Proximal Policy Optimization with Relative Pearson Divergence*

Taisuke Kobayashi

Abstract—The recent remarkable progress of deep reinforcement learning (DRL) stands on regularization of policy for stable and efficient learning. A popular method, named proximal policy optimization (PPO), has been introduced for this purpose. PPO clips density ratio of the latest and baseline policies with a threshold, while its minimization target is unclear. As another problem of PPO, the symmetric threshold is given numerically while the density ratio itself is in asymmetric domain, thereby causing unbalanced regularization of the policy. This paper therefore proposes a new variant of PPO by considering a regularization problem of relative Pearson (RPE) divergence, so-called PPO-RPE. This regularization yields the clear minimization target, which constrains the latest policy to the baseline one. Through its analysis, the intuitive threshold-based design consistent with the asymmetry of the threshold and the domain of density ratio can be derived. Through four benchmark tasks, PPO-RPE performed as well as or better than the conventional methods in terms of the task performance by the learned policy.

Index Terms—Reinforcement Learning, Machine Learning for Robot Control, Deep Learning Methods

I. INTRODUCTION

The next-generation robots aim to resolve complicated tasks with unknown dynamics: e.g. human assistance [1] and cloth manipulation [2]. Reinforcement learning (RL) [3] is the promising methodology for them due to model-free approach. Indeed, its extension using deep neural networks (DNNs) [4] to approximate policy and value functions, named deep RL (DRL) [5], has received a lot of attention.

The recent remarkable progress of DRL stands on policy regularization. By constraining the update amount and direction of policy, the policy regularization allows the policy to improve smoothly toward the optimal one. Several variations of regularization way have been proposed: e.g. hard/soft constraint by Kullback-Leibler (KL) divergence between the latest and baseline policies [2], [6]; maximization of entropy of policy [7]; indirect minimization of variance of temporal difference (TD) [8]; and so on. All of them contribute significantly to improving learning performance and are treated as representatives of DRL. Note that these recent studies have been systematized as policy-regularized RL [9].

Among the methods for policy regularization, this paper further focuses on proximal policy optimization (PPO) [10], which has been proposed as soft constraint version of KL divergence. Therefore, the first implementation of PPO is with a direct regularization of KL divergence between the latest and baseline policies. The gain of this regularization should be adjusted for facing situations, but the proposed way was in an ad-hoc manner, which could not fully optimize the gain. For more intuitive and practical implementation, another version, which is more popular than the first one, assumes the use of importance sampling and clips density ratio of the latest and baseline policies with a threshold. By doing so, the policy gradient vanishes when the density ratio is clipped, and that constrains the update of the policy. From here, PPO refers to this clipping version.

However, PPO has two theoretical problems at the cost of its intuitive implementation. First one comes from the modification from the soft constraint of KL divergence to clipping the density ratio, namely, its minimization target is unclear. Indeed, the follow-up study of PPO [11] analyzed its behavior and pointed out that PPO has no capability to make the latest policy bind to the baseline one. As another problem, the symmetric threshold is given numerically while the density ratio itself is in asymmetric domain. This gap may induce unbalanced regularization of the policy.

Hence, this paper tackles these two theoretical issues in PPO by considering a new regularization problem of relative Pearson (RPE) divergence [12], [13]. This RPE-divergence-based regularization yields the clear minimization target, which can softly constrain the latest policy to the baseline one. The symmetric density ratio can be obtained by adjusting a relativity parameter. To inherit the intuitive threshold-based design of PPO, the gain of RPE regularization is converted to the corresponding threshold through mathematical derivation. The proposed method, so-called PPO-RPE, therefore achieves both the intuitive implementation by the threshold and solving the theoretical issues in PPO.

To investigate the effects of solving the theoretical issues in PPO by PPO-RPE, four benchmark tasks are simulated in a Pybullet simulator [14]. With comparison to the conventional methods (i.e. PPO and PPO-RB [11]), PPO-RPE achieved the equivalent/superior task performance in all the tasks. During training the tasks, it is found that PPO-RPE regularizes the policy more than the conventional methods to explicitly minimize the RPE divergence between the latest and baseline policies. As a consequence, it can be indicated that PPO-RPE contributes to improvement of the learning performance by restricting the update of the latest policy in a theoretically rigorous way.

