The Sufficiency of Off-Policyness and Soft Clipping: PPO Is Still Insufficient according to an Off-Policy Measure

Xing Chen[1,4], **Dongcui Diao**[6], **Hechang Chen**[1,4,*], **Hengshuai Yao**[2,*], **Haiyin Piao**[3], **Zhixiao Sun** [3], **Zhiwei Yang**[1,4] , **Randy Goebel**[2,5], **Bei Jiang**[5,6], **Yi Chang**[1,4,*]

[1] School of Artificial Intelligence, Jilin University, Changchun, China
[2] Department of Computing Science, University of Alberta, Edmonton, Canada
[3] School of Electronics and Information, Northwestern Polytechnical University, Xian, China
[4] Engineering Research Center of Knowledge-Driven Human-Machine Intelligence, Ministry of Education, China
[5] Alberta Machine Intelligence Institute, University of Alberta, Edmonton, Canada
[6] Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, Canada
xingchen19@mails.jlu.edu.cn, {chenhc,yichang}@jlu.edu.cn, hengshu1@ualberta.ca

Abstract

The popular Proximal Policy Optimization (PPO) algorithm approximates the solution in a clipped policy space. Does there exist better policies outside of this space? By using a novel surrogate objective that employs the sigmoid function (which provides an interesting way of exploration), we found that the answer is "YES", and the better policies are in fact located very far from the clipped space. We show that PPO is insufficient in "off-policyness", according to an off-policy metric called DEON. Our algorithm explores in a much larger policy space than PPO, and it maximizes the Conservative Policy Iteration (CPI) objective better than PPO during training. To the best of our knowledge, all current PPO methods have the clipping operation and optimize in the clipped policy space. Our method is the first of this kind, which advances the understanding of CPI optimization and policy gradient methods. Code is available at https://github.com/raincchio/P3O.

Introduction

Real-world problems like medication dosing and autonomous driving pose a great challenge for Artificial Intelligence (AI) with an expectation to improve human life and safety. Applications like these require significant interaction with the environment to make learning algorithms effective. Humans can learn from others, by observing their experience to quickly pick up new skills, even without exposure on one's own. Subsequently, when there is a chance to practice, the skills obtained from previous experience or others can be quickly adapted and improved. But it is clear that we are still far from obtaining this remarkable learning ability in AI.

In our context, we consider problems where the environment state cannot be reset, which is also true in real life. In such situations, we can only sample a limited number of possible trajectories, which easily results in failure of learning.

Off-policy learning is one promising paradigm to address this challenge, and it provides an effective discipline of learning by sampling the potential trajectories starting from any state. This means we can evaluate a target policy using a behavior policy that generates experience (Precup, Sutton, and Dasgupta 2001; Sutton, Maei, and Szepesvári 2009). Moreover, the paradigm seems characteristic and general enough for obtaining skills from other sources. With off-policy learning, one agent can reuse experience from itself or even the other agents, where the samples are collected with methods that are different from real-time on-policy interaction. Off-policy learning holds significant promise, but it is tricky in practice. The mismatch between the distribution of the behavior policy and that of the target policy poses a big stability challenge for the learning process. Even for policy evaluation, following the temporal difference update easily diverges in the linear function approximation (Bertsekas 1995; Boyan and Moore 1994; Gordon 1995; Tsitsiklis and Van Roy 1997).

Precup, Sutton, and Singh (2000) et. al. were the first to use importance sampling for off-policy learning. They used an online updated product of importance sampling ratios to correct the distribution of the behavior policy to that of the target policy, and developed an algorithm that gives a consistent estimation for off-policy evaluation in the lookup table setting. [1] However, importance sampling suffers from high variances, especially when the behavior and target policies are very different. For policy gradient methods, it is hard to see there is a difference. In particular, the importance sampling ratios, widely adopted in recent popular policy gradient algorithms, can be problematic too.

In this context, the Conservative Policy Iteration (CPI) objective (Kakade and Langford 2002) is a key element in the recent spectrum of popular algorithms, including Trust Region Policy Optimization (TRPO) (Schulman et al. 2015), Proximal Policy Optimization (PPO) (Schulman et al. 2017) and many others. CPI is based on the importance sampling ratio between the new and the old policy, which can often cause high variances and lead to poor gradient estimation and unstable performances for policy gradient methods.

---

[1] There is another class of off-policy learning methods, which does not require importance sampling but to stabilize the underlying ordinary differential equation, e.g., see (Sutton, Maei, and Szepesvári 2009; Sutton et al. 2009).

TRPO avoids this problem by using a fixed threshold for the policy change. PPO applies a clipping method for the importance sampling ratio to ensure it is not too far from the current objective, and then the final objective is the minimum of the clipped and un-clipped objectives.

A more general topic than off-policy learning is sample-efficient learning. In deep reinforcement learning, sample efficiency is characterized by the following:

1) Efficient policy representation. Ensuring that the updated policy is close to the old policy is a good practice, although there is some approximation error (Tomar et al. 2022). The CPI with clipping used by PPO (Schulman et al. 2017), TPPO (Wang, He, and Tan 2020), and TR-PPO (Wang et al. 2019) ensures we consider only new policies that are not too far from the old policy. Sun et al. (2022) showed that ratio-regularizer can have a similar effect, and proposed a policy optimization method based on early stopping. 2) Convex optimization. Tomar et al. (2022) simplified the problem of maximizing the trajectory's return using convex optimization solvers which minimize the Bregman divergence. 3) Second-order methods that take advantage of the Hessian matrix. For example, identification of a trust-region is done by computing an approximation to the second-order gradient, such as TRPO (Schulman et al. 2015) and ACKTR (Wu et al. 2017). However, these methods usually have a high computation cost. 4) Off-policy learning. Besides the CPI objective-based methods, Wang et al. (2017) truncated the importance sampling ratio with bias correction and used a stochastic "dueling network" to help achieve stable learning. Furthermore, Haarnoja et al. (2018) proposed an off-policy formulation that reuses of previously collected data for efficiency. Other popular off-policy learning algorithms include soft actor-critic (Haarnoja et al. 2018) and TD3 (Fujimoto, Hoof, and Meger 2018).

