0}^{\infty}\gamma^t \bar{A}_{\pi',\pi}(s_t)\right] - \mathop{\mathbb{E}}_{\substack{s_0=s \\ \hat{\tau}\sim\pi}}\left[\sum_{t=0}^{\infty}\gamma^t \bar{A}_{\pi',\pi}(s_t)\right]\right| \leq \frac{4\gamma\epsilon}{(1-\gamma)^2}(\mathcal{D}_{TV}^{max}(\pi'\|\pi))^2 \tag{48}$$

And according to (Brillinger, 2018), we can bound $|\eta(s)|$ with:

$$\begin{aligned} |\eta(s)| &\leq |T(s)| + \frac{4\gamma\epsilon}{(1-\gamma)^2}(\mathcal{D}_{TV}^{max}(\pi'\|\pi))^2 \\ &\leq |T(s)| + \frac{2\gamma\epsilon}{(1-\gamma)^2}\mathcal{D}_{KL}^{max}(\pi'\|\pi)) \doteq |\eta(s)|_{max} \end{aligned} \tag{49}$$

Further, we can obtain:

$$\begin{aligned} |V_{\pi'}(s) + V_{\pi}(s)| &\leq |V_{\pi'}(s)| + |V_{\pi}(s)| \\ &= |V_{\pi'}(s)| - |V_{\pi}(s)| + 2|V_{\pi}(s)| \\ &\leq |V_{\pi'}(s) - V_{\pi}(s)| + 2|V_{\pi}(s)| \\ &\leq |\eta(s)|_{max} + 2|V_{\pi}(s)| \end{aligned} \tag{50}$$

Thus the following inequality holds:

$$\begin{aligned} &\|\boldsymbol{Y}_{\pi'} - \boldsymbol{Y}_{\pi}\|_{\infty} \\ &\leq \mathbf{max}_{s}\Big||V_{\pi'}(s) - V_{\pi}(s)| \cdot |V_{\pi'}(s) + V_{\pi}(s)|\Big| \\ &\leq \mathbf{max}_{s}\Big||\eta(s)|_{max} \cdot (|\eta(s)|_{max} + 2|V_{\pi}(s)|)\Big| \\ &= \mathbf{max}_{s}\Big||\eta(s)|_{max}^2 + 2|V_{\pi}(s)| \cdot |\eta(s)|_{max}\Big| \end{aligned} \tag{51}$$

Substitute Equation (51) into Equation (45) the upper bound of $\mathop{\mathbb{E}}_{s_0\sim\mu}[V_{\pi'}^2(s_0)]$ is obtained:

$$\mathop{\mathbb{E}}_{s_0\sim\mu}[V_{\pi'}^2(s_0)] \leq \mathop{\mathbb{E}}_{s_0\sim\mu}[V_{\pi}^2(s_0)] + \|\mu^{\top}\|_{\infty}\mathbf{max}_{s}\Big||\eta(s)|_{max}^2 + 2|V_{\pi}(s)| \cdot |\eta(s)|_{max}\Big| \tag{52}$$

The lower bound of $\mathcal{J}(\pi')^2$ can then be obtained according to (Achiam et al., 2017):

$$\mathcal{J}(\pi')^2 \geq \left(\mathbf{min}\left\{\mathbf{max}\left\{0,\ \mathcal{J}_{\pi',\pi}^{l}\right\}, \mathcal{J}_{\pi',\pi}^{u}\right\}\right)^2 \tag{53}$$

where

$$\mathcal{J}^{l}_{\pi',\pi} = \mathcal{J}(\pi) + \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi'}}\left[A_{\pi}(s,a) - \frac{2\gamma\epsilon^{\pi'}}{1-\gamma}\sqrt{\frac{1}{2}\mathcal{D}_{KL}(\pi'\|\pi)[s]}\right]$$

$$\mathcal{J}^{u}_{\pi',\pi} = \mathcal{J}(\pi) + \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi'}}\left[A_{\pi}(s,a) + \frac{2\gamma\epsilon^{\pi'}}{1-\gamma}\sqrt{\frac{1}{2}\mathcal{D}_{KL}(\pi'\|\pi)[s]}\right]$$

By substituting Equation (52) and Equation (53) into Equation (44) Proposition 4 is proved. □

C.4. Expectation Bound [Theorem 1, (Achiam et al., 2017)]

Proposition 5. *For any policies π', π, with $\epsilon^{\pi'} \doteq \max_s |\mathop{\mathbb{E}}_{a\sim\pi'}[A^{\pi}(s,a)]|$, and define $d^{\pi} = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P(s_t = s|\pi)$ as the discounted state distribution using π, then the following bound holds:*

$$\mathcal{J}(\pi') - \mathcal{J}(\pi) \geq \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi'}}\left[A^{\pi}(s,a) - \frac{2\gamma\epsilon^{\pi'}}{1-\gamma}\mathcal{D}_{TV}(\pi'\|\pi)[s]\right] \tag{54}$$

Proof. d^{π} we used is defined as

$$d^{\pi}(\hat{s}) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P(s_t = s|\pi). \tag{55}$$

Then it allows us to express the expected discounted total reward compactly as:

$$\mathcal{J}(\pi) = \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi\\ s'\sim P}}\left[R(s,a,s')\right], \tag{56}$$

where by $a \sim \pi$, we mean $a \sim \pi(\cdot|s)$, and by $s' \sim P$,we mean $s' \sim P(\cdot|s,a)$. We drop the explicit notation for the sake of reducing clutter, but it should be clear from context that a and s' depend on s.

Define $P(s'|s,a)$ is the probability of transitioning to state s' given that the previous state was s and the agent took action a at state s, and $\mu : \mathcal{S} \mapsto [0,1]$ is the initial augmented state distribution. Let $p^t_{\pi} \in \mathbb{R}^{|\mathcal{S}|}$ denote the vector with components $p^t_{\pi}(s) = P(s_t = s|\pi)$, and let $P_{\pi} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ denote the transition matrix with components $P_{\pi}(s'|s) = \int P(s'|s,a)\pi(a|s)da$; then $p^t_{\pi} = P_{\pi}p^{t-1}_{\pi} = P^t_{\pi}\mu$ and

$$\begin{aligned} d^{\pi} &= (1-\gamma)\sum_{t=0}^{\infty}(\gamma P_{\pi})^t\mu \\ &= (1-\gamma)(I-\gamma P_{\pi})^{-1}\mu \end{aligned} \tag{57}$$

This formulation helps us easily obtain the following lemma.

Lemma 1. *For any function $f : \mathcal{S} \mapsto \mathbb{R}$ and any policy π,*

$$(1-\gamma)\mathop{\mathbb{E}}_{s\sim\mu}[f(s)] + \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi\\ s'\sim P}}[\gamma f(s')] - \mathop{\mathbb{E}}_{s\sim d^{\pi}}[f(s)] = 0. \tag{58}$$

Proof. Multiply both sides of equation 57 by $(I-\gamma P_{\pi})$ and take the inner product with the vector $f \in \mathbb{R}^{|\mathcal{S}|}$. □

Combining Lemma 1 with equation 56, we obtain the following, for any function f and any policy π:

$$\mathcal{J}(\pi) = \mathop{\mathbb{E}}_{s\sim\mu}[f(s)] + \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi\\ s'\sim P}}\left[R(s,a,s') + \gamma f(s') - f(s)\right] \tag{59}$$

Then we will derive and present the new policy improvement bound. We will begin with a lemma:

Lemma 2. *For any function $f \mapsto \mathbb{R}$ and any policies π' and π, define*

$$L_{\pi,f}(\pi') \doteq \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi\\ s'\sim P}}\left[\left(\frac{\pi'(a|s)}{\pi(a|s)}-1\right)(R(s,a,s')+\gamma f(s')-f(s))\right], \tag{60}$$

and $\epsilon_f^{\pi'} \doteq \max_s |\mathop{\mathbb{E}}_{\substack{a\sim\pi'\\ s'\sim P}}[R(s,a,s')+\gamma f(s')-f(s)]|$. Then the following bounds hold:

$$\mathcal{J}(\pi')-\mathcal{J}(\pi) \geq \frac{1}{1-\gamma}\left(L_{\pi,f}(\pi')-2\epsilon_f^{\pi'}D_{TV}(d^{\pi'}||d^{\pi})\right), \tag{61}$$

$$\mathcal{J}(\pi')-\mathcal{J}(\pi) \leq \frac{1}{1-\gamma}\left(L_{\pi,f}(\pi')+2\epsilon_f^{\pi'}D_{TV}(d^{\pi'}||d^{\pi})\right), \tag{62}$$

where D_{TV} is the total variational divergence. Furthermore, the bounds are tight(when $\pi' = \pi$, the LHS and RHS are identically zero).