II. PRELIMINARIES

A. Policy-regularized reinforcement learning

The original purpose of RL [3] is that an agent optimizes its policy \( \pi \) in order to maximize the sum of rewards \( r \) (i.e. a
return $\sum_{k=0}^{\infty} \gamma^k r_k$ with $\gamma \in [0, 1)$ discount factor) received from an environment. To this end, numerous interactions, which exchange actions $a$ sampled from the agent with states $s$ sampled from the environment according to Markovian dynamics, are performed during learning $\pi$. This paper considers the minimization problem of the expected negative return at $t$ time step, in other words the expected negative action-value function $Q(s_t, a_t)$, as follows:

$$\pi^*(a_t | s_t) = \arg \min_{\pi} -E_{\pi_1 \sim \pi} \left[ Q(s_t, a_t) \right]$$  \hspace{1cm} (1)$$

where $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$ with $V(s_t)$ denotes the advantage function with the state-value function. $V(s_t)$ can be added to the minimization target since its gradient w.r.t policy is equal to zero. The optimal policy $\pi^*$ is obtained through this minimization problem.

Recently, to improve RL performance in terms of fast convergence and reducing variance of results, regularization terms for the policy $\Omega$ is additionally minimized [9]. That is, the following surrogated minimization target can unify most of the recent RL algorithms (e.g. [7], [8], [10]).

$$L^1(\pi) = -E_{\pi_1 \sim \pi} \left[ A(s_t, a_t) \right] + E_{\pi_2 \sim \pi} \left[ \Omega(s_t, a_t) \right]$$

$$\pi^* = \arg \min_{\pi} -E_{\pi_1 \sim \pi} \left[ A(s_t, a_t) \right] - E_{\pi_2 \sim \pi} \left[ \Omega(s_t, a_t) \right]$$  \hspace{1cm} (2)$$

where by introducing a baseline policy $\pi_b(a_t | s_t)$ (e.g. the old version of $\pi$ or the one outputted from slowly updated target network [15]) for importance sampling, a density ratio $\rho = \pi / \pi_b$ is derived in the minimization target, which is clipped in PPO [10], [11] (details are in the next section).

The optimal policy in the policy-regularized optimization problem, $\pi^*$, is obtained by minimizing $L^1(\pi)$.

Here, a policy-gradient method is generally employed for solving this optimization problem. That is, when the policy is parameterized by a parameter set $\theta$ (e.g. weights and biases in deep neural networks), $\theta$ is updated by its first-order gradient as follows:

$$\nabla_\theta L^1(\pi) = -E_{\pi_1 \sim \pi_b} \left[ \nabla_\theta \rho(s_t, a_t)A(s_t, a_t) \right]$$

$$\simeq -\rho(s_t, a_t)A^\dagger(s_t, a_t)\nabla_\theta \ln \pi(a_t | s_t)$$  \hspace{1cm} (3)$$

$$\theta \leftarrow \theta - \alpha \text{SGD}(\nabla_\theta L^1(\pi))$$  \hspace{1cm} (4)$$

where $A^\dagger = A^\dagger + \pi\nabla_\pi A^\dagger = A^\dagger + \rho\nabla_\rho A^\dagger$. If $A^\dagger$ has no direct computational graph with $\pi$, $A^\dagger = A^\dagger$. Monte-Carlo approximation, where only one action is sampled from the baseline policy, is applied to omit the use of closed-form solution of the expectation. In general DRL, $\theta$ is updated using one of the stochastic gradient descent (SGD) optimizers like [16].

**B. Proximal policy optimization**

In PPO [10], the initial version had the explicit regularization for KL divergence. However, its adaptive gain could not be designed properly, and this version did not achieve significant performance. To improve the performance with the practical implementation, the new version of PPO and its variant (named PPO-RB [11]) therefore give eq. (2) the following condition according to a threshold parameter $\epsilon$.

