Off-Policy Proximal Policy Optimization

Wenjia Meng[1], **Qian Zheng**[2,3], **Gang Pan**[2,3], **Yilong Yin**[1]*

[1] School of Software, Shandong University, Jinan, China
[2] The State Key Lab of Brain-Machine Intelligence, Zhejiang University, Hangzhou, China
[3] College of Computer Science and Technology, Zhejiang University, Hangzhou, China
wjmeng@sdu.edu.cn, qianzheng@zju.edu.cn, gpan@zju.edu.cn, ylyin@sdu.edu.cn

Abstract

Proximal Policy Optimization (PPO) is an important reinforcement learning method, which has achieved great success in sequential decision-making problems. However, PPO faces the issue of sample inefficiency, which is due to the PPO cannot make use of off-policy data. In this paper, we propose an Off-Policy Proximal Policy Optimization method (Off-Policy PPO) that improves the sample efficiency of PPO by utilizing off-policy data. Specifically, we first propose a clipped surrogate objective function that can utilize off-policy data and avoid excessively large policy updates. Next, we theoretically clarify the stability of the optimization process of the proposed surrogate objective by demonstrating the degree of policy update distance is consistent with that in the PPO. We then describe the implementation details of the proposed Off-Policy PPO which iteratively updates policies by optimizing the proposed clipped surrogate objective. Finally, the experimental results on representative continuous control tasks validate that our method outperforms the state-of-the-art methods on most tasks.

Introduction

Off-policy deep reinforcement learning has achieved huge success in domains, *e.g.*, games (Mnih et al. 2015), (Silver et al. 2016), (Silver et al. 2017), (Vinyals et al. 2019), (Schrittwieser et al. 2020), (Xing et al. 2021), (Meng et al. 2019), robotics (Kober, Bagnell, and Peters 2013), and continuous control tasks (Lillicrap et al. 2016), (Haarnoja et al. 2018), (Yang et al. 2022b). These off-policy deep reinforcement learning methods make use of off-policy data collected during the interaction between agent and environment to optimize policies (Degris, White, and Sutton 2012), (Silver et al. 2014), which are more sample efficient than on-policy methods only using on-policy data (Fujimoto, Hoof, and Meger 2018), (Mnih et al. 2016). With the utilization of off-policy data whose behavior policy differs from the target policy, these off-policy deep reinforcement learning methods can avoid the expensive cost on large amounts of on-policy interaction and are suitable for solving complex real-world sequential decision-making problems (Haarnoja et al. 2018), (Yang et al. 2022a), (Lillicrap et al. 2016), (Mnih et al. 2015).

Proximal Policy Optimization (PPO) (Schulman et al. 2017) is one of the most popular deep reinforcement learning methods, which optimizes policies by optimizing a clipped surrogate objective of policy performance. In order to further improve the sample efficiency of PPO, several works are proposed to achieve this goal from different perspectives. Specifically, Trust Region-Guided Proximal Policy Optimization (TRGPPO) (Wang et al. 2019) improves the sample efficiency of PPO by adaptively adjusting the clipping range within a trust region. Truly Proximal Policy Optimization (Wang, He, and Tan 2020) improves the sample efficiency of PPO by adopting a new clipping function to restrict the policy ratio, and substituting the triggering condition for clipping by a trust region-based one. Separated Trust Regions Policy Optimization (Zou et al. 2019) improves the sample efficiency of PPO by proposing a softer objective with more conservative constraints and building the separated trust-region for optimization. However, these methods ignore the perspective of *directly utilizing off-policy data to improve the sample efficiency of PPO* (Wang et al. 2019), (Wang, He, and Tan 2020), (Zou et al. 2019).

In this paper, we put forward an Off-Policy Proximal Policy Optimization (Off-Policy PPO) method that leverages off-policy data to further improve the sample efficiency of PPO. Specifically, we first propose a clipped surrogate objective that can use off-policy data and avoid the excessively large policy update. Next, we theoretically clarify the use of off-policy data during the optimization process of this objective does not harm the stability of PPO. We then introduce the implementation details of the proposed Off-Policy PPO, which include the whole procedure of this method and the network update procedure in this method. Our contributions are described as follows:

- We propose an Off-Policy Proximal Policy Optimization method (Off-Policy PPO) that introduces a clipped surrogate objective using off-policy data and iteratively utilizes off-policy data to optimize policies by maximizing this proposed clipped surrogate objective.
- We theoretically clarify that the stability of the proposed Off-Policy PPO by demonstrating the degree of the policy update distance in our method is the same as that in PPO. We also conduct experiments on a variety of representative continuous control tasks and the experimental results demonstrate that our method can achieve better

*Y. Yin is the corresponding author.

performance than state-of-the-art methods on most tasks.

Background & Notation

In this paper, we study the *Markov decision process* denoted by the tuple (S, A, P, ρ_0, r). S and A separately represent the state space and the action space; $P : S \times A \times S \to \mathbb{R}$ denotes the transition dynamics distribution; $\rho_0 : S \to \mathbb{R}$ and $r : S \times A \to \mathbb{R}$ represent distribution of the initial state s_0 and the reward function, respectively. During the interaction, the agent given a state s_t chooses an action a_t conforming to policy $\pi : S \times A \to [0, 1]$ at timestep t; environment yields a reward $r(s_t, a_t)$ and the next state s_{t+1}.

With above interaction, the discounted return from timestep t can be formulated as $R_t = \sum_{k=t}^{\infty} \gamma^{k-t} r(s_k, a_k)$, where γ is the discount factor. Based on such R_t, we next introduce the state value $V_\pi(s_t)$ given s_t, the action value $Q_\pi(s_t, a_t)$ given (s_t, a_t) and the corresponding advantage value $A_\pi(s_t, a_t)$ (Schulman et al. 2016):

$$V_\pi(s_t) = \mathbb{E}_{a_t, s_{t+1}, \cdots \sim \pi} \big[\sum_{k=t}^{\infty} \gamma^{k-t} r(s_k, a_k) \big], \qquad (1)$$

$$Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t+1}, a_{t+1}, \sim \pi} \big[\sum_{k=t}^{\infty} \gamma^{k-t} r(s_k, a_k) \big], \qquad (2)$$

$$A_\pi(s_t, a_t) = Q_\pi(s_t, a_t) - V_\pi(s_t). \qquad (3)$$

The standard reinforcement learning learns a policy π to maximize the policy performance objective (the discounted return from the start state) (Sutton and Barto 2018):

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \cdots}[R_0] = \mathbb{E}_{s_0, a_0, \cdots} \big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \big] \qquad (4)$$

where $s_0 \sim \rho_0, a_t \sim \pi(a_t|s_t), s_{t+1} \sim P(s_{t+1}|s_t, a_t)$.