Proof. First, for notational convenience, let $\delta_f(s,a,s') \doteq R(s,a,s')+\gamma f(s')-f(s)$. By equation 59, we obtain the identity

$$\mathcal{J}(\pi')-\mathcal{J}(\pi) = \frac{1}{1-\gamma}\left(\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'\\ s'\sim P}}[\delta_f(s,a,s')] - \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi\\ s'\sim P}}[\delta_f(s,a,s')]\right) \tag{63}$$

Now, we restrict our attention to the first term in equation 63. Let $\dagger\delta_f^{\pi'} \in \mathbb{R}^{|\mathcal{S}|}$ denote the vector of components, where $\dagger\delta_f^{\pi'}(s) = \mathop{\mathbb{E}}_{\substack{a\sim\pi'\\ s'\sim P}}[\delta_f(s,a,s')|s]$. Observe that

$$\begin{aligned}\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'\\ s'\sim P}}[\delta_f(s,a,s')] &= \left\langle d^{\pi'}, \dagger\delta_f^{\pi'}\right\rangle \\ &= \left\langle d^{\pi}, \dagger\delta_f^{\pi'}\right\rangle + \left\langle d^{\pi'}-d^{\pi}, \dagger\delta_f^{\pi'}\right\rangle\end{aligned}$$

With the Hölder's inequality; for any $p, q \in [1,\infty]$ such that $\frac{1}{p}+\frac{1}{q}=1$, we have

$$\left\langle d^{\pi}, \dagger\delta_f^{\pi'}\right\rangle + \left\|d^{\pi'}-d^{\pi}\right\|_p \left\|\dagger\delta_f^{\pi'}\right\|_q \geq \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'\\ s'\sim P}}[\delta_f(s,a,s')] \geq \left\langle d^{\pi}, \dagger\delta_f^{\pi'}\right\rangle - \left\|d^{\pi'}-d^{\pi}\right\|_p \left\|\dagger\delta_f^{\pi'}\right\|_q \tag{64}$$

We choose $p=1$ and $q=\infty$; With $\left\|d^{\pi'}-d^{\pi}\right\|_1 = 2D_{TV}(d^{\pi'}||d^{\pi})$ and $\left\|\dagger\delta_f^{\pi'}\right\|_\infty = \epsilon_f^{\pi'}$, and by the importance sampling identity, we have

$$\begin{aligned}\left\langle d^{\pi}, \dagger\delta_f^{\pi'}\right\rangle &= \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi'\\ s'\sim P}}[\delta_f(s,a,s')] \\ &= \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi\\ s'\sim P}}[\left(\frac{\pi'(a|s)}{\pi(a|s)}\right)\delta_f(s,a,s')]\end{aligned} \tag{65}$$

After bringing equation 65, $\left\|d^{\pi'}-d^{\pi}\right\|_1$, $\left\|\dagger\delta_f^{\pi'}\right\|_\infty$ into equation 64, then substract $\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi}\\ a\sim\pi\\ s'\sim P}}[\delta_f(s,a,s')]$, the bounds are obtained. The lower bound leads to equation 61, and the upper bound leads to equation 62. □

Then we will bound the divergence term, $||d^{\pi'} - d^{\pi}||_1$, i.e. $2D_{TV}(d^{\pi'}||d^{\pi})$.

Lemma 3. *The divergence between discounted future state visitation distributions, $||d^{\pi'} - d^{\pi}||_1$, is bounded by an average divergence of the policies π' and π:*

$$\|d^{\pi'} - d^{\pi}\|_1 \leq \frac{2\gamma}{1-\gamma} \mathop{\mathbb{E}}_{\hat{s}\sim d^{\pi}} \left[D_{TV}(\pi'||\pi)[s]\right], \tag{66}$$

where $D_{TV}(\pi'||\pi)[s] = \frac{1}{2}\sum_a |\pi'(a|s) - \pi(a|s)|$.

Proof. Firstly, we introduce an identity for the vector difference of the discounted future state visitation distributions on two different policies, π' and π. Define the matrices $G \doteq (I - \gamma P_{\pi})^{-1}, \bar{G} \doteq (I - \gamma P_{\pi'})^{-1}$, and $\Delta = P_{\pi'} - P_{\pi}$. Then:

$$\begin{aligned} G^{-1} - \bar{G}^{-1} &= (I - \gamma P_{\pi}) - (I - \gamma P_{\pi'}) \\ &= \gamma\Delta, \end{aligned} \tag{67}$$

left-multiplying by G and right-multiplying by $\bar{G}$, we obtain

$$\bar{G} - G = \gamma\bar{G}\Delta G. \tag{68}$$

Thus, the following equality holds:

$$\begin{aligned} d^{\pi'} - d^{\pi} &= (1-\gamma)\left(\bar{G} - G\right)\mu \\ &= \gamma(1-\gamma)\bar{G}\Delta G\mu \\ &= \gamma\bar{G}\Delta d^{\pi}. \end{aligned} \tag{69}$$

Using equation 69, we obtain

$$\begin{aligned} \|d^{\pi'} - d^{\pi}\|_1 &= \gamma\|\bar{G}\Delta d^{\pi}\|_1 \\ &\leq \gamma\|\bar{G}\|_1\|\Delta d^{\pi}\|_1, \end{aligned} \tag{70}$$

where $||\bar{G}||_1$ is bounded by:

$$\|\bar{G}\|_1 = \|(I - \gamma P_{\pi'})^{-1}\|_1 \leq \sum_{t=0}^{\infty} \gamma^t \|P_{\pi'}\|_1^t = (1-\gamma)^{-1}. \tag{71}$$

Next, we bound $\|\Delta d_1^{\pi}\|$ as following:

$$\begin{aligned} \|\Delta d^{\pi}\|_1 = &\sum_{s'} \left| \sum_{s} \Delta(s'|s) d^{\pi}(s) \right| \\ \leq &\sum_{s,s'} |\Delta(s'|s)| d^{\pi}(s) \\ = &\sum_{s,s'} \left| \sum_{a} P(s'|s,a)\left(\pi'(a|s) - \pi(a|s)\right) \right| d^{\pi}(s) \\ \leq &\sum_{s,a,s'} P(s'|s,a)|\pi'(a|s) - \pi(a|s)| d^{\pi}(s) \\ = &\sum_{s,a} |\pi'(a|s) - \pi(a|s)| d^{\pi}(s) \\ = &\ 2 \mathop{\mathbb{E}}_{s\sim d^{\pi}} [D_{TV}(\pi'||\pi)[s]]. \end{aligned} \tag{72}$$

By taking equation 72 and equation 71 into equation 70, this lemma is proved. □

The new policy improvement bound follows immediately.

Lemma 4. *For any function $f : \mathcal{S} \mapsto \mathbb{R}$ and any policies π' and π, define $\delta_f(s,a,s') \doteq R(s,a,s') + \gamma f(s') - f(s)$,*

$$\epsilon_f^{\pi'} \doteq \max_{\mathrm{s}} |\mathop{\mathbb{E}}_{\substack{a\sim\pi'\\ s'\sim P}}[\delta_f(s,a,s')]|,$$

$$L_{\pi,f}(\pi') \doteq \mathop{\mathbb{E}}_{\substack{s\sim d^\pi\\ a\sim\pi\\ s'\sim P}}\left[\left(\frac{\pi'(a|s)}{\pi(a|s)} - 1\right)\delta_f(s,a,s')\right], and$$

$$D^{\pm}_{\pi,f}(\pi') \doteq \frac{L_{\pi,f}(\pi')}{1-\gamma} \pm \frac{2\gamma\epsilon_f^{\pi'}}{(1-\gamma)^2}\mathop{\mathbb{E}}_{s\sim d^\pi}[D_{TV}(\pi'||\pi)[s]],$$

where $D_{TV}(\pi'||\pi)[s] = \frac{1}{2}\sum_a |\pi'(a|s) - \pi(a|s)|$ is the total variational divergence between action distributions at s. The following bounds hold:

$$D^{+}_{\pi,f}(\pi') \geq \mathcal{J}(\pi') - \mathcal{J}(\pi) \geq D^{-}_{\pi,f}(\pi').$$

Furthermore, the bounds are tight (when $\pi' = \pi$, all three expressions are identically zero)

Proof. Begin with the bounds from Lemma 2 and bound the divergence $D_{TV}(d^{\pi'}||d^\pi)$ by Lemma 3. □

The choice of $f = V_\pi$ in Lemma 4 leads to following inequality:

For any policies π', π, with $\epsilon^{\pi'} \doteq \max_s |\mathop{\mathbb{E}}_{a\sim\pi'}[A_\pi(s,a)]|$, the following bound holds:

$$\mathcal{J}(\pi') - \mathcal{J}(\pi) \geq \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^\pi\\ a\sim\pi'}}\left[A_\pi(s,a) - \frac{2\gamma\epsilon^{\pi'}}{1-\gamma}D_{TV}(\pi'||\pi)[s]\right]$$