In this paper, we aim to improve the policy representation due to clipping methods, which provides an improved control of the variances caused by importance sampling. Our goal is achieved by applying the preconditioning technique to the CPI objective, aided with a regularization loss in the policy change. Note that preconditioning is usually applied to an iterative method such as linear system solvers (Saad 2003; Yao and Liu 2008). Recently, there has been research on applying preconditioning in deep learning to accelerate the learning process, e.g., see (Li et al. 2016; Sappl et al. 2019). Our work is a new application of preconditioning to control variances in policy gradient estimation. Moreover, our preconditioning technique has an interesting property: it encourages exploration when the policy change is small and switches to exploitation when the policy change is large.

Background

Here we review the basis of Markov Decision Processes (MDPs) and recent popular algorithms TRPO and PPO. The key to TRPO and PPO are their objective functions, both of which are based on an approximation to the value function of a new policy.

Markov Decision Processes. An MDP is defined by $(\mathbb{S}, \mathbb{A}, \mathcal{P}, R, \lambda)$, where $\mathbb{S}$ is the state space, $\mathbb{A}$ is the action space, and for each $a \in \mathbb{A}$, $\mathcal{P}$ is a probability measure assigned to a state $s \in \mathbb{S}$, which we denote as $\mathcal{P}(\cdot|s, a)$. Define $R : \mathbb{S} \times \mathbb{A} \to \mathbb{R}$ as the reward function, where $\mathbb{R}$ is the real space. $\lambda \in (0, 1)$ is the discount factor. Here we consider stochastic policies; denote a stochastic policy by a probability measure π applied to a state s: $\pi(\cdot|s) \to [0, 1]$. At a time step t, the agent observes the current state s_t and takes an action a_t. The environment provides the agent with the next state s_{t+1} and a scalar reward $R_{t+1} = R(s_t, a_t)$. The main task of the agent is to find an optimal policy that maximizes the expected sum of discounted future rewards:

$$V_\pi(s) = \mathbb{E}_\pi \Big[\sum_{t=0}^{\infty} \lambda^t R_t \Big],$$

where $a_t \sim \pi(\cdot|s_t)$ and $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$ for all $t \geq 0$.

The state-action value function for the policy is defined similarly, except the initial action (at $t = 0$) is not necessarily chosen according to the policy:

$$Q_\pi(s, a) = \mathbb{E}_\pi \Big[\sum_{t=0}^{\infty} \lambda^t R_t \Big],$$

where $a_t \sim \pi(\cdot|s_t)$ for $t \geq 1$.

We will use A_π to denote *the advantage function*, which can be used to determine the advantage of an action a at a state s by $A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s)$. Note that if $a \sim \pi(\cdot|s)$, then the advantage is zero. So this measure computes the advantage of an action with respect to the action that is suggested by the current policy π. In the remainder of our paper, $\hat{A}$ is an approximation to the advantage, which is simply the difference between the state-action function and the value function both estimated by a specific algorithm.

Let ρ_0 be the initial state distribution. Let $\eta(\pi)$ be the expected discounted reward:

$$\eta(\pi) = \mathbb{E}_{s \sim \rho_0} \Big[V_\pi(s) \Big].$$

Now suppose we are interested in another policy $\tilde{\pi}$. Let $d_{\tilde{\pi}}$ be the stationary distribution of the policy. According to Kakade and Langford (2002), the expected return of $\tilde{\pi}$ can be calculated in terms of $\eta(\pi)$ and its advantage over π in a straight-forward way:

$$\eta(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a).$$

Here $\rho_{\tilde{\pi}}(s) = \sum_{t=0}^{\infty} \lambda^t d_{\tilde{\pi}}(s_t)$, $s_0 \sim \rho_0$ and the actions are chosen according to $\tilde{\pi}$, which is just the sum of discounted visitation probabilities. Schulman et al. (2015) approximated $\eta(\tilde{\pi})$ by replacing $\rho_{\tilde{\pi}}(s)$ with $\rho_\pi(s)$ in the right-hand side:

$$\hat{\eta}_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a|s) A_\pi(s, a).$$

Note that, in our algorithm, the new policy will be $\tilde{\pi}$. The benefit of this approximation is that the expected return of the new policy $\tilde{\pi}$ can be approximated based on the previous

samples and the old policy π. Note further, for any parameter value θ_0, because of how $\hat{\eta}$ is defined, we have

$$\hat{\eta}_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0}), \tag{1}$$

$$\nabla_\theta \hat{\eta}_{\pi_{\theta_0}}(\pi_\theta)|_{\theta=\theta_0} = \nabla_\theta \eta(\pi_\theta)|_{\theta=\theta_0}. \tag{2}$$

This means that a small gradient ascent update of θ_0 to improve $\hat{\eta}_{\pi_{\theta_0}}(\pi_{\theta_0})$ also improves $\eta(\pi_{\theta_0})$.