That is, the following surrogated density ratio $\rho^{\text{PPO}}$ replaces the original $\rho$ in eq. (2), instead of explicit definition of $\Omega$.

$$\rho^{\text{PPO}} = \begin{cases} -\eta \rho + (1 + \eta)(1 + \sigma \epsilon) \sigma(\rho - 1) \geq \epsilon & \text{ otherwise} \\ \rho & \end{cases}$$  \hspace{1cm} (5)$$

$$L^{\text{PPO}}(\pi) = -E_{\pi_1 \sim \pi_b} \left[ \rho^{\text{PPO}}(s_t, a_t)A(s_t, a_t) \right]$$  \hspace{1cm} (6)$$

where $\sigma$ denotes the sign of $A$, and $\eta \geq 0$ is for rollback to the baseline policy. If $\eta = 0$, this problem is equivalent to the original PPO.

With this condition for clipping if $\eta = 0$ or rollback if $\eta > 0$, PPO inhibits update of the latest policy $\pi$ away from the baseline $\pi_b$, although its specific regularization target represented by $\Omega$ is complicated. Note that it is reported that only the case with $\eta > 0$ enables $\pi$ to bind with $\pi_b$ [11].

**C. Relative Pearson divergence**

To measure the divergence between two probabilities, KL divergence is the most major metric. However, it is actually the special case of f-divergence, which is the generalized divergence [17], and therefore, we can measure the divergence using either of other f-divergences. To inherit the behaviors of PPO (eqs. (5) and (6)), the following conditions are required for selecting the specific f-divergence.

1. When $\rho = 1$, the term inside the expectation of the loss function should be $\rho A$.
2. Only when $\rho$ is on the threshold, the gradient must be zero and there is only one vertex.

In addition, the symmetry is also desired. Unfortunately, it is found that KL divergence cannot satisfy all of them simultaneously. Thus, the alternative divergence that satisfies all the above is heuristically given as follows.

Specifically, Pearson (PE) divergence, which is one of the f-divergences, is utilized while modifying it to the relative version, RPE divergence [12], [13]. The raw PE divergence between two policies, $\pi$ and $\pi_b$, is given as follows:

$$\text{PE}(\pi, \pi_b) = E_{\pi_1 \sim \pi_b} \left[ \frac{1}{2}(\rho(a) - 1)^2 \right]$$  \hspace{1cm} (7)$$

Since the mean of $\rho$ is always equal to 1 (i.e. $\int \pi_b(a | s)\rho(s, a)da = \int \pi(a | s)da = 1$), this divergence computes...
Algorithm 1 PPO-RPE

1: Set $\epsilon \in (0, 1)$ (0.1–0.3 is the recommended range)
2: Set $\beta \in [0, 1]$ (0.5 is the default value)
3: Initialize the policy $\pi$ with $\theta$ and the baseline policy $\pi_b$
4: Initialize optimizer as SGD with learning rate $\alpha$
5: while True do
6:   Get $s_t$ from the environment
7:   Sample $a_t$ from the baseline policy $\pi_b(a_t | s_t)$
8:   Get $s_{t+1}, r_t$ from the environment by acting on $a_t$
9:   $A = Q(s_t, a_t) - V(s_t)$
10:  $\sigma = \text{sign}(A)$
11:  $\rho = \frac{\pi(a_t | s_t)}{\pi_b(a_t | s_t)}$
12:  $\rho_\beta = \frac{1 - \beta + \beta \rho}{\rho}$
13:  $\text{denom} = \sigma \epsilon |\beta \sigma \epsilon - 2 \{1 - \beta (1 + \epsilon)\}$
14:  $\text{num} = A(\rho_\beta - 1) \left\{ \beta (\rho_\beta - 1) - \frac{2(1 - \beta)}{1 - \beta + \beta \rho} \right\}$
15:  $\tilde{A}_{\text{RPE}} = A - \frac{\text{num}}{\text{denom}}$
16:  $g = -\rho \tilde{A}_{\text{RPE}} \nabla_{\theta} \ln \pi(a_t | s_t)$
17:  $\theta \leftarrow \theta - \alpha \text{SGD}(g)$
18:  Update $\pi_b$ according to $\pi$ with $\theta$
19:  $t \leftarrow t + 1$
20: end while

the mean squared error of $\rho$ from its mean. This divergence is non-negative and vanishes if and only if $\pi = \pi_b$.