At this point, the Proposition 5 is proved. □

C.5. Proof of Theorem 1

With Proposition 2, Proposition 4, and Proposition 5, we have the following surrogate function of lower probability bound $\mathcal{B}_k(\pi')$:

$$\mathcal{B}_k(\pi') \geq \mathcal{J}^l_{\pi',\pi} - k\left(MV_{\pi',\pi} + VM_{\pi',\pi}\right) \tag{73}$$

where

$$\begin{aligned}
MV_{\pi',\pi} &= \frac{\|\mu^\top\|_\infty}{1-\gamma^2}\max_s\Bigg|\mathop{\mathbb{E}}_{\substack{a\sim\pi'\\ s'\sim P}}\left[A_\pi(s,a,s')^2\right] - \mathop{\mathbb{E}}_{\substack{a\sim\pi\\ s'\sim P}}\left[A_\pi(s,a,s')^2\right] + |H(s,a,s')|^2_{max} \\
&\quad + 2\mathop{\mathbb{E}}_{\substack{a\sim\pi'\\ s'\sim P}}[A_\pi(s,a,s')]\cdot|H(s,a,s')|_{max}\Bigg| + MV_\pi + \frac{2\gamma^2\|\mu^\top\|_\infty}{(1-\gamma^2)^2}\sqrt{\frac{1}{2}\mathcal{D}^{max}_{KL}(\pi'\|\pi)}\cdot\|\Omega_\pi\|_\infty \\
VM_{\pi',\pi} &= \|\mu^\top\|_\infty\max_s\Big||\eta(s)|^2_{max} + 2|V_\pi(s)|\cdot|\eta(s)|_{max}\Big| + \mathop{\mathbb{E}}_{s_0\sim\mu}[V^2_\pi(s_0)] \\
&\quad - \left(\min\left\{\max\left\{0,\ \mathcal{J}^l_{\pi',\pi}\right\}, \mathcal{J}^u_{\pi',\pi}\right\}\right)^2 \\
\mathcal{J}^l_{\pi',\pi} &= \mathcal{J}(\pi) + \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^\pi\\ a\sim\pi'}}\left[A_\pi(s,a) - \frac{2\gamma\epsilon^{\pi'}}{1-\gamma}\sqrt{\frac{1}{2}\mathcal{D}_{KL}(\pi'\|\pi)[s]}\right]
\end{aligned}$$

We define $\mathcal{M}^j_k(\pi) = \mathcal{J}^l_{\pi,\pi_j} - k\left(MV_{\pi,\pi_j} + VM_{\pi,\pi_j}\right)$, and it can be found that $\mathcal{B}_k(\pi_j) = \mathcal{M}^j_k(\pi_j)$. Then by Equation (73), we have $\mathcal{B}_k(\pi_{j+1}) \geq \mathcal{M}^j_k(\pi_{j+1})$ and the following holds:

$$\mathcal{B}_k(\pi_{j+1}) - \mathcal{B}_k(\pi_j) \geq \mathcal{M}^j_k(\pi_{j+1}) - \mathcal{M}^j_k(\pi_j) \tag{74}$$

Thus, by maximizing $\mathcal{M}_k^j$ at each iteration, we guarantee that the true lower probability bound of performance $\mathcal{B}_k$ is non-decreasing. So far Theorem 1 has been proved.

Remark 4. $\mathcal{M}_k^j(\pi)$ is the objective function in our optimization problem which we can guarantee its monotonic improvement theoretically. Thus the RHS greater than or equal to zero, which can lead to the monotonic improvement of lower probability bound $\mathcal{B}_k(\pi)$.

C.6. Additional Results

Lemma 5. *$\mathcal{B}_k(\pi_j) = \mathcal{M}_k^j(\pi_j)$, where $\mathcal{B}_k(\pi_j) = \mathcal{J}(\pi_j) - k\mathcal{V}(\pi_j)$ and $\mathcal{M}_k^j(\pi) = \mathcal{J}^l_{\pi,\pi_j} - k\left(MV_{\pi,\pi_j} + VM_{\pi,\pi_j}\right)$.*

Proof. To prove Lemma 5, we will show that (i) $\mathcal{J}^l_{\pi_j,\pi_j} = \mathcal{J}(\pi_j)$, and (ii) $\left(MV_{\pi,\pi_j} + VM_{\pi,\pi_j}\right) = \mathcal{V}(\pi_j)$.

Expectation Part For the same policy π_j, the following two conditions hold:

$$\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_j}\\ a\sim\pi_j}}\left[A_{\pi_j}(s,a)\right] = 0 \tag{75}$$

$$\mathcal{D}_{KL}(\pi_j\|\pi_j)[s] = 0 \ . \tag{76}$$

With equation 76 and equation 75, the following equality holds:

$$\begin{aligned}\mathcal{J}^l_{\pi_j,\pi_j} &= \mathcal{J}(\pi_j) + \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_j}\\ a\sim\pi_j}}\left[A_{\pi_j}(s,a) - \frac{2\gamma\epsilon^{\pi}}{1-\gamma}\sqrt{\frac{1}{2}\mathcal{D}_{KL}(\pi_j\|\pi_j)[s]}\right] \\ &= \mathcal{J}(\pi_j) \ .\end{aligned} \tag{77}$$

Similarly,

$$\begin{aligned}\mathcal{J}^u_{\pi_j,\pi_j} &= \mathcal{J}(\pi_j) + \frac{1}{1-\gamma}\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_j}\\ a\sim\pi_j}}\left[A_{\pi_j}(s,a) + \frac{2\gamma\epsilon^{\pi_j}}{1-\gamma}\sqrt{\frac{1}{2}\mathcal{D}_{KL}(\pi_j\|\pi_j)[s]}\right] \\ &= \mathcal{J}(\pi_j) \ .\end{aligned} \tag{78}$$

Variance Part For the same policy π_j, the following conditions hold:

$$\mathop{\mathbb{E}}_{\substack{a\sim\pi_j\\ s'\sim P}}\left[A_{\pi_j}(s,a,s')^2\right] - \mathop{\mathbb{E}}_{\substack{a\sim\pi_j\\ s'\sim P}}\left[A_{\pi_j}(s,a,s')^2\right] = 0 \tag{79}$$

$$\mathop{\mathbb{E}}_{\substack{a\sim\pi_j\\ s'\sim P}}\left[A_{\pi_j}(s,a,s')\right] = 0 \tag{80}$$

$$\forall s, \bar{A}_{\pi_j,\pi_j}(s) = 0 \ . \tag{81}$$

equation 76, equation 80 and equation 81 indicate that:

$$\begin{aligned}|H(s,a,s')|_{max} &= \left|\gamma\mathop{\mathbb{E}}_{\substack{s_0=s'\\ \hat{\tau}\sim\pi_j}}\left[\sum_{t=0}^{\infty}\gamma^t\bar{A}_{\pi_j,\pi_j}(s_t)\right] - \mathop{\mathbb{E}}_{\substack{s_0=s\\ \hat{\tau}\sim\pi_j}}\left[\sum_{t=0}^{\infty}\gamma^t\bar{A}_{\pi_j,\pi_j}(s_t)\right]\right| + \frac{2\gamma(1+\gamma)\epsilon}{(1-\gamma)^2}\mathcal{D}^{max}_{KL}(\pi_j||\pi_j) \\ &= 0\end{aligned} \tag{82}$$

$$\begin{aligned}|\eta(s)|_{max} &= \left|\mathop{\mathbb{E}}_{\substack{s_0=s\\ \hat{\tau}\sim\pi_j}}\left[\sum_{t=0}^{\infty}\gamma^t\bar{A}_{\pi_j,\pi_j}(s_t)\right]\right| + \frac{2\gamma\epsilon}{(1-\gamma)^2}\mathcal{D}^{max}_{KL}(\pi_j\|\pi_j) \\ &= 0 \ .\end{aligned} \tag{83}$$