The TRPO objective. TRPO, PPO and our algorithm P3O all aim to improve a policy incrementally by maximizing the advantage of the new policy over the old one, by considering the influence from importance sampling. Sample-based TRPO maximizes the following Conservative Policy Iteration (CPI) objective

$$L^{cpi}(\theta) = \hat{\mathbb{E}}_t \left[r_t(\theta) \hat{A}_{\pi_{old}}(s_t, a_t) \right],$$

where $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{old}}(a_t|s_t)$, by ensuring the difference between the new policy and the old policy is smaller than a threshold. Here the operator $\hat{\mathbb{E}}_t$ refers to an empirical average over a finite number of samples. Note that the importance sampling ratio $r_t(\theta)$ can be very large; to avoid this problem, TRPO uses a hard threshold for the policy change instead of a regularization because "it is difficult to choose a single regularization factor that would work for different problems", according to Schulman et al. (2017). TRPO then uses the trust region method which is a second-order method that maximizes the objective function with a quadratic approximation. Under that method, the advantage $\hat{A}_{\pi_{old}}$ is replaced by the $Q_{\pi_{old}}$ in TRPO.

The PPO objective. This policy objective function is used to define the PPO algorithm (Schulman et al. 2017), which was motivated to augment TRPO with a first-order method extension. The PPO algorithm first samples a number of trajectories using policy π_{old}, uniformly extending each trajectory with T time steps. For each trajectory, the advantage is computed according to

$$\hat{A}_t = \lambda^{T-t} V(s_T) + \sum_{k=t}^{T-1} \lambda^{k-t} r_k - V(s_t).$$

Given the advantages and the importance sampling ratios (to re-weigh the advantages), PPO maximizes the following objective:

$$L^{ppo}(\theta) = \hat{\mathbb{E}}_t \min \left\{ r_t(\theta) \hat{A}_t, clip(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right\},$$

The role of the $clip$ operator is to make the policy update more on-policy, and the min operator's role is to optimize the policy out of the clip range. The role of the two composed operators is to prevent the potential instability caused by importance sampling, and maintain performance across different tasks. Note that this is not the first time that importance sampling causes trouble for reinforcement learning. For the discrete-action problems, both off-policy evaluation and off-policy control are known to suffer from high variances due to the product of a series of such ratios, each of which can be bigger than expected, especially when the

behavior policy and the target policy are dissimilar, e.g., see (Precup, Sutton, and Dasgupta 2001; Sutton, Maei, and Szepesvári 2009).

We noticed some works in literature called PPO an on-policy algorithm. This might be a historical mistake. ***The nature of reinforcement learning is truly off-policy***. For example, the familiar class of algorithms including Q-learning (Watkins and Dayan 1992), experience replay (Lin 1992), DQN (Mnih et al. 2015), DDPG (Lillicrap et al. 2015), Distributional RL such as C51 (Bellemare, Dabney, and Munos 2017), QR-DQN (Dabney et al. 2018) and DLTV (Mavrin et al. 2019), Horde(Sutton et al. 2011), Unreal (Jaderberg et al. 2017), Rainbow (Hessel et al. 2018), LSPI (Lagoudakis and Parr 2003), LAM-API (Yao and Szepesvári 2012), Kernel regression MDPs (Grunewalder et al. 2012) and even MCTS (Gelly and Silver 2011), are all off-policy methods.

In off-policy learning, the agent needs to constantly improve its behavior using experience that is imperfect in the sense that it is not learning optimal. Importance sampling ratios arise because one needs to correct the weighting of the objectives from the behavior policy towards the weighting that would be otherwise under a target and improved policy, *e.g.*, see (Precup, Sutton, and Dasgupta 2001; Schulman et al. 2015). Such re-weighting often leads to high variances in the policy gradient estimations, e.g., see interesting discussions by Ilyas et al. (2020). PPO uses experience from the past (old policies) and also has importance sampling to correct the sample distribution. So clearly PPO is an off-policy algorithm.

Clipping the importance sampling ratio loses the gradient information of policies in an infinitely large policy space. We define $\tilde{\Pi}_\epsilon^+ = \{\pi; \frac{\pi(s,a)}{\pi_{old}(s,a)} > 1 + \epsilon\}$ and $\tilde{\Pi}_\epsilon^- = \{\pi; \frac{\pi(s,a)}{\pi_{old}(s,a)} < 1 - \epsilon\}$. Note that PPO's optimization never crosses into these two spaces for non-negative and negative Advantages, respectively. By clipping, PPO looses the gradient information for any policy in $\tilde{\Pi}_\epsilon^+$ and $\tilde{\Pi}_\epsilon^-$. We found that there are much better policies within these two policies that PPO fails to discover in our experiments. All the PPO algorithms we reviewed extend PPO in some way; however, they unanimously inherit the clipping operation from PPO.

Method

In this section, we propose a new objective function by pre-conditioning for better exploration in the parameter space. And we also use KL divergence to ensure small and smooth policy changes between the updates, balancing exploration and exploitation.

The Scopic Objective

Inspired by the value of the CPI and PPO objectives, we propose the following refined objective:

$$L^{sc}(\theta) = \hat{\mathbb{E}}_t \left[\sigma \left(\tau \left(r_t(\theta) - 1 \right) \right) \frac{4}{\tau} \hat{A}_t \right] \tag{3}$$

where σ is the sigmoid function and τ is the temperature. The advantage $\hat{A}_t$ is computed in the same way as in PPO. We term this new objective function the **Scopic** objective,
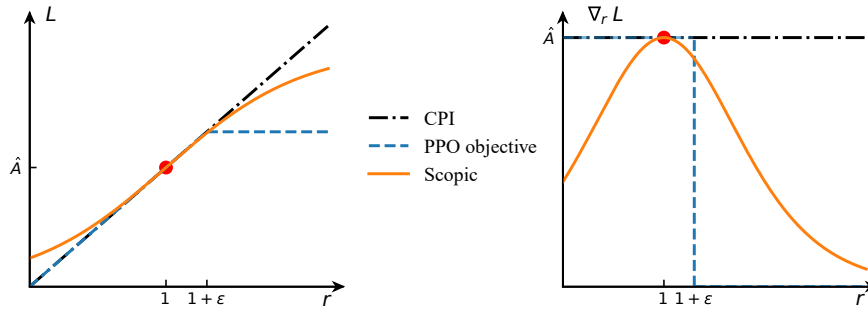
Figure 1: Left: The objective (L) versus the importance sampling ratio (r). Right: $\nabla_r L$, i.e, the gradient with respect to r. The L graph comparison includes the CPI objective (black), the PPO objective (blue), and the Scopic objective (orange). Both plots are for positive Advantage ($\hat{A}_t > 0$) and for a single sample and $\tau = 2$. The red point shows the starting point for the optimization, which is on-policy learning.