In the following, we first describe the Trust Region Policy Optimization (TRPO) which maximizes the policy performance ($\eta(\pi)$) by optimizing its surrogate objective with on-policy data. Next, we state the Proximal Policy Optimization (PPO) which proposes a clipped surrogate objective to avoid the excessively large policy update in TRPO.

Trust Region Policy Optimization (TRPO). To maximize the performance objective in Eq. (4), the TRPO method (Schulman et al. 2015) uses on-policy data to optimize policies by maximizing a surrogate objective function subject to a constraint on a *Kullback-Leibler* (KL) divergence:

$$\max_\pi \mathbb{E}_{s \sim \rho_{\pi_{\text{old}}}, a \sim \pi_{\text{old}}} \big[\frac{\pi(a|s)}{\pi_{\text{old}}(a|s)} A_{\pi_{\text{old}}}(s, a) \big] \qquad (5)$$

$$\text{subject to } \mathbb{E}_{s \sim \rho_{\pi_{\text{old}}}} \big[D_{KL}\big(\pi_{\text{old}}(\cdot|s) || \pi(\cdot|s)\big) \big] \leq \delta, \qquad (6)$$

where π_{old} is the current policy, δ denotes the bound, $D_{KL}\big(\pi_{\text{old}}(\cdot|s)||\pi(\cdot|s)\big)$ represents the KL deivergence between $\pi_{\text{old}}(\cdot|s)$ and $\pi(\cdot|s)$, $\rho_{\pi_{\text{old}}}$ denotes the discounted state distribution starting at initial state s_0 and following π_{old}: $\rho_{\pi_{\text{old}}}(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s|s_0, \pi_{\text{old}})$ (Sutton et al. 2000). However, without a constraint, the optimization of the surrogate objective function in Eq. (5) would lead to excessively large policy updates.

Proximal Policy Optimization (PPO). In order to avoid such a large policy update, Proximal Policy Optimization (PPO) (Schulman et al. 2017) puts forward a clipped surrogate objective and optimizes policies by maximizing this clipped surrogate objective. The clipped surrogate objective proposed in PPO can be expressed as:

$$\begin{aligned} L_{\text{PPO}}^{\text{CLIP}} = \mathbb{E}_{s \sim \rho_{\pi_{\text{old}}}, a \sim \pi_{\text{old}}} \big[\min \big(& \frac{\pi(a|s)}{\pi_{\text{old}}(a|s)} A_{\pi_{\text{old}}}(s, a), \\ & \text{clip}\big(\frac{\pi(a|s)}{\pi_{\text{old}}(a|s)}, 1 - \epsilon, 1 + \epsilon\big) A_{\pi_{\text{old}}}(s, a) \big) \big], \end{aligned} \qquad (7)$$

where ϵ is a hyperparameter. Note that clipped surrogate objective in Eq. (7) can help PPO avoid large policy updates by penalizing changes to the policy that moves $\frac{\pi(a|s)}{\pi_{\text{old}}(a|s)}$ away from 1 (Schulman et al. 2017). However, PPO faces the issue of high sample complexity due to the lack of utilization of off-policy data, which leads to great demand for on-policy interaction between agent and environment.

Off-Policy Proximal Policy Optimization

To tackle the sample inefficiency problem in the PPO method, we propose an Off-Policy Proximal Policy Optimization method (Off-Policy PPO) that employs off-policy data for policy optimization, as outlined in this section. Specifically, we first introduce the clipped surrogate objective using off-policy data in the proposed Off-Policy PPO. We next clarify the stability of the proposed Off-Policy PPO by clarifying our method makes an update close to the older policy, and the degree of this update distance is the same as that in PPO. Finally, we describe the implementation details of the proposed Off-Policy PPO.

Clipped Surrogate Objective Using Off-Policy Data

In this section, we describe the proposed clipped surrogate objective that utilizes off-policy data in Off-Policy PPO. To do this, we first present the optimization problem that maximizes the surrogate objective using off-policy data in Off-Policy TRPO (Meng et al. 2021). Using this surrogate objective, we then explain how we derive a clipped surrogate objective that effectively uses off-policy data while avoiding large policy updates.

Specifically, the optimization problem which can use off-policy data in the Off-Policy TRPO (Meng et al. 2021) is:

$$\max_\pi \mathbb{E}_{s \sim \rho_\mu, a \sim \mu} \big[\frac{\pi(a|s)}{\mu(a|s)} A_{\pi_{\text{old}}}(s, a) \big] \qquad (8)$$

$$s.t. \quad \overline{D}_{KL}^{\rho_\mu, \text{sqrt}}(\mu, \pi_{\text{old}}) \overline{D}_{KL}^{\rho_\mu, \text{sqrt}}(\pi_{\text{old}}, \pi) + \overline{D}_{KL}^{\rho_\mu}(\pi_{\text{old}}, \pi) \leq \delta, \qquad (9)$$

where μ represents the behavior policy and $\rho_\mu(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s|s_0, \mu)$, $\overline{D}_{KL}^{\rho_\mu}(\pi_{\text{old}}, \pi) := \mathbb{E}_{s \sim \rho_\mu}[D_{KL}(\pi_{\text{old}}(\cdot|s) \parallel \pi(\cdot|s))]$, $\overline{D}_{KL}^{\rho_\mu, \text{sqrt}}(\mu, \pi_{\text{old}}) := \mathbb{E}_{s \sim \rho_\mu}[\sqrt{D_{KL}(\mu(\cdot|s) \parallel \pi_{\text{old}}(\cdot|s))}]$, $\overline{D}_{KL}^{\rho_\mu, \text{sqrt}}(\pi_{\text{old}}, \pi) := \mathbb{E}_{s \sim \rho_\mu}[\sqrt{D_{KL}(\pi_{\text{old}}(\cdot|s) \parallel \pi(\cdot|s))}]$. However, without the constraint in Eq. (9), the maximization of the above surrogate objective utilizing off-policy data in Eq. (8) suffers from an excessively large policy update.