With equation 77, equation 78, equation 79, equation 82 and equation 83, we have the following condition hold:

$$
\begin{aligned}
& MV_{\pi_j,\pi_j} + VM_{\pi_j,\pi_j} \\
&= \frac{\|\mu^\top\|_\infty}{1-\gamma^2}\mathbf{max}_s \Bigg| \mathop{\mathbb{E}}_{\substack{a\sim\pi_j \\ s'\sim P}} \left[A_{\pi_j}(s,a,s')^2\right] - \mathop{\mathbb{E}}_{\substack{a\sim\pi_j \\ s'\sim P}} \left[A_{\pi_j}(s,a,s')^2\right] + |H(s,a,s')|^2_{max} \\
&\quad + 2\mathop{\mathbb{E}}_{\substack{a\sim\pi_j \\ s'\sim P}} \left[A_{\pi_j}(s,a,s')\right] \cdot |H(s,a,s')|_{max} \Bigg| + MV_{\pi_j} + \frac{2\gamma^2\|\mu^\top\|_\infty}{(1-\gamma^2)^2}\sqrt{\frac{1}{2}\mathcal{D}^{max}_{KL}(\pi_j\|\pi_j)} \cdot \|\Omega_{\pi_j}\|_\infty \\
&\quad + \|\mu^\top\|_\infty \mathbf{max}_s \Big| |\eta(s)|^2_{max} + 2|V_{\pi_j}(s)| \cdot |\eta(s)|_{max} \Big| - \mathbf{min}\ \left(\mathcal{J}(\pi)\right)^2 + \mathop{\mathbb{E}}_{s_0\sim\mu}[V^2_{\pi_j}(s_0)] \\
&= MV_{\pi_j} - \mathbf{min}\ \left(\mathcal{J}(\pi)\right)^2 + \mathop{\mathbb{E}}_{s_0\sim\mu}[V^2_{\pi_j}(s_0)] \\
&= \mathop{\mathbb{E}}_{s_0\sim\mu}[\mathop{\mathbb{V}ar}_{\hat{\tau}\sim\pi_j}[R_{\pi_j}(s_0)] - \mathop{\mathbf{min}}_{\mathcal{J}(\pi)\in[\mathcal{J}^l_{\pi_j,\pi_j},\mathcal{J}^u_{\pi_j,\pi_j}]} \left(\mathcal{J}(\pi)\right)^2 + \mathop{\mathbb{E}}_{s_0\sim\mu}[V^2_{\pi_j}(s_0)] \\
&= \mathop{\mathbb{E}}_{s_0\sim\mu}[\mathop{\mathbb{V}ar}_{\hat{\tau}\sim\pi_j}[R_{\pi_j}(s_0)] - \mathop{\mathbf{min}}_{\mathcal{J}(\pi)\in[\mathcal{J}(\pi_j),\mathcal{J}(\pi_j)]} \left(\mathcal{J}(\pi)\right)^2 + \mathop{\mathbb{E}}_{s_0\sim\mu}[V^2_{\pi_j}(s_0)] \\
&= \mathop{\mathbb{E}}_{s_0\sim\mu}[\mathop{\mathbb{V}ar}_{\hat{\tau}\sim\pi_j}[R_{\pi_j}(s_0)] + \mathop{\mathbb{E}}_{s_0\sim\mu}[V^2_{\pi_j}(s_0)] - \mathcal{J}(\pi_j)^2 \\
&= \mathcal{V}(\pi_j)
\end{aligned}
\tag{84}
$$

Summarize With equation 77 and equation 84, we have the following condition hold:

$$
\mathcal{B}_k(\pi_j) = \mathcal{J}(\pi_j) - k\mathcal{V}(\pi_j) = \mathcal{J}^l_{\pi_j,\pi_j} - k\left(MV_{\pi_j,\pi_j} + VM_{\pi_j,\pi_j}\right) = \mathcal{M}^j_k(\pi_j)\ , \tag{85}
$$

which proves the Lemma. □

D. APO Pseudocode

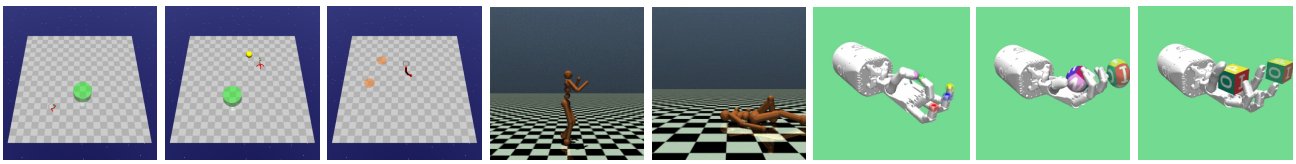
Algorithm 1 Absolute Policy Optimization

Input: Initial policy $\pi_0 \in \Pi_\theta$.
for $j = 0, 1, 2, \ldots$ **do**
  Sample trajectory $\tau \sim \pi_j = \pi_{\theta_j}$
  Estimate gradient $g \leftarrow \nabla_\theta O_{\pi,\pi_j}\big|_{\theta=\theta_j}$
  {Define $O_{\pi,\pi_j} = \left(\frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_j} \\ a\sim\pi}} \left[A_{\pi_j}(s,a) \right] - k \left(\overline{MV}_{\pi,\pi_j} + \overline{VM}_{\pi,\pi_j} \right) \right)$}
  Estimate Hessian $H \leftarrow \nabla_\theta^2 \mathbb{E}_{s\sim\pi_j}[\mathcal{D}_{KL}(\pi \| \pi_j)[s]]\big|_{\theta=\theta_j}$
  Solve convex programming {(Achiam et al., 2017)}

$$\theta^*_{j+1} = \arg\max_\theta g^\top(\theta - \theta_j)$$
$$\text{s.t. } \frac{1}{2}(\theta - \theta_j)^\top H(\theta - \theta_j) \le \delta$$

  Get search direction $\Delta\theta^* \leftarrow \theta^*_{j+1} - \theta_j$
  for $k = 0, 1, 2, \ldots$ **do** {Line search}
    $\theta' \leftarrow \theta_j + \xi^k \Delta\theta^*$ {$\xi \in (0,1)$ is the backtracking coefficient}
    if $\mathbb{E}_{s\sim\pi_j}[\mathcal{D}_{KL}(\pi_{\theta'} \| \pi_j)[s]] \le \delta$ **and** {Trust Region}
      $O_{\pi_{\theta'},\pi_j} \ge O_{\pi_j,\pi_j}$ **then** {Objective}
      $\theta_{j+1} \leftarrow \theta'$ {Update policy}
      break
    end if
  end for
end for

E. Expeiment Details



(a) Goal (b) Push (c) Chase (d) Humanoid (e) H.Standup (f) HandReach (g) H.M.Egg (h) H.M.Block

Figure 14: Tasks of continuous experiments

(a) Point (b) Swimmer (c) Arm3 (d) Drone (e) Hopper (f) Ant (g) Walker

Figure 15: Robots of continuous tasks benchmark GUARD.

E.1. GUARD Environment Settings

Goal Task In the Goal task environments, the reward function is:

$$r(x_t) = d^g_{t-1} - d^g_t + \mathbf{1}[d^g_t < R^g] \,,$$

where d^g_t is the distance from the robot to its closest goal and R^g is the size (radius) of the goal. When a goal is achieved, the goal location is randomly reset to someplace new while keeping the rest of the layout the same.

Push Task In the Push task environments, the reward function is

$$r(x_t) = d^r_{t-1} - d^r_t + d^b_{t-1} - d^b_t + \mathbf{1}[d^g_t < R^g] \,,$$

where d^r and d^b are the distance from the robot to its closest goal and the distance from the box to its closest goal, and R^g is the size (radius) of the goal. The box size is 0.2 for all the Push task environments. Like the goal task, a new goal location is drawn each time a goal is achieved.

Chase Task In the Chase task environments, the reward function is

$$r(x_t) = d^r_{t-1} - d^r_t + \mathbf{1}[d^g_t < R^g] \,,$$

where d^r is the distance from the robot to its closest goal and R^g is the size (radius) of the goal. Those targets continuously move away from the robot at a slow speed. The dense reward component provides a bonus for minimizing the distance between the robot and the targets. The targets are constrained to a circular area.

The test suites of APO and PAPO continuous experiments are summarized in Table 2 and Table 3, respectively.

State Space The internal state spaces describe the state of the robots, which can be obtained from standard robot sensors (accelerometer, gyroscope, magnetometer, velocimeter, joint position sensor, joint velocity sensor and touch sensor). The details of the internal state spaces of the robots in our test suites are summarized in Table 4.

Control Space For all the experiments, the control space of all robots are continuous, and linearly scaled to [-1, +1].

E.2. Policy Settings

The hyper-parameters used in our experiments are listed in Table 5 as default.