short for **s**igmoidal **c**onservative **p**olicy **i**teration **o**bjective without **c**lipping. Intuitively, according to the Scopic objective, the agent learns to maximize the scaled advantages of the new policy over the old one whilst maintaining stability by feeding the importance sampling ratio to the sigmoid function. Theoretically, by using the sigmoid, the importance sampling ratio is allowed to range from zero to infinity while the output is still in a small range, $[\sigma(-\tau), 1]$. So the new policy is allowed to be optimized in a policy space that is much larger than the clipped surrogate objective. Because the PPO objective is clipped, PPO would not have any information such as the gradient for new policies whose importance sampling ratios over the current policy are beyond the two policy spaces as defined in the PPO objective.

In addition to constraining the importance sampling ratio range in a "soft" way, there is an interesting property of the Scopic objective that is very beneficial for reinforcement learning. *The input of the sigmoid is zero if there is no change in the policy at a state.* Note that the gradient of the sigmoid achieves the maximum in this case. This means when policy change is zero or little, the sigmoid strives for a big parameter update and hence further *exploration* in the policy space. The effect of a big change in θ leads to a big change in π_θ as well, and thus the action selection has a big change, meaning that our method effectively adapts the parameter update magnitude to explore the action space. On the other hand, when the new policy changes greatly from the old policy, the gradient of the sigmoid grows small which gives the parameter little update. The effect is that the agent will focus on a close neighborhood of the policy and use the knowledge built in the policy, which leads to *exploitation*. So by using the Scopic objective, the agent learns to balance exploration and exploitation automatically via the gradient magnitude that is adapted by the sigmoid function. This method of exploration is novel for reinforcement learning and it has not been explored in previous research to the best of our knowledge.

Existing methods of exploration are mostly based on the novelty of states and actions, typically have some roots in the count-based methods such as UCT (Kocsis and Szepesvári

2006) and UCB-1 (Auer 2002). The count-based methods have a wide applications in computer games, *e.g.*, the use of UCT in AlphaGo (Silver et al. 2016), and the contextual bandits (Bubeck, Cesa-Bianchi et al. 2012) in recommendation systems (Li et al. 2010); *etc*. The count-based methods, by definition, only apply to discrete spaces. However, it is possible to extend to some smoothed versions for the continuous case, such as kernel regression UCT (Yee, Lisy, and Bowling 2016); such methods depend on a choice of kernels and a regression procedure that is performed on a data set of samples. Plappert et al. (2018) proposed a method that adds Gaussian noise to parameters for exploration and Mavrin et al. (2019) discussed parameter uncertainty versus intrinsic uncertainty, and their method implements the UCB principle without counting, by using the distribution information of the value function for uncertainty estimation. In general, the discussion on "should we be optimistic or pessimistic in the face of uncertainty" attracts lots of interests from the literature, e.g., see (Ciosek et al. 2019; Zhang and Yao 2019; Keramati et al. 2020; Zhou, Wang, and Feng 2020; Kuznetsov et al. 2020; Zhou et al. 2021). Our approach via the sigmoidally preconditioned objective is different from the above methods, and it balances exploration and exploitation given an online sample and it does not involve other samples, which is very computationally efficient.

Preconditioning At first sight, the Scopic objective term $4/\tau$ may appear odd. Here we explain this choice and what the preconditioner is. First note the important case when there is no change in the new policy. This momentary on-policy learning can be recovered by $\tau = 2$, when the input of the sigmoid function is zero. This means that the learning reduces to that of on-policy, at least for the first mini-batch update. Consider the definition of $\hat{\eta}$ in Eq.1 and Eq.2, for any parameter value θ_0. For the choice of term of $4/\tau$, we can derive

$$L^{sc}(\theta_0) = \hat{\eta}(\theta_0), \qquad \nabla_\theta L^{sc}(\theta)|_{\theta=\theta_0} = \nabla_\theta \hat{\eta}(\theta)|_{\theta=\theta_0}.$$

This ensures that the gradient descent update of the Scopic objective will improve $\hat{\eta}$ and hence η for the case of on-policy learning.
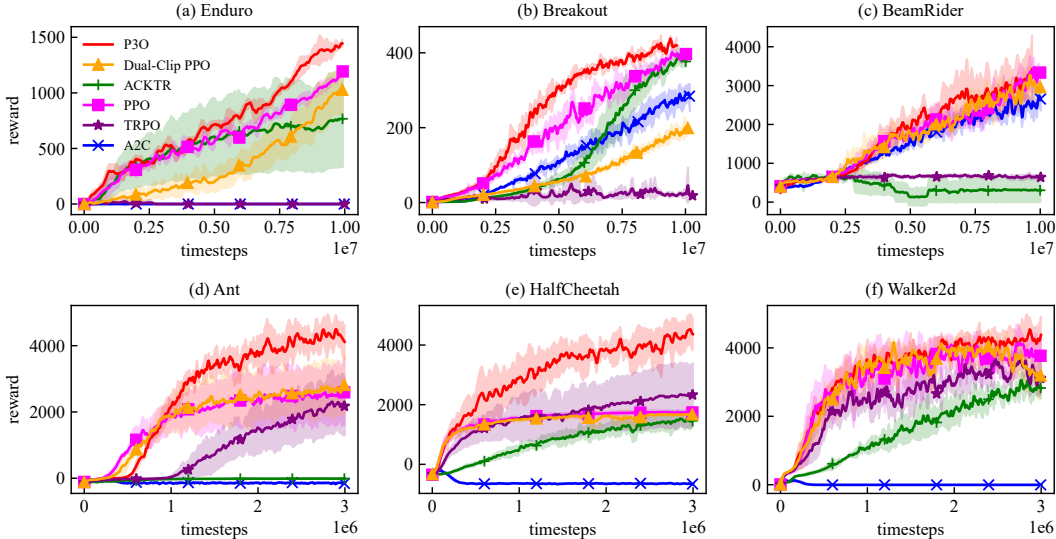
Figure 2: Performance of our P3O versus five baselines for discrete tasks (first row) and continuous tasks (second row).