To address this issue, a straightforward solution is to apply the clipping strategy from PPO to adjust the surrogate objective in Eq. (8). To clarify the clipped objective, we begin by presenting the surrogate objective in Eq. (8):

$$L_\mu(\pi) = \mathbb{E}_{s\sim\rho_\mu, a\sim\mu}\big[\frac{\pi(a|s)}{\mu(a|s)} A_{\pi_{\text{old}}}(s,a)\big]. \tag{10}$$

With $L_\mu(\pi)$ in Eq. (10), the corresponding clipped surrogate objective using off-policy data is:

$$\begin{aligned} \overline{L}_\mu(\pi) = \mathbb{E}_{s\sim\rho_\mu, a\sim\mu}\big[\min\big[\frac{\pi(a|s)}{\mu(a|s)} A_{\pi_{\text{old}}}(s,a), \\ \text{clip}\big(\frac{\pi(a|s)}{\mu(a|s)}, 1-\epsilon, 1+\epsilon\big) A_{\pi_{\text{old}}}(s,a)\big]\big]. \end{aligned} \tag{11}$$

Note that in Eq. (11), the policy ratio $\frac{\pi(a|s)}{\mu(a|s)}$ is generally either less than $1-\epsilon$ or greater than $1+\epsilon$. Consequently, the target policy $\pi(a|s)$ often remains unchanged and does not undergo any updates during the optimization process of the clipped surrogate objective in Eq. (11).

In order to address this issue, we propose a clipped surrogate objective which scales the lower and upper bound ($(1-\epsilon)$, $(1+\epsilon)$) in Eq. (11) by a factor of $\frac{\pi_{\text{old}}(a|s)}{\mu(a|s)}$:

$$\begin{aligned} L^{\text{CLIP}}_{\text{Off-Policy PPO}}(\pi) = \mathbb{E}_{s\sim\rho_\mu, a\sim\mu}\big[\min\big[r_\pi(s,a) A_{\pi_{\text{old}}}(s,a), \\ \text{clip}\big(r_\pi(s,a), l_{s,a}, h_{s,a}\big) A_{\pi_{\text{old}}}(s,a)\big]\big], \end{aligned} \tag{12}$$

where $r_\pi(s,a) = \frac{\pi(a|s)}{\mu(a|s)}$, $l_{s,a} = \frac{\pi_{\text{old}}(a|s)}{\mu(a|s)}(1-\epsilon)$, $h_{s,a} = \frac{\pi_{\text{old}}(a|s)}{\mu(a|s)}(1+\epsilon)$.

Stability Analysis

In this section, we analyze the stability of the proposed Off-Policy PPO by clarifying our method makes an update close to the older policy, and the degree of this update distance is the same as that in PPO. In order to clarify this, we first describe the optimal policy set in Lemma 1, which maximizes the proposed clipped objective in Eq. (12). We then clarify the maximum KL divergence between current policy π_{old} and the optimal policy π_{new} in the proposed Off-Policy PPO is consistent with that in PPO in Theorem 1.

With the surrogate objective in Eq. (12), we denote Π_{new} as the optimal policy set maximizing this objective in Lemma 1. Note that advantage value $A_{\pi_{\text{old}}}(s,a)$ is simplified and denoted as A in Lemma 1.

Lemma 1. Π_{new}=$\{\pi|$ *for all state and action pair* (s,a) *that* $A<0$, $\pi(a|s) \le \mu(a|s) l_{s,a}$*; for all state and action pair* (s,a) *that* $A>0$, $\pi(a|s) \ge \min(\mu(a|s)h_{s,a}, 1)\}$.

Proof. Firstly, we prove that a policy π^* meeting the conditions in Π_{new} is the optimal solution maximizing the objective in Eq. (12). In order to prove this, we need to show that, given any state and action pair (s,a), the policy π^* meeting the condition in Π_{new} satisfies: $L^{s,a}_\mu(\pi^*) \ge L^{s,a}_\mu(\pi)$ for any π. Note that $L^{s,a}_\mu(\pi)$ denotes the surrogate objective given any (s,a) under the policy π: $L^{s,a}_\mu(\pi) = \min\big[\ r_\pi(s,a)A, \text{clip}\big(r_\pi(s,a), l_{s,a}, h_{s,a}\big)A\big]$.

If $A<0$, $L^{s,a}_\mu(\pi)$ could be written as:

$$L^{s,a}_\mu(\pi) = \begin{cases} l_{s,a}A, & r_\pi(s,a) \le l_{s,a} \\ r_\pi(s,a)A, & r_\pi(s,a) > l_{s,a}. \end{cases} \tag{13}$$

$L^{s,a}_\mu(\pi^*)$ can be written as $L^{s,a}_\mu(\pi^*) = \min\big[\ r_{\pi^*}(s,a)A, \text{clip}\big(r_{\pi^*}(s,a), l_{s,a}, h_{s,a}\big)A\big] = l_{s,a}A$, where π^* meeting the condition in Π_{new} satisfies $\pi^*(a|s) \le \mu(a|s)l_{s,a}$ when $A<0$.

Thus, if $A<0$, $L^{s,a}_\mu(\pi) \le l_{s,a}A = L^{s,a}_\mu(\pi^*)$ for any π.

If $A>0$, $L^{s,a}_\mu(\pi)$ could be written as:

$$L^{s,a}_\mu(\pi) = \begin{cases} h_{s,a}A, & r_\pi(s,a) \ge h_{s,a} \\ r_\pi(s,a)A, & r_\pi(s,a) < h_{s,a}. \end{cases} \tag{14}$$

$L^{s,a}_\mu(\pi^*)$ can be written as $L^{s,a}_\mu(\pi^*) = \min\big[\ r_{\pi^*}(s,a)A, \text{clip}\big(r_{\pi^*}(s,a), l_{s,a}, h_{s,a}\big)A\big] = h_{s,a}A$, where π^* satisfies $\pi^*(a|s) \ge \mu(a|s)h_{s,a}$ when $A>0$, which is due to that π^* meeting the conditon in Π_{new} satisfies $\pi^*(a|s) \ge \min(\mu(a|s)h_{s,a}, 1)$ and $\pi^*(a|s) < 1$.

Thus, if $A>0$, $L^{s,a}_\mu(\pi) \le h_{s,a}A = L^{s,a}_\mu(\pi^*)$ for any π. Based on such fact, we have proven that a policy π^* meeting the conditions in Π_{new} is the optimal solution.

Secondly, we prove that a policy π_0 not meeting conditions in Π_{new} is not the optimal solution of maximizing the objective in Eq. (12). In order to prove this, we construct a policy π^* satisfying conditions in Π_{new}. Then, $L^{s,a}_\mu(\pi_0) \le L^{s,a}_\mu(\pi^*)$ for any state and action pair (s,a). Based on such fact, we have proven that a policy not meeting the conditions in Π_{new} is not the optimal solution of maximizing the objective in Eq. (12).