Table 2: The test suites environments of APO continuous experiments

Task Settings		Moving Area			Task			Dimension	
		Ground	Aerial	Fixed	Goal	Push	Chase	Low	High
GUARD Robot	Arm3 ($\mathbb{R}^3$)			✓	✓			✓	
	Drone ($\mathbb{R}^4$)		✓		✓			✓	
	Point ($\mathbb{R}^2$)	✓			✓	✓		✓	
	Swimmer ($\mathbb{R}^2$)	✓			✓	✓		✓	
	Hopper ($\mathbb{R}^5$)	✓			✓	✓		✓	
	Ant ($\mathbb{R}^8$)	✓				✓	✓		✓
	Walker ($\mathbb{R}^{10}$)	✓				✓	✓		✓
Mujoco Robot	Humanoid ($\mathbb{R}^{17}$)	✓							✓
	HumanoidStandup ($\mathbb{R}^{17}$)	✓							✓

Table 3: The test suites environments of PAPO continuous experiments

Task Settings		Moving Area			Task		Dimension	
		Ground	Aerial	Fixed	Goal	Chase	Low	High
GUARD Robot	Arm3 ($\mathbb{R}^3$)			✓	✓		✓	
	Drone ($\mathbb{R}^4$)		✓		✓		✓	
	Point ($\mathbb{R}^2$)	✓				✓	✓	
	Swimmer ($\mathbb{R}^2$)	✓				✓	✓	
	Hopper ($\mathbb{R}^5$)	✓				✓	✓	
	Ant ($\mathbb{R}^8$)	✓			✓	✓		✓
	Walker ($\mathbb{R}^{10}$)	✓			✓	✓		✓
Mujoco Robot	Humanoid ($\mathbb{R}^{17}$)	✓						✓
	HumanoidStandup ($\mathbb{R}^{17}$)	✓						✓
Gymnasium Robot	HandReach ($\mathbb{R}^{24}$)			✓				✓
	HandManipulateEgg ($\mathbb{R}^{24}$)			✓				✓
	HandManipulateBlock ($\mathbb{R}^{24}$)			✓				✓

Our experiments use separate multi-layer perceptrons with $tanh$ activations for the policy network and value network. Each network consists of two hidden layers of size (64,64). Policy networks and value networks are trained using $Adam$ optimizer. Policy networks are trained with a learning rate of 1e-3 while value networks are trained with 3e-4.

We apply an on-policy framework in our experiments. During each epoch the agent interacts B times with the environment and then performs a policy update based on the experience collected from the current epoch. The maximum length of the trajectory is set to 1000. The steps in each epoch are set to 30000. The total epoch number N is set to 200 in continuous tasks and 500 in atari tasks as default.

The policy update step is based on the scheme of TRPO, which performs up to 100 steps of backtracking with a coefficient of 0.8 for line searching.

For all experiments, we use a discount factor of $\gamma = 0.99$, an advantage discount factor $\lambda = 0.97$, and a KL-divergence step size of $\delta_{KL} = 0.02$.

Other unique hyper-parameters for each algorithm follow the original paper to attain best performance.

Each model is trained on a server with a 48-core Intel(R) Xeon(R) Silver 4214 CPU @ 2.2.GHz, Nvidia RTX A4000 GPU with 16GB memory, and Ubuntu 20.04.

Table 4: The internal state space components of different test suites environments.

Internal State Space	Point	Swimmer	Walker	Ant	Drone	Hopper	Arm3
Accelerometer ($\mathbb{R}^3$)	✓	✓	✓	✓	✓	✓	✓
Gyroscope ($\mathbb{R}^3$)	✓	✓	✓	✓	✓	✓	✓
Magnetometer ($\mathbb{R}^3$)	✓	✓	✓	✓	✓	✓	✓
Velocimeter ($\mathbb{R}^3$)	✓	✓	✓	✓	✓	✓	✓
Joint position sensor ($\mathbb{R}^n$)	$n = 0$	$n = 2$	$n = 10$	$n = 8$	$n = 0$	$n = 6$	$n = 3$
Joint velocity sensor ($\mathbb{R}^n$)	$n = 0$	$n = 2$	$n = 10$	$n = 8$	$n = 0$	$n = 6$	$n = 3$
Touch sensor ($\mathbb{R}^n$)	$n = 0$	$n = 4$	$n = 2$	$n = 8$	$n = 0$	$n = 1$	$n = 1$

Table 5: Important hyper-parameters of different algorithms in our experiments

Policy Parameter		A2C	TRPO	PPO	APO	PAPO	ESPO	α-PPO	V-MPO
Epochs in continuous tasks	N_1	200	200	200	200	200	200	200	200
Epochs in discrete tasks	N_2	500	500	500	500	-	-	-	-
Steps per epoch	B	30000	30000	30000	30000	30000	30000	30000	30000
Maximum length of trajectory	L	1000	1000	1000	1000	1000	1000	1000	1000
Discount factor	γ	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Advantage discount factor	λ	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
backtracking steps		-	100	-	100	-	-	-	-
backtracking coefficient		-	0.8	-	0.8	-	-	-	-
Target KL	δ_{KL}	-	0.02	0.02	0.02	0.02	-	0.02	0.02
Stopping Threshold	δ	-	-	-	-	-	0.25	-	-
Clip Ratio	ϵ	-	-	0.2	-	0.2	-	0.2	-
Probability factor	k	-	-	-	7	7	-	-	-
Policy network hidden layers		(64, 64)	(64, 64)	(64, 64)	(64, 64)	(64, 64)	(64, 64)	(64, 64)	(64, 64)
Policy network iteration		80	-	80	-	80	80	80	80
Policy network optimizer		Adam	-	Adam	-	Adam	Adam	Adam	Adam
Policy learning rate		3e-4	-	3e-4	-	3e-4	3e-4	3e-4	3e-4
Value network hidden layers		(64, 64)	(64, 64)	(64, 64)	(64, 64)	(64, 64)	(64, 64)	(64, 64)	(64, 64)
Value network iteration		80	80	80	80	80	80	80	80
Value network optimizer		Adam	Adam	Adam	Adam	Adam	Adam	Adam	Adam
Value learning rate		1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3

E.3. Normalized Score Settings

We intend to use 100% as the TRPO baseline measure. Thus the score we used is slightly changed from the normalization algorithm proposed by (van Hasselt et al., 2015) with the form as follow:

$$\Delta_1 \doteq score_{agent} - score_{random}$$
$$\Delta_2 \doteq score_{TRPO} - score_{random}$$
$$score_{normalized} = \frac{\Delta_2}{\Delta_1} \; if \; \Delta_1 < 0 \; and \; \Delta_2 < 0 \; else \; \frac{\Delta_1}{\Delta_2}$$

The difference is that we take the inverse of the metrics used in (van Hasselt et al., 2015) when $\Delta_1 < 0 \; and \; \Delta_2 < 0$. For further explanation, we need to address the positive and negative cases of Δ_1 and Δ_2:

- $\Delta_1 > 0 \; and \; \Delta_2 > 0$ In this case, $\frac{\Delta_1}{\Delta_2}$ can effectively demonstrate the ability of algorithms.
- $\Delta_1 < 0 \; and \; \Delta_2 < 0$ Practically in this case, assuming $score_{agent} > score_{TRPO}$, $score_{normalized}$ should be greater than 1 to demonstrate that agent is more capable than baseline TRPO. However, we will get a decimal if we obey the original score algorithms in (van Hasselt et al., 2015) which is incorrect. Thus we take the inverse of it in this situation.
- $\Delta_1 < 0 \; and \; \Delta_2 > 0$ In this case we will get a negative number which is reasonable to show the negative effects in terms of reward enhancement.
- $\Delta_1 > 0 \; and \; \Delta_2 < 0$ The changed normalization algorithm is still incorrect in this situation. However, we have not encountered such cases in all of our atari game statistics.