When the denominator $\pi_b$ has small value, the numerical computation of PE divergence would diverge. For numerical stability, RPE divergence first introduces the following mixture probability with a mixture ratio $\beta \in [0, 1]$.

$$\pi_\beta(a | s) = \beta \pi(a | s) + (1 - \beta) \pi_b(a | s)$$ (8)

Using this, RPE divergence is defined as follows:

$$\text{PE}_\beta(\pi, \pi_b) = \mathbb{E}_{a \sim \pi} \left[ \frac{1}{2} \left( \frac{\rho(a)}{\beta \rho(a) + (1 - \beta)} - 1 \right)^2 \right]$$ (9)

where the relative density ratio between $\pi$ and $\pi_\beta$, $\rho_\beta = \pi / \pi_\beta$, is within $[0, 1/\beta]$. The numerical stability is, therefore, guaranteed as expected, while KL divergence is numerically unstable due to the use of logarithm and raw density ratio. In particular, when $\beta = 0.5$, the range of the density ratio is bounded in $[0, 2]$ while it has the mean on 1, that is, it has a symmetric distribution around one (see Fig. 1). This symmetry of the relative density ratio yields the asymmetric thresholds in the proposed method.

III. PROXIMAL POLICY OPTIMIZATION WITH RELATIVE PEARSON DIVERGENCE: PPO-RPE

A. Overview

The issues in PPO and PPO-RB are listed below:

- No explicit regularization term is given, thereby making the problem mathematically difficult to understand what divergence policy is regularized in terms of.
- No capability to bind the latest policy to the baseline policy has been reported in [11], where ad-hoc implementations have been proposed to solve it.
- Inconsistency between symmetric design of the threshold and asymmetric domain of the density ratio would induce unbalanced regularization.

To resolve them by a mathematically rigorous way, this paper proposes a proximal policy optimization with relative Pearson divergence, so-called PPO-RPE. Specifically, PPO-RPE defines the minimization problem with RPE divergence regularization. Since the regularization target is mathematically given, the capability to bind the latest policy to the baseline one can easily be guaranteed. In addition, the gain of its regularization is adaptively designed based on a symmetric threshold for the relative density ratio in RPE divergence, which makes the threshold for the raw density ratio asymmetric.

Here, the overall implementation of PPO-RPE is summarized in Alg. 1 while the details are described from the next sections. In addition, all the examples of the negative loss functions (i.e. $\rho A^1$) for the proposed and conventional methods are illustrated in Fig. 2. As can be seen in this figure, the previous methods (i.e. PPO and PPO-RB) regularize only when $\rho$ exceeds one side ($\rho > 1 + \epsilon$ when $A > 0$ and $\rho < 1 - \epsilon$ when $A < 0$). In that case, the latest policy $\pi$ can leave from the baseline one $\pi_b$, without constraint to $\pi_b$ for example when $\rho < 1$ and $A > 0$. In contrast, PPO-RPE has the capability to bind $\pi$ to $\pi_b$ in accordance with the asymmetric threshold for $\rho$. 

![Negative loss functions for the respective methods: PPO and PPO-RB.](image)
B. Minimization target

As a first step to explicitly regard PPO-RPE as a subclass of policy-regularized RL described in eq. (3), the regularization term for PPO-RPE, \( \Omega^{\text{RPE}} \), is defined. Note that the arguments \((s_t, a_t)\) in \( A \), \( \rho \), and \( \beta \) are omitted from here for brevity.