The Scopic loss can be viewed as a preconditioning technique. Let $p(\theta) = \sigma(\tau(r_t(\theta) - 1))$ then the gradients of the Scopic objective and the CPI objective are as follows:

$$\nabla_\theta L^{sc} = \hat{\mathbb{E}}_t[4p(\theta)(1 - p(\theta))\nabla_\theta r_t(\theta)\hat{A}_t],$$

$$\nabla_\theta L^{cpi} = \hat{\mathbb{E}}_t[\nabla_\theta r_t(\theta)\hat{A}_t].$$

So the stochastic gradient ascent update for the Scopic objective is a modification from that of the CPI objective. This is similar to preconditioning in iterative methods (Saad 2003), but note here that the preconditioner is stochastic and applies to the stochastic gradient. Figure 1 shows the objective function and the gradient for CPI, the PPO objective and our Scopic objective.

Surrogate Function   The Surrogate function was first introduced in the TRPO paper. The core idea is that when we need to maximize an objective, we can maximize a lower bound instead. We know the PPO algorithm optimizes a proxy function smaller than CPI with a simple analysis. However, from the perspective of optimization, we can view the optimization process of PPO as modifying a one-step on-policy policy gradient to a multi-step mini-batch stochastic optimization, and then using gradient clipping to ensure the stability of optimization. This process includes both on-policy (for the first mini-batch) and off-policy processes. Our method is also a surrogate function. When $\tau > 2$, the Scopic objective is a lower bound of the CPI objective. In experiments, we show that using the sigmoid function is better than gradient clipping for the CPI objective.

KL Divergence

The Scopic objective function can facilitate more exploration, but sometimes the current policy may only need a little exploration. Therefore, for a better balance between exploration and exploitation, we consider minimizing the KL divergence between the new policy and the old policy. This operation ensures that learning is close to on-policy learning and that the importance sampling ratio will be close to one, especially when the learning rate is decayed. In particular, KL divergence can work well if the current policy is good enough. Our method has two networks: a policy network and a value network. The final objective for our policy network is

$$L^{p3o}(\theta) = \hat{\mathbb{E}}_t\Big[\sigma\left(\tau\left(r_t(\theta) - 1\right)\right)\frac{4\hat{A}_t}{\tau} - \beta KL\left(\pi_{\theta_{old}}(\cdot|s_t), \pi_\theta(\cdot|s_t)\right)\Big],$$

where $\beta \geq 0$ is the regularizer. The value networks is trained with the TD(0) algorithm (Sutton and Barto 2018), with the TD update being:

$$\hat{\nabla}_w L^{vf}(w) = \left[r_t + \lambda_v V(s_{t+1}) - V(s_t)\right]\nabla_w V_w(s_t),$$

because the objective function is $L^{vf}(w) = \hat{\mathbb{E}}_t[r_t + \lambda V_w(s_{t+1}) - V_w(s_t)]$.

Our Preconditiond Proximal Policy Optimization (P3O) algorithm [2] reduces to the gradient ascent maximizing L^{p3o} and the gradient descent minimizing L^{vf}.

Empirical Evaluation

We tested the performance of our P3O algorithm versus baselines in both continuous- and discrete tasks in OpenAI Gym (Brockman et al. 2016) and the Arcade Learning Environment (Bellemare et al. 2013). The tasks include Ant-v2, HalfCheetah-v2, and Walker2d-v2 for continuous tasks, for which the policy is parameterized using a Gaussian distribution. Discrete tasks include Enduro-v4, Breakout-v4, and BeamRider-v4. The observations on the discrete environments is shown in a four stacking frames RGB image of the

---

[2] We noted another algorithm also called P3O by (Fakoor, Chaudhari, and Smola 2020).

screen, and the policy is parameterized using Softmax distribution. In addition to TRPO and PPO, we also include A2C (Mnih et al. 2016), ACKTR (Wu et al. 2017), and DualClip-PPO (Ye et al. 2020) as baselines. We evaluate the episodic accumulated reward during the training process of each algorithm. We run each algorithm in the six environments with four random seeds and set the training time steps to be ten million for the discrete tasks and three million for continuous tasks. Both PPO and our P3O do not use augmented data over iterations. Instead, both algorithms use the data from the latest policy.

Performance Comparison

In Figure 2, the learning curves of TRPO and A2C are very flat: showing ineffectiveness for these discrete environments. The TRPO's performance was similar to the empirical results in (Wu et al. 2017). The poor performance may be because it is hard to set a proper parameter for the KL-divergence constraint since it varies in different environments. ACKTR is a second-order, natural gradient algorithm with a much higher computational cost per time step. However, it still did not outperform PPO which is a first-order method. Dualclip-PPO inherits the clipping operation from PPO objective and adds another max operator with an additional parameter (Ye et al. 2020). The algorithm was applied to the game of Honor of Kings and achieved competitive plays against human professionals. However, there was no baseline comparison. In our experiments, the algorithm performed close to or worse than than PPO. That PPO is better than all the other four baselines shows that it is indeed important to control the high variance issue of importance sampling. Our P3O outperformed all the baselines including the best performing PPO for the tasks. In the next subsection, we consider reasons why this happened.