F. Total Experiment results

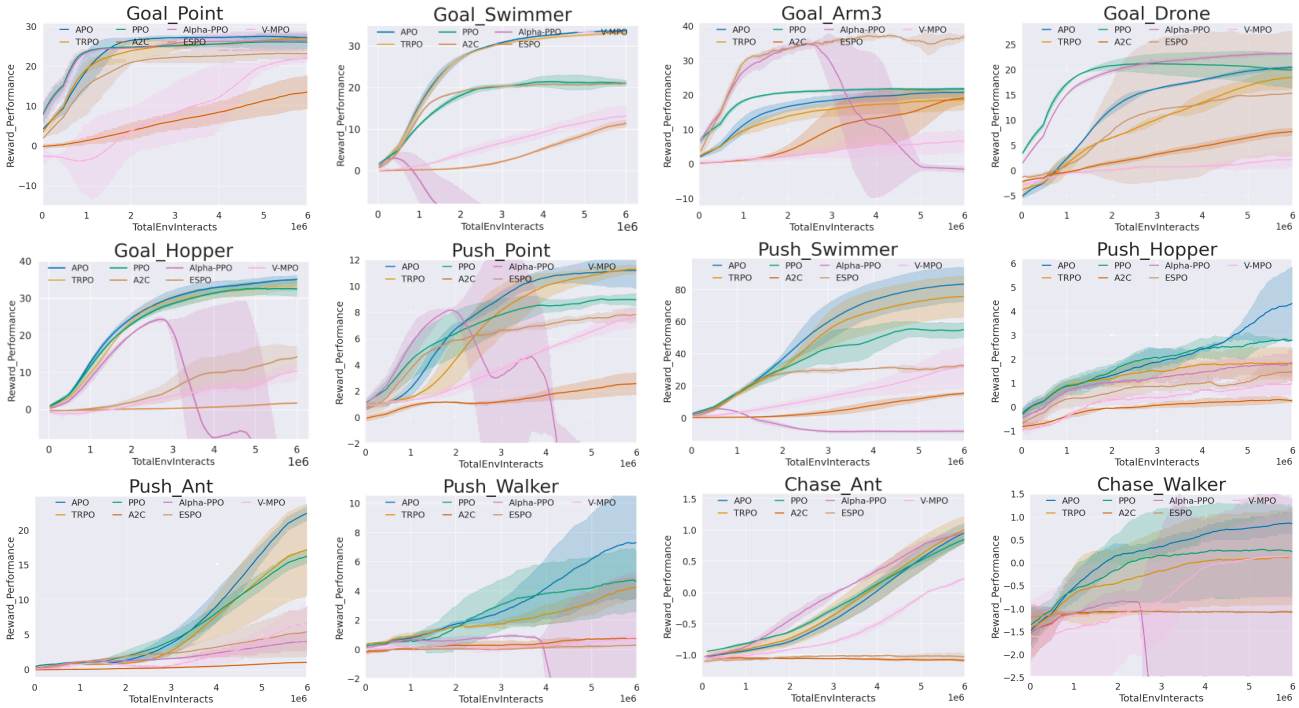


Figure 16: All continuous GUARD tasks expected performance learning curves for APO test suites.

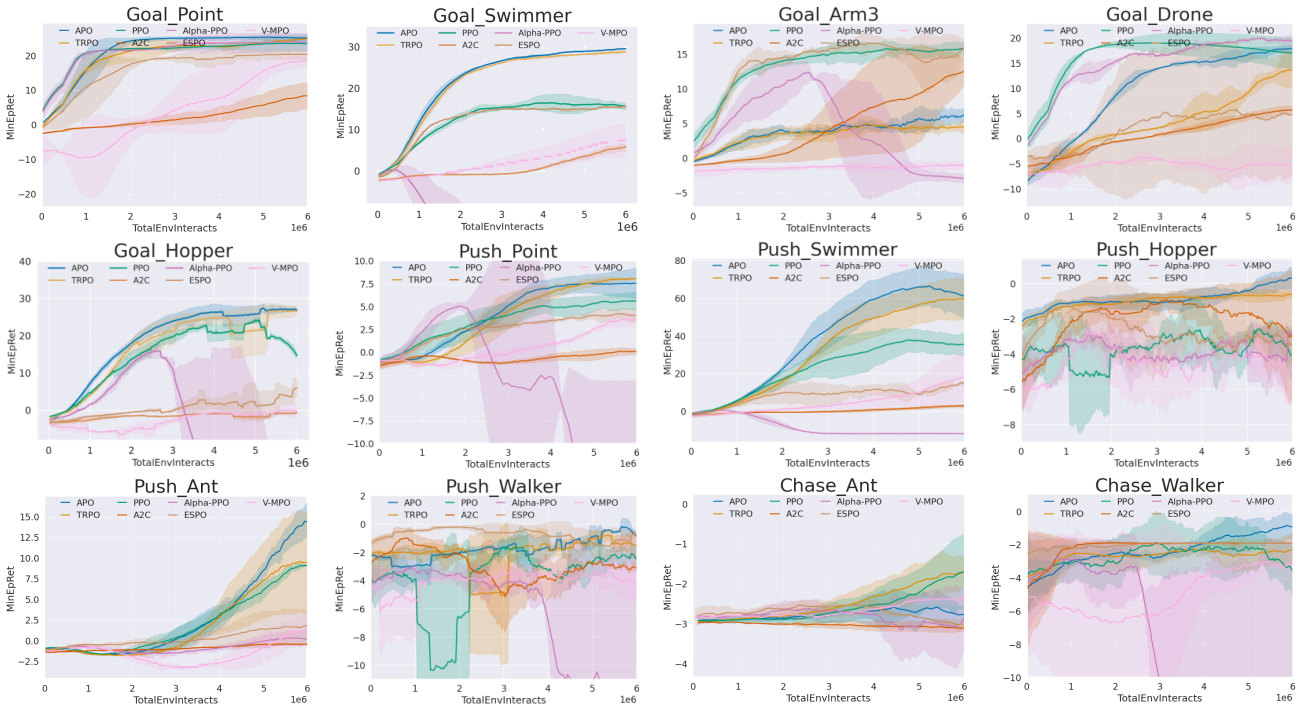


Figure 17: All continuous GUARD tasks worst performance learning curves for APO test suites.

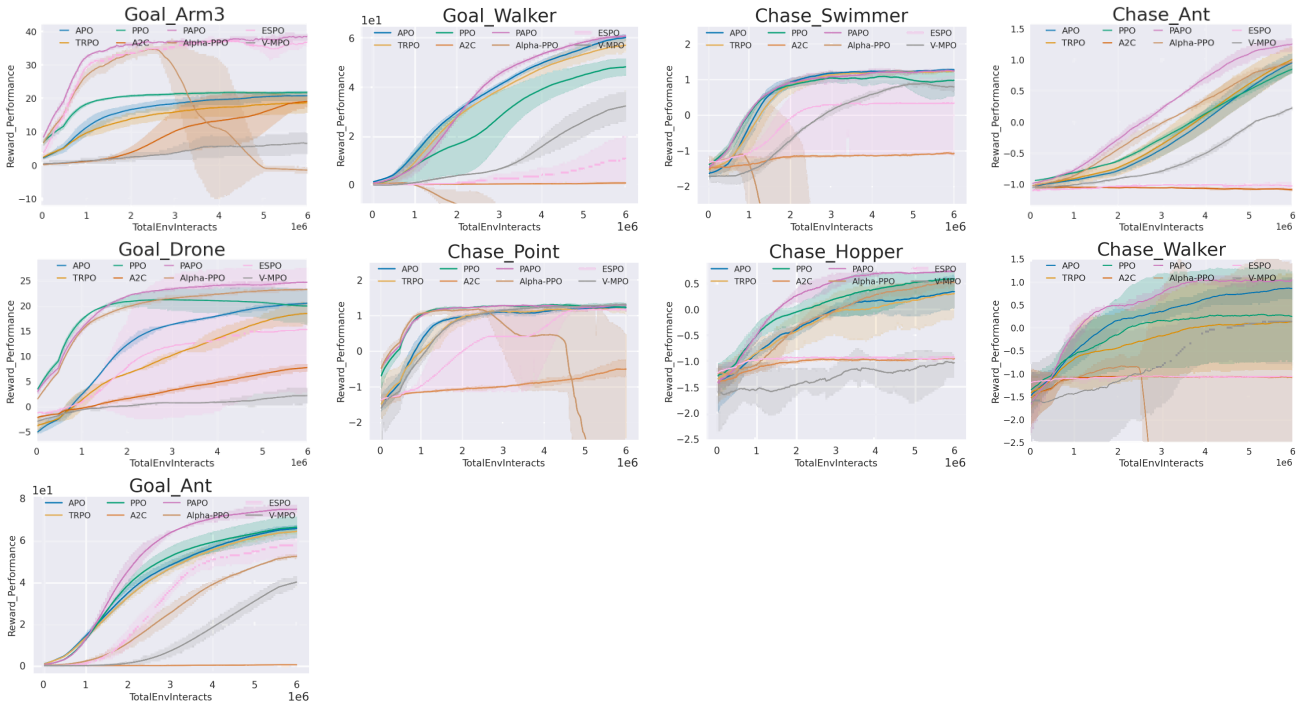


Figure 18: All continuous GUARD tasks expected performance learning curves for PAPO test suites.