Finally, combining the above results, we prove that Π_{new} described in Lemma 1 contains all the optimal solutions of maximizing the surrogate objective in Eq. (12). □

Based on the optimal policy set Π_{new} in Lemma 1, we clarify the degree of the policy update distance in Off-Policy PPO is the same as that in PPO. Specifically, we demonstrate that the maximum KL divergence between the new policy and the old one in our method is equivalent to that in PPO, as illustrated in Theorem 1.

Theorem 1. *Let* $\pi^{\textit{Off-Policy PPO}}_{new} \in \Pi_{new}$ *denote the optimal pilicy in Off-Policy PPO, which achieves the minimum KL divergence over all optimal policies, i.e.,* $D_{KL}(\pi_{old}(\cdot|s_t), \pi^{\textit{Off-Policy PPO}}_{new}(\cdot|s_t)) \le D_{KL}(\pi_{old}(\cdot|s_t), \pi(\cdot|s_t))$ *for* $\pi \in \Pi_{new}$ *at any timestep* t *, and let* π^{PPO}_{new} *have the similar definition for PPO, we have* $\max_t D_{KL}(\pi_{old}(\cdot|s_t), \pi^{\textit{Off-Policy PPO}}_{new}(\cdot|s_t)) = \max_t D_{KL}(\pi_{old}(\cdot|s_t), \pi^{PPO}_{new}(\cdot|s_t))$ *for all timestep* t.

Proof. In order to simplify the expression in the proof, we separately denote $D_{\text{KL}}(\pi_{\text{old}}(\cdot|s_t), \pi^{\text{Off-Policy PPO}}_{\text{new}}(\cdot|s_t))$, $D_{\text{KL}}(\pi_{\text{old}}(\cdot|s_t), \pi^{\text{PPO}}_{\text{new}}(\cdot|s_t))$ as $D^{s_t}_{\text{KL}}(\pi_{\text{old}}, \pi^{\text{Off-Policy PPO}}_{\text{new}})$, $D^{s_t}_{\text{KL}}(\pi_{\text{old}}, \pi^{\text{PPO}}_{\text{new}})$ in the following.

In the proof, we need to prove that $D^{s_t}_{\text{KL}}(\pi_{\text{old}}, \pi^{\text{Off-Policy PPO}}_{\text{new}}) = D^{s_t}_{\text{KL}}(\pi_{\text{old}}, \pi^{\text{PPO}}_{\text{new}})$ for any timestep t. Specifically, we prove this in two cases, *i.e.*, $A_{\pi_{\text{old}}}(s_t, a_t) \le 0$ and $A_{\pi_{\text{old}}}(s_t, a_t) > 0$ for any timestep t. We denote $A_{\pi_{\text{old}}}(s_t, a_t)$ as A_t for simplicity in the following.

In the case $A_t \leq 0$, we prove that $D_{\text{KL}}^{s_t}(\pi_{\text{old}}, \pi_{\text{new}}^{\text{Off-Policy PPO}}) = D_{\text{KL}}^{s_t}(\pi_{\text{old}}, \pi_{\text{new}}^{\text{PPO}})$. If $A_t \leq 0$, the optimal policy $\pi_{\text{new}}^{\text{Off-Policy PPO}}$ can be derived by solving the following constraint optimization problem according to Lemma 1:

$$\begin{aligned} \min_{\pi} & \sum_a \pi_{\text{old}}(a|s_t)\log\frac{\pi_{\text{old}}(a|s_t)}{\pi(a|s_t)} \\ s.t. \quad & \pi(a_t|s_t) \leq \mu(a_t|s_t) l_{s_t,a_t}, \\ & \sum_a \pi(a|s_t) = 1, \\ & \pi(a|s_t) > 0, \end{aligned} \tag{15}$$

where a_t denotes the action at timestep t.

By using the Karush-Kuhn-Tucker (KKT) conditions (Gordon and Tibshirani 2012), we get:

$$\pi_{\text{new}}^{\text{Off-Policy PPO}}(a|s_t) = \begin{cases} \dfrac{\pi_{\text{old}}(a|s_t)(1-\mu(a_t|s_t)l_{s_t,a_t})}{1-\pi_{\text{old}}(a_t|s_t)}, & a \neq a_t \\ \mu(a_t|s_t)l_{s_t,a_t}, & a = a_t. \end{cases} \tag{16}$$

The corresponding KL divergence is

$$\begin{aligned} & D_{\text{KL}}^{s_t}(\pi_{\text{old}}, \pi_{\text{new}}^{\text{Off-Policy PPO}}) \\ & = (1-\pi_{\text{old}}(a_t|s_t))\log\frac{1-\pi_{\text{old}}(a_t|s_t)}{1-\pi_{\text{old}}(a_t|s_t)(1-\epsilon)} \\ & \qquad - \pi_{\text{old}}(a_t|s_t)\log(1-\epsilon), \end{aligned} \tag{17}$$

which equals $D_{\text{KL}}^{s_t}(\pi_{\text{old}}, \pi_{\text{new}}^{\text{PPO}})$ described in Eq. (26) of appendix in (Wang et al. 2019) due to the lower bound in PPO is $1-\epsilon$.

In the case $A_t > 0$, we prove that $D_{\text{KL}}^{s_t}(\pi_{\text{old}}, \pi_{\text{new}}^{\text{Off-Policy PPO}}) = D_{\text{KL}}^{s_t}(\pi_{\text{old}}, \pi_{\text{new}}^{\text{PPO}})$. If $A_t > 0$, according to Lemma 1, the constraint optimization problem is:

$$\begin{aligned} \min_{\pi} & \sum_a \pi_{\text{old}}(a|s_t)\log\frac{\pi_{\text{old}}(a|s_t)}{\pi(a|s_t)} \\ s.t. \quad & \pi(a_t|s_t) \geq \min(\mu(a_t|s_t)h_{s_t,a_t}, 1), \\ & \sum_a \pi(a|s_t) = 1, \\ & \pi(a|s_t) > 0. \end{aligned} \tag{18}$$

By using the KKT conditions, we get:

$$\begin{aligned} & \pi_{\text{new}}^{\text{Off-Policy PPO}}(a|s_t) \\ & = \begin{cases} \dfrac{\pi_{\text{old}}(a|s_t)(1-\min(\mu(a_t|s_t)h_{s_t,a_t},1))}{1-\pi_{\text{old}}(a_t|s_t)}, & a \neq a_t \\ \min(\mu(a_t|s_t)h_{s_t,a_t},1), & a = a_t. \end{cases} \end{aligned} \tag{19}$$