DEON Off-Policy Measure and Policy Space Comparison

In order to understand the performance comparison between PPO and our P3O, we compared the maximum **deviation from on-policy learning** (DEON) measure, defined by $y = \max(|r - 1|)$, where the maximum was taken over the importance sampling ratio r minus one, absolute, in the collected trajectories of samples. The results of this measure are computed during the training process of PPO and P3O, and compared in Figure 3. It shows that the deviation of our P3O from on-policy learning is much bigger than PPO during training: *P3O is more off-policy than PPO*. This means P3O explores in a much bigger policy space than PPO. The clipping and minimum operations in the PPO objective prevent the algorithm from exploring the policy spaces $\tilde{\Pi}_\epsilon^+$ and $\tilde{\Pi}_\epsilon^-$ in the case of non-negative and negative Advantages. Together with the performance comparison in Figure 2, this shows that the new Scopic objective via sigmoidal preconditioning that is used by our P3O is a very effective way of conducting exploration in the parameter space which results in efficient exploration of the action space.

Note that, for discrete tasks, all the importance sampling ratios finally converged close to one due to the use of the de-
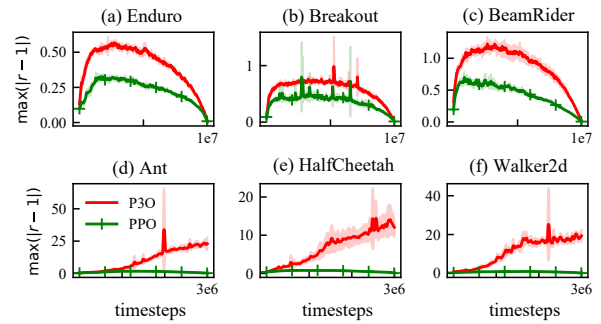

Figure 3: The DEON off-policy measure of our P3O vs. PPO during training. For PPO, the metric is computed without clipping the importance sampling ratio. Both algorithms use their *raw* importance sampling ratios computed during their individual training. P3O's deviations are much higher than PPO, which confirms that P3O explores in a policy space that is much bigger than PPO; this is because the clipping in the PPO's objective makes it fail to discover better policies beyond the clip range.

caying learning rate. This is interesting because it shows the learning rate decay can give us an on-policy learning algorithm in the long run. For continuous tasks where the fixed learning rate is used, the DEON measure of P3O is still increasing in the end, while for PPO it drops close to zero. In particular, the average DEON measure is up to as big as 60.0 for P3O which performs better than PPO. This shows clipping the importance sampling ratio in the range $[1 - \epsilon, 1 + \epsilon]$ by the PPO objective is far from sufficient to cover good policies. In our experiments, we set $\epsilon = 0.2$, as used in the PPO paper (Schulman et al. 2017). The DEON metric being still large in the end of learning means P3O is still exploring. The larger policy search space and the consistent exploration leads to a bigger improvement in performance for continuous tasks than for discrete tasks (See Figure 2 and note the continuous tasks have a much coarser scale in the y-axis).

Previously, there have been a few measures of policy dissimilarity, especially based on the L_1 distance. For example, Harutyunyan et al. (2016) and Munos et al. (2016) used $\left\| \pi(\cdot|s) - \mu(\cdot|s) \right\|_1$ to measure the dissimilarity of the two policies at a state s. We believe they are also the first to propose the notion of "off-policyness." This idea can be also extended to the action dissimilarity at a state (Meng et al. 2020), $|\pi(a|s) - \mu(a|s)|$. However, the L_1 and absolute-value based measures are not sufficiently sensitive. Consider two cases, (1) $\pi(a|s) = 0.10001, \mu(a|s) = 0.00001$; (2) $\pi(a|s) = 0.2, \mu(a|s) = 0.1$. The dissimilarity according to the absolute-value measure for the two cases is both 0.1. However, apparently the policies deviate more in the first case. The behaviour policy μ is a rare event for a at s. This suggests that, in the last iteration, we have insufficient samples for the state and action. However, in the second case, there is still a significant percentage of samples. For measuring off-policyness, it appears we need more sensitive metrics. This can be captured well by our DEON measure, which is based on the ratio of the two policies at a state. In
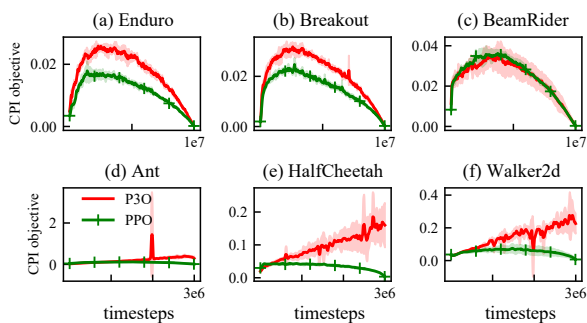
Figure 4: Plotting the CPI objective during training of P3O and PPO. The CPI objective (see the Background Section) has no clipping involved for PPO as well. P3O maximizes the CPI objective much better than PPO except for Beam-Rider.
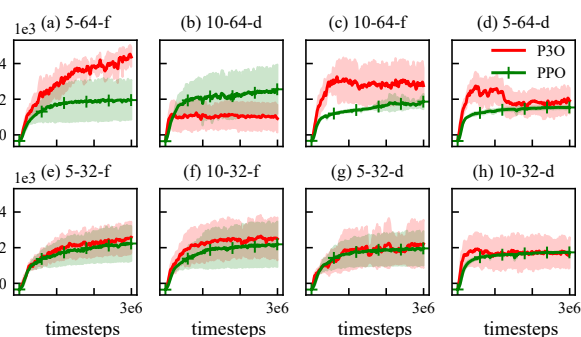