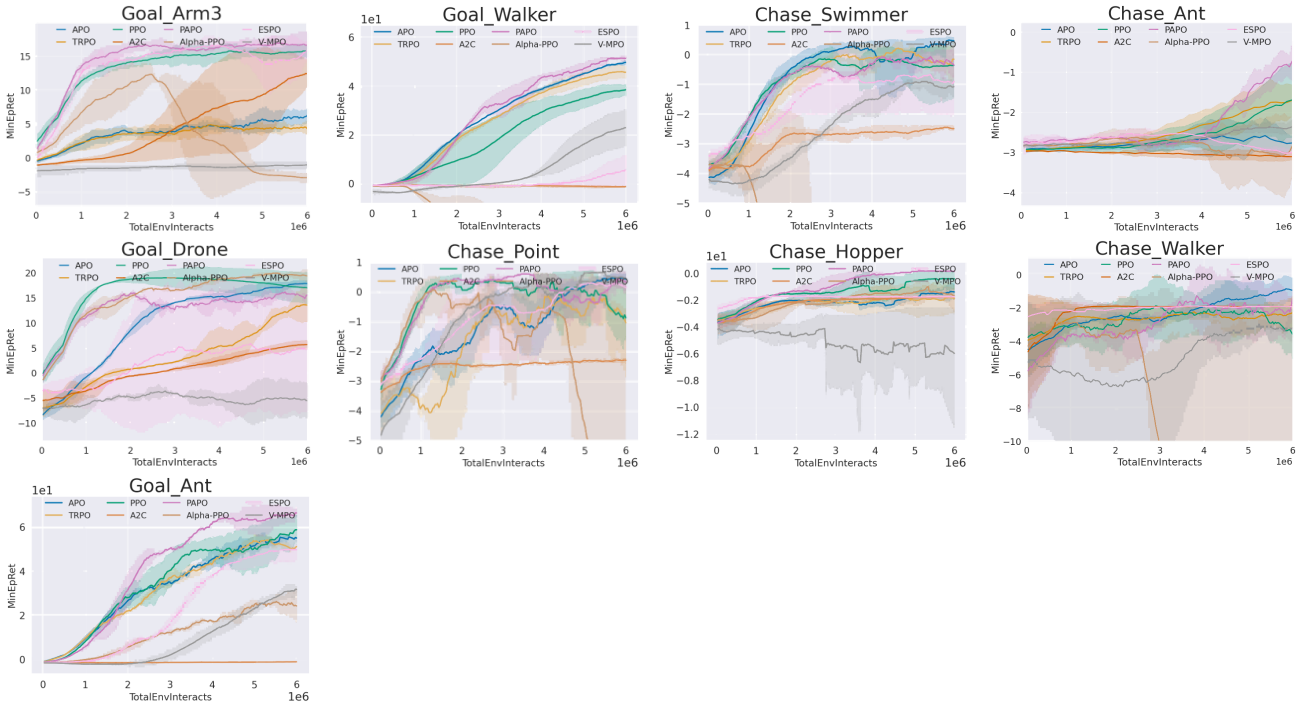


Figure 19: All continuous GUARD tasks worst performance learning curves for PAPO test suites.