When $A_t > 0$ and $\mu(a_t|s_t)h_{s_t,a_t} \leq 1$, the KL divergence is

$$\begin{aligned} & D_{\text{KL}}^{s_t}(\pi_{\text{old}}, \pi_{\text{new}}^{\text{Off-Policy PPO}}) \\ & = (1-\pi_{\text{old}}(a_t|s_t))\log\frac{1-\pi_{\text{old}}(a_t|s_t)}{1-\pi_{\text{old}}(a_t|s_t)(1+\epsilon)} \\ & \qquad - \pi_{\text{old}}(a_t|s_t)\log(1+\epsilon), \end{aligned} \tag{20}$$

which equals $D_{\text{KL}}^{s_t}(\pi_{\text{old}}, \pi_{\text{new}}^{\text{PPO}})$ described in Eq. (28) of appendix in (Wang et al. 2019) due to the upper bound in PPO is $1+\epsilon$. When $A_t > 0$ and $\mu(a_t|s_t)h_{s_t,a_t} > 1$, $D_{\text{KL}}^{s_t}(\pi_{\text{old}}, \pi_{\text{new}}^{\text{Off-Policy PPO}}) = +\infty = D_{\text{KL}}^{s_t}(\pi_{\text{old}}, \pi_{\text{new}}^{\text{PPO}})$ (Wang et al. 2019).

Algorithm 1: Off-Policy PPO

```
Require: Environment $E$, trace-decay parameter $\lambda$, discount factor $\gamma$, learning rate $\alpha$, trajectory length $K$, replay memory $R$, epoch number $N$, minibatch size $M$.
  Initialize policy network $\pi_\theta$ and state value network $V_w$.
  repeat
    // Collect data
    Collect $K$ transitions during interaction with environment $\mathcal{S}_{0:K} = \{s_0, a_0, r_0, \mu(\cdot|s_0), \cdots, s_K, a_K, r_K, \mu(\cdot|s_K)\}$ according to behavior policy $\mu$.
    Add the collected data into replay memory $R$.

    // Update networks
    Sample off-policy data $T_{0:K}$ from replay memory $R$.
    Obtain $\{V_w(s_j)\}_{j=0}^{K}$ by $V_w$.
    Obtain V-trace target $v_j = V_w(s_j) + \delta_j V + \gamma c_j(v_{j+1} - V_w(s_{j+1}))$.
    Obtain advantage value $\{A(s_j, a_j)\}_{j=0}^{K-1} = \{r_j + \gamma v_{j+1} - V_w(s_j)\}_{j=0}^{K-1}$.
    for epoch=$1, 2, \cdots, N$ do
      Acquire data $D_{0:K}$ by randomly shuffling $T_{0:K}$.
      for $i = 1, 2, \cdots, K/M$ do
        Obtain $M$ minibatch from shuffled data $D_{i*M-M:i*M-1}$.
        Update $\pi_\theta$ by maximizing objective in Eq. (12): $\theta \leftarrow \theta_{\text{old}} + \alpha\nabla_\theta L_{\text{Off-Policy PPO}}^{\text{CLIP}}(\pi_\theta)$.
        Update $V_w$ by minimizing the mean squared error between V-trace target $v$ and state value $V_w(s)$.
      end for
    end for
  until policy $\pi_\theta$ converges
```

Combining above results on two cases ($A_t \leq 0$ and $A_t > 0$), we have proven $D_{\text{KL}}^{s_t}(\pi_{\text{old}}, \pi_{\text{new}}^{\text{Off-Policy PPO}}) = D_{\text{KL}}^{s_t}(\pi_{\text{old}}, \pi_{\text{new}}^{\text{PPO}})$ for any timestep t. Based on such fact, we can conclude that $\max_t D_{\text{KL}}^{s_t}(\pi_{\text{old}}, \pi_{\text{new}}^{\text{Off-Policy PPO}}) = \max_t D_{\text{KL}}^{s_t}(\pi_{\text{old}}, \pi_{\text{new}}^{\text{PPO}})$ for all timestep t.

As far, we prove that $\max_t D_{\text{KL}}(\pi_{\text{old}}(\cdot|s_t), \pi_{\text{new}}^{\text{Off-Policy PPO}}(\cdot|s_t)) = \max_t D_{\text{KL}}(\pi_{\text{old}}(\cdot|s_t), \pi_{\text{new}}^{\text{PPO}}(\cdot|s_t))$ in Theorem 1. □

Implementation Details

In this section, we introduce the implementation details of the proposed Off-Policy PPO, which iteratively optimizes policies by maximizing the proposed clipped surrogate objective in Eq. (12). Specifically, the whole procedure of the proposed Off-Policy PPO is described in Algorithm 1.

In Algorithm 1, we first initialize policy network π_θ and state value network V_w, respectively. Us-