Figure 5: Hyper-parameter sensitivity studies of PPO (green) and our P3O (red) on HalfCheetah, with two schedulings of each of the three dimensions: the number of epochs on the samples, batch size, and learning rate(decay or fixed). In terms of the best case, P3O (group h) performed better than PPO (group c; consistent with the original PPO paper) with higher mean reward and lower variances.

particular, the DEON measure is 10000 for the first case, and 1.0 for the second case.

CPI Objective Comparison

This experiment was motivated by noting that both the PPO objective and the Scopic objective originate from the CPI objective. As shown by the TRPO and PPO algorithms, directly maximizing the CPI objective is problematic because of high variances of importance sampling. PPO and our P3O can be viewed as special methods for maximizing the CPI objective. So one important question is how much is the CPI objective maximized in either of the two algorithms? We thus calculated the CPI objective (without any clipping or sigmoid preconditioning) in the training process of the algorithms. The result is shown in Figure 4. For both the discrete and continuous tasks (except BeamRider), P3O consistently maximizes the CPI objective better than PPO. This means the Advantage of the new policy is consistently larger than the old policy with sigmoid preconditioning than with clipping. For the discrete tasks, the CPI objective of P3O finally converged close to that of PPO. This is also because the decay learning rate was used for discrete tasks, which became really small in the end. This leads to the importance sampling ratio being one and the Scopic objective reducing to the CPI objective. The CPI of continuous tasks is even much bigger. Because of the use of the fixed learning rate, P3O still actively explores even in the end of learning and keeps discovering new policies whose Advantage is much bigger than the old one.

Sensitivity to Hyper-Parameters

The hyper-parameter studies were performed over three dimensions (number of updates, batch size, learning rate). The numbers of epoch updates were either 5 or 10. The batch size were either 32 or 64. The learning rate were either constant 10^{-4} or the decay scheduling. The decay schedule started with a learning rate of 3×10^{-4} and decayed linearly, which was used in OpenAI's PPO implementation. This leads to eight hyper-parameter combinations whose results are shown in Figure 5. Group (c) performed the best for

PPO (consistent with the best result in the PPO paper) and Group (h) was the best for P3O. This shows PPO prefers the decay schedule while our method prefers the fixed learning rate for continuous tasks. For the best hyper-parameter group for both algorithms, P3O's variance is much lower than PPO.

Conclusion

We proposed a new surrogate objective that applies the sigmoid function to the importance sampling ratio. Our surrogate objective enables us to find much better policies outside of the clipped policy space, and can be viewed as a "soft clipping" technique, with a nice exploration property. This extends our understanding of the PPO algorithm and many later developments based on it, and suggests we should look into optimizing the CPI objective beyond the clipped policy space. We found that PPO is insufficient in *off-policyness*, and our P3O deviates more from on-policy learning than PPO, according to a measure of off-policyness during training which is called DEON. We can use this metric to measure the policy disparity introduced by the importance sampling method. We compared our P3O algorithm with five recent deep reinforcement learning baselines in both discrete and continuous environments. Results show that our method achieves better performance than the baselines.

Acknowledgments

References

Auer, P. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov): 397–422.

Bellemare, M. G.; Dabney, W.; and Munos, R. 2017. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, 449–458.

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

Bertsekas, D. P. 1995. A Counterexample to Temporal Differences Learning. *Neural Computation*, 7(2): 270–279.

Boyan, J. A.; and Moore, A. W. 1994. Generalization in Reinforcement Learning: Safely Approximating the Value Function. In *International Conference on Neural Information Processing Systems*, 369–376.

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

Bubeck, S.; Cesa-Bianchi, N.; et al. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 1–122.

Ciosek, K.; Vuong, Q.; Loftin, R.; and Hofmann, K. 2019. Better Exploration with Optimistic Actor Critic. In *Advances in Neural Information Processing Systems*, volume 32.

Dabney, W.; Rowland, M.; Bellemare, M.; and Munos, R. 2018. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Fakoor, R.; Chaudhari, P.; and Smola, A. J. 2020. P3o: Policy-on policy-off policy optimization. In *Uncertainty in Artificial Intelligence*, 1017–1027.

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

Gelly, S.; and Silver, D. 2011. Monte-Carlo tree search and rapid action value estimation in computer Go. *Artificial Intelligence*, 175(11): 1856–1875.

Gordon, G. J. 1995. Stable Function Approximation in Dynamic Programming. Technical report, Carnegie Mellon University, USA.

Grunewalder, S.; Lever, G.; Baldassarre, L.; Pontil, M.; and Gretton, A. 2012. Modelling transition dynamics in MDPs with RKHS embeddings. *arXiv preprint arXiv:1206.4655*.

Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. 2018. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

Harutyunyan, A.; Bellemare, M. G.; Stepleton, T.; and Munos, R. 2016. Q(lambda) with Off-Policy Corrections. In *Algorithmic Learning Theory*, 305–320.

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

Ilyas, A.; Engstrom, L.; Santurkar, S.; Tsipras, D.; Janoos, F.; Rudolph, L.; and Madry, A. 2020. A Closer Look at Deep Policy Gradients. In *International Conference on Learning Representations*, volume 32.