With \( \pi_\beta \) in eq. (8) and \( \rho_s = \pi / \pi_\beta = \rho / (1 - \beta + \beta \rho) \), RPE-divergence-based \( \Omega^{\text{RPE}} \) is given as follows:

\[
\Omega^{\text{RPE}} = C \left( \rho^2 - 1 \right)^2 \rho = C \frac{1 - \beta + \beta \rho}{\rho} \left( \rho^2 - 1 \right)^2 \tag{10}
\]

where \( C \) denotes the gain of this regularization. The expectation w.r.t \( \pi \) of this regularization is equivalent to eq. (9) amplified by \( C \).

Finally, by substituting eq. (10) for eq. (2), we have the following loss function to be minimized for the policy.

\[
L^{\text{RPE}}(\pi) = -\mathbb{E}_{a_t \sim \pi_\beta} \left[ \rho A - C(1 - \beta + \beta \rho)(\rho^2 - 1) \right] = -\mathbb{E}_{a_t \sim \pi_\beta} \left[ \rho A^{\text{RPE}} \right] \tag{11}
\]

where the regularization term is included in \( A^{\text{RPE}} \). Note that \( \rho A^{\text{RPE}} / A \) can be regarded as \( \rho^{\text{RPE}} \) corresponding to PPO.

C. Policy gradient

To analyze the behavior of PPO-RPE and to design \( C \) (see the next section), the policy gradient is analytically derived. Since we know \( \nabla_{\theta} A = \rho \nabla_{\theta} \ln \pi \) and the expectation of the loss function is approximated by Monte Carlo method, what we have to derive is the gradient of \( L^{\text{RPE}} \) w.r.t the density ratio, which is equivalent to the negative surrogated advantage function: \( \nabla_{\rho} L^{\text{RPE}} \approx -A^{\text{RPE}} - \rho \nabla_{\rho} A^{\text{RPE}} = -A^{\text{RPE}} \). Now, \( A^{\text{RPE}} \) can be computed analytically as follows:

\[
A^{\text{RPE}} \approx A - C \beta (\rho^2 - 1)^2 - 2C(1 - \beta + \beta \rho)(\rho^2 - 1) \frac{\partial \rho_s}{\partial \beta} = A \left[ 1 - \frac{C \rho}{A(\rho^2 - 1)} \left\{ \beta(\rho^2 - 1) + \frac{2(1 - \beta)}{1 - \beta + \beta \rho} \right\} \right] \tag{12}
\]

where \( \partial \rho_s / \partial \beta = (1 - \beta) / (1 - \beta + \beta \rho)^2 \). This is substituted for eq. (3), and used for updating the parameter set \( \theta \) in eq. (4).

The second term in the square bracket is for the regularization, and vanishes if and only if \( \rho = \rho_s = 1 \) (i.e. \( \pi = \pi_\beta \)) regardless of \( C \) and \( \beta \). Therefore, we can see that, with this policy gradient, the policy \( \pi \) is appropriately updated toward maximization of the advantage function \( A \) (i.e. the action-value function \( Q \)) while being regularized to the baseline policy \( \pi_b \).

D. Threshold-based design of gain for regularization

The optimal gain of the regularization \( C \) is difficult to be designed due to its non-intuitiveness. In addition, the constant \( C \) is insufficient to adapt the agent to various environments without tuning \( C \) for each environment. Therefore, inspired by PPO, the threshold-based design of \( C \) is derived.

First of all, we focus on the fact that the density ratio \( \rho \) is asymmetric around one, and therefore, the symmetric threshold for it as conducted in PPO would cause unbalanced regularization. Instead, PPO-RPE sets the threshold for the relative density ratio \( \rho_s \), which can yield the symmetric domain if \( \beta = 0.5 \) (see Fig. 1). Given the threshold parameter \( \epsilon \in (0, 1) \), the following condition is considered.

\[
\rho^* = \frac{(1 - \beta)(1 + \sigma \epsilon)}{1 - \beta(1 + \sigma \epsilon)} = 1 + \frac{\sigma \epsilon}{1 - \beta(1 + \sigma \epsilon)} \tag{13}
\]

where \( \rho^* \) can be easily derived using the fact \( \rho^* = \rho / (1 - \beta + \beta \rho) \). Note that the term \( \sigma \epsilon / (1 - \beta(1 + \sigma \epsilon)) \) in \( \rho^* \) yields the asymmetric threshold for \( \rho \), and if \( \beta = 0 \) with no relativity, it is reverted to be symmetric.