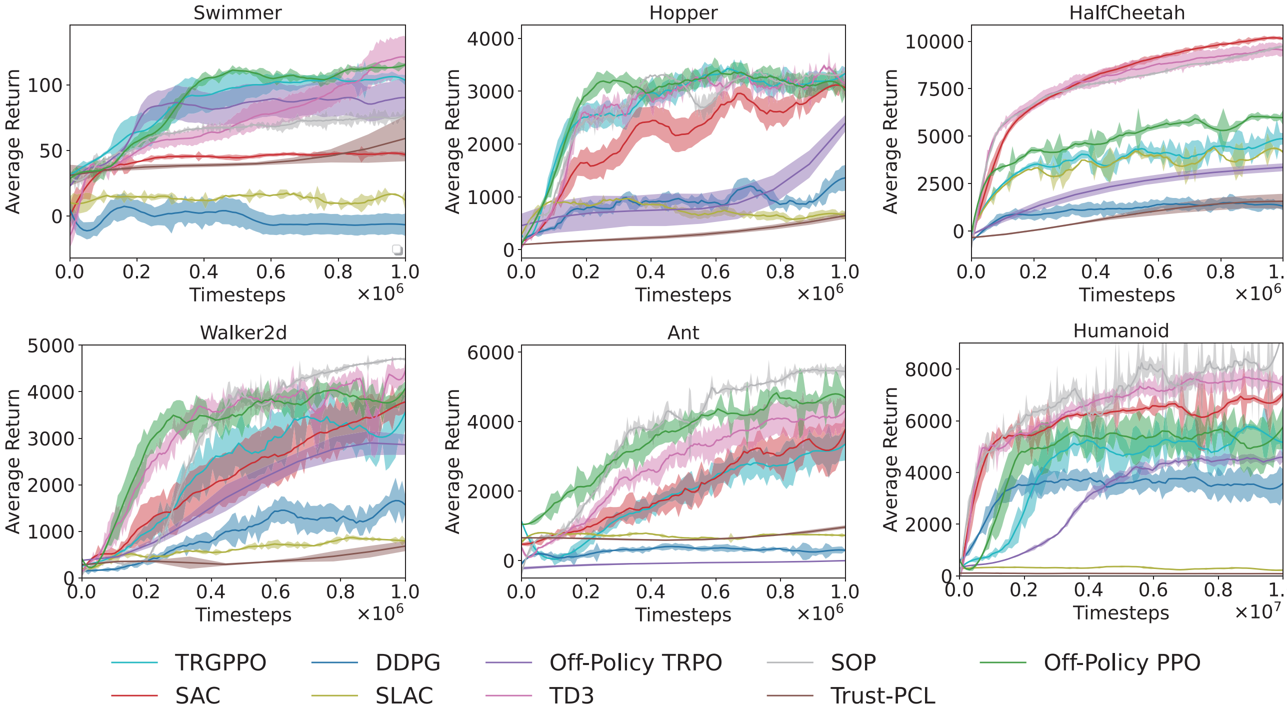

Figure 1: Training curve comparison between the proposed Off-Policy PPO and other state-of-the-art methods.

ing these networks, we then collect K transitions ($\mathcal{S}_{0:K} = \{s_0, a_0, r_0, \mu(\cdot|s_0), \cdots, s_K, a_K, r_K, \mu(\cdot|s_K)\}$) and add these data into memory. Next, we sample off-policy data ($T_{0:K}$) from memory R and use these data to update policy and state value networks.

During the network update procedure, we first use data $T_{0:K}$ to estimate state values $\{V_w(s_j)\}_{j=0}^{K}$ and V-trace target (Espeholt et al. 2018) $v_j = V_w(s_j) + \delta_j V + \gamma c_j(v_{j+1} - V_w(s_{j+1}))$ where $\delta_j V = \rho_j(r_j + \gamma V_w(s_{j+1}) - V_w(s_j))$, $\rho_j = \min(1, \frac{\pi_{\theta_{\text{old}}}(a_j|s_j)}{\mu(a_j|s_j)})$, and $c_j = \min(1, \frac{\pi_{\theta_{\text{old}}}(a_j|s_j)}{\mu(a_j|s_j)})$. With these values, we next obtain advantage values $\{A(s_j, a_j)\}_{j=0}^{K-1} = \{r_j + \gamma v_{j+1} - V_w(s_j)\}_{j=0}^{K-1}$. Finally, we optimize policy network π_θ and state value network V_w with N epochs. In every epoch, we shuffle the off-policy data $T_{0:K}$ and obtain the minibatch from the shuffled data $D_{0:K}$. With such minibatch, we optimize π_θ by maximizing objective in Eq. (12): $\theta \leftarrow \theta_{\text{old}} + \alpha \nabla_\theta L^{\text{CLIP}}_{\text{Off-Policy PPO}}(\pi_\theta)$ and optimize V_w by minimizing the mean squared error between V-trace target v and $V_w(s)$.

Experiments

In this section, we perform experiments to evaluate the proposed Off-Policy Proximal Policy Optimization (Off-Policy PPO) on a variety of representative continuous control tasks. We first introduce the experimental setup which comprises networks, hyperparameters, and experimental tasks. We next compare our method and the state-of-the-art methods, *i.e.*, TRGPPO (Wang et al. 2019), Soft Actor-Critic (SAC) (Haarnoja et al. 2018), DDPG (Lillicrap et al. 2016), SLAC (Lee et al. 2020), Off-Policy TRPO (Meng et al. 2021), TD3 (Fujimoto, Hoof, and Meger 2018), SOP (Wang et al. 2020), and Trust-PCL (Nachum et al. 2018), to validate that our method can achieve better or comparable performance than these methods. We then study the effectiveness of our method on using off-policy data by comparing our method with PPO. Next, we study on KL divergence curves between our method, PPO, and SAC to evaluate the stability of our method in practice. Finally, we study the effectiveness of our method on sample efficiency by comparing our method to other methods from the aspect of timesteps to reach a threshold on the continuous control tasks.

Setup

In the experiments, we adopt a policy network and state value network to separately approximate the Gaussian policy distribution and the state value. These networks are multi-layer neural networks comprising two hidden layers with 64 neurons. We use the Tanh as the activation function of these networks. For hyperparameters, the trace-decay parameter λ is 0.95 and the discount factor γ is 0.99. The length of transitions (K) is set to be 1024. We use the Adam optimizer with learning rate $\alpha = 3 \times 10^{-4}$. The epoch number N is 10. The minibatch size M is set to be 32. Experimental tasks consist of six representative continuous control tasks from OpenAI Gym (Brockman et al. 2016) and MuJoCo (Todorov, Erez, and Tassa 2012), which cover