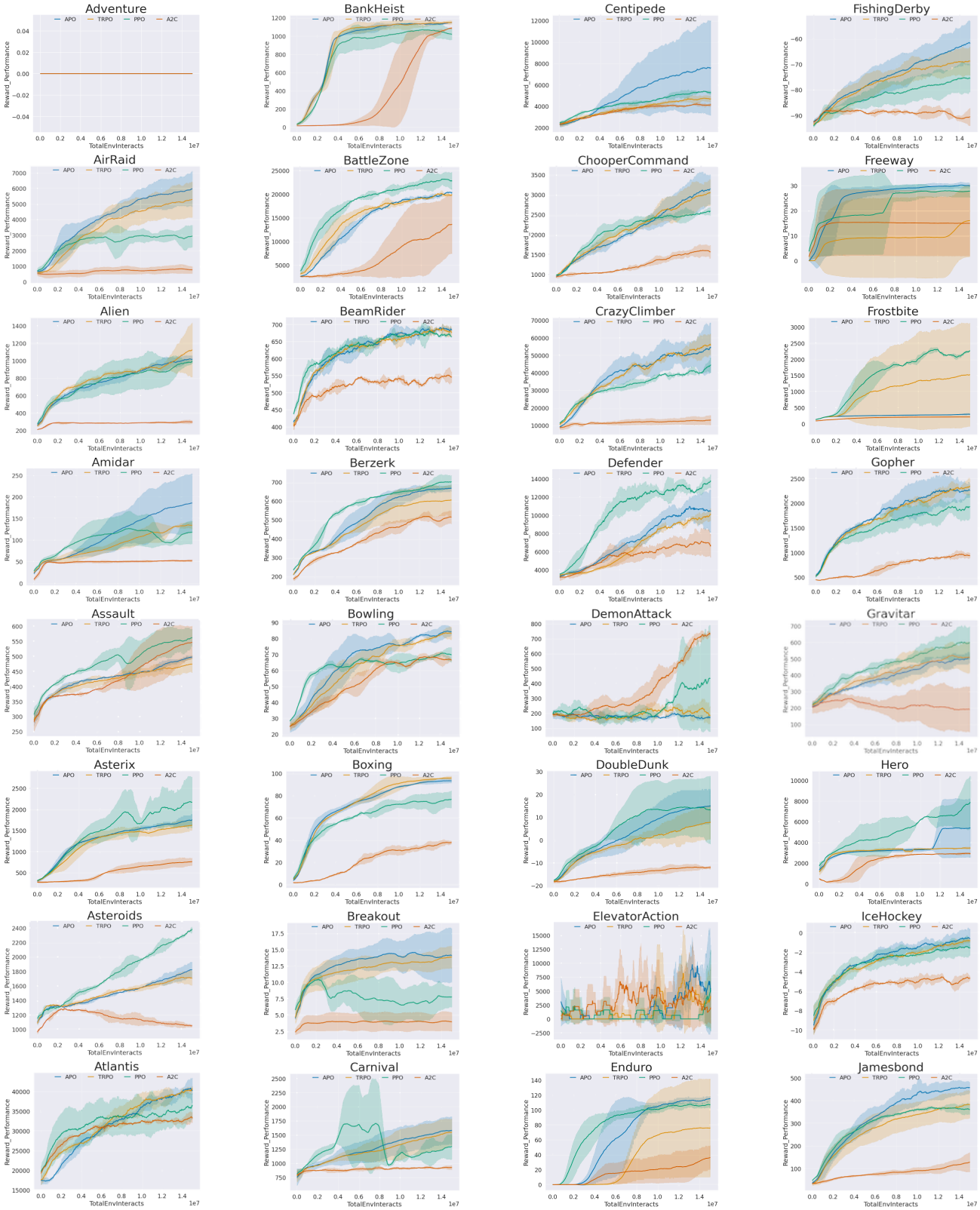


Figure 20: Comparison of expected performance on Atari Game No.1 - No. 32

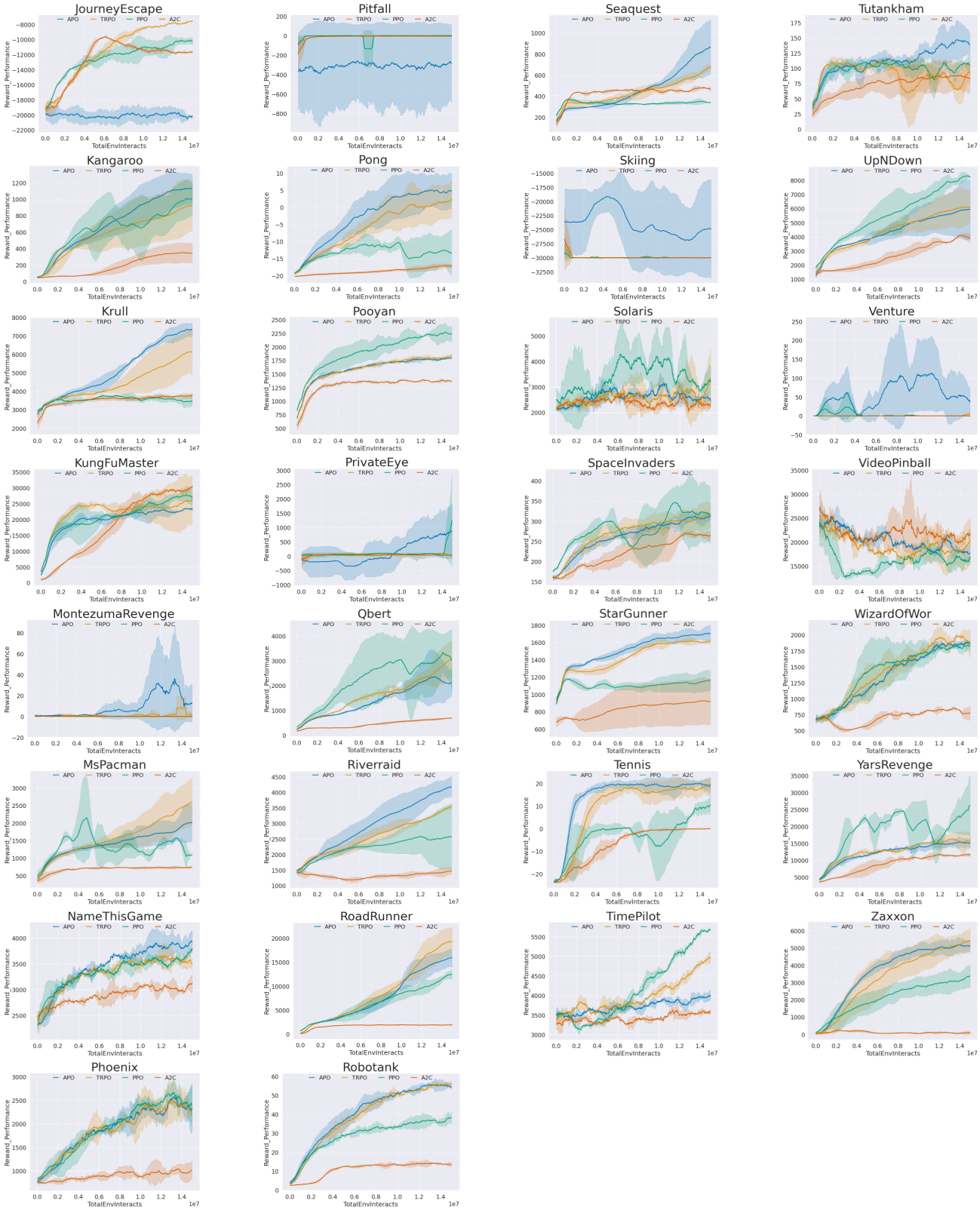


Figure 21: Comparison of expected performance on Atari Game No.33 - No. 62

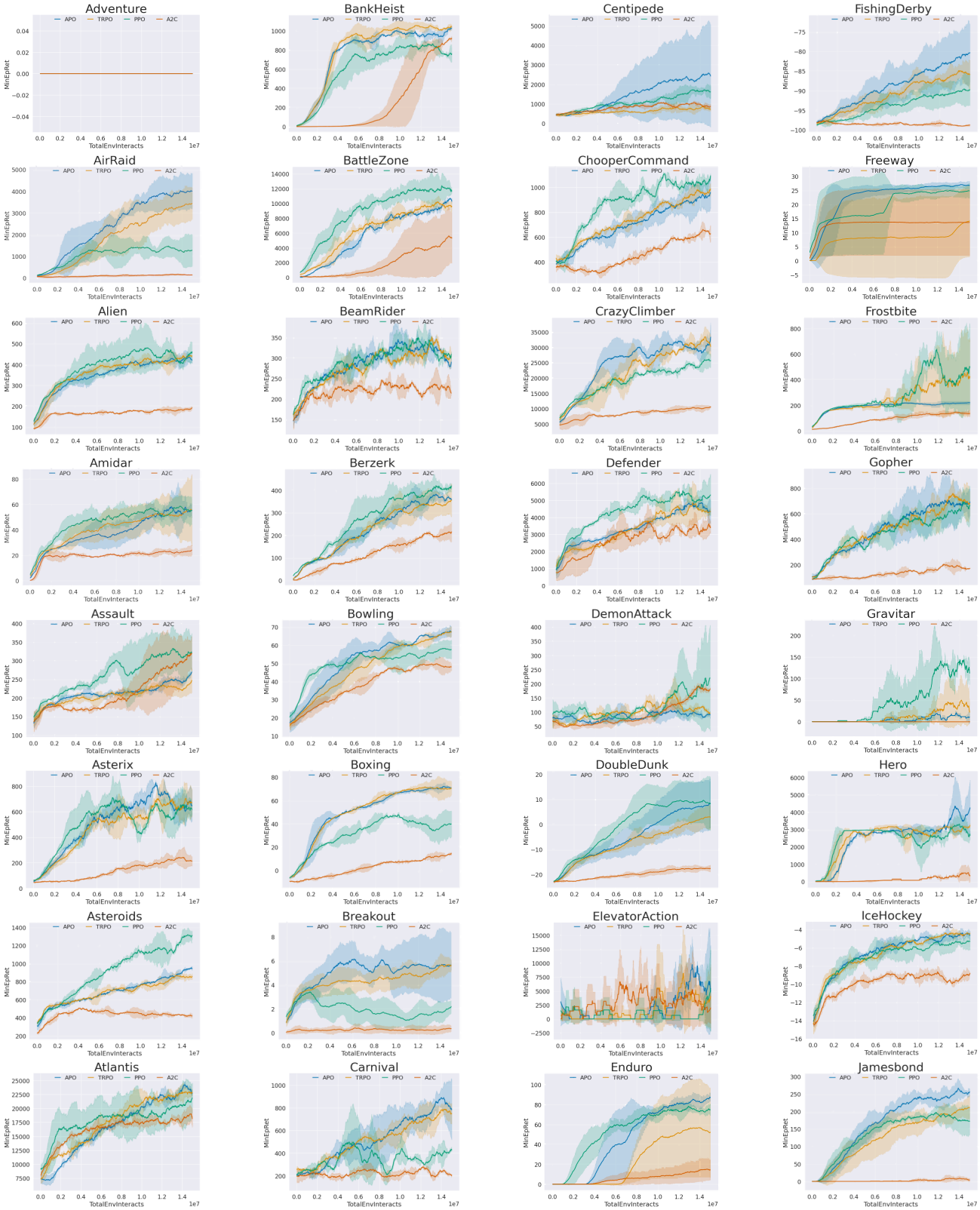


Figure 22: Comparison of worst performance on Atari Game No.1 - No. 32

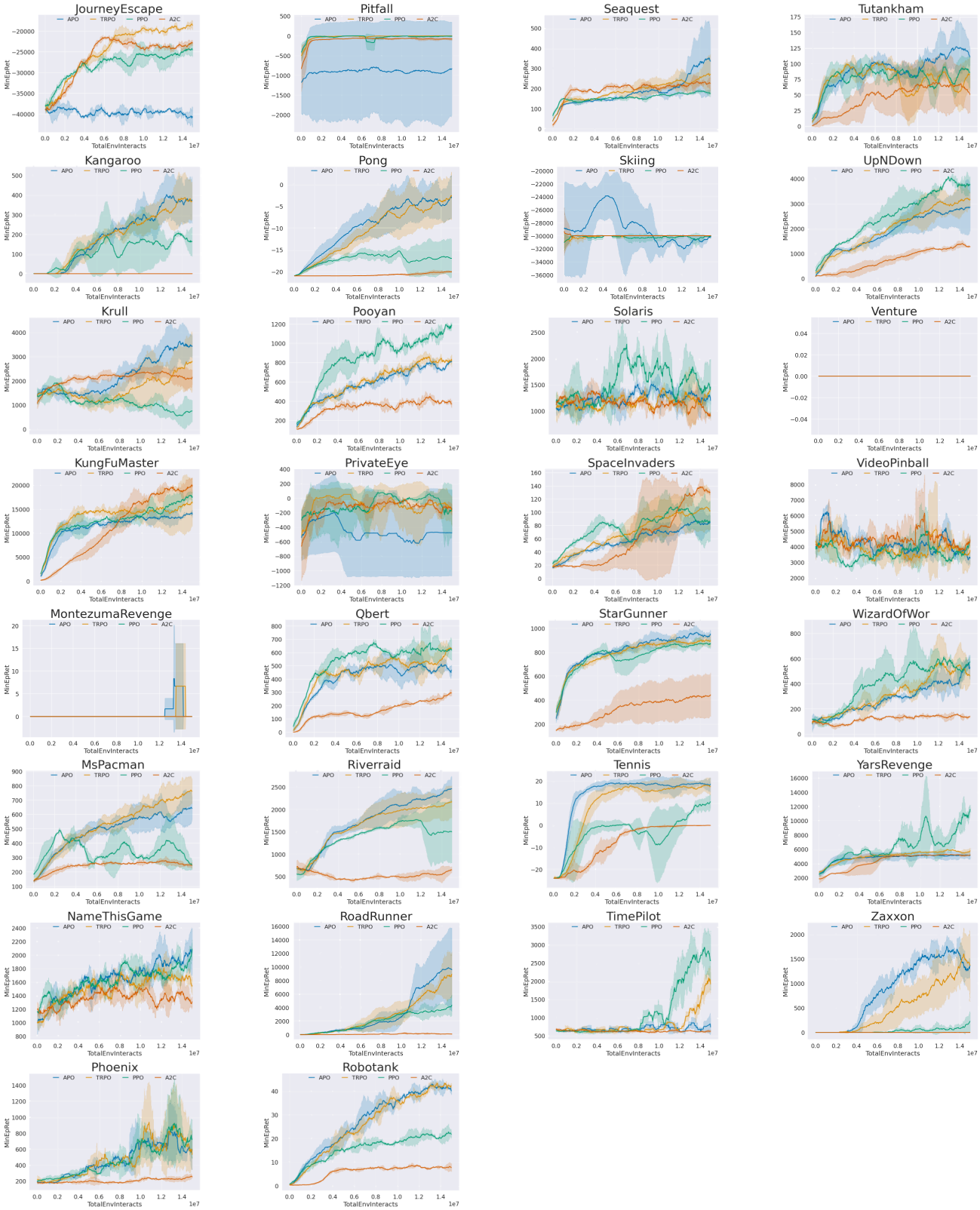


Figure 23: Comparison of worst performance on Atari Game No.33 - No. 62