At \( \rho^* \) (and \( \rho^* \)), to bind the latest policy \( \pi \) to the baseline \( \pi_b \), \( A^{\text{RPE}} \) is desired to be zero. That is, \( C \) can be derived from the terms in the square bracket of eq. (12) as follows:

\[
0 = 1 - \frac{C}{A} (\rho^2 - 1) \left\{ \beta(\rho^2 - 1) + \frac{2(1 - \beta)}{1 - \beta + \beta \rho^*} \right\} = 1 - \frac{C}{A} \sigma \epsilon \left\{ \beta \sigma \epsilon + 2(1 - \beta) \frac{1 - \beta(1 + \sigma \epsilon)}{1 - \beta} \right\}
\]

\[
C = \frac{1}{\sigma \epsilon [\beta \sigma \epsilon + 2(1 - \beta(1 + \sigma \epsilon))] + 1} \tag{14}
\]

This equation shows that \( C \) depends on the scale of \( A \), namely, it has adaptability to various environments with different scales of reward function.

By substituting this for eq. (12), PPO-RPE achieves RPE-divergence-based regularization with the threshold-based gain.

\[
A^{\text{RPE}} = A - \frac{A(\rho^2 - 1) \left\{ \beta(\rho^2 - 1) - \frac{2(1 - \beta)}{1 - \beta + \beta \rho} \right\} \sigma \epsilon [\beta \sigma \epsilon - 2(1 - \beta(1 + \sigma \epsilon))] + 1}{\sigma \epsilon [\beta \sigma \epsilon - 2(1 - \beta(1 + \sigma \epsilon))] + 1} \tag{16}
\]

Finally, the parameter set \( \theta \) is updated according to SGD with the policy gradient \( -\rho A^{\text{RPE}} \nabla_{\theta} \ln \pi(a_t | s_t) \).

IV. SIMULATIONS

A. Conditions

Four benchmark tasks are simulated by Open AI Gym with Pybullet dynamical engine [14], [20]. The agent performs the task using the learned policy 50 times to compute the sum of rewards for each, and their median is used as the score for each condition. Totally, 10 trials for each condition are conducted with different random seeds.

| Symbol | Meaning | Value |
|--------|---------|-------|
| \( \gamma \) | Discount factor | 0.99 |
| \( \alpha \) | Learning rate | 3e-4 |
| \( \tau, \nu \) | Hyperparameters for [18] | (0.3, 1.0) |
| \( \lambda_{\text{max}}^1, \lambda_{\text{max}}^2 \) | Hyperparameters for [19] | (0.5, 0.95, 10) |
| \( \epsilon \) | Threshold parameter | 0.1 |
| \( \beta_{\text{DE}} \) | Gain for entropy regularization | 0.01 |
| \( \beta_{\text{TD}} \) | Gain for TD regularization | 0.01 |
In this paper, the policy $\pi$ is parameterized by student-t distribution [21], aiming for more robust updates of $\pi$ than one parameterized by normal distribution. Neural networks for approximating the parameters in $\pi$ contains $L = 5$ fully connected layers with $N = 100$ neurons and pairs of layer normalization [22] and Swish activation function [23] for nonlinearity. For the baseline policy $b$, the policy before update, $\pi(a|s; \theta_{old})$, is employed. That is, the proposed and conventional methods add the regularization to the latest policy so that a single parameter update does not cause an extreme change in the policy.

To learn the value function, a target network with $t$-soft update [18] is employed for stable and efficient improvement. Learning of policy is accelerated using adaptive eligibility traces [19], instead of experience replay [24] for reducing memory cost. In addition, the policy entropy regularization based on SAC [7] with a regularization weight $\beta$ are combined. As an optimizer employed in eq. (4), a robust SGD, i.e., LaProp [25] with t-momentum [16] and d-AmsGrad [26] (so-called td-AmsProp), is employed with their default parameters except the learning rate. Note that these implementations are built on PyTorch [27].