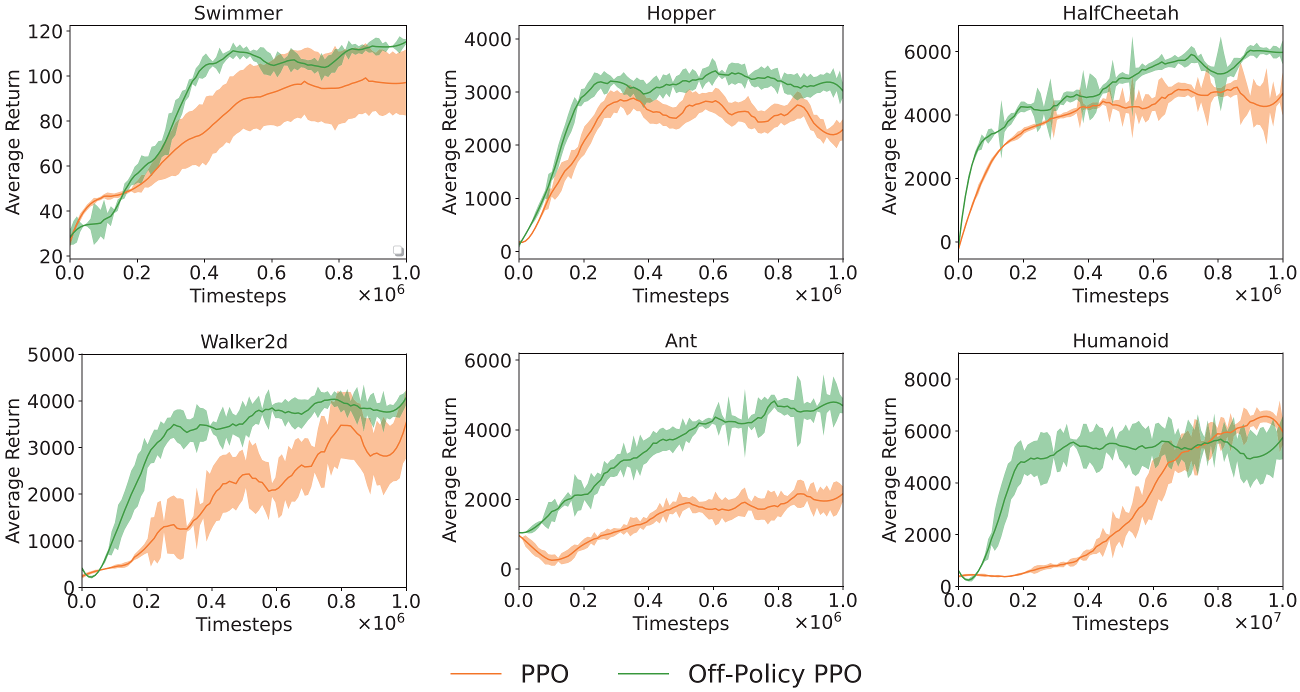

Figure 2: Training curve comparison between the proposed Off-Policy PPO and PPO during training.

simple and complex tasks: Swimmer, Hopper, HalfCheetah, Walker2d, Ant, and Humanoid. We adopt a commonly used version, i.e., v2, for these tasks. The experiments are performed on a GPU server that has four Nvidia RTX 3090. The results reported in the experiments are averaged over the top three seeds of ten random seeds. For the implementations of these compared methods: we use the original implementation codes given by the authors of most state-of-the-art methods (TRGPPO, SAC, SLAC, Off-Policy TRPO, TD3, SOP, and Trust-PCL) and we use the implementation in https://github.com/chainer/chainerrl for DDPG.

Comparison with State-of-the-art Methods

In this section, we compare our method with state-of-the-art methods, *i.e.*, TRGPPO, SAC, DDPG, SLAC, Off-Policy TRPO, TD3, SOP, and Trust-PCL, to validate that our method can outperform these methods on most tasks.

The training curve comparison between our method and state-of-the-art methods is shown in Figure 1. We observe that, when compared to other methods, our method requires fewer timesteps to achieve the same return on the majority of tasks, *i.e.*, Swimmer, Hopper, Walker2d, and Ant. On HalfCheetah and Humanoid, the proposed Off-Policy PPO needs fewer timesteps to achieve the same return than most methods, *i.e.*, TRGPPO, DDPG, SLAC, Off-Policy TRPO, and Trust-PCL. It can be observed that the final return achieved by our method is higher or comparable when compared to other methods on these tasks. Notice that our method obtains the highest returns among TRGPPO, DDPG, SLAC, Off-Policy TRPO, and Trust-PCL on HalfCheetah and Humanoid. Results in Figure 1 illustrate that our method can surpass these state-of-the-art methods on most tasks.

Study on Using Off-Policy Data

In this section, we study the effectiveness of our method on using off-policy data by comparing our method using off-policy data with PPO only using on-policy data.

The results are shown in Figure 2. Note that our method needs fewer timesteps to achieve the same return when compared to PPO on most tasks and this phenomenon is particularly evident on complex tasks (Walker2d, Ant, and Humanoid), which is due to that our method can use off-policy data to optimize policies. We can also note that our method using off-policy data achieves higher final returns than PPO only using on-policy data on most tasks. The experimental results from Figure 2 validate the effectiveness of our method on using off-policy data.

Study on KL Divergence

In this section, we study on KL divergence by comparing our method, PPO, and SAC to evaluate the stability of our method in practice. SAC is chosen to be compared due to this method is the state-of-the-art off-policy method.

The KL divergence comparison is shown in Figure 3. The KL divergence reported in Figure 3 denotes KL divergence between π_{old} and π_{new} in each policy update in practice. Note that we choose representative tasks, *i.e.*, Hopper, HalfCheetah, and Walker2d, due to their popularity (Todorov, Erez, and Tassa 2012). From Figure 3, we can observe that the KL divergence of SAC is larger than these of PPO and Off-Policy PPO. It is noticeable that the KL divergence of SAC on Hopper and Walker2d has an increasing trend and that of

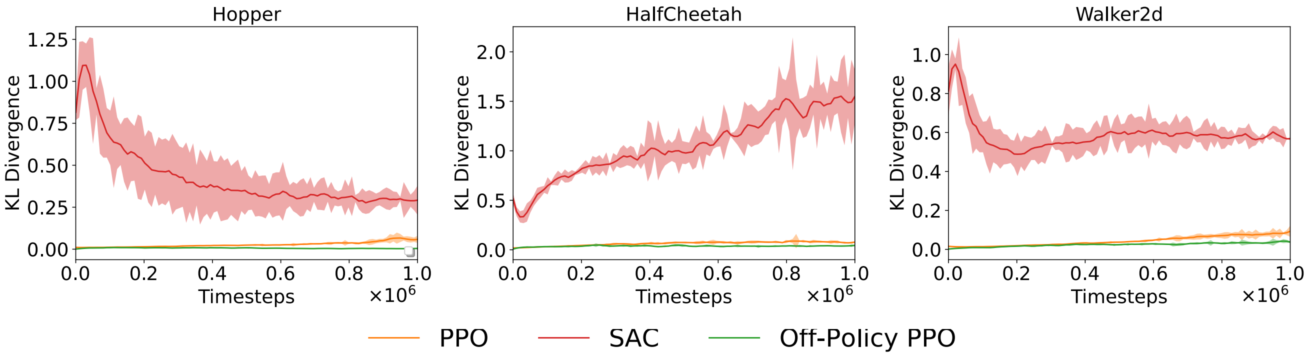

Figure 3: KL divergence comparison among the proposed Off-Policy PPO, PPO, and SAC during training.

	Timesteps to reach a threshold ($\times 10^3$)								
Tasks	PPO	TRGPPO	SAC	DDPG	SLAC	Off-TRPO	TD3	SOP	Trust-PCL	Our Method
Swimmer	520	340	/	/	/	810	740	/	/	**340**
Hopper	/	440	920	/	/	/	380	355	/	**210**
HalfCheetah	140	140	70	/	160	/	**45**	45	/	70
Walker2d	750	590	700	/	/	/	285	365	/	**220**
Ant	/	860	680	/	/	/	455	**285**	/	310
Humanoid	6800	3400	1000	/	/	/	**950**	1200	/	2800

Table 1: Comparison of timesteps to reach a threshold within one million timesteps (except Humanoid with ten million) during training. The thresholds for these tasks (Swimmer, Hopper, HalfCheetah, Walker2d, Ant, and Humanoid) separately are 90, 3000, 3000, 3000, 3000, and 5000. We denote Off-Policy TRPO as Off-TRPO for short. For each task, the minimum result is indicated in boldface. / indicates that the method did not reach a threshold within fixed timesteps.