Jaderberg, M.; Mnih, V.; Czarnecki, W. M.; Schaul, T.; Leibo, J. Z.; Silver, D.; and Kavukcuoglu, K. 2017. Reinforcement Learning with Unsupervised Auxiliary Tasks. In *International Conference on Learning Representations*, volume 11.

Kakade, S.; and Langford, J. 2002. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, volume 27.

Keramati, R.; Dann, C.; Tamkin, A.; and Brunskill, E. 2020. Being optimistic to be conservative: Quickly learning a CVaR policy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 4436–4443.

Kocsis, L.; and Szepesvári, C. 2006. Bandit Based Monte-Carlo Planning. In *Proceedings of the 17th European Conference on Machine Learning*, 282–293.

Kuznetsov, A.; Shvechikov, P.; Grishin, A.; and Vetrov, D. 2020. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *International Conference on Machine Learning*, 5556–5566.

Lagoudakis, M. G.; and Parr, R. 2003. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4: 1107–1149.

Li, C.; Chen, C.; Carlson, D.; and Carin, L. 2016. Preconditioned stochastic gradient Langevin dynamics for deep neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*, volume 30.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670.

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

Lin, L.-J. 1992. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3): 293–321.

Mavrin, B.; Yao, H.; Kong, L.; Wu, K.; and Yu, Y. 2019. Distributional Reinforcement Learning for Efficient Exploration. In *Proceedings of the 36th International Conference on Machine Learning*, 4424–4434.

Meng, W.; Zheng, Q.; Yang, L.; Li, P.; and Pan, G. 2020. 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 *Proceedings of The 33rd 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.

Munos, R.; Stepleton, T.; Harutyunyan, A.; and Bellemare, M. 2016. Safe and Efficient Off-Policy Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 29.

Plappert, M.; Houthooft, R.; Dhariwal, P.; Sidor, S.; Chen, R. Y.; Chen, X.; Asfour, T.; Abbeel, P.; and Andrychowicz, M. 2018. Parameter Space Noise for Exploration. In *International Conference on Learning Representations*, volume 33.

Precup, D.; Sutton, R. S.; and Dasgupta, S. 2001. Off-Policy Temporal Difference Learning with Function Approximation. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 417–424.

Precup, D.; Sutton, R. S.; and Singh, S. P. 2000. Eligibility Traces for Off-Policy Policy Evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 759–766.

Saad, Y. 2003. *Iterative Methods for Sparse Linear Systems*. Other Titles in Applied Mathematics. SIAM, second edition. ISBN 978-0-89871-534-7.

Sappl, J.; Seiler, L.; Harders, M.; and Rauch, W. 2019. Deep learning of preconditioners for conjugate gradient solvers in urban water related problems. *arXiv preprint arXiv:1906.06925*.

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.; 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.

Sun, M.; Kurin, V.; Liu, G.; Devlin, S.; Qin, T.; Hofmann, K.; and Whiteson, S. 2022. You May Not Need Ratio Clipping in PPO. *arXiv preprint arXiv:2202.00079*.

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

Sutton, R. S.; Maei, H.; and Szepesvári, C. 2009. A Convergent O(n) Temporal-difference Algorithm for Off-policy Learning with Linear Function Approximation. In *Advances in Neural Information Processing Systems*, volume 21.

Sutton, R. S.; Maei, H. R.; Precup, D.; Bhatnagar, S.; Silver, D.; Szepesvári, C.; and Wiewiora, E. 2009. Fast Gradient-Descent Methods for Temporal-Difference Learning with Linear Function Approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 993–1000.

Sutton, R. S.; Modayil, J.; Delp, M.; Degris, T.; Pilarski, P. M.; White, A.; and Precup, D. 2011. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 761–768.

Tomar, M.; Shani, L.; Efroni, Y.; and Ghavamzadeh, M. 2022. Mirror Descent Policy Optimization. In *International Conference on Learning Representations*, volume 32.

Tsitsiklis, J.; and Van Roy, B. 1997. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5): 674–690.

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*, volume 32.

Wang, Z.; Bapst, V.; Heess, N.; Mnih, V.; Munos, R.; Kavukcuoglu, K.; and de Freitas, N. 2017. Sample Efficient Actor-Critic with Experience Replay. In *International Conference on Learning Representations*, 201–207.

Watkins, C. J.; and Dayan, P. 1992. Q-learning. *Machine learning*, 8(3): 279–292.

Wu, Y.; Mansimov, E.; Grosse, R. B.; Liao, S.; and Ba, J. 2017. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *Advances in neural information processing systems*, 30: 5279–5288.

Yao, H.; and Liu, Z.-Q. 2008. Preconditioned temporal difference learning. In *Proceedings of the International Conference on Machine Learning*, 1208–1215.

Yao, H.; and Szepesvári, C. 2012. Approximate policy iteration with linear action models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1212–1218.

Ye, D.; Liu, Z.; Sun, M.; Shi, B.; Zhao, P.; Wu, H.; Yu, H.; Yang, S.; Wu, X.; Guo, Q.; et al. 2020. Mastering complex control in moba games with deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 6672–6679.

Yee, T.; Lisy, V.; and Bowling, M. 2016. Monte Carlo Tree Search in Continuous Action Spaces with Execution Uncertainty. In *Proceedings of International Joint Conference on Artificial Intelligence*, 690–696.

Zhang, S.; and Yao, H. 2019. Quota: The quantile option architecture for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 5797–5804.

Zhou, F.; Wang, J.; and Feng, X. 2020. Non-crossing quantile regression for distributional reinforcement learning. In *Advances in Neural Information Processing Systems*, 15909–15919.

Zhou, F.; Zhu, Z.; Kuang, Q.; and Zhang, L. 2021. Non-decreasing Quantile Function Network with Efficient Exploration for Distributional Reinforcement Learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3455–3461.