The common parameters for the above implementations are summarized in Table I. Two conventional methods [10], [11] and the proposed method, PPO-RPE, are compared throughout the simulations as follows:

1) PPO [10]: $\eta = 0$ for no rollback
2) PPO-RB [11]: $\eta = 0.3$ as recommended value
3) PPO-RPE (proposa): $\beta = 0.5$ for symmetry

Note that the threshold parameter $\epsilon = 0.1$, which was chosen from the range recommended in [10] (i.e. $[0.1, 0.3]$), is commonly set as listed in Table I.

---

**B. Results**

To evaluate the task performance by the learned policy, the test results after learning are also depicted in Fig. 3. Learning curves gained from 10 trials for each condition are also illustrated in Fig. 4. Note that since the experience replay [24] was not employed unlike the major implementations, the learning tendency is very different. In addition, the trajectories of the amount of regularization (i.e. $\sigma(\rho - \rho^1)$) are shown in Fig. 5.

First of all, as can be seen in Fig. 3, PPO-RPE performed as well or better than the other methods. PPO and PPO-RB deteriorated learning speed in the respective tasks (see Fig. 4(a) and (b)). In contrast, PPO-RPE shows its excellent learning speed in both tasks. Even in high-dimensional locomotion tasks, PPO-RPE is also effective, and its regularization effect is especially remarkable to improve the task performance in HalfCheetah task (see Fig. 4(c)), which requires conservative learning to avoid local optima.

In Fig. 5 we can find that PPO-RPE has stronger constraint from the latest to baseline policies than the others. This is because PPO-RPE always regularizes the policy except for $\pi = b$, while PPO (and PPO-RB) does not if the policy is within the threshold. Nevertheless, it is noted that all PPO variants serve as a safeguard against extreme cases, as imagined from the small mean of $\sigma(\rho - \rho^1)$ during training.

---

**V. Conclusion**

This paper proposed PPO-RPE, which is a variant of PPO integrated with RPE divergence regularization. In the standard PPO, the regularization target is not explicitly given, inducing undesired behaviors like no capability to bind policy with its baseline. Although the variant of PPO, named PPO-RB, provides the ad-hoc solution for this problem, it still remains another issue about inconsistency between symmetric threshold and asymmetric domain of density ratio, which would make regularization unbalanced. To resolve both of them in a mathematically rigorous way, the minimization problem with RPE divergence, gain of which was designed based on the symmetric threshold for the relative density ratio, was considered as PPO-RPE. The threshold-based gain as well as the standard PPO yields the regularization adaptive to the scale of the main minimization target (i.e. the negative advantage function). As a result, PPO-RPE with stronger constraint to the baseline outperformed the conventional methods by enhancing learning stability.

Although the threshold allows us to design the gain of regularization intuitively, this may still have different optimal values depending on the average divergence between the latest and baseline policies. In the near future, therefore, the divergence between their marginal policies will be investigated to design the threshold for further improvement. As another perspective of PPO-RPE for robotic applications, the ability to appropriately bind the policy updates is expected to facilitate the management of the behaviors during the learning process. By taking advantage of this benefit, application research of the proposed method will be promoted.
Fig. 4. Learning curves of four benchmark tasks: the sum of rewards at each episode are given as the score; the corresponding shaded areas show the 95% confidence intervals; only PPO-RPE succeeded in improving the score from the early stage in both (a) and (b); in (c), PPO-RPE was able to gradually improve the task performance over the others.

Fig. 5. Trajectories of the amount of regularization: instead of each regularization term \( \Omega \), \( \sigma (\rho - \rho^l) \) is illustrated; PPO-RPE has stronger regularization than the others due to the explicit constraint between the latest and baseline policies.

REFERENCES

[1] H. Modares, I. Ranatunga, F. L. Lewis, and D. O. Popa, “Optimized assistive human–robot interaction using reinforcement learning,” IEEE transactions on cybernetics, vol. 46, no. 3, pp. 655–667, 2015.