SAC on HalfCheetah has a decreasing trend, which is most likely due to that large policy updates of SAC on Hopper and Walker2d mainly exist in the early training stage and these of SAC on HalfCheetah mainly exist in the late training stage. From Figure 3, it can be observed that the proposed Off-Policy PPO has nearly the same KL divergence as PPO during the whole training process. The similar KL divergence curves between our method and PPO in Figure 3 demonstrate that the proposed Off-Policy PPO does not harm the stability in practice.

Study on Sample Efficiency

In this section, we study the effectiveness of our method on sample efficiency by comparing our method and other methods in terms of timesteps to reach a threshold over training.

The comparison is represented in Table 1. We separately set the threshold values of Swimmer, Hopper, HalfCheetah, Walker2d, Ant, and Humanoid as $90, 3000, 3000, 3000, 3000, 5000$, which refer to the threshold values in (Wang et al. 2019). As shown in Table 1, when compared with other methods, our method requires fewer or comparable timesteps to reach these thresholds on most tasks. Note that Off-Policy TRPO did not reach thresholds within the fixed timesteps on most tasks, which is probably because the update interval value in this method is relatively large. DDPG and Trust-PCL did not reach thresholds within the fixed timesteps on most tasks, which is probably due to that these two methods may need more timesteps during training to achieve better performance. The results from Table 1 validate that our method achieves higher sample efficiency than other methods on most tasks.

Conclusion and Future Work

In this paper, we propose an Off-Policy Proximal Policy Optimization (Off-Policy PPO) method, which improves the sample efficiency of PPO by utilizing off-policy data. We provide a novel idea for improving the sample efficiency of PPO. Specifically, we propose a clipped surrogate objective using off-policy data, which is based on the surrogate objective in Off-Policy TRPO. Then, we theoretically clarify the stability of optimizing the proposed clipped surrogate objective using off-policy data. Next, we describe the algorithm of the proposed Off-Policy PPO which iteratively optimizes policies by maximizing the proposed clipped surrogate objective in detail. Finally, experimental results on a variety of domains validate that our method outperforms state-of-the-art methods on most tasks. These experimental results also evaluate our method in terms of using off-policy data and sample efficiency.

Although this algorithm is appealing to sequential-decision-making problems having difficulty in collecting data, it is interesting to improve the performance of this algorithm in rare scenarios where the quality of off-policy data is poor, *e.g.*, $\pi_{\text{old}}(\cdot|s) \ll \mu(\cdot|s)$, in the future work.

Acknowledgments

This work is supported by STI 2030 Major Projects (2021ZD0200400), Natural Science Foundation of China (No. 62206158, No. 61925603), Natural Science Foundation of Shandong Province (No. ZR2022QF097), Major Basic Research Project of Natural Science Foundation of Shandong Province (No. ZR2021ZD15), The Fundamental Research Funds of Shandong University. We are very grateful to Guoqiang Wu for his great help with this paper.

References

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

Degris, T.; White, M.; and Sutton, R. S. 2012. Off-policy actor-critic. In *International Conference on Machine Learning*, 179–186.

Espeholt, L.; Soyer, H.; Munos, R.; Simonyan, K.; Mnih, V.; Ward, T.; Doron, Y.; Firoiu, V.; Harley, T.; Dunning, I.; et al. 2018. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, 1407–1416.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 1587–1596.

Gordon, G.; and Tibshirani, R. 2012. Karush-kuhn-tucker conditions. *Optimization*, 10(725/36): 725.

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

Kober, J.; Bagnell, J. A.; and Peters, J. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11): 1238–1274.

Lee, A. X.; Nagabandi, A.; Abbeel, P.; and Levine, S. 2020. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. In *Advances in Neural Information Processing Systems*, 741–752.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2016. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*.

Meng, W.; Zheng, Q.; Shi, Y.; and Pan, G. 2021. An off-policy trust region policy optimization method with monotonic improvement guarantee for deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(5): 2223–2235.

Meng, W.; Zheng, Q.; Yang, L.; Li, P.; and Pan, G. 2019. Qualitative measurements of policy discrepancy for return-based deep Q-network. *IEEE Transactions on Neural Networks and Learning Systems*, 31(10): 4374–4380.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, 1928–1937.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Nachum, O.; Norouzi, M.; Xu, K.; and Schuurmans, D. 2018. Trust-PCL: An Off-Policy Trust Region Method for Continuous Control. In *International Conference on Learning Representations*.

Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *International Conference on Machine Learning*, 1889–1897.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M. I.; and Abbeel, P. 2016. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *International Conference on Learning Representations*.

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

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. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587): 484–489.

Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, 387–395.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. *Nature*, 550(7676): 354–359.

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

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 1057–1063.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 5026–5033.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.

Wang, C.; Wu, Y.; Vuong, Q.; and Ross, K. 2020. Striving for simplicity and performance in off-policy DRL: Output normalization and non-uniform sampling. In *International Conference on Machine Learning*, 10070–10080.

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

Wang, Y.; He, H.; Tan, X.; and Gan, Y. 2019. Trust Region-Guided Proximal Policy Optimization. In *Advances in Neural Information Processing Systems*, 626–636.

Xing, D.; Liu, Q.; Zheng, Q.; and Pan, G. 2021. Learning with Generated Teammates to Achieve Type-Free Ad-Hoc Teamwork. In *International Joint Conference on Artificial Intelligence*, 472–478.

Yang, L.; Ji, J.; Dai, J.; Zhang, L.; Zhou, B.; Li, P.; Yang, Y.; and Pan, G. 2022a. Constrained Update Projection Approach to Safe Policy Optimization. In *Advances in Neural Information Processing Systems*.

Yang, L.; Zhang, Y.; Zheng, G.; Zheng, Q.; Li, P.; Huang, J.; and Pan, G. 2022b. Policy optimization with stochastic mirror descent. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 8823–8831.

Zou, L.; Zhuang, Z.; Cheng, Y.; Wang, X.; and Zhang, W. 2019. Separated trust regions policy optimization method. In *International Conference on Knowledge Discovery & Data Mining*, 1471–1479.