[2] Y. Tsurumine, Y. Cui, E. Uchibe, and T. Matsubara, “Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation,” Robotics and Autonomous Systems, vol. 112, pp. 72–83, 2019.

[3] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT press, 2018.

[4] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” nature, vol. 521, no. 7553, pp. 436, 2015.

[5] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot et al., “Mastering the game of go with deep neural networks and tree search,” nature, vol. 529, no. 7587, p. 484, 2016.

[6] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, “Trust region policy optimization,” in International conference on machine learning, 2015, pp. 1889–1897.

[7] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” arXiv preprint arXiv:1801.01290, 2018.

[8] S. Parisi, V. Tangkaratt, J. Peters, and M. E. Khan, “Td-regularized actor-critic methods,” Machine Learning, pp. 1–35, 2019.

[9] M. Geist, B. Scherrer, and O. Pietquin, “A theory of regularized markov decision processes,” arXiv preprint arXiv:1901.11275, 2019.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” arXiv preprint arXiv:1707.06347, 2017.

[11] Y. Wang, H. He, and X. Tan, “Truly proximal policy optimization,” in Uncertainty in Artificial Intelligence. PMLR, 2020, pp. 113–122.

[12] M. Yamada, T. Suzuki, T. Kanamori, H. Hachiya, and M. Sugiyama, “Relative density-ratio estimation for robust distribution comparison,” Neural computation, vol. 25, no. 5, pp. 1324–1370, 2013.

[13] M. Sugiyama, S. Liu, M. C. Du Plessis, M. Yamanaka, M. Yamada, T. Suzuki, and T. Kanamori, “Direct divergence approximation between probability distributions and its applications in machine learning,” Journal of Computing Science and Engineering, vol. 7, no. 2, pp. 99–111, 2013.

[14] E. Coumans and Y. Bai, “Pybullet, a python module for physics simulation for games, robotics and machine learning,” GitHub repository, 2016.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., “Human-level control through deep reinforcement learning,” nature, vol. 518, no. 7540, pp. 529–533, 2015.

[16] W. E. L. Ilboudo, T. Kobayashi, and K. Sugimoto, “Robust stochastic gradient descent with student-t distribution based first-order momentum,” IEEE Transactions on Neural Networks and Learning Systems, pp. 1–14, 2020.

[17] F. Liese and I. Vajda, “On divergences and informations in statistics and information theory,” IEEE Transactions on Information Theory, vol. 52, no. 10, pp. 4394–4412, 2006.

[18] T. Kobayashi and W. E. L. Ilboudo, “t-soft update of target network for deep reinforcement learning,” Neural Networks, 2021.

[19] T. Kobayashi, “Adaptive and multiple time-scale eligibility traces for online deep reinforcement learning,” arXiv preprint arXiv:2008.10040, 2020.

[20] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “Openai gym,” arXiv preprint arXiv:1606.01540, 2016.

[21] T. Kobayashi, “Student-t policy in reinforcement learning to acquire global optimum of robot control,” Applied Intelligence, pp. 1–13, 2019.

[22] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” arXiv preprint arXiv:1607.06450, 2016.

[23] S. Ellwinger, E. Uchibe, and K. Doya, “Sigmoid-weighted linear units for neural network function approximation in reinforcement learning,” Neural Networks, vol. 107, pp. 3–11, 2018.

[24] L.-J. Lin, “Self-improving reactive agents based on reinforcement learning, planning and teaching,” Machine learning, vol. 8, no. 3-4, pp. 293–321, 1992.

[25] L. Ziyin, Z. T. Wang, and M. Ueda, “Laprop: a better way to combine momentum with adaptive gradient,” arXiv preprint arXiv:2002.04939, 2020.

[26] T. Kobayashi, “Towards deep robot learning with optimizer applicable to non-stationary problems,” arXiv preprint arXiv:2007.15890, 2020.

[27] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” in Advances in Neural Information Processing Systems Workshop, 2017.